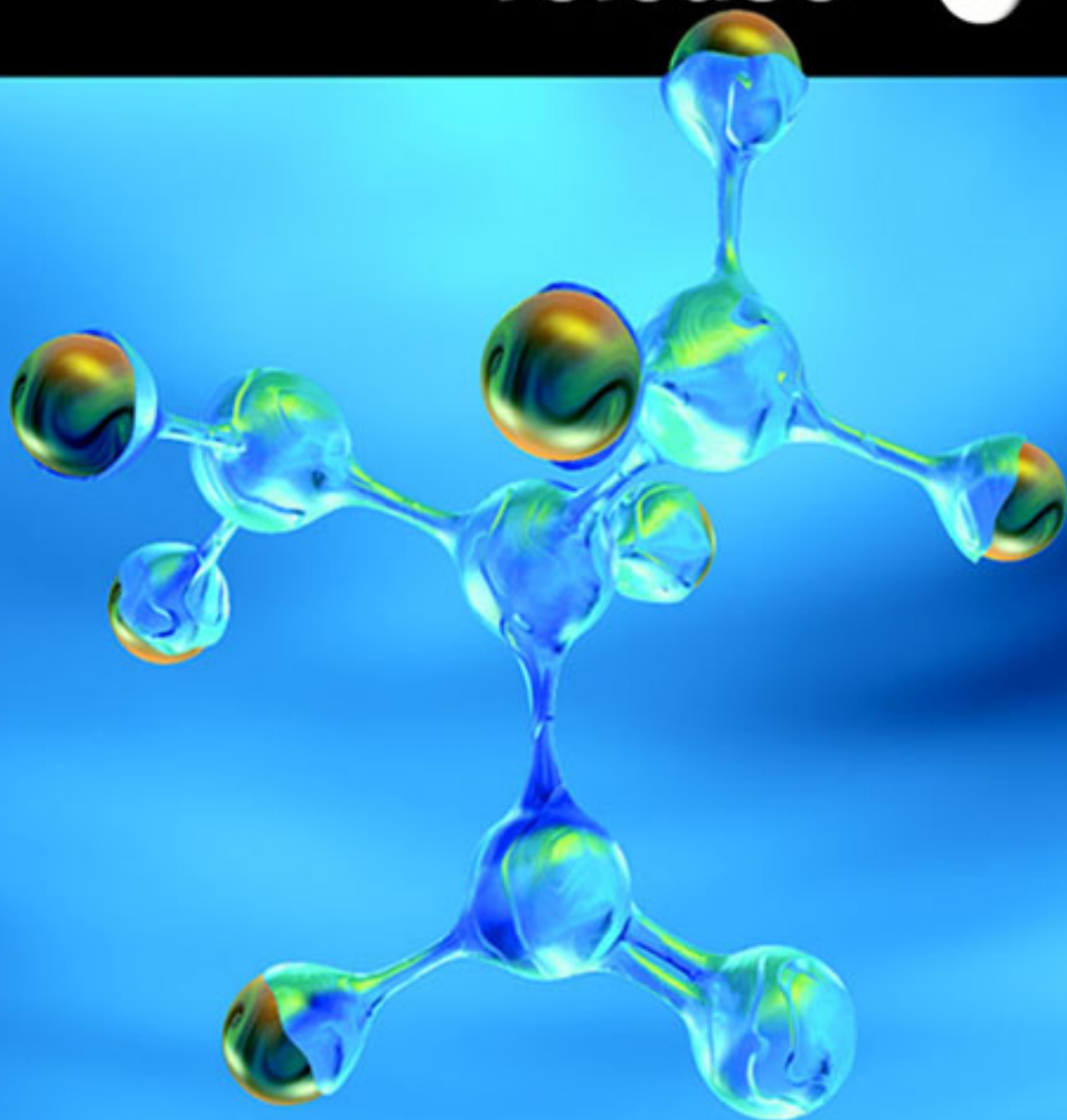


# HyperChem

release 8



**TOOLS FOR MOLECULAR MODELING**

# **HyperChem<sup>®</sup> Release 7 for Windows<sup>®</sup>**

## **Getting Started**

# Copyright © 2002 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperGauss, HyperDFT, HyperNMR, ChemPlus, HyperChem Data, and HyperChem Raytrace are trademarks of Hypercube, Inc.

## Third Party Trademarks

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows , Windows NT, and Windows 95 are trademarks of Microsoft Corporation.

MM2 is a trademark of Dr. Norman Allinger, University of Georgia.

CHARMm is a registered trademark of Accelrys Inc.

IBM is a registered trademark of International Business Machines Inc.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

<b>Introduction</b>	<b>1</b>
<b>Chapter 1 Installing HyperChem</b>	<b>3</b>
Introduction . . . . .	3
Readme File . . . . .	4
System Requirements . . . . .	4
Required Hardware and Software. . . . .	4
Operating System Software. . . . .	4
Computer . . . . .	4
Memory. . . . .	4
Storage/Disk space . . . . .	4
CD-ROM . . . . .	4
Display . . . . .	5
Pointing Devices . . . . .	5
Optional Hardware . . . . .	5
Registering HyperChem . . . . .	5
Licensing Options . . . . .	5
Software-Licensed Standalone Installation . . . . .	6
How the Software-license works . . . . .	6
Installation Instructions . . . . .	7
Licensing HyperChem . . . . .	7
Installing the license key . . . . .	8
Software-Licensed Network Installation . . . . .	8
Installation Instructions . . . . .	9
1. Install the Sentinel License Manager Software . . . . .	9
2. Fingerprinting the computer; obtaining a license key . . . . .	9



3. Installing HyperChem 7 on Client machines. . . . .	11
Hardware-Licensed Standalone Installation . . . . .	12
How the Hardware Lock Works . . . . .	13
Before Installing the Hardware Lock . . . . .	13
Types of Standalone Hardware Locks . . . . .	13
Replacing the Hardware Lock . . . . .	14
Installation Instructions . . . . .	14
Installing the Hardware Lock . . . . .	15
Converting a Hardware Lock to a Software Lock . . . . .	16
Hardware-Licensed Network Installation . . . . .	16
Installation Instructions . . . . .	16
1. Install the SuperPro Server Software . . . . .	16
3. Installing HyperChem 7 on Client machines. . . . .	18
Evaluation Installation . . . . .	19
Windows NT, 2000 or XP Professional Issues . . . . .	19
Technical Support . . . . .	20
Obtaining Support from Your Dealer . . . . .	20
Obtaining Support from Hypercube, Inc. . . . .	20
HyperChem Discussion Group . . . . .	21

## **Chapter 2 Learning HyperChem 23**

Introduction . . . . .	23
Before You Begin . . . . .	23
Using On-line Help . . . . .	24

## **Tutorial 1 HyperChem Basics 25**

### **Lesson 1 Getting Started 27**

Skills Covered in This Lesson . . . . .	27
Starting HyperChem . . . . .	27
Parts of the HyperChem Window. . . . .	28
Title Bar . . . . .	28
Menu Bar . . . . .	28
Tool Bar . . . . .	28
Workspace . . . . .	29
Status Line . . . . .	29
Help . . . . .	29
Control Menu Button . . . . .	29
Maximize/Minimize Buttons . . . . .	29

Using the Mouse . . . . .	29
Keyboard Alternatives . . . . .	31
Keyboard Shortcuts . . . . .	32
Opening a Sample File . . . . .	33
Scrolling Through the List of Files . . . . .	34
Changing Directories . . . . .	35
Selecting a File . . . . .	35
Using Display Settings . . . . .	37
Using Labels . . . . .	38
Using Different Molecular Renderings . . . . .	39
Exiting HyperChem . . . . .	43
For More Information . . . . .	44

## **Lesson 2 Basic Drawing and Editing Techniques 45**

Skills Covered in This Lesson. . . . .	45
Reopening HyperChem . . . . .	45
Expanding the HyperChem Window . . . . .	45
Drawing Atoms . . . . .	46
Drawing Bonds . . . . .	48
Selecting Atoms . . . . .	51
Selecting Groups of Atoms . . . . .	53
Deleting Atoms . . . . .	55
Copying Atoms to the Clipboard . . . . .	55
Clearing the HyperChem Workspace . . . . .	56
For More Information . . . . .	56

## **Lesson 3 Creating Small Molecules in 2D and 3D 57**

Skills Covered in This Lesson. . . . .	57
Reopening HyperChem . . . . .	57
Drawing a 2D Sketch . . . . .	58
Saving Your Work in a File . . . . .	58
Changing Bond Orders . . . . .	60
Making Rings Aromatic . . . . .	62
Labeling Atoms . . . . .	63
Editing Individual Atoms . . . . .	64
Invoking the Model Builder . . . . .	65
Saving the Structure . . . . .	67
For More Information . . . . .	67

## **Lesson 4 Translating, Rotating, and Scaling Molecules 69**

Skills Covered in This Lesson . . . . .	69
Reopening a HIN file. . . . .	69
XY Translation . . . . .	70
Z Translation . . . . .	72
Using the Zoom Tool . . . . .	73
Centering and Scaling . . . . .	74
XY Rotation . . . . .	75
Z Rotation . . . . .	76
Z Clipping . . . . .	76
Clearing the Workspace . . . . .	79
Practice Exercises . . . . .	79
For More Information . . . . .	80

## **Lesson 5 Measuring Structural Properties 81**

Skills Covered in This Lesson . . . . .	81
Creating a 2D Sketch. . . . .	81
Editing the Sketch . . . . .	82
Building the 3D Structure . . . . .	83
Atom Characteristics. . . . .	84
Measuring Bond Distances . . . . .	85
Measuring Bond Angles . . . . .	86
Measuring Torsion Angles . . . . .	87
Measuring Nonbonded Distances. . . . .	88
Hydrogen Bonds . . . . .	89
Practice Exercises . . . . .	90
For More Information . . . . .	90

## **Lesson 6 Creating a Polypeptide 91**

Skills Covered in This Lesson . . . . .	91
Using the Amino Acids Dialog Box . . . . .	91
Building and Displaying a Polypeptide . . . . .	92
Creating a Zwitterion . . . . .	93
Site-specific Mutagenesis . . . . .	94
Saving the Structure . . . . .	97
Practice Exercises . . . . .	97
For More Information . . . . .	98

## **Lesson 7 Selecting and Displaying Subsets 99**

Skills Covered in This Lesson . . . . .	99
Creating an Alpha Helix . . . . .	99
Labeling the Ends . . . . .	100
Selecting the Backbone . . . . .	102
Coloring the Backbone . . . . .	104
Displaying Only the Backbone . . . . .	104
Displaying the Side Chains . . . . .	105
Complementing a Subset . . . . .	106
Rectangular Selection . . . . .	107
Spherical Selection . . . . .	108
Naming a Selection . . . . .	110
Selecting a Ring . . . . .	111
Selecting a Side Chain . . . . .	111
For More Information . . . . .	112

## **Lesson 8 Working with Macromolecules 113**

Skills Covered in This Lesson . . . . .	113
Reading a Brookhaven PDB Format File . . . . .	113
Opening the PDB file . . . . .	114
Removing the Water Molecules . . . . .	115
Displaying the Structure . . . . .	116
Displaying the Primary Structure. . . . .	116
Displaying the Side Chains . . . . .	117
Displaying the Alpha Helix . . . . .	118
Displaying the Disulfide Bridges. . . . .	119
Selecting the Ring Structure . . . . .	121
Displaying Hydrogen Bonds . . . . .	122
Site-specific Mutagenesis . . . . .	123
Rotating a Side Chain . . . . .	124
For More Information . . . . .	126

## **Tutorial 2 HyperChem Calculations 127**

## **Lesson 9 Minimizing the Energy of a System 129**

Skills Covered in This Lesson . . . . .	129
Keeping a Log File . . . . .	130
Choosing the Force Field . . . . .	130
Building Chair Cyclohexane . . . . .	132

Measuring Structural Properties of Chair Cyclohexane . . . . .	133
Performing a Single Point Calculation . . . . .	134
Entering Comments in the Log File . . . . .	134
Optimizing the Structure . . . . .	135
Setting Optimization Variables. . . . .	135
Running the Calculation . . . . .	136
Measuring Properties of the Minimized System . . . . .	136
Transforming from Chair to Boat Cyclohexane . . . . .	137
Measuring the Axial Hydrogens . . . . .	139
Optimizing the Boat Cyclohexane . . . . .	140
Remeasuring the Axial Hydrogens . . . . .	141
Creating Twist Boat Cyclohexane. . . . .	141
Optimizing Twist Boat Cyclohexane. . . . .	143
Analyzing the Results . . . . .	143
Stopping the Log File . . . . .	144
Practice Exercises . . . . .	145
For More Information . . . . .	145

## **Lesson 10 Simulating Dynamic and Equilibrium Behavior 147**

Skills Covered in This Lesson . . . . .	147
Exercise 1: Creating the Isolated Alanine Zwitterion. . . . .	148
Sketching the Alanine Zwitterion . . . . .	148
Editing the Structure . . . . .	149
Assigning Atomic Charges . . . . .	151
Choosing the Force Field . . . . .	152
Building and Exploring the 3D Structure . . . . .	153
Performing a Single Point Calculation . . . . .	155
Optimizing the Isolated Molecule . . . . .	156
Saving the Structure . . . . .	157
Exercise 2: Solvating the Structure . . . . .	158
Setting up Periodic Boundary Conditions . . . . .	159
Specifying the Periodic Box Size . . . . .	159
Displaying the Solvated System . . . . .	160
Adjusting Cutoffs and Dielectric Options . . . . .	161
Optimizing the Solvated Molecule . . . . .	162
Exercise 3: Using Superposition . . . . .	163
Deleting the Water Molecules . . . . .	163
Merging the Two Systems . . . . .	165
Superposing Molecules . . . . .	166
Exercise 4: Simulated Annealing . . . . .	168
Setting up Playback Dynamics . . . . .	170

Setting up Averaging from Molecular Dynamics . . . . .	171
Proceeding with Dynamics . . . . .	173
Reoptimizing the New Structure. . . . .	174
Exercise 5: Langevin and Monte Carlo Simulations. . . . .	174
Part 1: Langevin Dynamics . . . . .	174
Part 2: Monte Carlo Simulation . . . . .	175
To set up the Monte Carlo simulation: . . . . .	175
Setting up Monte Carlo Playback . . . . .	176
Setting up Averaging from Monte Carlo. . . . .	176
Proceeding with Simulation and Analysis . . . . .	177
Advanced Exercises . . . . .	177
For More Information . . . . .	177

## **Lesson 11 Molecular Orbital Calculations 179**

Skills Covered in This Lesson. . . . .	179
Creating the Water Molecule. . . . .	179
Using Structure Alignment . . . . .	180
Displaying Atomic Charges . . . . .	182
Calculating a Wave Function. . . . .	182
Plotting the Electrostatic Potential . . . . .	184
Plotting the Total Charge Density . . . . .	188
Plotting the Total Spin Density . . . . .	189
Plotting Individual Molecular Orbitals. . . . .	189
Saving the Molecule . . . . .	196
Advanced Exercises . . . . .	196
For More Information . . . . .	196

## **Lesson 12 Interaction of Water with N-methylacetamide 197**

Skills Covered in This Lesson. . . . .	197
Creating the NMA Molecule . . . . .	197
Changing from cis- to trans-NMA . . . . .	199
Optimizing the NMA Structure . . . . .	201
Calculating Atomic Charges . . . . .	202
Solvating the Carboxyl Group . . . . .	203
Merging Two Structures . . . . .	203
Moving the Water Molecule . . . . .	204
Optimizing a Subset of the System . . . . .	205
Recomputing Hydrogen Bonds . . . . .	206
Characterizing the Interacting System . . . . .	206
Advanced Exercises . . . . .	207

<b>Lesson 13</b>	<b>Electronic Properties of Proteins</b>	<b>209</b>
	Skills Covered in This Lesson . . . . .	209
	Before You Begin . . . . .	210
	Reading In and Displaying the Molecule . . . . .	210
	Choosing the Area of Interest . . . . .	211
	Choosing a Classical-quantum Boundary . . . . .	213
	Calculating a Wave Function . . . . .	214
	Using the Results in Other Calculations. . . . .	217
	Practice Exercises . . . . .	217
	For More Information . . . . .	218
<b>Tutorial 3</b>	<b>Ab Initio Calculations</b>	<b>219</b>
<b>Lesson 14</b>	<b>Protonation of Water</b>	<b>221</b>
	Skills Covered in This Lesson . . . . .	221
	Setup Drawing. . . . .	221
	Creating H <sub>3</sub> O <sup>+</sup> . . . . .	222
	Choosing a Basis Set . . . . .	223
	Minimum Energy Structure . . . . .	224
	Protonation Energy . . . . .	224
	Practical Exercises. . . . .	225
	For More Information . . . . .	226
<b>Lesson 15</b>	<b>Vibrations and Transition States of Ammonia</b>	<b>227</b>
	Skills Covered in This Lesson . . . . .	227
	Creating Ammonia . . . . .	227
	Choosing a Basis Set . . . . .	228
	Vibrational Analysis on Pyramidal Ammonia . . . . .	229
	Optimizing Planar Ammonia . . . . .	231
	Vibrational Analysis on Planar Ammonia . . . . .	232
	Finding a Transition State by Eigenvector Following. . . . .	234
	Finding a Transition State by Synchronous Transit . . . . .	235
	Practical Exercises. . . . .	239
	For More Information . . . . .	240
<b>Lesson 16</b>	<b>Lowest Excited Electronic State of Ethylene</b>	<b>241</b>
	Skills Covered in This Lesson . . . . .	241

Optimizing the Ground State of Ethylene . . . . .	241
Orbitals of Ground State Ethylene . . . . .	242
CI and the Electronic Spectrum of Ethylene . . . . .	244
Geometry and Energy of Triplet Ethylene . . . . .	246
Practical Exercises . . . . .	248
For More Information . . . . .	248

## **Tutorial 4 HyperChem & DDE 251**

### **Lesson 17 Driving HyperChem from Excel 253**

Skills Covered in This Lesson. . . . .	253
Lesson Requirements . . . . .	254
Exercise 1: Calculations on Sets of Molecules . . . . .	254
Opening HyperChem and Excel . . . . .	255
Setting up the Computation . . . . .	256
Running the Macro . . . . .	258
Graphing the Results . . . . .	259
Exercise 2: Manipulating Molecules with Excel . . . . .	260
Modifying the Macro . . . . .	260
Running the Macro . . . . .	262
Exercise 3: Plotting Energy vs. Torsion Angle. . . . .	263
Creating the Hydrogen Peroxide Molecule . . . . .	263
Modifying the Macro . . . . .	263
Creating a Worksheet . . . . .	264
Setting up the Computation . . . . .	266
Running the Macro . . . . .	266
Graphing the Results . . . . .	266
For More Information . . . . .	267

### **Lesson 18 Driving HyperChem from Visual Basic 269**

Skills Covered in This Lesson. . . . .	269
Lesson Requirements . . . . .	269
Opening HyperChem and Visual Basic . . . . .	269
Creating a Form . . . . .	270
Customizing the Interface . . . . .	272
Entering Visual Basic Code . . . . .	273
Creating the Executable File . . . . .	275
Saving the Visual Basic Files . . . . .	275
Adding test.exe to the Start... menu . . . . .	276



Creating Other Programs . . . . .	276
For More Information . . . . .	276
<b>Tutorial 5 Some Newer Capabilities</b>	<b>277</b>
<b>Lesson 19 DFT Polarizability of H<sub>2</sub></b>	<b>279</b>
Skills Covered in This Lesson . . . . .	279
Creating H <sub>2</sub> . . . . .	279
Choosing an Exchange-Correlation Potential . . . . .	280
Single Point DFT Calculations in an Electric Field . . . . .	284
Polarizability Tensor Calculations . . . . .	285
<b>Lesson 20 Annotations</b>	<b>287</b>
Skills Covered in This Lesson . . . . .	287
Create a Molecular System . . . . .	287
Rectangle Annotations . . . . .	288
Line Annotations . . . . .	292
Text Annotations . . . . .	293
Saving Annotations . . . . .	297
<b>Lesson 21 TNDO Parameter Files</b>	<b>299</b>
Skills Covered in This Lesson . . . . .	299
Default CNDO/INDO Parameters . . . . .	299
Inspecting TNDO Parameters . . . . .	301
Copying TNDO Parameters to a New Set . . . . .	302
Editing TNDO Parameters . . . . .	302
<b>Index</b>	<b>305</b>

# Introduction

Welcome to HyperChem<sup>®</sup>, the molecular modeling and simulation software that lets you perform complex chemical calculations on your desktop computer. HyperChem comes with five manuals: *Getting Started*, *Reference*, *Modules*, *CDK*, and *Computational Chemistry*.

*Getting Started* contains the following parts:

- Chapter 1, “*Installing HyperChem*,” contains information on installation procedures, system requirements, starting HyperChem, and product support.
- Chapter 2, “*Learning HyperChem*,” contains four tutorials. The first tutorial introduces the HyperChem interface and menu system; the second focuses on chemical calculations; the third focuses on *ab initio* quantum mechanical calculations; the fourth gives instruction in communicating with HyperChem from other software.
- An index, which covers the topics discussed in *Learning HyperChem*.

*Reference* contains the following parts:

- “*Reference*,” which discusses HyperChem features and general procedures. Use this manual to find in-depth information on any topic except those described in the Modules manual.
- A comprehensive glossary, which lists important terms and concepts used in the HyperChem documentation.
- A master index, which covers *Getting Started*, *Reference*, *Modules*, *CDK*, and *Computational Chemistry*.

*Modules* contains the following parts:

- “*HyperChem Data*,” which discusses a module for performing database operation with HyperChem. Use this part to find out about retrieving and storing molecules and results from HyperChem in a database.
- “*HyperNMR*,” which discusses a module for calculating NMR parameters and spectra. Use this part to find out about calculating chemical shifts and coupling constants or simulating a one-dimensional NMR spectra.

*Chemist’s Developer Kit (CDK)* contains the following parts:

- “*Customizing HyperChem*,” which discusses how you can use your own menus and scripts to customize your use of HyperChem.
- “*Interfacing to HyperChem*”, which shows you how to have other programs such as Word or Excel or programs written in C++, Visual Basic, etc. communicate with HyperChem to avail themselves of HyperChem functionality.

*Computational Chemistry* contains the following parts:

- The *Practical Guide* gives you an overview and introduction to the calculations you can run with HyperChem.
- *Theory and Methods* is a technical reference, giving you detailed information on the specific implementation of calculations in HyperChem. You can find information here on customizing HyperChem.
- An index, which covers the topics discussed in *Computational Chemistry*.

HyperChem’s standard manual set also includes on-line documentation. Use the HyperChem on-line Help to give you quick access to information as you work in HyperChem.

Readme.txt contains information on late developments not included in the printed manuals.

# Chapter 1

## Installing HyperChem

### Introduction

This chapter contains information on the following:

- System Requirements
- Registering HyperChem
- Licensing Options
- Software-Licensed Standalone Installation
  - How the Software-license works
  - Installation Instructions
- Software-Licensed Network Installation
  - Installation Instructions
  - Network Version
- Hardware-Licensed Standalone Installation
  - How the Hardware Lock Works
  - Before Installing the Hardware Lock
  - Installation Instructions
  - Installing the Hardware Lock
  - Converting a Hardware Lock to a Software Lock
- Hardware-Licensed Network Installation
  - Installing the Network License Server Software
- Evaluation Installation
- Windows NT, 2000 or XP Professional Issues
- Technical Support

## Readme File

After you install HyperChem, be sure to read the readme.txt file. The readme.txt file contains information on late developments that might not be included in the HyperChem documentation set.

To access the readme.txt file, open the HyperChem Release 7 folder on the Windows Start Menu and choose Readme.

## System Requirements

### Required Hardware and Software

The following are required to run HyperChem.

#### Operating System Software

Microsoft® Windows 95, 98, ME, NT4 SP4, 2000, or Windows XP should be installed on your computer.

#### Computer

An IBM compatible PC with an Intel® Pentium II or compatible CPU (or appropriate emulator such as Virtual PC).

#### Memory

HyperChem requires a minimum of 32Mb of RAM. It is recommended that you have 128Mb or more of RAM for better performance.

#### Storage/Disk space

The full installation of HyperChem requires about 150 Mb of disk space. The minimal installation requires about 52 MB. Additional disk space will be required depending on the computational run-time requirements.

#### CD-ROM

A CD-ROM or compatible DVD device is required to install the HyperChem software.

## Display

The HyperChem recommends video displays capable of at least 1024 x 768 resolution at High Color (16 bit depth). The minimum display capabilities supported are 640 x 480 resolution at 8 bit, 256 color depth. It is recommended that you use more than 256 colors.

Visualization of the "Save As HTML" pages created by HyperChem and the Raytrace... feature require High Color (16 bit depth).

## Pointing Devices

A Microsoft mouse or other pointing device compatible with Microsoft Windows is required.

## Optional Hardware

You can use the Microsoft Windows settings to add and choose a printer if you want to print the molecular images created with HyperChem.

## Registering HyperChem

Your HyperChem package includes a license agreement that you should read carefully. It explains our obligations to you, and yours to us. After reading the agreement, fill out the registration card and mail it back to us.

## Licensing Options

The unified product CD-ROM offers a variety of licensing options that must be specified during the installation of the product according to the type of license you have received from Hypercube, Inc.

- Software-licensed standalone installation
- Software-licensed network installation
- Hardware-licensed stand-alone installation
- Hardware-licensed network installation

- Evaluation installation

## Software-Licensed Standalone Installation

The standalone software-licensed installation option allows HyperChem to function on a single computer using a locking mechanism that is specific to one unique system. If the software is moved or reinstalled on a different computer, the HyperChem will not work. The purchase of a standalone software-locked version requires the user to contact Hypercube for a license code during installation.

For users wishing to install a single license of HyperChem on a single computer, the software-licensed standalone version of HyperChem is the recommended choice.

### How the Software-license works

The software license is a device that locks the HyperChem software to the hardware of the PC to which it is licensed.

A *locking code* based on the computer hardware is generated by HyperChem during every execution. HyperChem will then compare the locking code with the currently installed license. If no license has been installed, a Client Activator application will start and prompt the user to license the software. The Client Activator application will display the Locking Code to the user and allow the user to open a file containing the license.

If HyperChem has already been licensed, but the current locking code does not correspond to the license, the Client Activator application will start and display the new Locking Code. This indicates that the hardware environment has changed and the user must relicense HyperChem. The user then should contact Hypercube, Inc. via email at [license@hyper.com](mailto:license@hyper.com), to request an initial license or license transfer. Any change to a computer that effects the Locking Code requires that the user relicense and submit a License Transfer Agreement (LTA).

Hypercube, Inc. will generate a *license key* based on the locking code and other information and send an email to the user with the license key attached. The user will then run start eh Client Activator again and supply the license key. When HyperChem confirms

that the current Locking Code corresponds to the license installed, HyperChem will execute normally.

## Installation Instructions

Standalone installation is carried out as follows:

1. Install the software-locked standalone version of HyperChem on your local PC.
2. Learn the locking codes (fingerprint of your local PC) of our local PC by either trying to execute HyperChem or by running the echoid.exe program included with your installation.
3. E-mail [license@hyper.com](mailto:license@hyper.com) with your HyperChem serial number and the locking codes generated in the previous step.
4. Hypercube, Inc. will e-mail back to you the license key (as a file) that will unlock your software. Place this file (license.lic) in an appropriate place on your PC such as in the HyperChem Program folder.
5. Try executing HyperChem again. This time open the license.lic file at the appropriate time as requested to permanently install your license. From then on, your local copy of HyperChem will run as long as the license.lic file remains in the specified folder.

## Licensing HyperChem

When licensing HyperChem for the first time, execute HyperChem from the Start Menu. The Client Activator will start and present the option to Activate Now. Select this option to display the locking code. Send an email message to [license@hyper.com](mailto:license@hyper.com) with the following information:

- Name
- Email address
- Postal Address
- Telephone number
- Fax number
- Serial Number
- Locking code
- Name of dealer where purchased



Alternately, you may also visit [www.hyper.com/support/licensing](http://www.hyper.com/support/licensing) to obtain a license.

### **Installing the license key**

Once you receive the license key by e-mail, save it as a file on the hard disk. Run HyperChem. If the the screen prompts you to 'Activate Now' the product, click on the Activate Now button. The next screen will display the lock code and an empty text box with the icon of a "folder" next to it. Click on the folder icon. This will allow you to browse the files on your current drive, navigate to the folder where you saved the license file and click on the name of the license file and then on the OK button. The license key will be displayed in the text box. Click on the Next button and finish the procedure. This will install the license on your computer, allowing HyperChem to run.

## **Software-Licensed Network Installation**

The installation of the soft key version of HyperChem 7 for Networks is a three stage process. The first stage requires the installation of the license server software, *SentinelLM*. The second stage is obtaining and installing a license key to activate the product. The third stage is installation of HyperChem 7 on client machines.

*Note:* Before you begin, please make sure that the network configuration necessary for the License Manager to function correctly is installed.

First, verify that the TCP/IP protocol is installed on the network server and client workstations. Also make sure that for Windows 95, 98 and ME computers the 'Client for Microsoft Networks' is installed and bound to the TCP/IP protocol. If not, contact Hypercube, Inc. to verify a custom set of protocols.

Also, if HyperChem clients are to be used from outside the local subnet of the license server, please verify that the names of the computer to be used as a license server and all the clients have computer names that are properly resolved into IP addresses. If you are using the Domain Name System (DNS) to issue permanent IP addresses, the DNS server will also provide name resolution. However, networks using the Dynamic Host Configuration Protocol (DHCP) to issue temporary IP addresses may use several methods for name resolution. For further details concerning name res-

olution, please consult the Name Resolution Overview page at [www.hyper.com/support/TechPub](http://www.hyper.com/support/TechPub).

## Installation Instructions

### 1. Install the Sentinel License Manager Software

On the computer which is to act as a license key server, insert the HyperChem 7 CD to start the Setup program. If the initial install screen does not appear, start the Setup program by double clicking on the 'setup.exe' filename or the setup icon from the CD.

Proceed, following the on-screen instructions. Select to install the 'Purchased Product' for product type and 'Networked' for license type. Then select to install the 'Network License Server' and select 'Software-locked' for the license server type.

After entering your registration and dealer information, you will be presented with a choice for the destination folder. Choose a name for the License server directory (e.g. C:\Hyper7\_Server) and click Finish. This will install the SentinelLM software components on the target machine. Once this step is completed, the setup program will exit. (This document assumes that C:\Hyper7\_Server is the folder to which the license server application is installed from this point forward.)

Using a file browser open the license server folder (C:\Hyper7\_Server). You will find folders named 'Admin' and 'Server'. The 'Admin' folder contains utilities for the network administrator to install the licence and maintain the license server and the 'Server' folder contains the server application.

### 2. Fingerprinting the computer and obtaining a license key

Please make sure that the IP address and Hostname of the License server are permanent. Run the 'echoid.exe' program contained in the License server folder (usually 'C:\Hyper7\_Server\admin'). This program generates two locking codes. Please note these locking codes and send them to [license@hyper.com](mailto:license@hyper.com) along with the following information:

- Contact name
- Email address

- Postal address
- Telephone number
- Fax number
- Name of dealer where purchased
- Serial number
- Locking code
- Number of network licenses purchased
- A list of IP subnets (maximum of 7) from which the client computers will access the license server.

On receipt of this information, Hypercube, Inc. will generate a license key and send it to you via e-mail. When you receive the email containing the license file, copy the attached file to a safe location on your computer, and install the license following the steps here.

#### A. Start the License Manager Service

To install and start the license server under Windows NT, 2000 or XP, execute 'loadls.exe' from license server folder (usually 'C:\Hyper7\_Server\server\winnt'). This utility will prompt you for the location of the 'lservnt.exe' server file. Click 'Add' to install and start the SentinelLM service.

To start the license server under Windows 95, 98 or ME, first start the 'MS-DOS Prompt' and change to the license server folder (usually 'C:\Hyper7\_Server\server\win9x'). Then execute 'lserv9x.exe' to start the server.

#### B. Install the license

At the command prompt change to the License Manager administration tools folder (usually 'C:\Hyper7\_Server\admin'). Then install the license by typing 'lslic -F licensefile.lic' where 'licensefile.lic' is the license file you received from Hypercube, Inc. (Be sure to type a capital letter for the -F command.)

#### C. Confirm the installation

Two utilities are provided for querying the License Manager Software to confirm the proper installation of the license. A graphical program, 'wlmadmin.exe' will scan the subnet for SentinelLM servers and return the number of licenses installed, current usage, and other diagnostic information. A command-line utility, 'lsmon.exe' gives similar information. These utilities are installed

on the license server computer, typically in the 'C:\Hyper7\_Server\admin' folder, and on the client computers, typically in the 'C:\Hyper7\admin' folder. Both the client and server should show the HyperChem Product and the number of licenses available.

If the client computer does not show available licenses issued by the server, then the broadcast method used by the License Manager may not work. You will need to include the LSHOST or LSFORCEHOST environment variable on the client computer to direct HyperChem to the location of the server.

**Important:** If a router (gateway) is present between the license server computer and the HyperChem clients, the licensing mechanism will not be able to use the broadcast method of locating the license server. Other network equipment may also block broadcast calls. If this is the case, make sure that the name resolution is possible between the two subnets. For further information, contact Hypercube, Inc. Technical Support.

For further information on the Rainbow Technologies License Manager administration tools, please consult the accompanying SentinelLM System Administrator's Guide (lmsa61.pdf). Use Adobe Acrobat Reader to view the Guide.

### 3. Installing HyperChem 7 on Client machines

1. Place the HyperChem 7 CD in you CD-ROM drive
2. If the Windows Autorun feature is turned on, the installation options will be displayed automatically. If the installation options do not appear, open the Windows Start Menu and choose Run. The Run dialog box opens.
3. Enter the following command in the open text box (assuming D is your CD-ROM drive):  
`D:setup`
4. Choose OK. The HyperChem setup program initializes and the HyperChem setup dialog box opens
5. Follow the on screen instructions for product choice and installation.

The HyperChem setup program will detect existing HyperChem installation and offer you the choice of uninstalling or keeping the older version of the program.

On each of the client machines install HyperChem from the HyperChem 7 CD following the on screen instructions. Select to install the 'Purchased Product' for product type and 'Networked' for license type. Then select 'HyperChem client' component and 'From Software locked Server'. Select the type of install, Typical, Minimal or Custom. Choose the folder into which you want to install HyperChem (the default is C:\Hyper7). Click finish to proceed with the installation. After the installation is done setting up HyperChem, installations for POV-Ray and Acrobat Reader will follow.

*Note:* The license server components should not be installed on the client machines.

Once the license server is activated with the proper license key, HyperChem can be used from any of the client machines. If the license key server is not on the same subnet as the client machines, acquiring a license by the client machine can be done by any of the following methods:

A. Set the LSHOST Environment Variable to the IP address of the license server.

On a Windows 95, 98 or ME client, set the environment variable by including the line 'SET LSHOST=<IP address of server>' in the autoexec.bat file of the client computer. You will need to restart the computer before this change will go into effect.

On a Windows NT, 2000 or XP client set the environment variable by selecting 'Properties' from My Computer and selecting the 'Advanced' tab. Click on 'Environment Variables...' and then click New under 'User variables.' The variable name should be LSHOST and the value is the IP address of the license server computer.

B. Alternatively, create a file named "LSHOST", containing the IP address or hostname of the license server machine. Place this file in the directory containing HyperChem program executables.

Start HyperChem by selecting the icon from the Start Menu.

## Hardware-Licensed Standalone Installation

If your copy of HyperChem is a hardware locked copy, you will find a small device with an electrical connector at each end, enclosed with your HyperChem manuals and disks. This device is a hardware lock and is also known as a dongle. HyperChem

requires the hardware lock to be connected to the parallel or USB port of your computer for normal operation.

## How the Hardware Lock Works

The hardware lock is a copy protection device supplied with HyperChem. HyperChem interrogates the hardware lock to verify its presence. The hardware lock should not affect any program except HyperChem, nor should it affect any peripheral devices connected to the same parallel interface port or other USB ports. You do not need to remove the hardware lock to run other programs.

**Important:** The parallel port hardware lock works on the assumption that the parallel interface is a resource that can be shared by multiple devices. However, some new peripheral devices, and their support software, work on the assumption that the port should *not* be shared with other devices. If you have such a peripheral (examples include some printers and backup devices), its software may complain because HyperChem is already using the port; if that peripheral is connected first, it may not allow HyperChem to communicate with the hardware lock. If this happens, you may need to add another parallel port to your computer so that your peripheral device does not need to share a port with other software and hardware, or you might use a software-locked version or network version of HyperChem or network version of the peripheral so that another port is not needed. Otherwise, you may need to disable the device's driver while you use HyperChem so that it is not trying to take control over the port.

## Before Installing the Hardware Lock

### Types of Standalone Hardware Locks

The hardware locks are available for either the parallel port or USB ports. The locks work identically and either one may be used. The hardware lock requires the Sentinel System Driver version 5.39.2 or later from Rainbow Technologies to function properly with HyperChem. This driver is installed during the Standalone, Hardware-Licensed HyperChem installation.

## Replacing the Hardware Lock

Only one hardware lock comes with a HyperChem package. If the device is lost, stolen or destroyed, you must buy another copy of HyperChem to replace it. We suggest that you insure the device for its replacement cost: the full price of your HyperChem package.

If your hardware lock is damaged or fails in service, contact your HyperChem dealer. You must return the original lock to receive a replacement.

*Caution:* The computer, peripheral devices or hardware lock may be damaged when parallel port devices are connected or disconnected while the computer is running.

- Always turn off the computer and peripheral devices before installing or removing the hardware lock.
- Do not connect or disconnect a peripheral device when the computer is running. If the peripheral is disconnected during operation, low voltages and signals passing through the parallel or USB port interface could damage your computer, peripheral devices, data or lock.
- Do not turn a peripheral device on or off while it is connected to your computer; this can cause a hardware lock failure.

## Installation Instructions

*Important:* Make sure that the hardware lock is not present on the parallel port or USB port while installing HyperChem. Also, if you are updating the hardware lock driver, called the Sentinel System Driver, make sure to remove the hardware lock. The driver installation may cause the hardware lock to become damaged.

1. Insert the HyperChem 7 CD into a CD or compatible drive.
2. If the Windows Autorun feature is enabled, the installation options will be displayed automatically. If the installation options do not appear, open the Windows Start Menu and choose Run. The Run dialog box opens.
3. Enter the following command in the open text box:  
`D:setup`
4. Choose OK. The HyperChem setup program initializes and the HyperChem installation options are displayed.

5. Choose HyperChem Installation and follow the on screen instructions for product choice and installation type. You must select the Standalone, Hardware-licensed version of HyperChem.

On Windows 95, 98 and ME, the installation will require the computer to be rebooted. The hardware lock drivers will be installed when Windows is restarted. On Windows NT, 2000 and XP, the installation proceeds without rebooting the computer. The hardware lock may be installed after the HyperChem installation is complete.

## Installing the Hardware Lock

To install the parallel port hardware lock, follow these steps:

1. Make sure that HyperChem is completely installed *before* installing the hardware lock. Remember to restart the computer to ensure the Sentinel System Driver is correctly installed on Windows 95, 98 and ME.
2. Shut down the computer and be sure to turn off the computer and all peripheral devices (e.g. printer, etc.).
3. Attach the male computer connector of the hardware lock (at the end labeled “Computer”) to your computer’s parallel port connector.

*Note:* If your parallel port connectors are full, disconnect one of the peripheral devices, and connect the hardware lock to the open parallel port. Next, reconnect the disconnected peripheral device at the female end of the hardware lock, If you like, you can connect the hardware lock in-line with any device connected to the parallel port on your computer (printer, plotter, etc.)

To install the USB port hardware lock, follow these steps:

1. Make sure that HyperChem is completely installed *before* installing the hardware lock. Remember to restart the computer to ensure the Sentinel System Driver is correctly installed on Windows 95, 98 and ME.
2. Attach the USB lock to your computer’s USB port connector. The hardware lock should be recognized as a Sentinel SuperPro device.



## Converting a Hardware Lock to a Software Lock

In the event that you wish to convert your stand-alone hardware locked version of HyperChem to a software locked version, you should contact our sales department and arrange to have the hardware lock returned in exchange for a single software lock. You will need to generate the machine code by running the Hlicense.exe program located in the License sub-folder of the HyperChem folder. E-mail the machine code and the HyperChem serial number to [license@hyper.com](mailto:license@hyper.com). On receipt of the hardware lock, a license key will be e-mailed back to you to activate HyperChem.

## Hardware-Licensed Network Installation

The hardware-licensed version of HyperChem for Networks uses a hardware lock to authenticate HyperChem running on client computers. The hardware lock is a physical device, placed on either the parallel port or USB port of any Windows or Novell NetWare

The hardware-licensed version of HyperChem for Networks is available with 1, 2, 3, 5, 10, 25, 50 and unlimited licenses. This version requires the installation of the Sentinel System Driver and SuperPro Server software on the license server computer. HyperChem 7 on client machines also must have the Sentinel System Driver installed to be able to access the hardware lock.

*Note:* If using a Windows PC as a license server, verify that the TCP/IP protocol is installed on the network server and client workstations. If installing on a Novell Netware license server, only the IPX/SPX protocol is supported.

Also, if HyperChem clients are to be used from outside the local subnet of the license server or through a router (gateway), please contact Hypercube, Inc. Technical Support for specific installation instructions.

## Installation Instructions

### 1. Install the SuperPro Server Software

#### A. Install the SuperPro Server software

On the computer which is to act as a license key server, insert the HyperChem 7 CD to start the Setup program. If the initial install

screen does not appear, start the Setup program by double clicking on the 'setup.exe' filename or the setup icon from the CD.

Proceed, following the on-screen instructions. Select to install the 'Purchased Product' for product type and 'Networked' for license type. Then select to install the 'Network License Server' and select 'Hardware-licensed' for the license server type.

After entering your registration and dealer information, you will be presented with a choice for the destination folder. Choose a name for the License server directory (e.g. C:\Hyper7\_Server) and click Finish. This will install the Sentinel System Driver and the SuperPro Server application. (This document assumes that C:\Hyper7\_Server is the folder to which the license server application is installed from this point forward.)

**Important:** Make sure that the Sentinel System Driver is completely installed *before* attaching the parallel port or USB port hardware lock. Damage to the hardware lock is possible if the lock is present during the driver installation. Rebooting the computer will be required on Windows 95, 98 and ME, to install the Sentinel System Driver. Also, if you are updating the Sentinel System Driver, make sure to remove the hardware lock.

#### B. Installing the Hardware lock

When installing the parallel port hardware lock, you should shut down the server and all parallel port devices. Place the hardware lock on the license server computer and restart the system. The USB hardware lock may be installed while the system is running.

#### C. Start the SuperPro Server

To install and start the license server under Windows NT, 2000 or XP, execute 'loadlserv.exe' from license server folder (usually 'C:\Hyper7\_Server\server\winnt'). This utility will prompt you for the location of the 'lservnt.exe' server file. Click 'Install Service' to install and start the SentinelLM service.

To start the license server under Windows 95, 98 or ME, first start the 'MS-DOS Prompt' and change to the license server folder (usually 'C:\Hyper7\_Server\server\win9x'). Then execute 'spnsrv9x.exe' to start the server.

#### D. Confirm the installation

A monitor utility is provided to query the SuperPro Server for current license usage. The utility will scan the subnet and report the

number of licenses that are installed and in use on all running servers in the subnet.

**Important:** If a router (gateway) is present between the license server computer and the HyperChem clients, the licensing mechanism will not be able to use the broadcast method of locating the license server. Other network equipment may also block broadcast calls. If this is the case, make sure that the name resolution is possible between the two subnets. For further information, contact Hypercube, Inc. Technical Support.

For further information on the Rainbow Technologies SuperPro server and monitor utility, please consult the accompanying Sentinel SuperPro System Administrator's Guide (SentinelSP Sys Admin Guide.pdf). Use Adobe Acrobat Reader to view the Guide.

### 3. Installing HyperChem 7 on Client machines

1. Place the HyperChem 7 CD in you CD-ROM drive
2. If the Windows Autorun feature is turned on, the installation options will be displayed automatically. If the installation options do not appear, open the Windows Start Menu and choose Run. The Run dialog box opens.
3. Enter the following command in the open text box:

D:setup

4. Choose OK. The HyperChem setup program initializes and the HyperChem setup dialog box opens
5. Follow the on screen instructions for product choice and installation.

The HyperChem setup program will detect existing HyperChem installation and offer you the choice of uninstalling or keeping the older version of the program.

On each of the client machines install HyperChem from the HyperChem 7 CD following the on screen instructions. Select to install the 'Purchased Product' for product type and 'Networked' for license type. Then select 'HyperChem client' component and 'From a Hardware-licensed Server'. Select the type of install, Typical, Minimal or Custom. Choose the folder into which you want to install HyperChem (the default is C:\Hyper7). Click finish to proceed with the installation. After the installation is done setting

up HyperChem, installations for POV-Ray and Acrobat Reader will follow.

*Note:* The license server components should not be installed on the client machines.

Once the license server is activated with the proper hardware lock, HyperChem can be used from any of the client machines. To direct HyperChem to a specific server and negating the use of broadcast calls to locate a license, set the environment variable, NSP\_HOST, to the IP address of the license server as follows:

On a Windows 95, 98 or ME client, set the environment variable by including the line 'SET NSP\_HOST=<IP address of server>' in the autoexec.bat file of the client computer. You will need to restart the computer before this change will go into effect.

On a Windows NT, 2000 or XP client set the environment variable by selecting 'Properties' from My Computer and selecting the 'Advanced' tab. Click on 'Environment Variables...' and then click New under 'User variables.' The variable name should be NSP\_HOST and the value is the IP address of the license server computer.

Start HyperChem by selecting the icon from the Start Menu.

## Evaluation Installation

A special Evaluation option is available for those users who wish to run HyperChem for a before purchasing the software. The Evaluation is a fully functional version of HyperChem that contains a free time-limited license. This option is intended for product evaluation, and will expire after the Evaluation period is over. After expiration, HyperChem will no longer function. The setup screen will guide through the Evaluation version installation.

## Windows NT, 2000 or XP Professional Issues

On the Windows NT, 2000 and XP operating systems, HyperChem 7 requires Administrator privileges for installation. Unless the user has adequate privileges to modify system files, the user must log off and log on as Administrator and install HyperChem.

**Important:** For any user that intends to use HyperChem, the HyperChem Registry settings must be set up. The HyperChem user Registry is set up by executing

D:\install\Hyper7\NewUser.exe

from the Run... dialog box from the Windows Start Menu, where D: is the letter of your CD drive.

## Technical Support

### Obtaining Support from Your Dealer

Your HyperChem dealer is your main source for technical support – initially and as you continue to work with HyperChem. All authorized dealers have access to Hypercube's Scientific and Technical Support Staff and can consult with them for help in solving any problems you have.

It's your dealer's responsibility to offer complete installation and configuration services. But getting HyperChem up and running is only the beginning of your dealer's commitment to your success with HyperChem. You can also contract with your dealer for these additional services: training, telephone and on-site support, and customizing HyperChem for your application.

### Obtaining Support from Hypercube, Inc.

Before contacting your dealer, you can often save time by checking some common causes of hardware and software problems. For example, if you are having trouble with the distribution media, find out if your device can read other media of the same kind.

If you are experiencing problems using HyperChem, please visit the HyperCube, Inc. support web site at:

**[www.hyper.com/support](http://www.hyper.com/support)**

If you think that you have found a bug or error in the software, or if you have suggestions for improvements or features that you would like to see in it, please contact us by E-mail at

**[support@hyper.com](mailto:support@hyper.com)**

or by the routes for support described in the information in the product package.

## **HyperChem Discussion Group**

The HyperChem Discussion Group is designed to encourage users to share their experiences and ideas to improve their use of HyperChem. Information on scientific and technical issues, such as Visual Basic tools, HyperChem structures, and product support, can be posted to this group.



# Chapter 2

## Learning HyperChem

### Introduction

This chapter outlines how HyperChem works; it has four main parts:

- Tutorial 1, *HyperChem Basics*, introduces you to the HyperChem user interface and teaches you how to use the menus and basic menu items for building and working with molecules.
- Tutorial 2, *HyperChem Calculations*, focuses on chemical calculations. It provides step-by-step instructions for performing single point calculations, optimization/minimization, and molecular dynamics using both semi-empirical quantum mechanics and molecular mechanics methods.
- Tutorial 3, *Ab Initio Calculations*, focuses on *ab initio* quantum mechanical calculations for performing single point calculations, geometry optimization, vibrational analysis, and electronic spectral calculations.
- Tutorial 4, *HyperChem and DDE*, shows you how to create links with other Windows applications using the Dynamic Data Exchange (DDE) feature of Microsoft® Windows™.

### Before You Begin

This chapter assumes that you know how to perform basic operations with your computer, such as starting applications from Microsoft Windows; sizing, moving, and scrolling through windows; and using menus.



Before you begin, make sure that the installation and configuration procedures in Chapter 1, “*Installing HyperChem,*” is complete.

## Using On-line Help

Whenever you want more information about how to use HyperChem, you can access on-line help in the following ways:

- L-click Help on the menu bar.
- L-click on the menu or menu item you want information on and press [F1].
- Press [Shift] + [F1] to get the Help cursor, and then L-click on the tool you want information on.
- For most dialog box options, L-click on the option and then press [F1].

# Tutorial 1

## HyperChem Basics

Tutorial 1 introduces you to the HyperChem user interface and discusses using tools, menus, and menu items to build, edit, display, and manipulate molecules. Tutorial 1 has eight lessons, as described in the following table:

Lesson	Information covered	Time to complete
Getting Started	Opening HyperChem; introduction to the user interface; opening and closing a file, and using labels and molecular renderings.	20 minutes
Basic Drawing and Editing Techniques	Drawing, selecting, copying, and deleting atoms and bonds.	10 minutes
Creating Small Molecules in 2D and 3D	Drawing and editing a 2D sketch of a molecule and transforming it into a 3D model.	10 minutes
Translating, Rotating, and Scaling Molecules	Using the tools in the tool bar to move and manipulate molecular structures.	10 minutes
Measuring Structural Properties	Measuring and adjusting the geometry of molecular structures.	35 minutes

---

<b>Lesson</b>	<b>Information covered</b>	<b>Time to complete</b>
Creating a Polypeptide	Creating biopolymers using the amino acid and nucleic acid databases.	15 minutes
Selecting and Displaying Subsets	Using specific selection techniques to select subsets for movement, display, calculation, and analysis.	15 minutes
Working with Macromolecules	Working with molecules from the Brookhaven Protein Data Bank; performing site-specific mutagenesis.	35 minutes

---

# Lesson 1

## Getting Started

### Skills Covered in This Lesson

- Starting HyperChem
- Using the mouse
- Identifying parts of the HyperChem window
- Using HyperChem menus
- Opening a file
- Labeling a molecule
- Using molecular renderings
- Exiting HyperChem

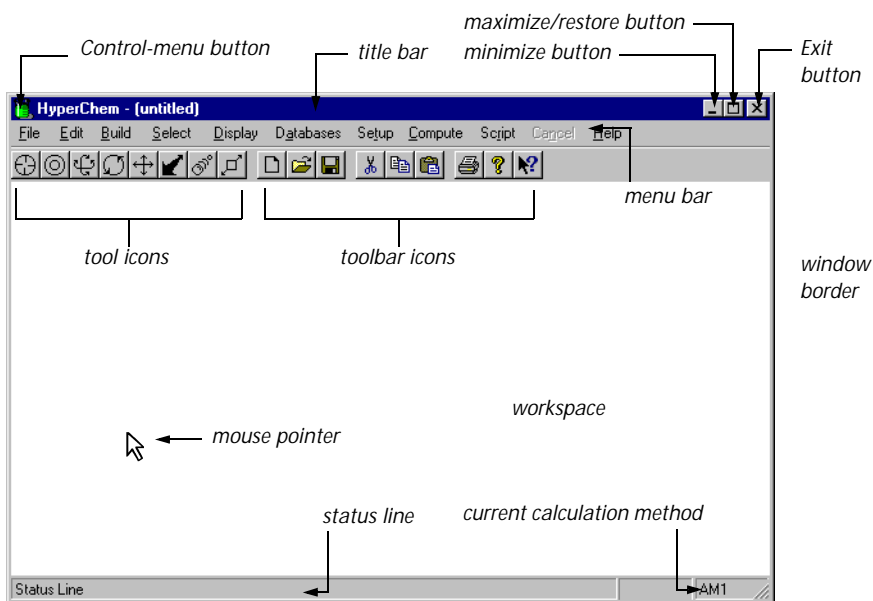
Lesson 1 explains HyperChem basics. First you start the program and identify parts of the HyperChem window. Then you select and open a file containing the atom types and coordinates for the 60-carbon spherical molecule, buckminsterfullerene.

### Starting HyperChem

To start HyperChem:

1. Click on the Start button, then Programs, then HyperChem Release 7. You should see a green beaker icon labelled HyperChem Professional. Click on this item to start HyperChem. Depending on the version of Windows you are using you may also see the green beaker icon in other contexts.

The HyperChem window opens. Depending upon the version of HyperChem you are using, the window may look slightly different from that below. The differences are not important in this context.



## Parts of the HyperChem Window

Familiarize yourself with the different parts of the HyperChem window. Each item is described in this section.

### Title Bar

The title bar shows the name of the file you are working on. If you are working on a file not previously saved, the name appears as untitled.

### Menu Bar

The menu bar contains the names of the different HyperChem menus: File, Edit, Build, Select, Display, Databases, Setup, Compute, Annotations, Cancel, Script, and Help.

### Tool Bar

The left side of the Tool bar contains the eight tools icons that you use to draw, select, display, and move atoms and molecules. To the

right are the four tool icons associated with annotations. To the right of this are the toolbar icons which provide shortcuts for operations such as reading and writing files, copying and pasting structures, and using the on-line help.

## Workspace

The workspace is where HyperChem displays the current molecular system.

## Status Line

The status line shows information such as the number of atoms in the molecule that is currently displayed, the status of a calculation, or an energy or gradient value. When you choose a menu item, a brief description of the item appears in the status line.

## Help

You use the Help menu to gain access to on-line help.

## Control Menu Button

The Control menu contains commands to resize, move, maximize, and close the HyperChem window. It also contains the Switch command, which lets you activate other windows.

## Maximize/Minimize Buttons

This button expands a window to its maximum size. Some windows take up the entire screen when they're maximized; others change just slightly. You can restore a window to its original size by clicking the Maximize button again.

## Using the Mouse

You communicate with HyperChem mostly by using the mouse, or another pointing device that emulates a mouse. As you move the mouse, the cursor in the window matches the movement. You

communicate with your computer by pressing a mouse button while the cursor is in a particular area of the window.

The following terms describe using the mouse with HyperChem:

<b>Point</b>	Move (slide) the mouse so that the cursor points at what you want to select in the HyperChem window.
<b>L-click</b>	Press and release the left mouse button.
<b>R-click</b>	Press and release the right mouse button. Generally, R-click has the opposite effect of L-click.
<b>Double-click</b>	Press and release the left mouse button twice, quickly.
<b>L-drag or R-drag</b>	Hold down the left or right mouse button, and move (slide) the cursor to a new position in the workspace. Release the mouse button.
<b>LR-drag</b>	Press and hold down the left mouse button, then the right mouse button, and move the cursor to a new position in the workspace. Release both mouse buttons.
<b>RL-drag</b>	Same as LR-drag, but press the right mouse button first.

The mouse cursor changes shape depending on where it is in the window and on what mode HyperChem is in.

To see how the HyperChem cursor changes:

1. Point to the Drawing tool and L-click.



2. Move the cursor to the workspace.

The cursor takes the shape of the Drawing tool.

3. Point to the Selection tool and L-click.



4. Move the cursor into the workspace.

It takes the shape of the Selection tool.

5. Move the cursor to the Select menu on the menu bar.

The cursor turns back into an arrow.

Later in this tutorial, you learn more about these and other cursor shapes.

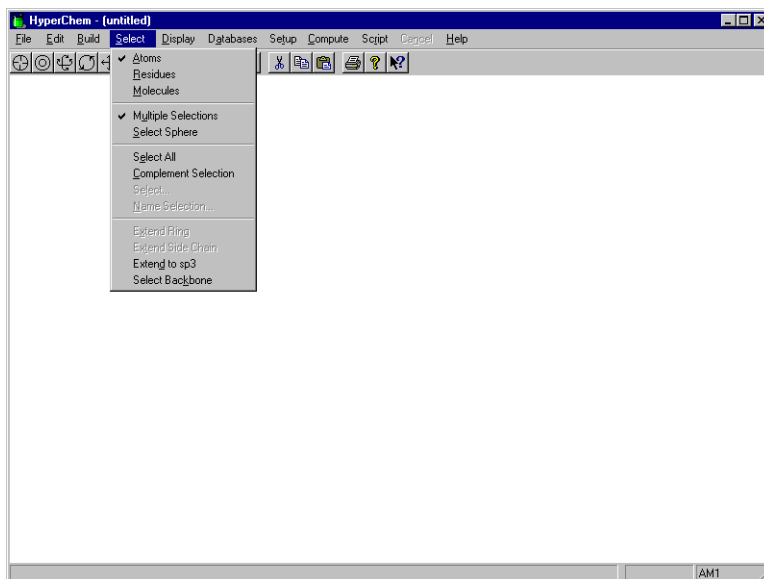
## Keyboard Alternatives

HyperChem provides standard Windows keyboard alternatives to using the mouse.

To open a menu using the keyboard alternative:

1. Hold down the [Alt] but[Alt]ton[D] and simultaneously press the [S] key.

The Select menu opens.



Each menu in the title bar has an underlined letter. When you press the [Alt] button and this letter, the menu opens.

To close the menu:

1. Press the [Alt] or [Esc] key.

The Select menu closes.



To select a menu item:

1. Hold down the [Alt] button and simultaneously press the [S] key.

The Select menu opens. Each menu item in the menu has an underlined letter.

2. Press [A].

This is the same as using the mouse to choose Atom on the Select menu.

*Note:* The *HyperChem Release 5 for Windows Reference Manual* lists keyboard alternatives.

## Keyboard Shortcuts

HyperChem also has various keyboard shortcuts. Use the following shortcuts when you want to avoid using the equivalent menus or menu items:

[Ctrl] + [N]	Same as New on the File menu.
[Ctrl] + [O]	Same as Open on the File menu.
[Ctrl] + [S]	Same as Save on the File menu.
[Ctrl] + [X]	Same as Cut on the Edit menu.
[Ctrl] + [C]	Same as Copy on the Edit menu.
[Ctrl] + [V]	Same as Paste on the Edit menu.
[Alt] + [F4]	Same as Exit on the File menu
[Ctrl] + [D]	Same as Delete on the Annotations menu.
[Ctrl] + [K]	Same as Select Next on the Annotations menu.
[Ctrl] + [L]	Same as Select Previous on the Annotations menu.
[Ctrl] + [F]	Same as Move Forward on the Annotations menu.
[Ctrl] + [B]	Same as Move Backward on the Annotations menu.
[Spacebar]	Pressing the space bar is the same as using Scale to Fit on the Display menu. This menu item centers and scales the molecule or selected items in the workspace.
F[F2]	Same as Last Rendering on Display menu.

[F3]	Same as Show Isosurface on Display menu.
[F4]	Same as Isosurface... on the Display menu.
[F9]	Same as Copy Image on the Edit menu.
[Esc]	Same as the Cancel menu item.

## Opening a Sample File

You can work with molecules in at least three ways using HyperChem:

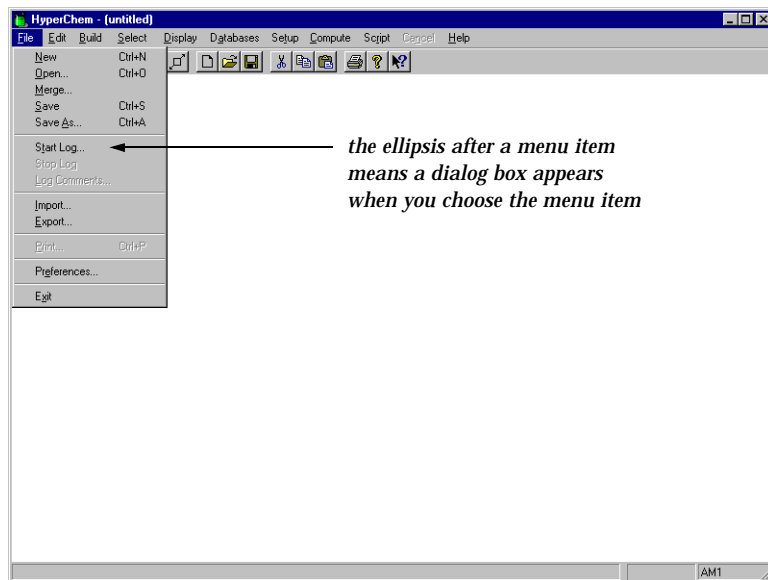
- Using the tools and editing features, you create a two-dimensional (2D) sketch of a molecule, and then convert it with the HyperChem Model Builder into a three-dimensional (3D) representation.
- Selecting residues sequentially from the HyperChem amino acid and nucleotide libraries to construct proteins and nucleic acids.
- Reading in atom types and molecular coordinates that have been saved in the HyperChem input file format (HIN file), the Brookhaven Protein Data Bank format (ENT file), or other file format.

In this exercise, to become more familiar with how HyperChem works, you read in molecular coordinates from a HIN file.

To open a HIN file:

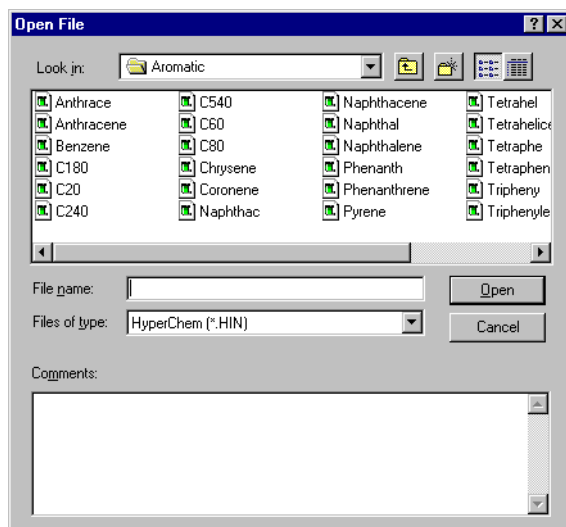
1. Move the pointer to File on the menu bar at the top of the HyperChem window.
2. L-click on File.

The File menu opens. An ellipsis appears after some menu items. This means a dialog box opens when you choose these menu items.



3. L-click on Open.

The Open File dialog box appears:

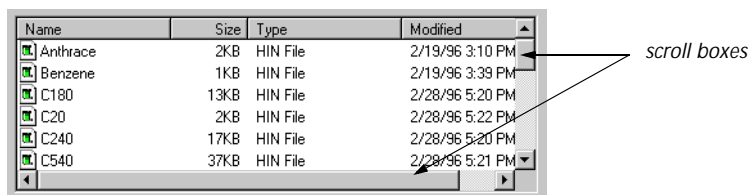


## Scrolling Through the List of Files

A list of files is located in the middle of the dialog box.

To scroll through the list:

1. Point the cursor on a scroll box and L-drag it up and down or left and right.



HyperChem scrolls through the list of filenames.

2. L-click on the up and down arrow keys to scroll one line at a time.

## Changing Directories

If the file you want is not in the current directory, the Directories list box lets you change to a different directory or disk drive.

## Selecting a File

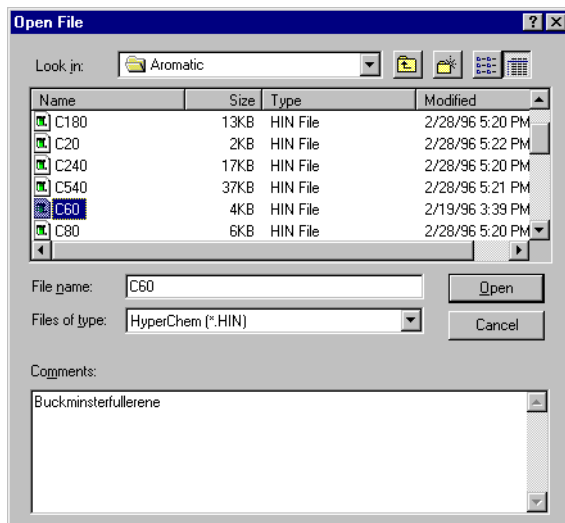
To select a file:

1. In the Files list box, go to the Samples/Aromatic directory and L-click on c60.hin.

HyperChem displays that name in the File Name text box.

*Note:* You can also enter a filename by entering it in the File Name text box.

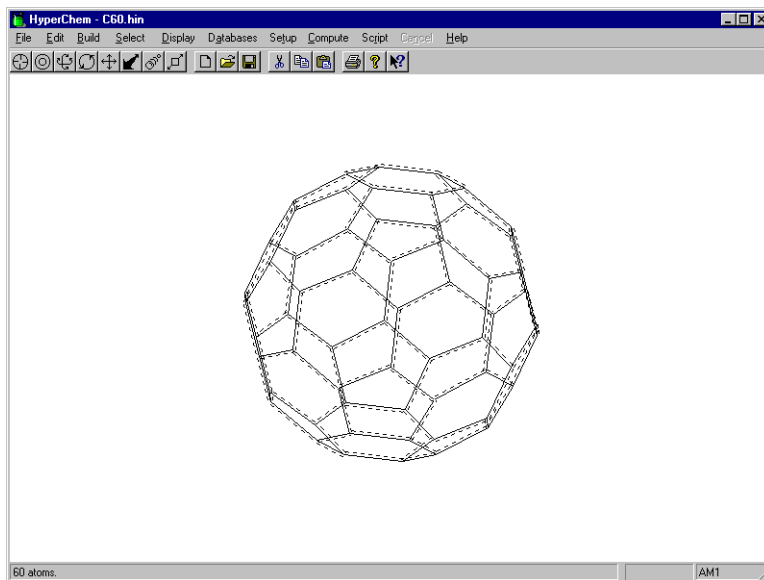
When a filename is highlighted, any comments contained within the file appears in the Comments text box at the bottom of the Open File dialog box.



2. L-click the Open button.

The dialog box disappears. The cursor turns briefly into an hour-glass icon to let you know that HyperChem is reading in the file. As HyperChem reads the file, the number of atoms appears in the status line at the bottom of the HyperChem window.

The molecule appears:



*Note:* You can also open a file by double-clicking on a filename.

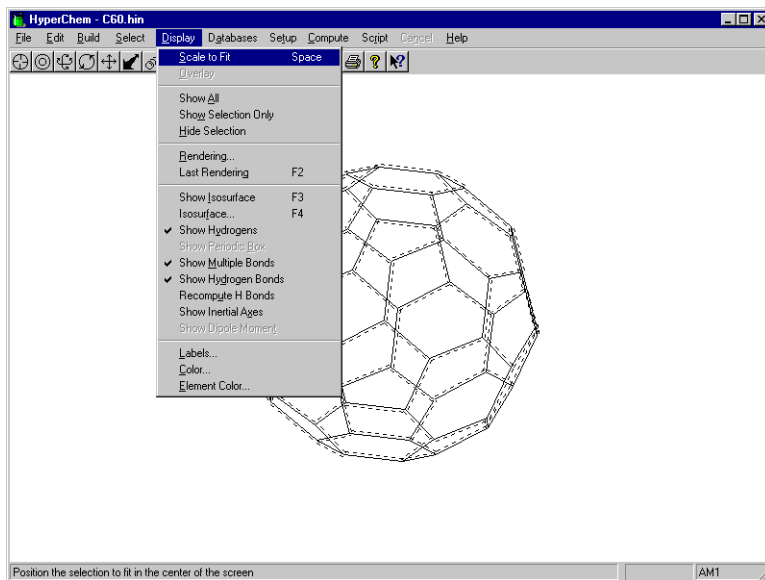
## Using Display Settings

HyperChem automatically uses the display settings from the last session. You choose the display settings by using menu items on the Display menu.

To open the Display menu:

1. L-click on Display on the menu bar.

The Display menu opens:



## Using Labels

If the molecule is displayed with labels, remove them using steps 1 and 2. Otherwise, skip to “To use labels.”

To remove the labels:

1. L-click on Labels.

The Labels dialog box appears. Even if the currently displayed molecule is labeled, the dialog box opens with the default setting of None for both atom and residue label types.

2. L-click on OK.

The dialog box closes and the labels are removed.

To use labels:

1. L-click on Labels.

The labels dialog box appears.

2. Choose Symbol as the atom label type, and then choose OK.

The dialog box closes and the molecule is labeled by atomic symbol.

3. Choose Labels on the Display menu.

Although the molecule is labeled, the dialog box opens with the default setting of None for both atom and residue label types.

4. L-click on OK.

The dialog box closes and the labels are removed.

## Using Different Molecular Renderings

You can use various molecular renderings when you display a molecular system.

To change the molecular rendering:

1. Choose Rendering on the Display menu.

The Rendering dialog box opens:



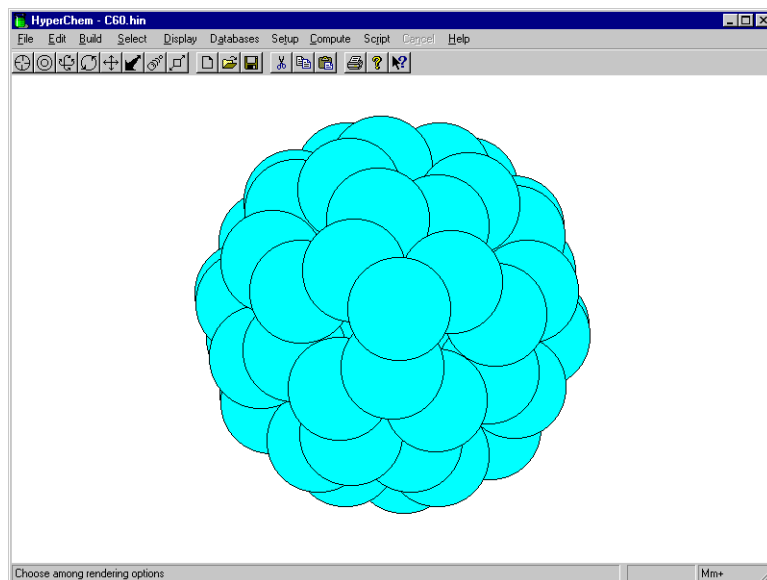
2. Choose Balls in the list, and then click on the “tab” at the top labelled Balls to show the Balls Options property sheet:



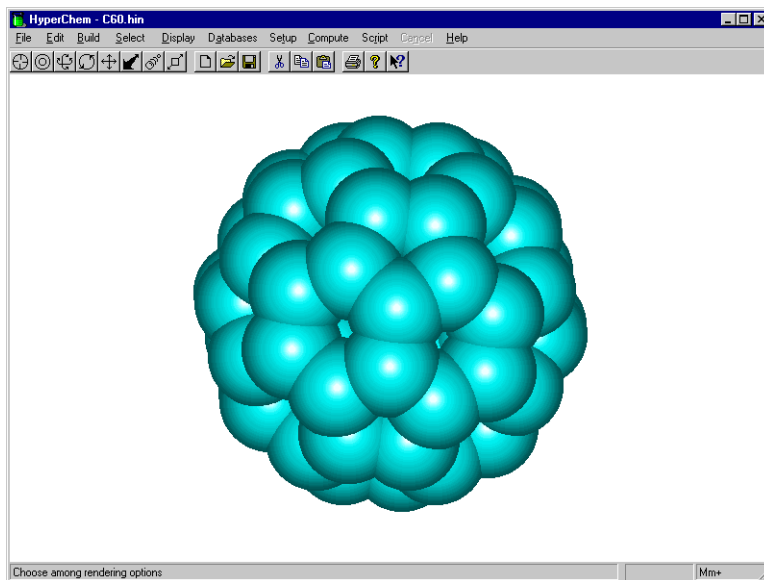


3. If Shading is turned on (marked with a ✓), click on it to turn the option off (no ✓). Then choose OK.

This gives a crude, but quickly drawn, space-filled representation of the molecule. The molecule is drawn using nonintersecting, nonshaded circles.

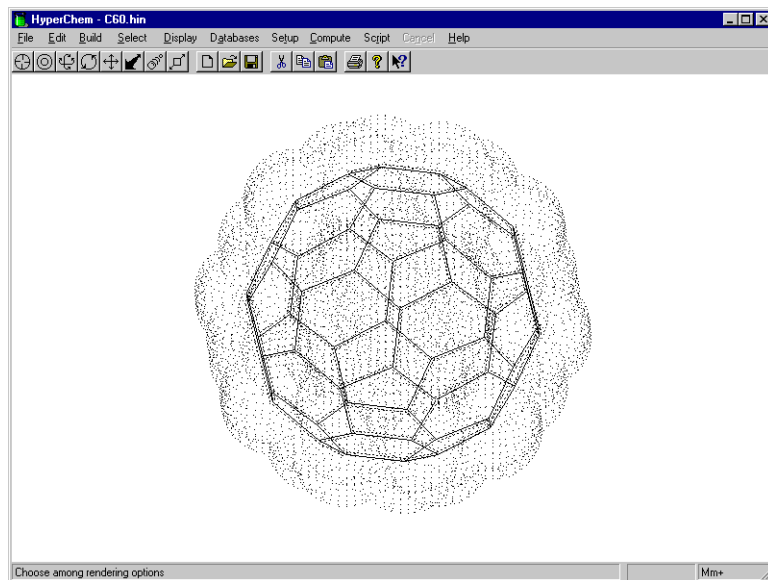


4. Reopen the Rendering dialog box. Choose Spheres, and then choose OK.



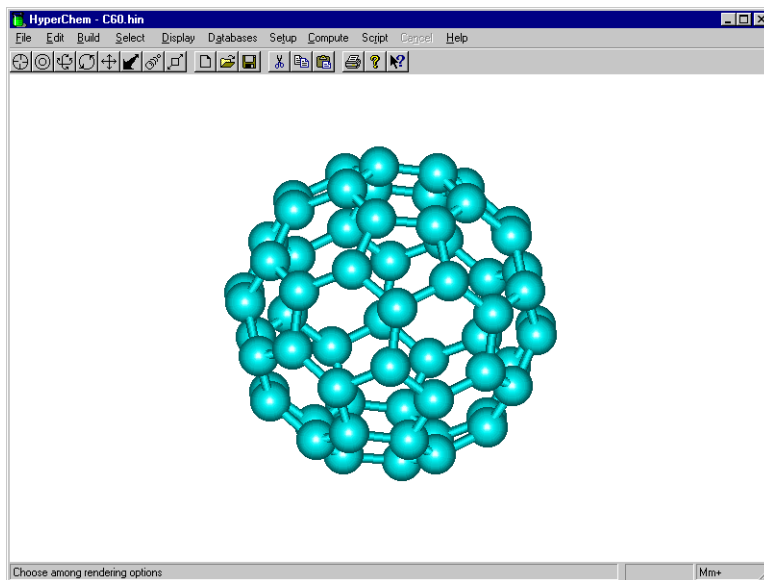
This gives a space-filled representation that simulates CPK models. It uses shaded spheres to represent atoms and calculates intersections between bonded spheres.

5. Choose Rendering on the Display menu to open the dialog box once more.
6. Choose Sticks & Dots. Then choose OK to close the dialog box.



This representation gives you a good idea of the shape and volume occupied by the molecule.

7. Choose Rendering on the Display menu to open the dialog box once more.
8. Choose Balls and Cylinders. Then go to the Balls property sheet by clicking on the Balls tab at the top of the dialog box, and turn on the Shading and Highlight options (both marked with ✓). Then choose OK to close the dialog box.



9. Press the [F2] key to restore the previous rendering options. This is the equivalent of selecting Last Rendering on the Display menu, and returns the display to the sticks and dots that were used before the most recent change to the rendering options.
10. Open the Rendering dialog box again, choose Sticks, and then choose OK.

The molecule appears in its original stick rendering.

## Exiting HyperChem

For now, exit HyperChem.

To exit HyperChem:

1. Select Exit on the File menu.

If the program warns you about saving changes to the file, click on "No".

The window closes.

## **For More Information**

For more detailed information on how to get started with HyperChem, see Chapter 1, "Introduction," of the *HyperChem for Windows Reference Manual*.

For information on how to use Microsoft Windows, see Help on the Start button.

# Lesson 2

## Basic Drawing and Editing Techniques

### Skills Covered in This Lesson

- Drawing atoms and bonds
- Selecting atoms and bonds
- Deleting, clearing, and copying

In the previous lesson, you read in a molecule from a HIN file. Lesson 2 teaches you basic drawing and editing techniques that you use later to create a 2D sketch of a molecule.

### Reopening HyperChem

To reopen HyperChem:

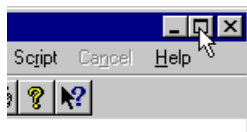
1. Point to the HyperChem icon in the WorkSpace.
2. Double-click on the icon.

The HyperChem window opens.

### Expanding the HyperChem Window

To expand the window to full-screen display:

1. L-click on the Maximize button in the top-right corner of the HyperChem window.



Now you have a larger workspace to draw in.

## Drawing Atoms

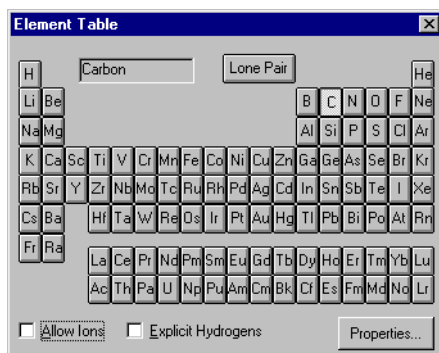
To draw an atom:

1. Open the Element Table dialog box.

There are two ways to do this:

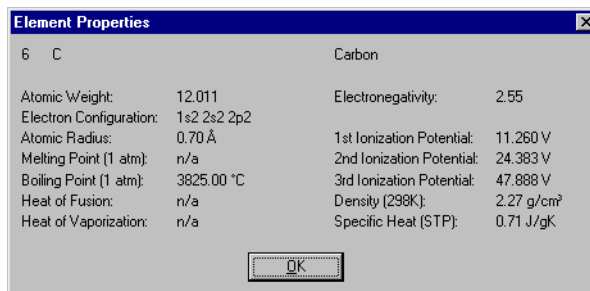
- Select Default Element on the Build menu, or
- Double-click on the Drawing tool.

The Element Table dialog box appears:



The Default Element dialog box lets you choose the default element from the periodic table.

2. If you click on the Properties... button, a box showing the physical properties of the currently-selected element are displayed. Or, if you shift-click on an element button, the properties for that element are displayed. Try this for carbon (C):



Then click on 'OK' to make the box disappear.

- If Allow Arbitrary Valence or Explicit hydrogens is on (marked with a check: ✓), L-click on these options to turn them off.
- Select Carbon in the list of default elements, then close the element dialog box.

The dialog box disappears and the default element is set to carbon.

*Note:* You can leave the Default Element dialog box open and move it so you can see the HyperChem workspace. This is useful when you want to draw molecules with many heteroatoms.

- L-click on the Drawing tool and move the cursor into the workspace.

The cursor changes to the shape of the Drawing tool, indicating you are in drawing mode.

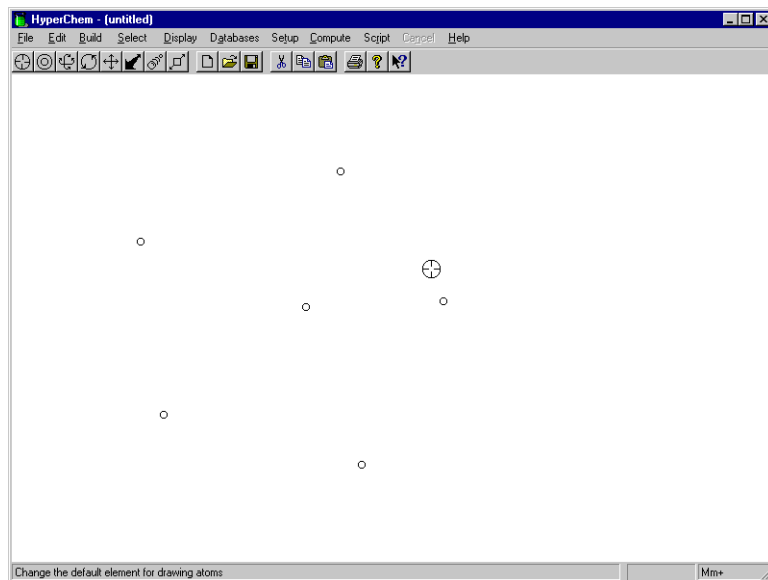
- L-click in the lower-left corner of the workspace.

A small circle appears representing an unbonded carbon atom.

- Draw several more atoms at various locations in the workspace.

Your workspace should look something like this:





## Drawing Bonds

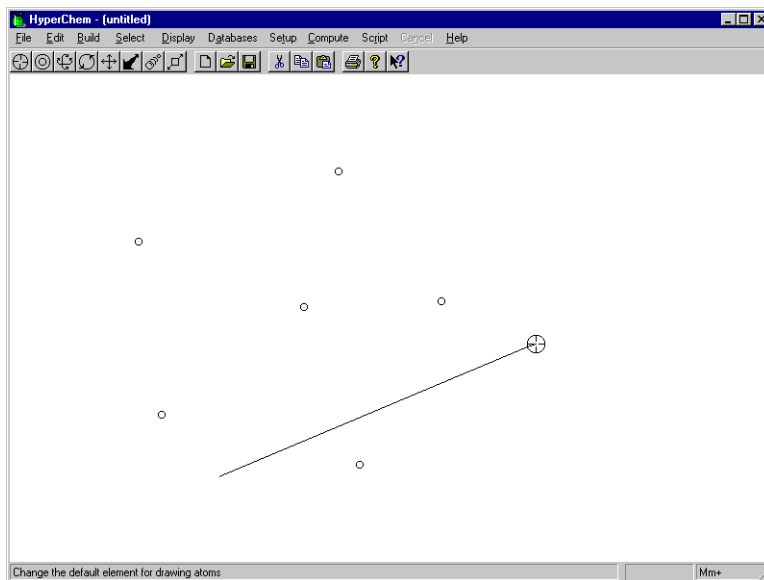
Normally, instead of drawing unbonded atoms, you draw bonded atoms:

To draw a bond:

1. Move the cursor above the first carbon you drew.
2. Press and hold down the left mouse button.

This is the location of the first atom of the bond you are drawing.

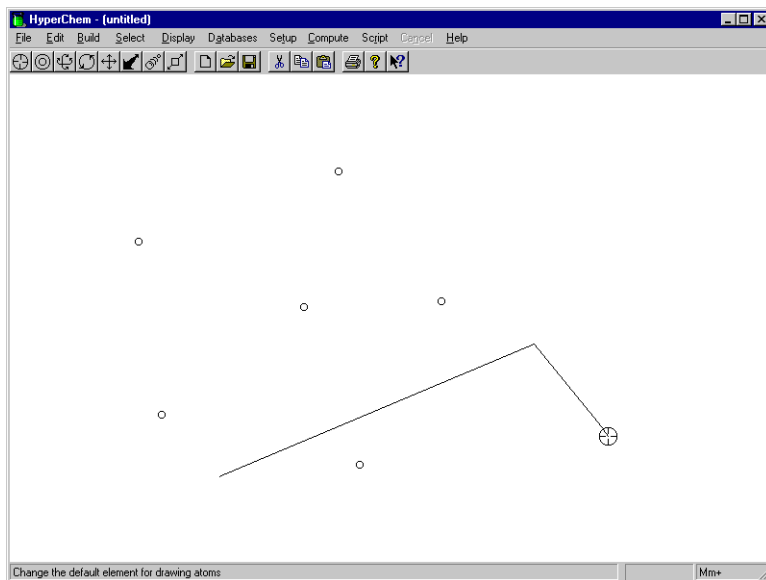
3. Continue to hold down the left mouse button and drag (L-drag) diagonally toward the top of the workspace.



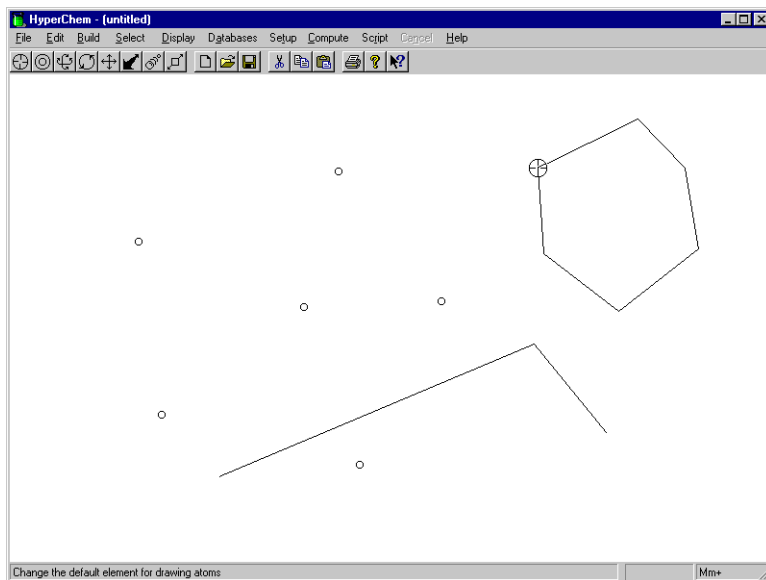
4. Release the mouse button.

This is the location of the second atom in the bond. A line representing a bond between two carbon atoms appears.

5. With the cursor still on the top end of the bond, L-drag diagonally toward the lower-right corner of the workspace.



6. Release the mouse button.  
This is the location of the third atom.
7. Starting in an empty area of the workspace, draw six bonds to form a ring.



Now that you know how to draw atoms and bonds, you learn basic selection techniques.

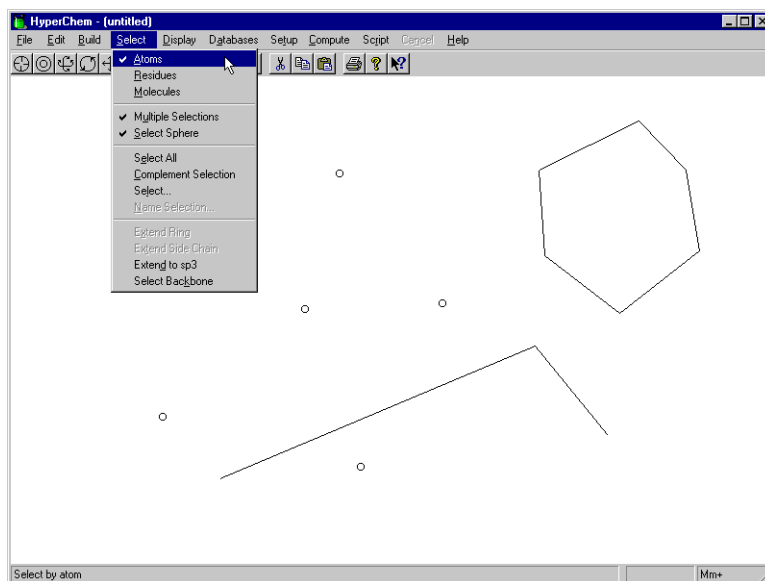
## Selecting Atoms

In this exercise, you learn basic selection techniques by selecting atoms. First, you must set the level of selection (atoms, residues, or molecules) that you want.

To set the selection level to atoms:

1. L-click on the Select menu.

The Select menu opens.



2. If Atoms is not selected (marked with a check: ✓), L-click on Atoms to select it.

Next, make sure Multiple Selections is off.

To turn off Multiple Selections:

1. L-click on the Select menu.

2. If Multiple Selections is on (marked with a check: ✓), L-click on it to turn it off; otherwise press [Esc] or L-click on the title bar to close the Select menu.

Now, when you select an atom or group of atoms, the previous selection is canceled.

To select atoms:

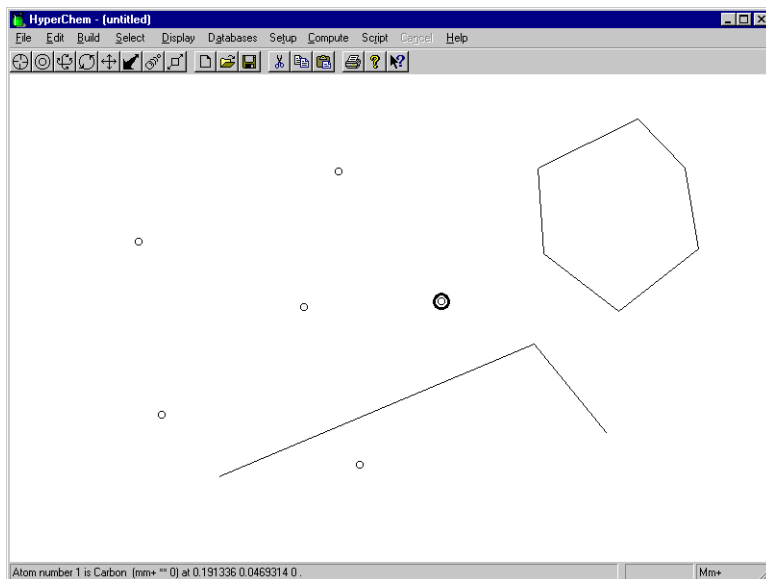
1. L-click on the Selection tool and move the cursor into the workspace.



The cursor assumes the shape of the Selection tool. This means HyperChem is in selection mode.

2. L-click on one of the atoms you created.

This highlights the atom. When you select items, they are highlighted in the workspace.



**Note:** You can set the type of highlight you want in the Preferences dialog box. Preferences is on the File menu.

3. L-click on the middle of one of the bonds you created.

The bond is highlighted.

The atoms at each end of the bond are selected, highlighting the line that represents the bond.

4. R-click in an empty area of the workspace.

All atoms and bonds are deselected. You learn more about how to deselect atoms later.

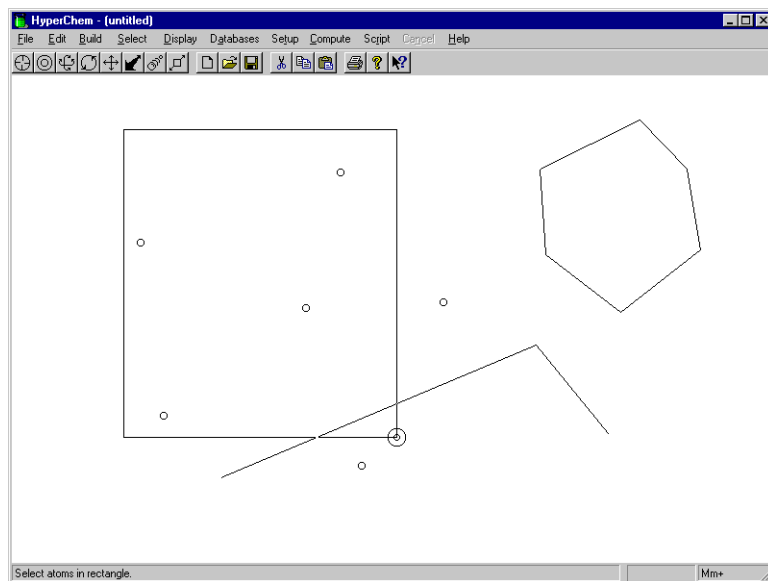
## Selecting Groups of Atoms

To select a group of atoms using rectangular selection:

1. On the Select menu, make sure that Select Sphere is not turned on.
2. Pick a point in an empty part of the workspace toward the top-left corner.
3. LR-drag diagonally, toward the lower-right corner of the workspace.

HyperChem displays a rectangle representing the border of the selection area.

4. Continue LR-dragging until a few atoms are enclosed in the rectangle, like this:



5. Release the mouse buttons.

All atoms in the selection area are highlighted.

To select a second group of atoms:

1. Select Multiple Selections on the Select menu.

When Multiple Selections is turned on, new selections are added to previous selections.

2. LR-drag as you did before to select a second group of atoms.
3. Release the mouse buttons and the second group is highlighted.

**Note:** To select all atoms in the workspace, L-click in an empty area of the workspace when you are in selection mode.

To deselect atoms:

1. R-click on an unbonded atom.

The atom is no longer highlighted.

2. R-click on the middle of a selected bond.

The atoms at the end of the bond (and any bonds to them) are no longer highlighted.

To deselect all selected atoms:

1. R-click in an empty area of the workspace.

The rest of the selected atoms are deselected.

In Lesson 7, “Selecting and Displaying Subsets,” you learn more about selecting.

## Deleting Atoms

To delete a single atom or a bond:

1. L-click on the Drawing tool to get into drawing mode.
2. R-click on the atom you want to delete, or on the middle of the bond you want to delete.

The atom or bond disappears.

To delete multiple atoms or bonds:

1. L-click on the Selection tool to get into selection mode.
2. LR-drag to select two or three objects in the workspace.
3. Select Clear on the Edit menu.

A dialog box appears asking if you want to delete the selection.

4. Choose Yes.

## Copying Atoms to the Clipboard

To copy:

1. L-click on the Selection tool to get into selection mode.
2. L-click on a bond or an atom.
3. Select Copy on the Edit menu.

A copy of the atom or bond you selected is saved on the Clipboard.

4. Select Paste on the Edit menu.

A copy of the bond or atom is pasted into the workspace.



## Clearing the HyperChem Workspace

To clear the workspace:

1. Select New on the File menu.

A window appears asking if you want to save the current changes.

2. Choose No.

HyperChem clears the remaining atoms and bonds from the screen.

## For More Information

For more information on basic drawing and editing techniques, see Chapter 5, “Constructing Molecules,” in the *HyperChem Release 5 for Windows Reference Manual*.

# Lesson 3

## Creating Small Molecules in 2D and 3D

### Skills Covered in This Lesson

- Drawing a 2D sketch of a molecule
- Editing bonds and atoms
- Using the 3D Model Builder
- Saving a structure

Now that you can draw atoms and bonds, you are ready to draw a 2D sketch of a molecule. Although HyperChem lets you draw molecules of any size, for practical reasons we recommend you limit these drawings to small and medium-size molecules.

In this exercise, you draw the backbone of 1-hydroxy-3-phenyl-2-propene. Only the connectivity of the molecule is relevant. Don't worry about angles between bonds or the length of the bonds you are drawing.

After you draw the 2D sketch, you modify the drawing and use the HyperChem Model Builder to produce a 3D representation of the molecule.

### Reopening HyperChem

1. Reopen HyperChem, if necessary.
2. Expand the HyperChem window to full size.

## Drawing a 2D Sketch

To draw the 2D sketch:

1. Double-click on the Drawing tool.

The Element Table dialog box opens.

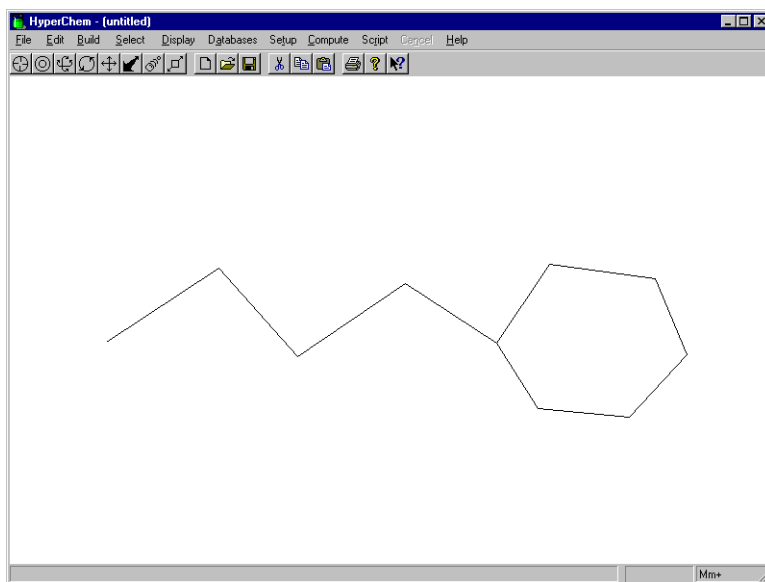
2. Turn Allow Arbitrary Valence on and Explicit Hydrogens off.

When you draw with Explicit Hydrogens turned off, hydrogen atoms are not automatically added.

3. Choose Carbon, then close the dialog box.

Carbon is set as the default element and the dialog box closes.

4. Draw the following:



## Saving Your Work in a File

Before you modify this sketch, save your work. This way, you won't have to redraw the sketch if you make a mistake when you modify it.

To save your work:

1. Select Save on the File menu.

Because this is the first time you have saved this sketch, HyperChem displays the Save File dialog box.



2. If you wish, select a directory to save the file in by L-clicking on the directory folders.
3. Make sure HIN is selected in the File type box.
4. Move the cursor into the File Name field.

Notice that the cursor changes from an arrow to a vertical line.

5. L-click in the field
6. Enter **propene**.

The filename propene.hin appears in the text box:



7. L-click on OK.

HyperChem saves your drawing on disk in a file named propene.hin. The filename appears in the title bar at the top of the screen.

The next time you use Save to save this file, the dialog box is not displayed.

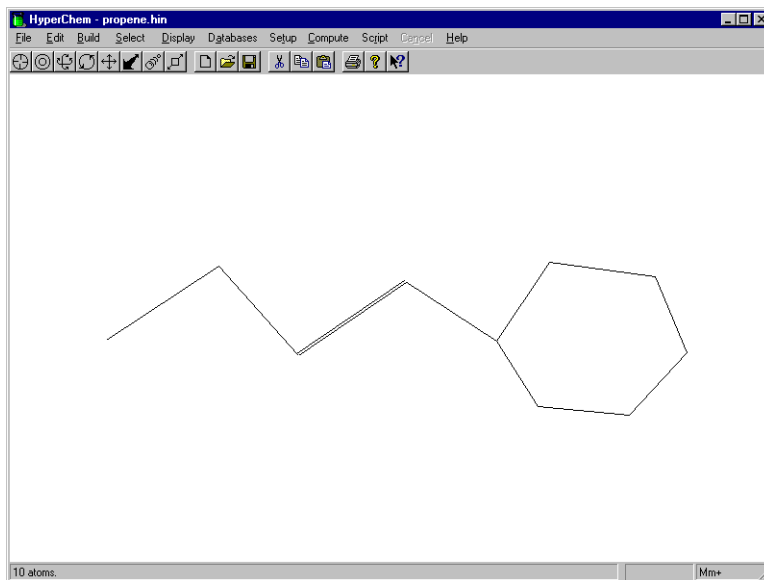
## Changing Bond Orders

You can now modify the sketch to prepare it for the HyperChem Model Builder.

For the structure you are building, you need to increase the bond order of a bond.

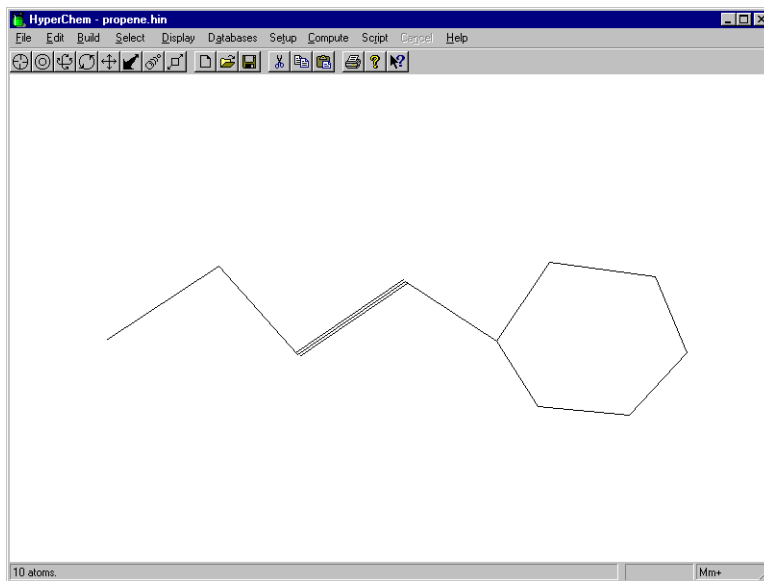
To change bond orders:

1. Point to the middle of the third bond from the left and L-click.  
The bond order increases from a single to a double.



2. L-click on the bond again.

The double bond increases to a triple bond.



3. R-click on the bond.

R-click reduces the bond order.

4. R-click again on the bond.

It changes back to a single bond.

5. The molecule you are building requires one double bond, so L-click on the middle of the bond again.

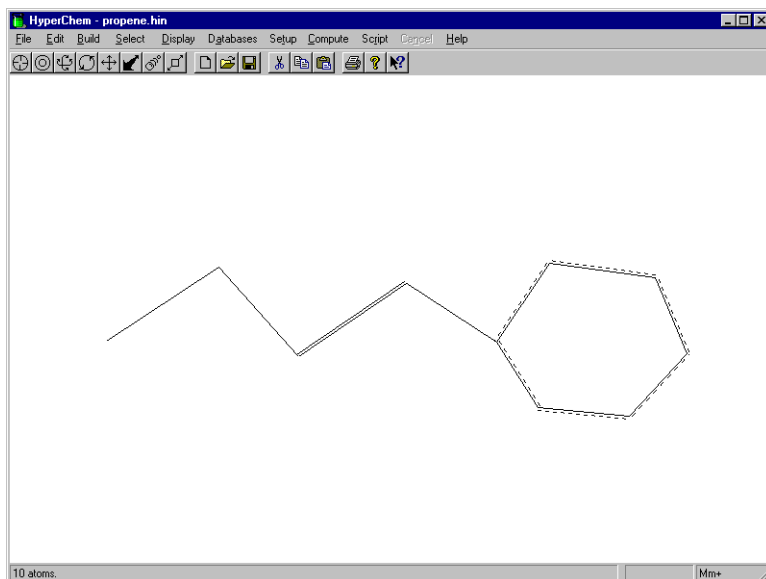
## Making Rings Aromatic

The six-membered ring in the sketch is currently a cyclohexane ring with all single bonds. For the structure you are building, you need to turn it into a benzene ring. To turn the ring into a phenyl substituent, you have to explicitly indicate that the ring is aromatic.

To make rings aromatic:

1. Double-click on any bond of the ring.

The broken lines indicate that the ring is aromatic:



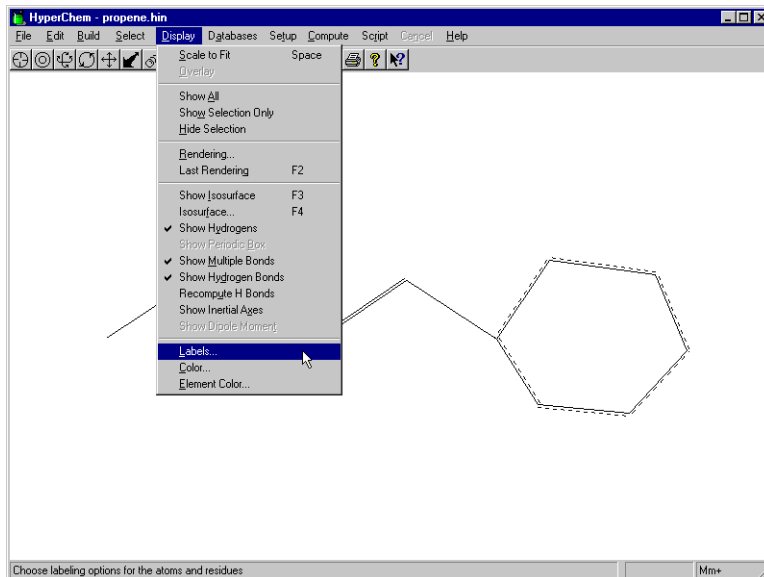
**Note:** If you double-click on a bond that is not in a ring, only the bond becomes aromatic.

# Labeling Atoms

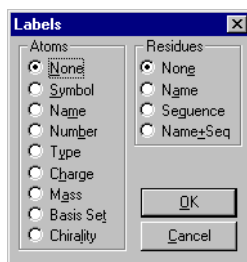
Before you start editing atoms, make it easier to see what you're doing by labeling the atoms.

To label atoms:

1. Select Labels on the Display menu.



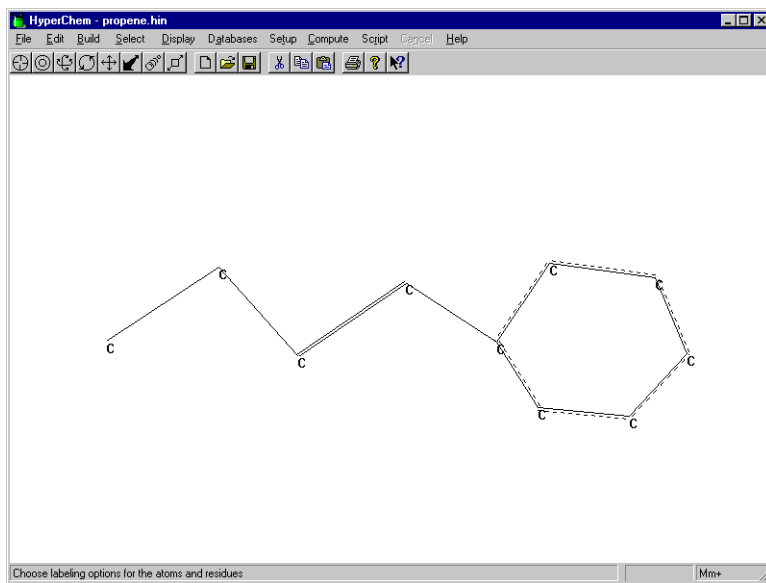
The Labels dialog box appears:



2. In the Atoms option box, L-click on Symbol.  
The radio button next to Symbol is filled in.
3. L-click on OK.



The Labels dialog box disappears and all atoms are labeled with the atomic symbol C.



## Editing Individual Atoms

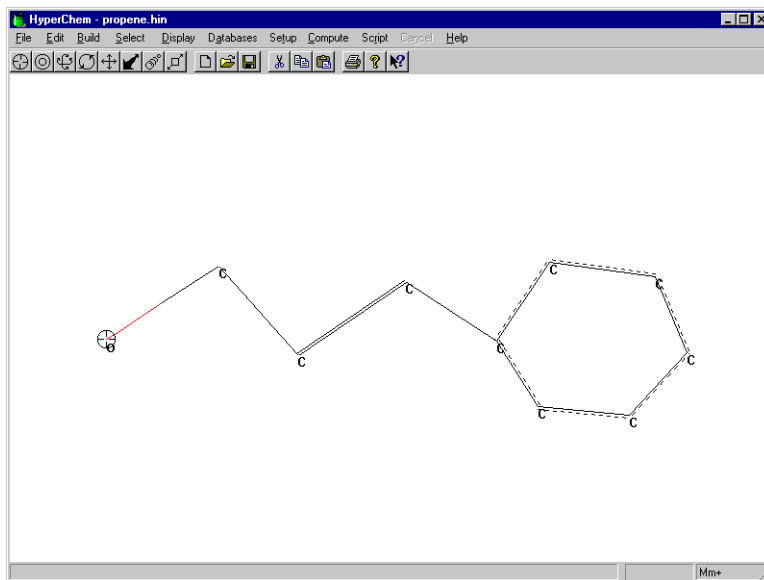
In Lesson 1 you set the Default Element to carbon, so that all atoms in the drawing are carbon. To correct the topology of this molecule, you need to change one of these carbon atoms to oxygen.

To edit individual atoms:

1. Double-click on the Drawing tool to open the Element Table dialog box.
2. Choose Oxygen, then close the dialog box.

The dialog box disappears and oxygen is set as the new default element.

3. L-click on the carbon atom at the left end of the aliphatic chain.



The atom symbol C changes to an O and the oxygen end of the bond turns from cyan to red (if you are using the default colors).

The 2D drawing of 1-hydroxy-3-phenyl-2-propene is complete.

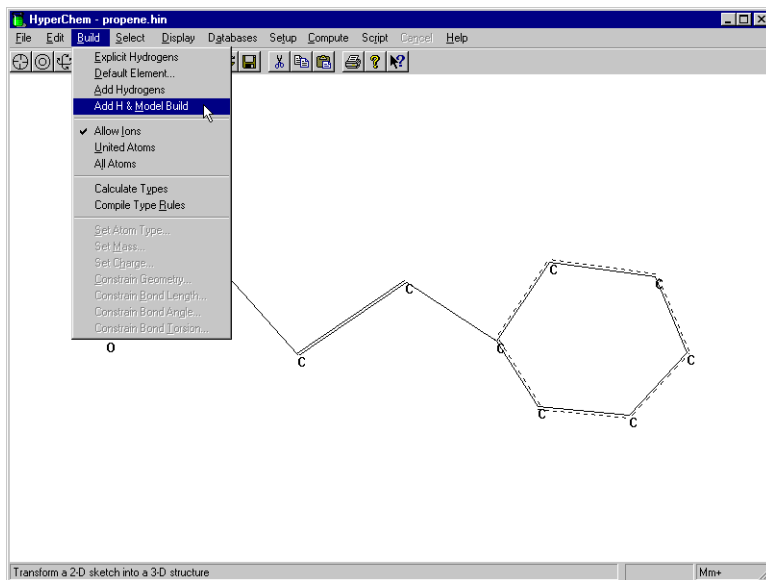
See “Element Color, Display menu” in the *HyperChem for Windows Reference Manual* for information on the default element colors.

## Invoking the Model Builder

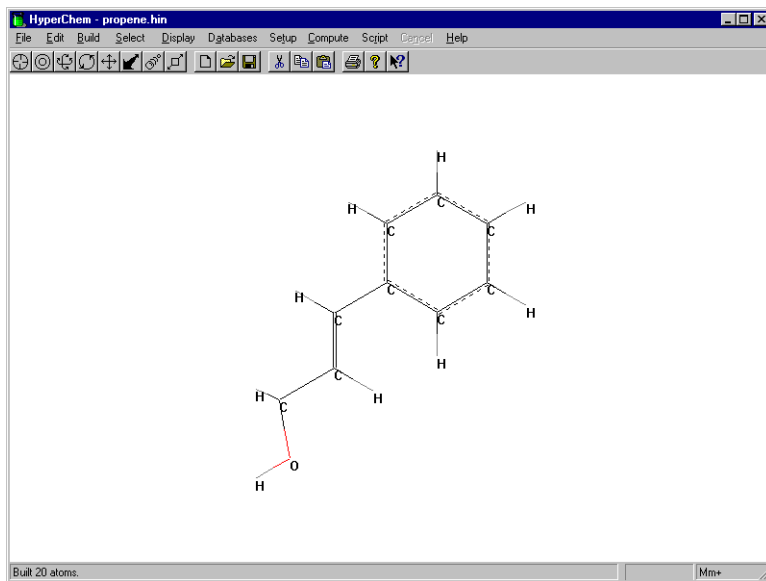
To convert your 2D sketch into a 3D structure, you use the HyperChem Model Builder.

To invoke the Model Builder:

1. Select Add H and Model build on the Build menu.



HyperChem produces a 3D representation of the molecule and automatically adds hydrogens:



2. If hydrogens are not displayed, select Show Hydrogens on the Display menu.

## Saving the Structure

To save the structure:

1. Select Save on the File menu.

HyperChem saves the new 3D structure in the file propene.hin.

## For More Information

For more information on creating small molecules in 2D and 3D, see Chapter 5, “Building Molecules,” and Chapter 8, “Constructing Molecules,” in the *HyperChem for Windows Reference Manual*.



# Lesson 4

## Translating, Rotating, and Scaling Molecules

### Skills Covered in This Lesson

- Translating about the x, y, and z axes
- Rotating about the x, y, and z axes
- Zooming in and out on molecules
- Changing the Z clipping slab

Lesson 4 explains moving molecules using the Rotation, Translation, and Zoom tools.

### Reopening a HIN file

If the file you saved in Lesson 3, propene.hin, is still displayed in the workspace, skip to the next section. Otherwise, reopen the file.

To reopen the file:

1. If necessary, reopen HyperChem.
2. Choose Open on the File menu.  
The Open File dialog box appears.
3. Using the Directories list box, move to the directory in which you stored propene.hin in Lesson 3.
4. Scroll through the list of files and double-click on propene.hin.

Notice when you double-clicked on the filename that you didn't need to L-click on OK.

The dialog box disappears. After a few moments, the molecule appears in the workspace.

To remove atom labelling:

1. Select Labels on the Display menu.

The Labels dialog box opens.

2. Select None as the atom label, and then choose OK.

This makes the screen less cluttered.

## XY Translation

You use the XY translation tool to move molecules or selected atoms along the plane of the computer screen.

You can control this tool different ways, just as you can most tools in the tool bar.

In this exercise, you do XY translation first by L-click-dragging in the workspace, and later by specifying a value in the Translation dialog box.

To use the XY translation tool:

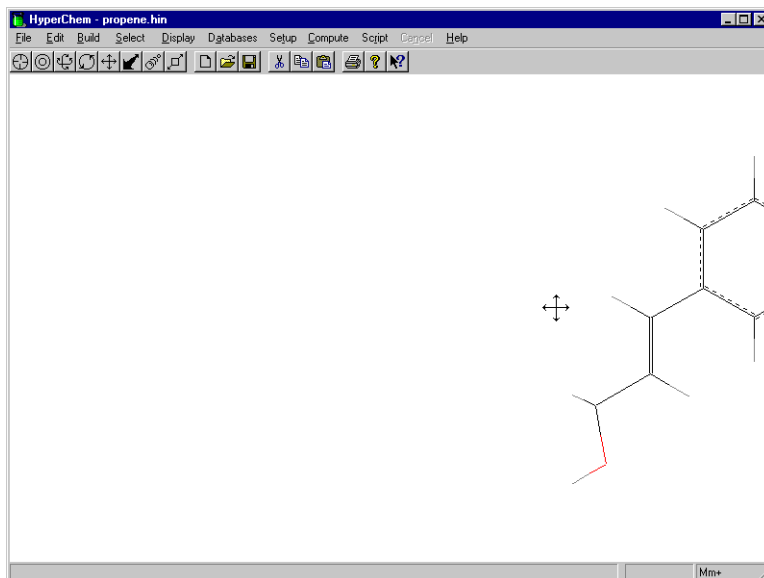
1. L-click on the XY translation tool.



2. Move the cursor into the workspace.

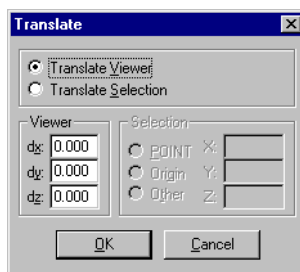
The cursor assumes the shape of the XY translation tool.

3. L-drag until the molecule looks like this:



4. Open the Translate dialog box by either
  - double-clicking on the XY translation tool, or
  - selecting Translate on the Edit menu.

The Translate dialog box appears:



5. L-click in the dx text box and enter **-5.0**.
6. L-click in the dy text box and enter **0.0**.
7. Press the [Tab] key.

You can skip to the next text box by pressing the [Tab] key.

8. Enter **0.0** in the dz text box and L-click on OK.



The molecule moves to the left side of the screen by 5.0 Ångstroms.

## Z Translation

In the previous exercise, you moved the molecule back and forth across the plane of the computer screen by L-dragging the XY translation tool across the workspace and by opening the Translation dialog box.

In this exercise, you move the molecule toward and away from you (along the z axis) using the Z translation tool.

To use the Z translation tool:

1. On the Display menu, choose Rendering and make sure Perspective is on in the Rendering dialog box.

With Perspective view, all molecules appear to be in perspective: closer atoms appear larger. You must turn Perspective on before you use the Z translation tool to see any effect.

2. Choose OK to close this dialog box.
3. L-click on the Z translation tool.



4. Move the pointer into the workspace.  
The pointer assumes the shape of the Z translation tool.
5. L-drag slowly toward the bottom of the workspace.  
The molecule moves closer to you.
6. Continue moving the molecule until it completely disappears.  
The molecule no longer appears because you have moved it out of the Z clipping slab. Later on you learn how to adjust the Z clipping plane.
7. Open the Translate dialog box by either
  - double-clicking on the Z translation tool, or
  - selecting Translate on the Edit menu.

8. In the dx and dy text box, enter **0**.
9. In the dz text box, enter **-5** and choose OK.

HyperChem moves the molecule 5Å away from you and the molecule reappears in the workspace. (You may need to repeat this step to get the molecule to reappear, if you had moved it much closer to you in step 5.)

10. Press the [Spacebar] to re-center the molecule in the workspace.

## Using the Zoom Tool

To increase or decrease the magnification of a molecule, use the Zoom tool. This is useful, for instance, when you want to examine structural details of a large molecule.

So far you have controlled tools in the tool bar by L-dragging in the workspace, or by specifying a particular value in the dialog boxes.

You can also control tools, such as the Zoom tool, by using the [Shift] key and the tool icon.

To use the Zoom tool:

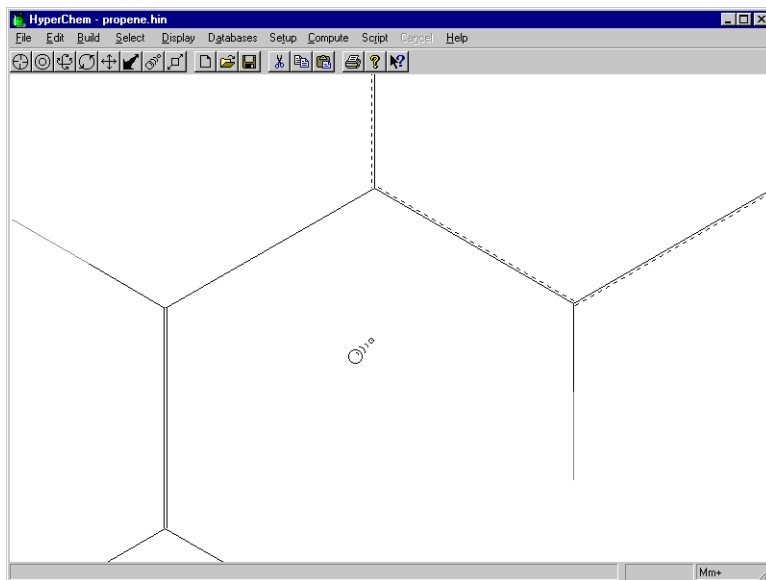
1. Hold the [Shift] key down and L-click near the top right of the Zoom tool icon.



The molecule gets smaller.

2. Continue holding down the [Shift] key and L-click near the bottom left of the icon.

Zoom in on the molecule until it looks like this:



Remember, you can also control the Zoom tool by

- using the Zoom dialog box, or
- L-dragging the Zoom tool in the workspace.

## Centering and Scaling

When a molecular system is first loaded from the file system, it is centered in the workspace in a standard size. After you use the Translation or Zoom tools on the molecule, you can return it to its original size and center it in the workspace. Scale to Fit uses the current selection, or the whole system if there is no selection.

To center and scale the molecule:

1. Select Scale to Fit on the Display menu, or alternatively, press the space bar.

The molecule returns to its original size and is positioned at the center of the workspace.

**Note:** You can get the same zoom factor as Scale to Fit without centering the system by entering a value of 1 in the Zoom dialog box.

## XY Rotation

To rotate one or more molecules around the x or y axis or both axes, use the XY rotation tool.

To rotate the molecule along the x and y axes:

1. L-click on the XY rotation tool.



2. Move the cursor into the workspace.
3. Use the Labels dialog on the Display menu to add symbols to the atoms, for clarity.

4. L-drag horizontally.

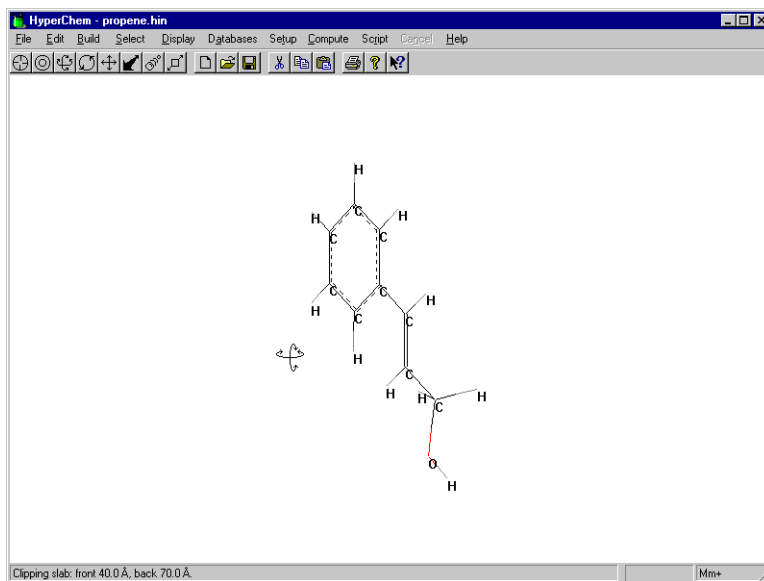
The molecule rotates around the y axis.

5. L-drag vertically.

The molecule rotates around the x axis.

6. L-drag diagonally to rotate the molecule around both x and y axes.

7. Rotate the molecule until it looks like this:



## Z Rotation

With the Z rotation tool, you can rotate molecules around the z axis.

To rotate the molecule along the z axis:

1. L-click on the Z rotation tool.



2. L-drag horizontally to the right in the workspace.

The molecule moves in the clockwise direction.

3. L-drag horizontally to the left.

The molecule moves in the counterclockwise direction.

*Note:* L-click-dragging vertically has no effect.

## Z Clipping

The Z clipping tool helps you look inside molecules by showing you atoms within a “slice” of a molecule.

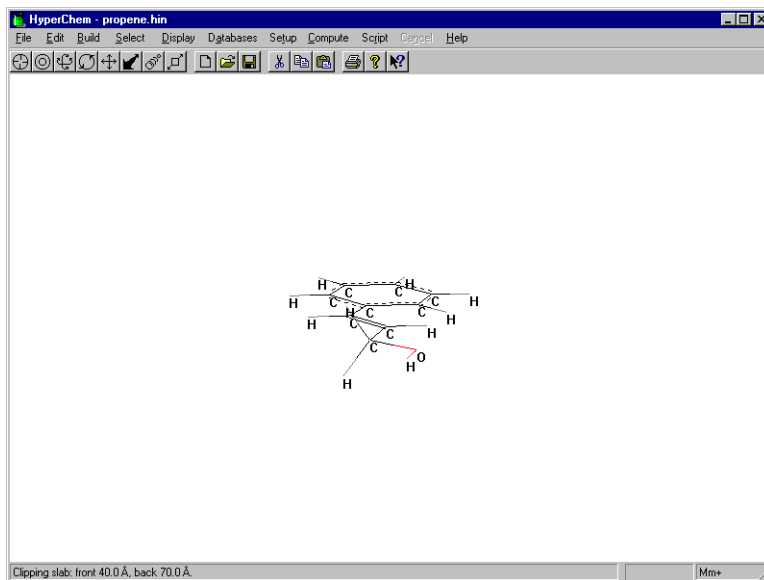
The “slice” or *clipping slab* for the molecular system has a front and back plane that are parallel to each other and the computer screen. Only atoms between the planes are visible.

The default clipping slab sets the clipping planes so that you can see the complete molecule or molecular system.

To help you see where the clipping slab is, the Z Clip dialog box shows you a view from above the system with the clipping planes behind and in front of the molecule. It also lets you change the values interactively or set explicit values for both planes.

To change the clipping slab using the Z Clip dialog box:

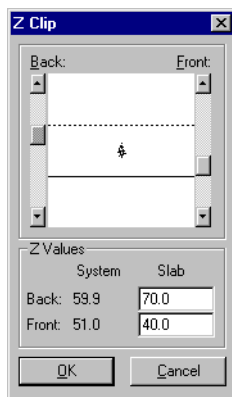
1. First, rotate the molecule so that the benzene ring is oriented either towards or away from the front clipping plane, like this:



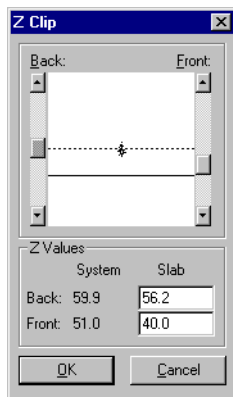
2. Double-click on the Z clipping tool.



The dialog box appears:



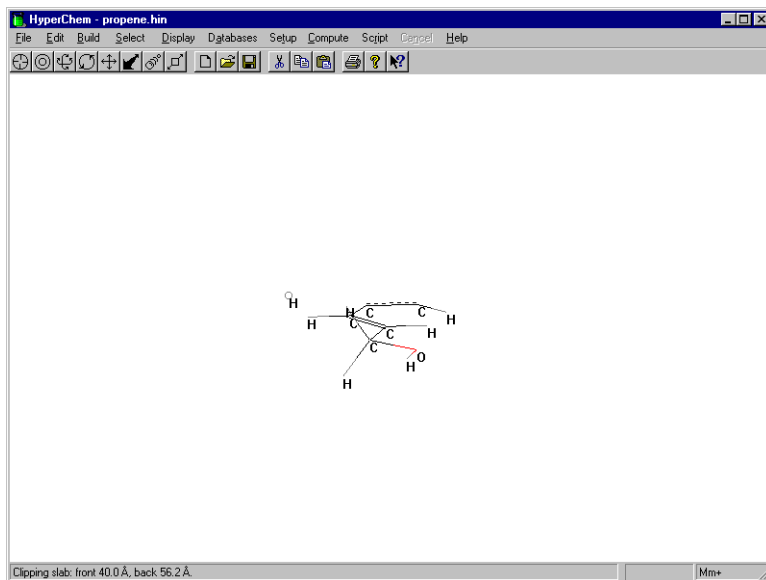
3. Point to the scroll box that controls the back clipping slab.
4. L-drag the scroll box down until a back part of the molecule is clipped.



When you drag the scroll box, the value changes in the Slab text box for the back clipping plane.

5. L-click on OK.

The dialog box disappears. The back part of the molecule is clipped in the workspace.



6. Double-click on the Z clipping tool to open the Z Clip dialog box again.
7. Point the cursor on the up arrow in the scroll bar for the back clipping plane.

8. L-click on the arrow until the plane is behind the molecule, and then choose OK.

The whole molecule is visible again.

You can also change the clipping slab without using the Z Clip dialog box.

To change the clipping slab using the mouse:

1. L-click on the Z clipping tool.
2. L-drag up in the workspace.

In the status line, the value of the front clipping plane increases.

3. Continue L-dragging up until the front part of the molecule is clipped.

This happens when the clipping plane reaches about 53Å.

4. L-drag down until the whole molecule is visible again.

The front clipping plane is in front of the molecule again.

## Clearing the Workspace

To clear the workspace:

1. Select New on the File menu.
2. If HyperChem asks if you want to save your changes, choose No.

This choice clears the screen.

## Practice Exercises

Try some of these practice exercises:

1. Rotate and translate a molecule while Show Inertial Axes on the Display menu is on.
2. If you have several molecules on the screen you can select one of them by choosing Molecules on the Select menu, and rotate it alone around its center of mass by using R-drag instead of L-drag.



*For More Information*

3. Select a side chain of a molecule, and use the Z rotation tool to rotate the side chain by using R-drag.
4. Change the Z clipping slab by entering specific values in the Z Clip dialog box.

## **For More Information**

For more information on translating, rotating, and scaling molecules, see Chapter 3, “Tools,” in the *HyperChem for Windows Reference Manual*.

# Lesson 5

## Measuring Structural Properties

### Skills Covered in This Lesson

- Measuring bonds, angles, torsion angles, and nonbonds
- Displaying atom characteristics
- Using builder constraints

Lesson 5 describes measuring structural properties by using different selection techniques. By selecting different parts of a molecule or molecular structure, you can measure bond lengths and angles, as well as display atom characteristics, such as charges and x, y, and z coordinates.

### Creating a 2D Sketch

To get into drawing mode:

1. If necessary, reopen HyperChem.
2. Select the Drawing tool.

To make sure the Default element is carbon:

1. Double-click on the Drawing tool.  
The Element Table dialog box opens.
2. Select Carbon if it is not already selected, and then close the dialog box.

To label the atoms by number:

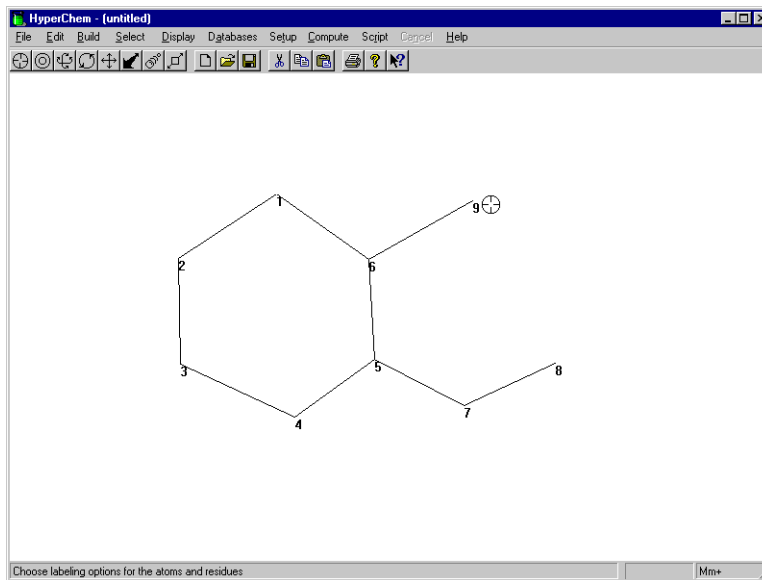
1. Select Labels on the Display menu.

The Labels dialog box opens.

2. Select Number as the atom label, and then choose OK.

This step makes it easier for you to see what you are doing as you draw and edit the 2D structure.

3. Draw the 2D structure in the order shown below:



## Editing the Sketch

To make the ring aromatic:

1. Double-click on the ring.

The ring appears with broken lines.

To edit the atoms:

1. Set the Default Element to nitrogen.
2. L-click on atom #8 to change it into a nitrogen.
3. Set the Default Element to oxygen.
4. L-click on atom #9 to change it into an oxygen.

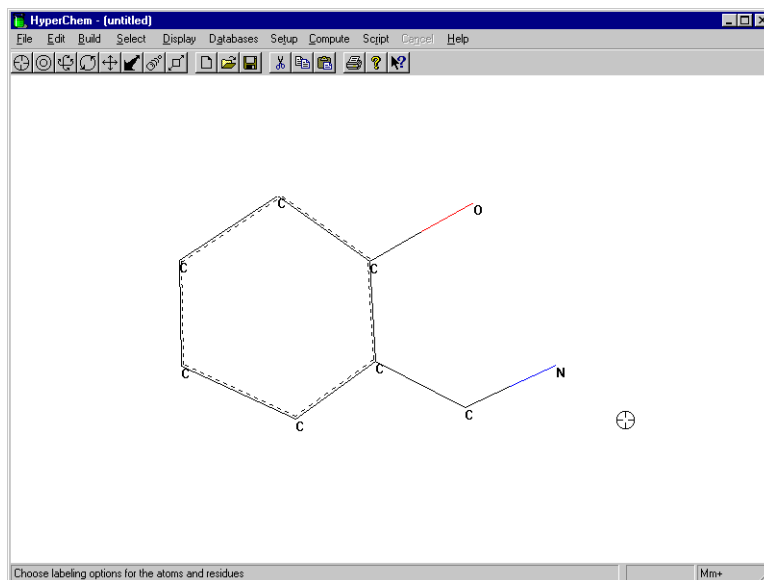
To label the atoms by symbol:

1. Select Labels on the Display menu.

The Labels dialog box opens.

2. Select Symbol as the atom label, and then choose OK.

Your drawing should look like this:



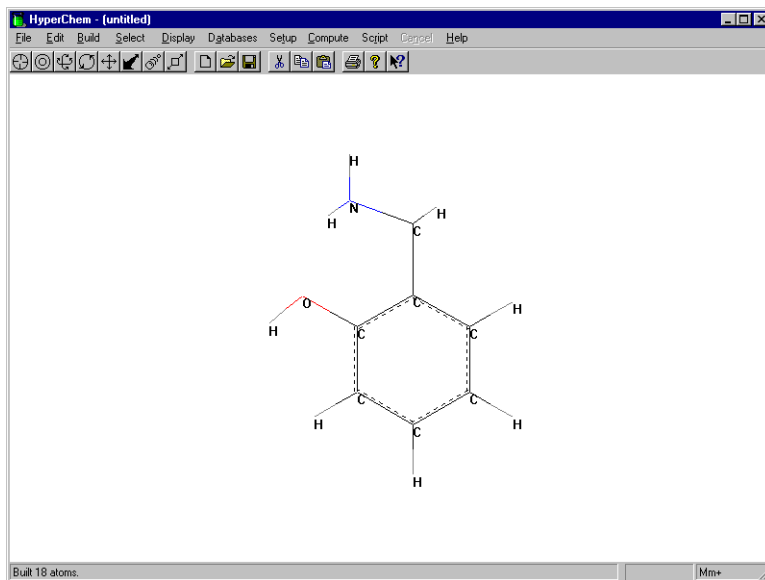
## Building the 3D Structure

To transform the drawing into a 3D structure:

1. Check that Explicit Hydrogens on the Build menu is not selected.
2. Double-click on the Selection tool.

Instead of selecting Add H & Model Build on the Build menu, you can also invoke the Model Builder by double-clicking on the Selection tool.

The structure should look like this:



(Your structure may be oriented somewhat differently from the above, depending on how your original atoms were positioned. The Model Builder sometimes reorients a structure if it differs significantly from the original structure.)

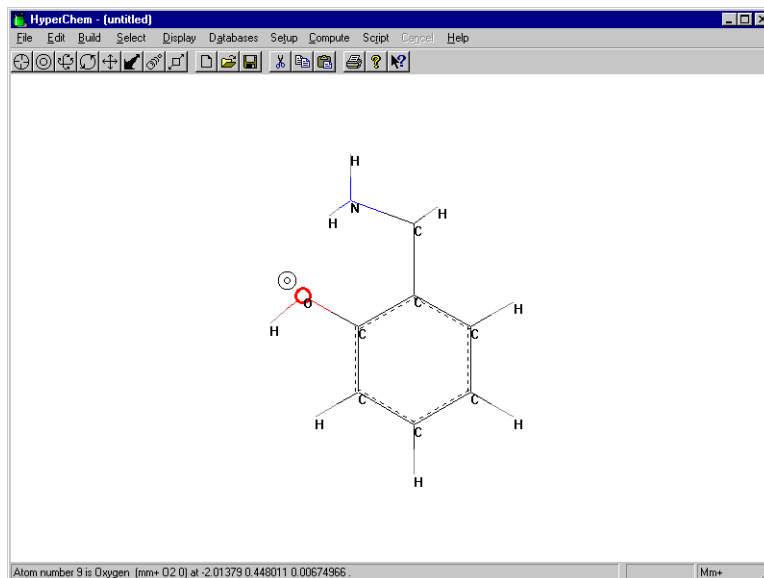
The molecule is built with certain default structural parameters. The following exercises show you how to measure those parameters.

## Atom Characteristics

To get information on atom characteristics:

1. On the Select menu, make sure that Atoms is chosen and that Multiple Selections is not.
2. L-click on the Selection tool.
3. L-click on the oxygen atom.

The selected atom is highlighted and the atom characteristics appear in the status line.



The status line shows the atom number, the atom type, and the charge for the current molecular mechanics force field. It also shows the x, y, and z coordinates of the atom.

The Build menu items Set Atom Type, Set Charge, and Constrain Geometry become active. This lets you specify non-default atomic properties for the Model Builder.

4. Select other atoms and compare atom characteristics shown in the status line.

## Measuring Bond Distances

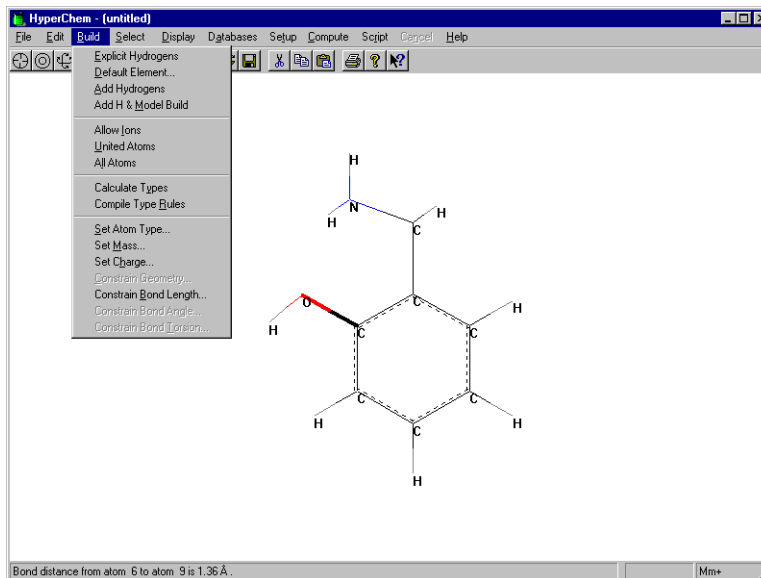
If you select a bond rather than an atom, information on the bond appears in the status line. HyperChem has a library of default bond lengths between atoms of a particular type and hybridization. When a bond length is not in the default library, HyperChem uses an average of the covalent radii of the two atoms for Model Building.

To measure a bond distance:

1. L-click on the carbon-oxygen bond.

The bond is highlighted and the bond distance appears in the status line.

2. L-click on the Build menu.



When you select a bond, the Build menu item Constrain bond length becomes active. This lets you specify non-default bond lengths for the Model Builder.

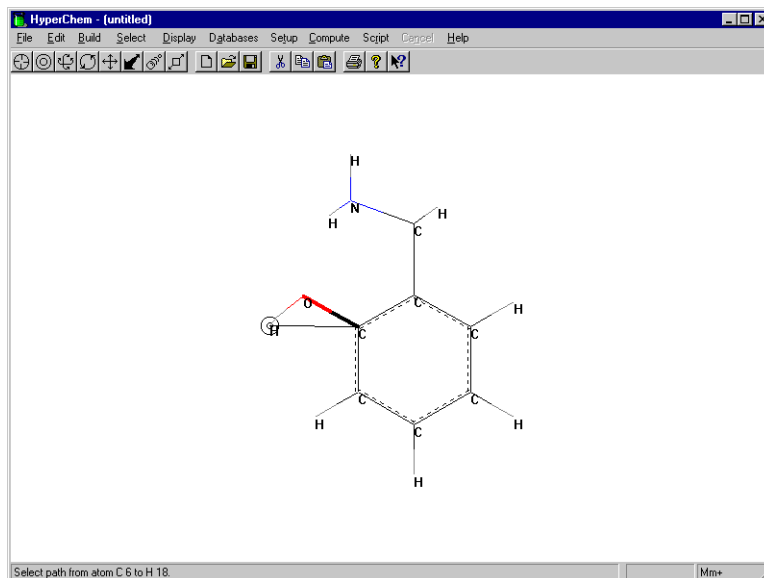
## Measuring Bond Angles

To measure a bond angle, L-drag between the two terminal atoms that are bonded to the common third atom.

To measure a bond angle:

1. L-drag from the carbon atom bonded to oxygen to the hydrogen atom bonded to oxygen.

Note the 'gravity' that attracts the end of the dragged line to nearby atoms.



2. Release the mouse button.

The angle is highlighted and the value of the angle appears in the status line. The default angles chosen by the Model Builder for this structure are based on hybridization and are either tetrahedral (109 degrees), trigonal (120 degrees), or linear (180 degrees).

*Note:* When you select a bond angle, Constrain Bond Angle on the Build menu becomes active. This lets you specify non-default bond angles for the builder.

3. Try measuring other angles in the structure.

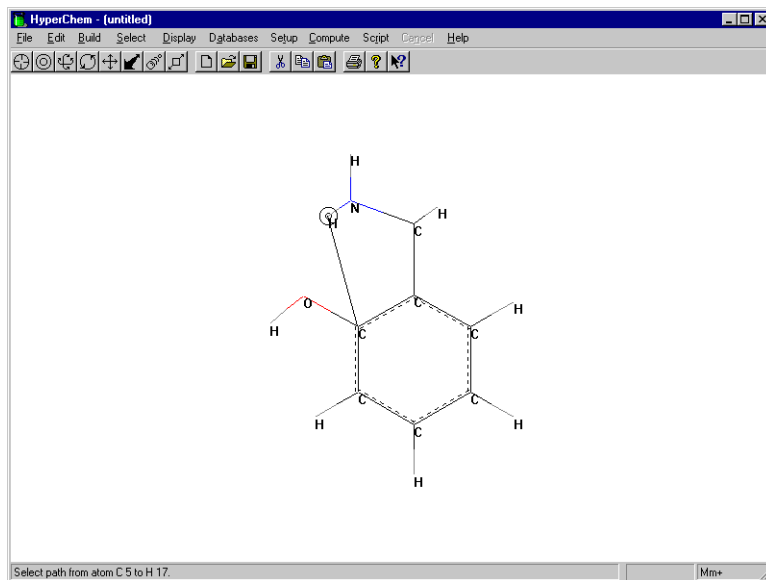
## Measuring Torsion Angles

You measure a torsion angle by dragging the select cursor between the two terminal atoms of a four-atom torsion.

To measure a torsion angle:

1. L-drag between the ring carbon atom and the hydrogen attached to the nitrogen atom to select the torsion shown below:





2. Release the mouse button.

The torsion is highlighted and the status line shows that it has a value of  $-60$  degrees (Gauche).

*Note:* When you select a torsion angle, Constrain Bond Torsion on the Build menu becomes active. This lets you specify non-default bond torsions for the builder.

3. Experiment with L-dragging between third-neighbor (vicinal) atoms and compare values for the angles.

## Measuring Nonbonded Distances

To measure a nonbonded distance:

1. Turn on Multiple Selections on the Select menu.

Before you measure a nonbonded distance, you must turn on Multiple Selections. This lets you select more than one item at a time. Otherwise, when you select a second item, the previous item becomes deselected.

2. R-click in an empty area of the workspace.

All previous selections are deselected.

3. L-click, sequentially, on any two nonbonded atoms.  
The distance appears in the status line.
4. Measure several pairs of nonbonded atoms and compare the distances.

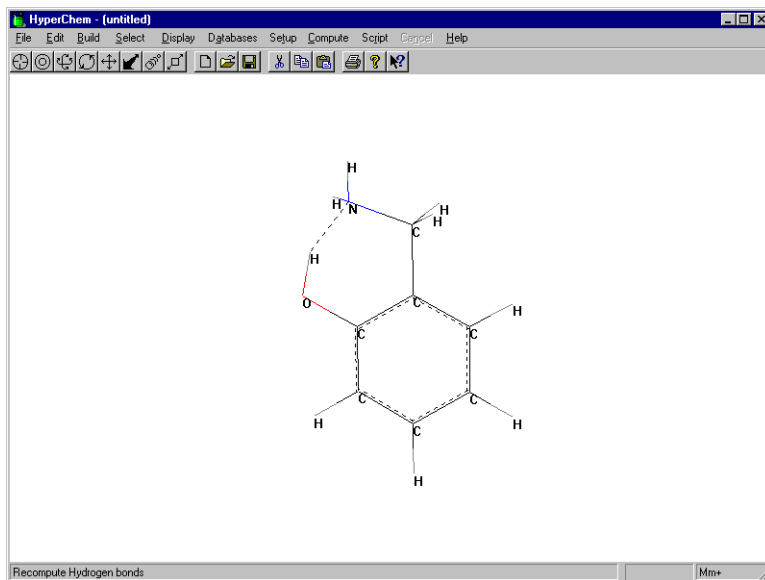
## Hydrogen Bonds

To help you confirm favorable conditions for hydrogen bonding, HyperChem calculates and displays hydrogen bonds.

A hydrogen bond is formed if the hydrogen-donor distance is less than 3.2 Ångstroms and the angle made by covalent bonds to the donor and acceptor atoms is less than 120 degrees. You may need to rotate the  $\text{-NH}_2$  group or the  $\text{-OH}$  group (see “Rotating a Side Chain” on page 124) in order to meet these conditions before you can display a hydrogen bond for this structure.

To confirm conditions for hydrogen bonding:

1. R-click on an empty area in the workspace.  
All previous selections are deselected.
2. Turn on Show Hydrogen Bonds on the Display menu.
3. Choose Recompute H Bonds on the Display menu.  
HyperChem displays the hydrogen bond as a broken line.



Hydrogen bonds are not automatically calculated at every configuration. When you change the molecule's geometry, you must recompute the hydrogen bonds.

## Practice Exercises

Try these practice exercises:

1. Measure improper angles and improper torsion angles.
2. Set builder constraints using Constrain Geometry, Constrain Bond Length, Constrain Bond Angle, and Constrain Bond Torsion on the Build menu.
3. Modify a model-built structure using Set Bond Length and Set Bond Angle on the Edit menu.

## For More Information

For more information on measuring structural properties, see "Measuring with the Selection Tool" in Chapter 3 of the *HyperChem for Windows Reference Manual*. For information on using builder constraints, see Chapter 5, "Building Molecules," in the same manual.

# Lesson 6

## Creating a Polypeptide

### Skills Covered in This Lesson

- Chaining residues together
- Creating a zwitterion
- Site-specific mutagenesis

Thus far in this tutorial you have learned how to build a molecule from scratch, and how to display a molecule by reading in coordinates from a HIN file. In this lesson, you create a polypeptide by sequentially selecting residues from the HyperChem amino acid library. The structure you build is *bradykinin*, a hormone-like peptide that inhibits inflammatory reactions.

### Using the Amino Acids Dialog Box

Clear the workspace, either by restarting HyperChem or by selecting New from the File menu.

To open the Amino Acids dialog box:

1. Select Amino Acids on the Databases menu. The Amino Acids dialog box will appear:



The Amino Acids dialog box is a *persistent* dialog box. It stays open while you chain residues together to build a polypeptide.

## Building and Displaying a Polypeptide

The first step in building the polypeptide is to determine the secondary conformation. In this exercise, you set the options to build a beta sheet structure.

To set the conformation:

1. Select Beta Sheet as the conformation.

This selection automatically sets both the phi and psi angle to 180 degrees.

2. Set the omega angle to 180 degrees for a *trans* peptide bond.

Normally, the peptide angle omega that connects residues is *trans*.

To build a polypeptide, start at the N-terminal end and chain together residues until you reach the C-terminal end.

To select the residues for the polypeptide:

1. L-click sequentially on Arg-Pro-Pro-Gly-Phe-Ser-Pro-Phe-Arg.

When you choose Pro, HyperChem “beeps.” This means that it cannot use the desired conformation. Proline can only adopt certain phi and psi angles that are not alpha or beta. As you build the polypeptide, HyperChem forces those angles to allowed values.

To close the dialog box:

1. L-click on the close-box in the top-left corner of the Amino Acids dialog box.

The polypeptide appears in the workspace. The structure automatically appears with labels because of the Labels option you set in the last lesson.

To unlabel the molecule:

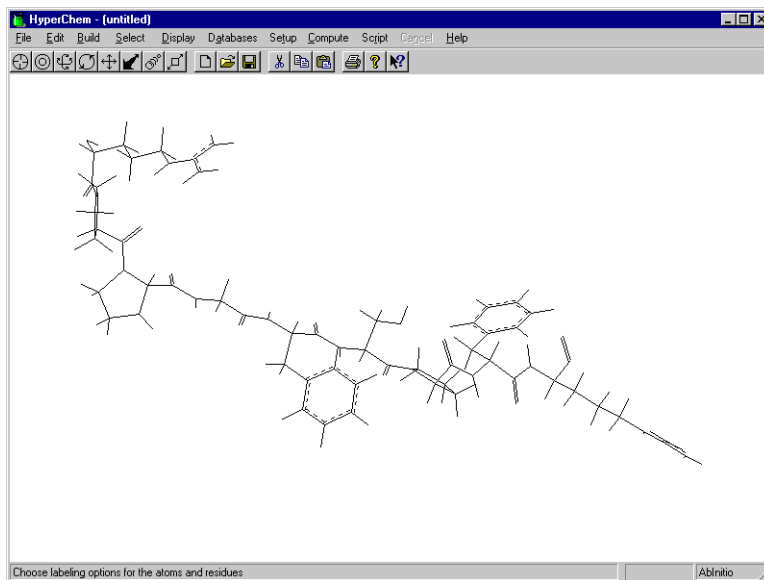
1. Select Labels on the Display menu. Make sure you choose None for both Atoms and Residues and then choose OK.

## Creating a Zwitterion

The peptide you created has a bare residue at each end. This is exactly how it would exist if it were in the middle of a peptide chain. The N-terminal end has  $\text{HN-}$  and the C-terminal end has  $\text{-CO}$ . In this exercise, you convert the ends of the peptide to zwitterionic form by modifying the N- and C- terminal residues.

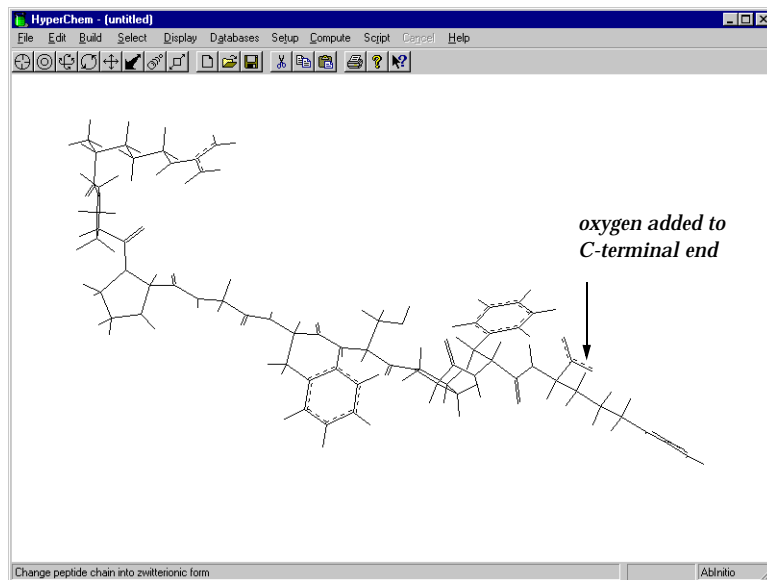
To create the zwitterion:

1. Translate and rotate the peptide until it looks like this:



2. Choose Make Zwitterion on the Databases menu.

HyperChem adds an oxygen at the C-terminal end:



HyperChem also adds protons to the N-terminal end.

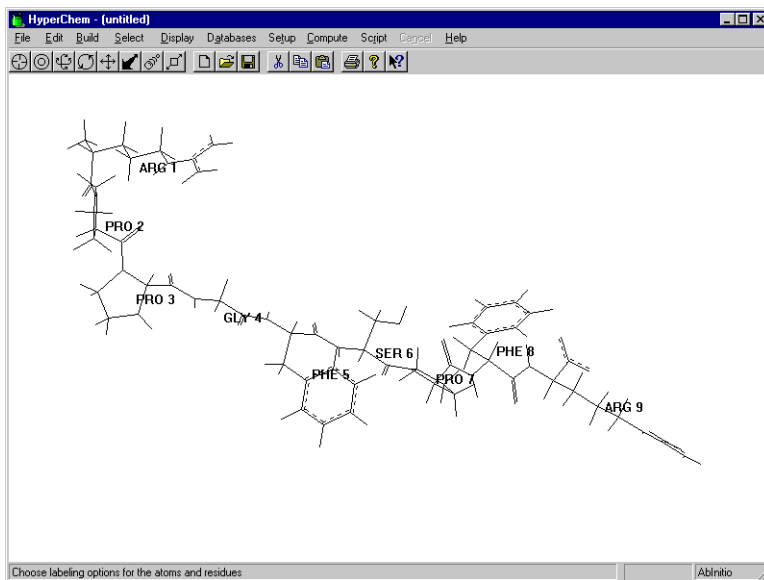
## Site-specific Mutagenesis

Site-specific mutagenesis plays an important role in protein engineering. Substituting a particular amino acid at a critical site can change the structure and properties of a protein, and therefore, its function.

To replace a residue:

First, label the peptide by residue to facilitate seeing what you are modifying.

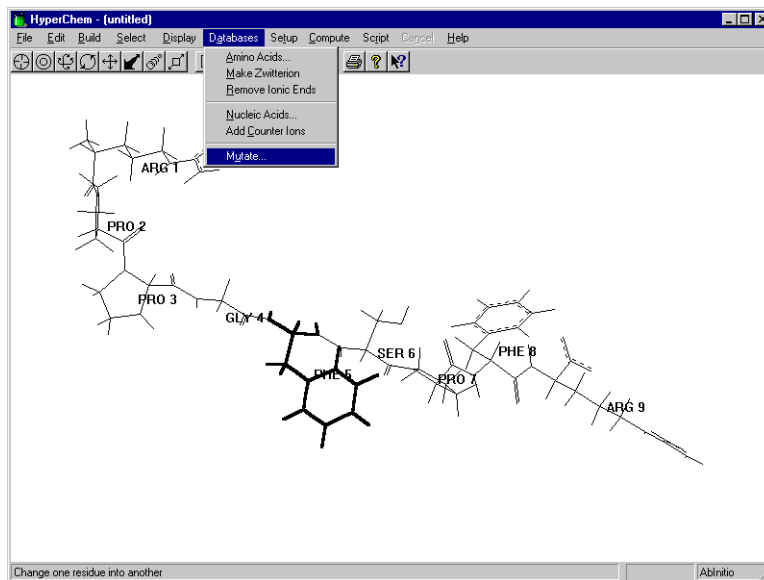
1. Select Labels on the Display menu.
2. L-click on Name+Seq as the option for labeling residues, and then choose OK.



In earlier exercises, you used the Label menu item to label a molecule by atom. You can also use this menu item to label residues.

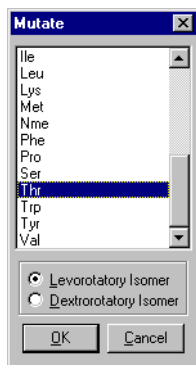
3. Choose Residues on the Select menu.
4. L-click on the Selection tool.
5. R-click in an empty area of the workspace to make sure nothing is selected.
6. L-click on PHE 5.  
The residue is highlighted.
7. Select Mutate on the Databases menu.



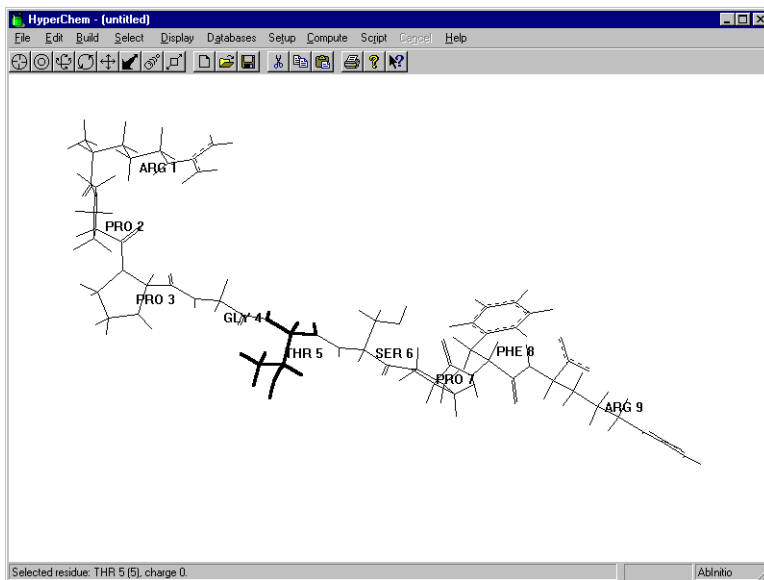


**Note:** The Mutate menu item is inactive (grayed out) in the Databases menu until you select a residue.

8. In the Mutate dialog box, scroll down the list of residues and choose Thr, then choose OK.



The selected phenylalanine is replaced by threonine:



## Saving the Structure

To save the structure:

1. Choose Save on the File menu.  
The Save File dialog box opens.
2. In the Directories list box, change to the directory where you want to save this file.
3. Enter **polyp** in the filename box, and then choose OK.
4. Choose New on the File menu to clear the workspace.

## Practice Exercises

1. Create a nucleic acid using Nucleic Acids on the Databases menu.

## **For More Information**

For more information on how to create polypeptides and nucleic acids, see “Databases Menu” in Chapter 5, “Building Molecules,” in the *HyperChem for Windows Reference Manual*.

# Lesson 7

## Selecting and Displaying Subsets

### Skills Covered in This Lesson

- Creating an alpha helix
- Selecting subsets such as backbones, rings, and side chains
- Labeling and coloring subsets

HyperChem can perform specific actions on selected atoms, residues, and molecules. These actions include moving molecules, changing display conditions, and performing calculations. When you select an object, it takes on a new appearance, usually a different color, and HyperChem recognizes that you want to change or analyze the selected atoms.

In this lesson, you learn how to select and display different subsets of the alpha-helical conformation of the polypeptide (PHE)<sub>6</sub>.

### Creating an Alpha Helix

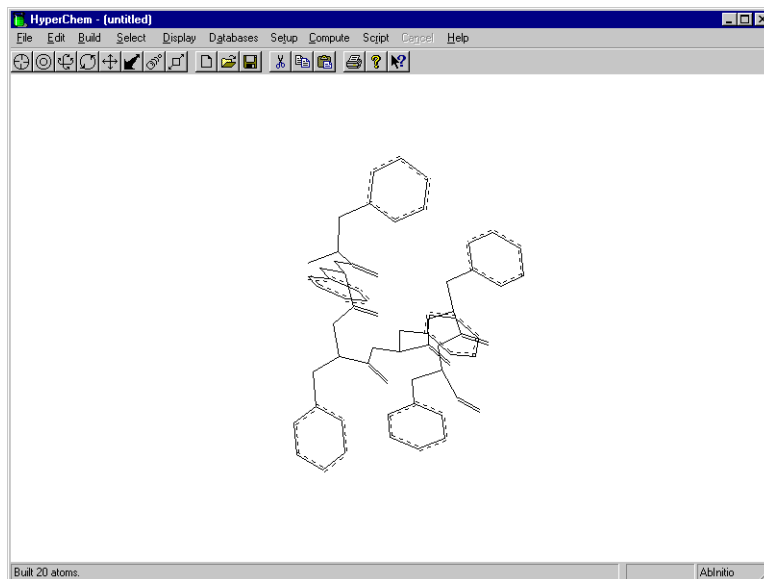
Clear the workspace, either by restarting HyperChem or by selecting New from the File menu.

To create the alpha helix:

1. Select Amino Acids on the Databases menu.  
The Amino Acids dialog box opens.
2. Select Alpha helix as the type of conformation.
3. Set Omega to 180 degrees.  
This defines *trans* peptide bonds.

4. L-click on Phe six times.
5. Double-click on the control box in the top-left corner of the dialog box, or choose Close.

The dialog box closes and the structure appears in the work-space:



(Your structure may be oriented differently from the above.)

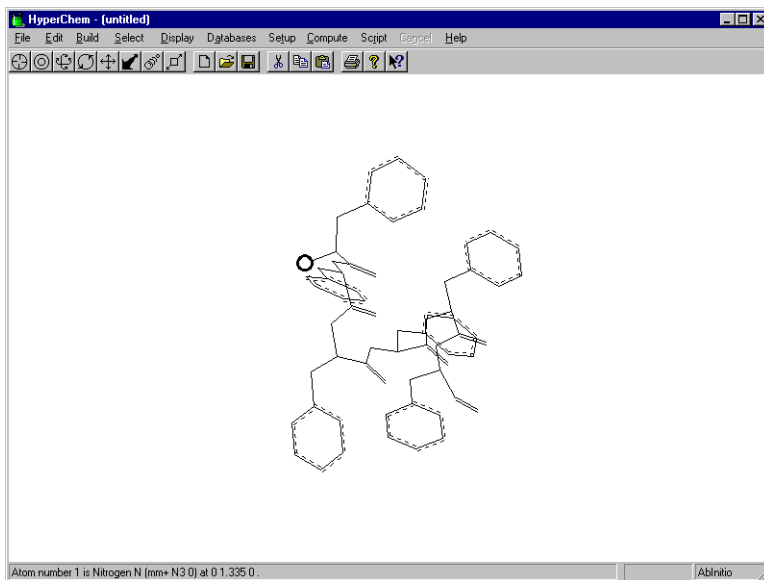
6. If the structure appears with hydrogens and labels, remove them from the display.

## Labeling the Ends

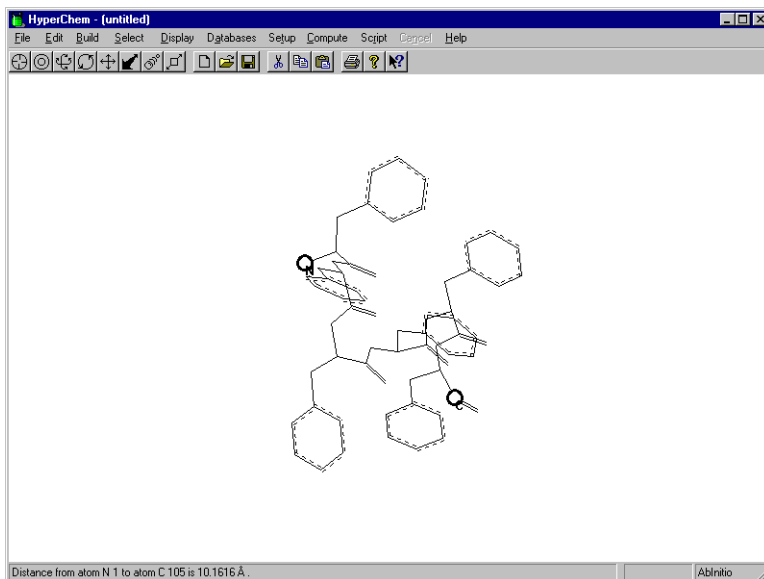
To better visualize the N-terminal and C-terminal ends of the polypeptide, you can label the ends.

To label the ends:

1. On the Select menu, make sure Atoms and Multiple Selections are on.
2. L-click on the Selection tool.
3. L-click on the N-terminal nitrogen atom on the left side of the polypeptide to select the N-terminal end:



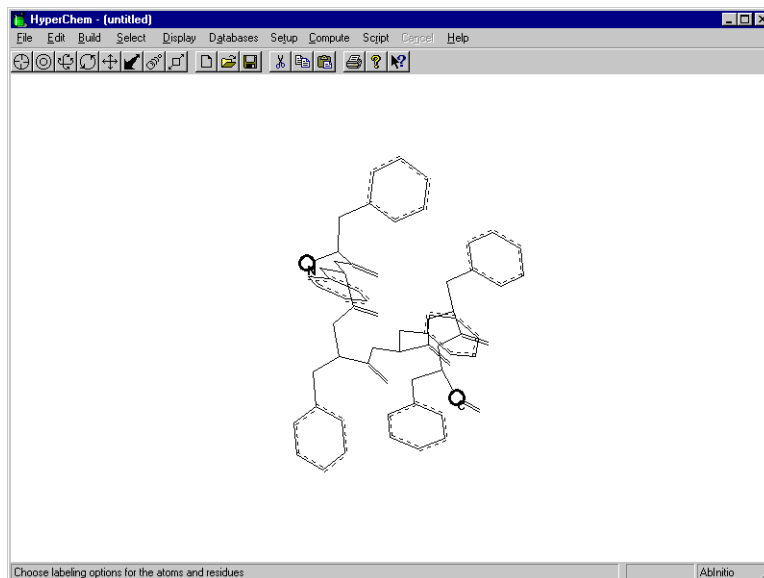
4. L-click on the C-terminal carbon atom on the right side of the polypeptide to select the C-terminal end:



5. Select Labels on the Display menu.  
The Labels dialog box appears.

6. Select Symbol as the atom label type, and then choose OK.

The N-terminal nitrogen and C-terminal carbon atom should be labeled like this:



## Selecting the Backbone

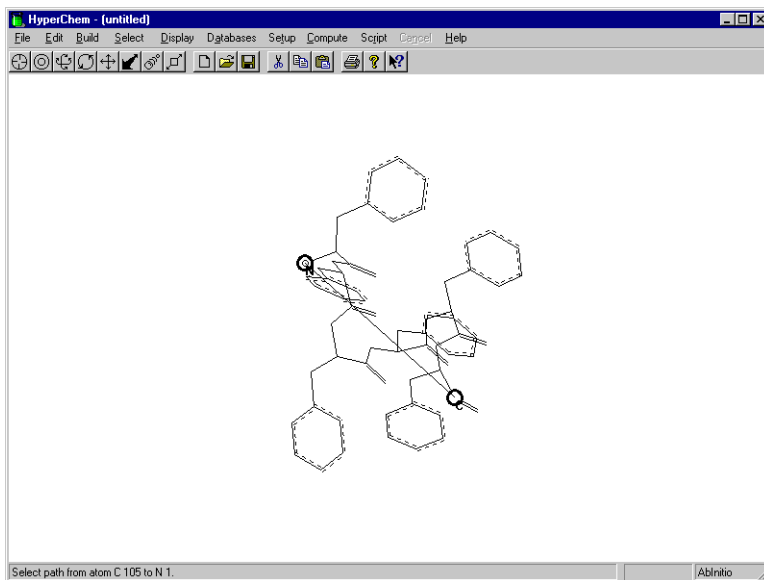
To investigate the alpha-helical nature of this polypeptide, it helps to visualize the backbone.

The first step is to select the backbone. You can do this two ways:

- By selecting the shortest point between the two terminal ends
- By using Select Backbone on the Select menu

To select the shortest point between the terminal ends:

1. If necessary, turn off Show Hydrogens on the Display menu.
2. If you are not in selection mode, L-click on the Selection tool.
3. L-drag from the carbonyl carbon atom of the C-terminal end to the nitrogen atom of the N-terminal end:



This selects the backbone of the polypeptide.

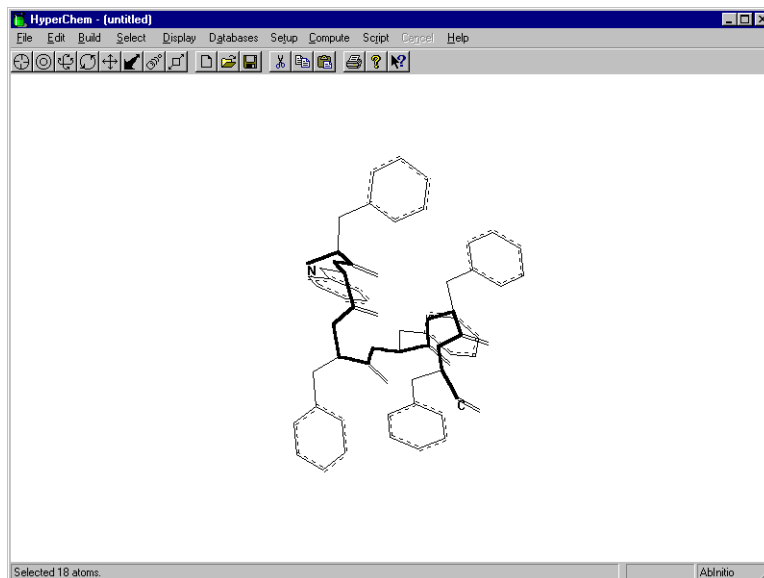
To use the Select Backbone command:

1. First, L-click on the Selection tool and R-click on an empty area of the workspace.

This deselects the previous selection.

2. Choose Select Backbone on the Select menu.





## Coloring the Backbone

To color the backbone:

1. Select Color on the Display menu.  
The Selection Color dialog box appears.
2. Select Yellow, then OK.
3. R-click on an empty area.  
HyperChem displays the backbone in yellow.

## Displaying Only the Backbone

To display only the backbone:

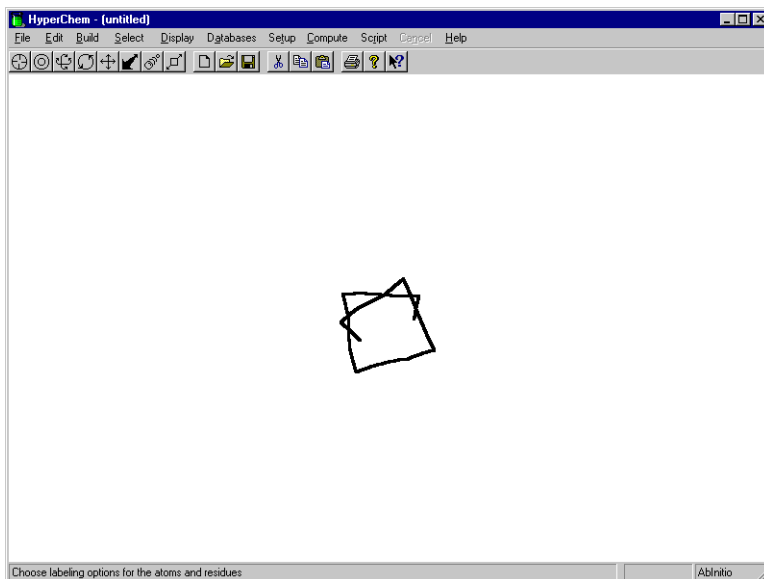
1. Choose Show Selection Only on the Display menu.  
Only the backbone and the labels are displayed.
2. Choose Labels on the Display menu.  
The Labels dialog box appears.
3. Select None for the residues label type, then choose OK.

Only the backbone and the terminal ends appear.

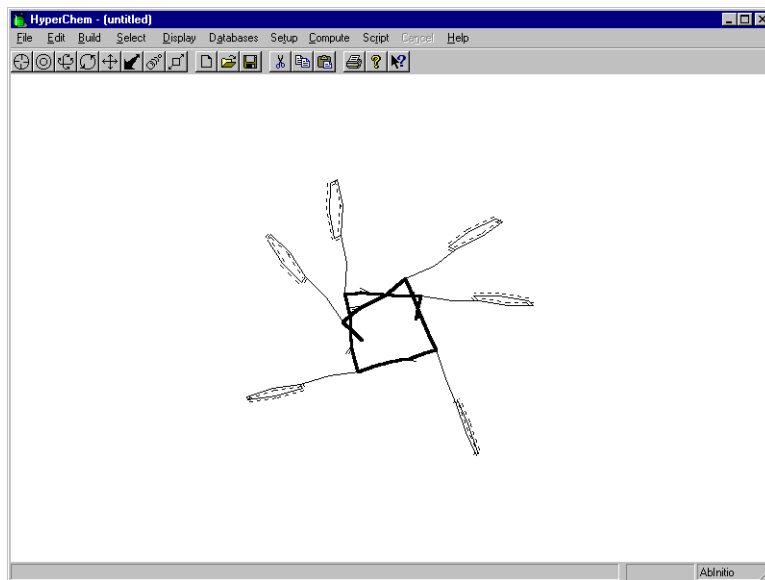
## Displaying the Side Chains

To display the side chains:

1. Rotate and translate the backbone so that it is on end and you are looking down its long axis, like this:



2. Choose Show All on the Display menu.



Later in this lesson, you also learn how to select a side chain.

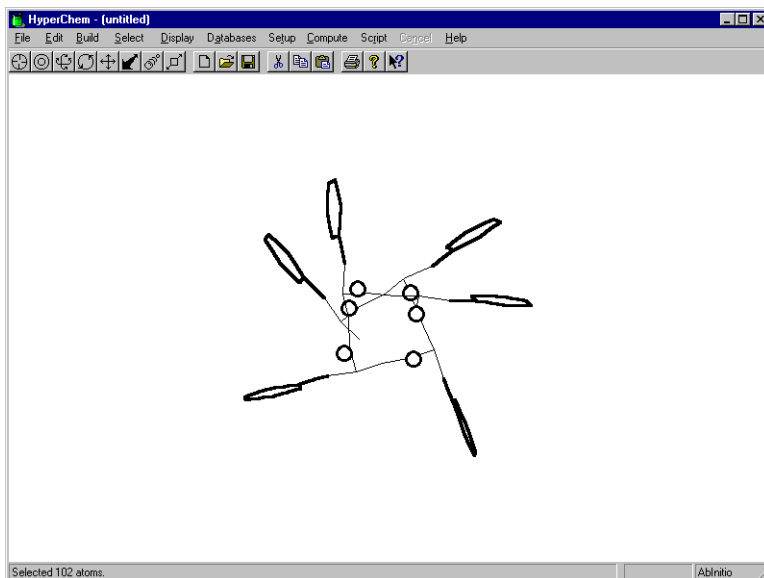
## Complementing a Subset

Sometimes, it might be helpful to select or display only the parts of a molecular system not currently selected. You can do this using Complement Selection on the Select menu.

To select and display the complement of a selected subset:

1. Choose Complement Selection on the Select menu.

This selects all parts of the polypeptide except the backbone.



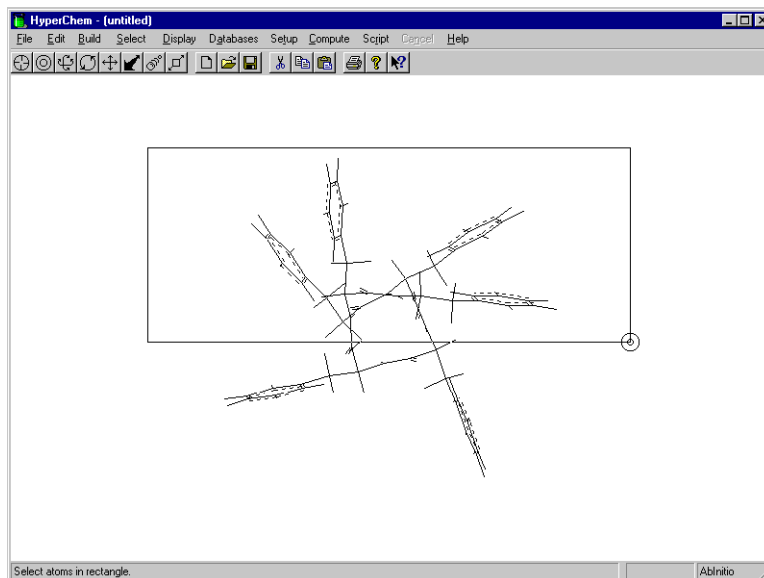
2. Choose Show Selection only on the Display menu and R-click the Selection tool on an empty area of the workspace.

This displays all parts of the polypeptide except the backbone.

## Rectangular Selection

To select all atoms in a 3D rectangle:

1. Select Show Hydrogens and then click on Show All on the Display menu.
2. Make sure Select Sphere on the Select menu is not turned on.
3. L-click on the Selection tool to get into selection mode.
4. Starting in the top-left corner of the workspace, LR-drag towards the bottom-right corner like this:

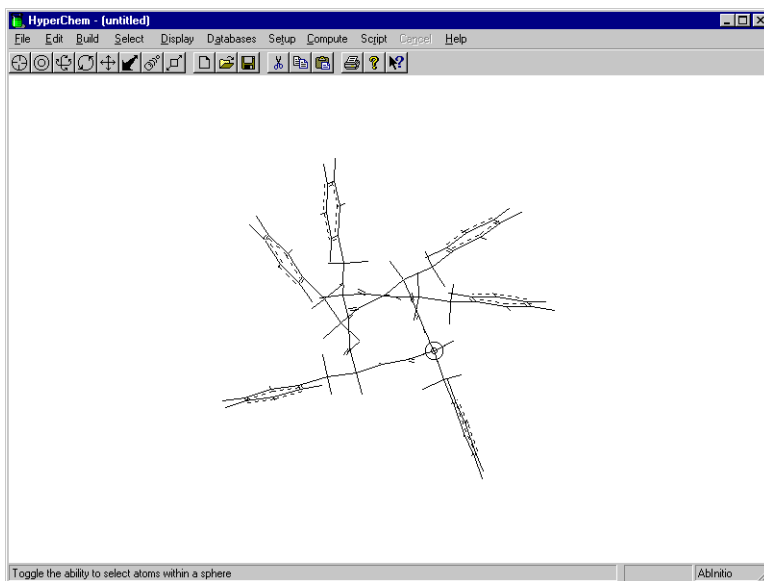


5. Release both mouse buttons.  
All atoms seen in the rectangular area are highlighted.

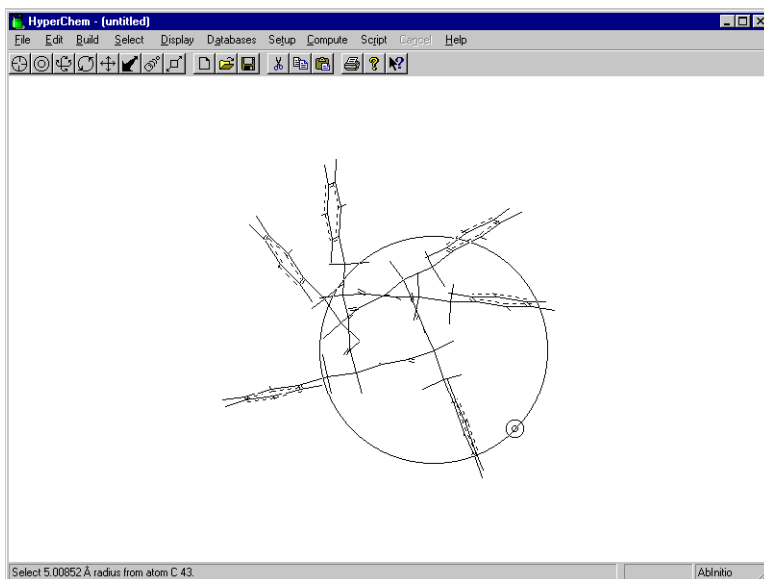
## Spherical Selection

To select atoms within a volume about a given atom:

1. R-click in an empty area of the workspace.  
This deselects all items in the workspace.
2. Choose Select Sphere on the Select menu.
3. Place the select cursor over the nitrogen atom of the N-terminal end.



4. LR-drag away from the nitrogen atom.  
The status line shows the radius of the selection sphere.
5. Continue until you create a radius of about 5.0 Ångstroms.



6. Release both mouse buttons.

HyperChem selects all atoms within a distance of about 5.0 Ångstroms of the N-terminal end of the molecule.

## Naming a Selection

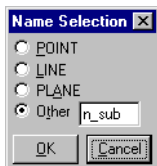
Often it is helpful to name a selection so that you can conveniently display it later. In this exercise, you assign a name to all atoms included in the current spherical selection.

To name a selection:

1. Choose Name Selection on the Select menu.

The Name Selection dialog box appears.

2. Make sure Other is selected and enter **n\_sub** in the Other text box.



3. L-click on OK.

HyperChem saves all the highlighted atoms in a selection named n\_sub.

4. R-click in an empty area of the workspace.

This step deselects the selection.

5. Choose Select on the Select menu.

The Select dialog box appears.

6. Choose By Name.

7. Select n\_sub from the Names list, and then choose OK.

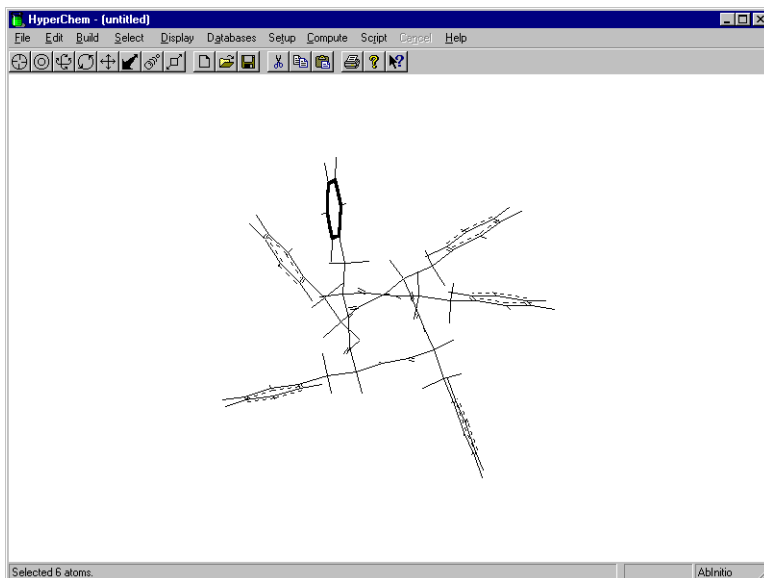
The selection is selected again.

## Selecting a Ring

To select a ring:

1. R-click in an empty area of the workspace.  
All items in the workspace are deselected.
2. Double-click on any bond in a phenyl ring.

The ring is selected:



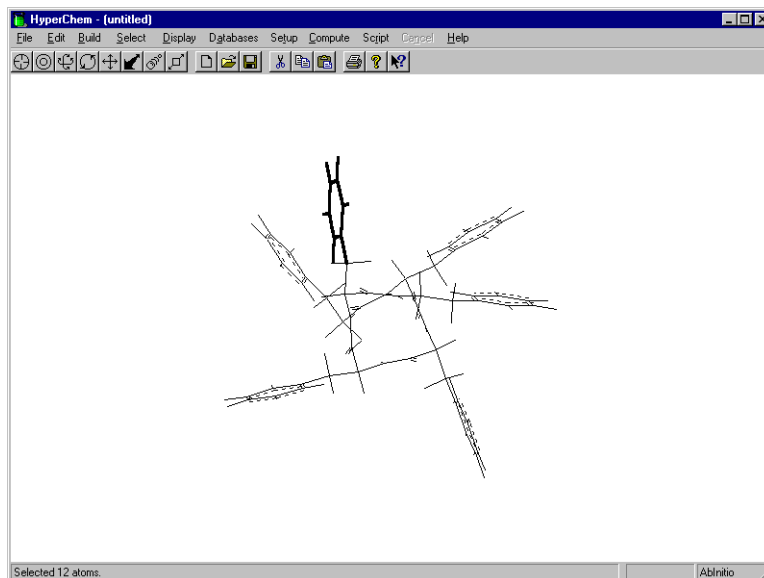
3. R-click in a empty area to deselect the ring.
4. Try selecting other rings in the polypeptide.

## Selecting a Side Chain

To select a side chain:

1. Point the cursor on a bond that is not part of a ring.
2. Double-click on the end of the bond closest to the ring.





The side chain is selected.

3. To end this lesson, choose Exit on the File menu and do not save the current changes.

## For More Information

For more information on selecting and displaying subsets, see “Selection Tool” in Chapter 3, and “Select Menu” in Chapter 6 of the *HyperChem for Windows Reference Manual*.

# Lesson 8

## Working with Macromolecules

### Skills Covered in This Lesson

- Reading and opening a PDB file
- Displaying structural characteristics of macromolecules
- Hydrogen bonding
- Site-specific mutagenesis
- Rotating a side chain

An important feature of HyperChem is the ability to manipulate and compute properties of macromolecules, such as proteins and nucleic acids.

So far you have used HyperChem to build and study small molecules. Now you study a macromolecule stored in Brookhaven Protein Data Bank (PDB) file format.

In this lesson, you work with the PDB file for bovine pancreatic trypsin inhibitor (BPTI).

### Reading a Brookhaven PDB Format File

You can read files in the Brookhaven PDB format into HyperChem. Files from the Brookhaven Protein Data Bank have a *.ent* extension. HyperChem can read PDB files from the distribution tape provided by Brookhaven or another source. These files must first be transferred from the tape to a medium (usually disk) accessible to HyperChem.

## Opening the PDB file

To open the PDB file:

1. If necessary, open HyperChem.
2. Choose Open on the File menu.

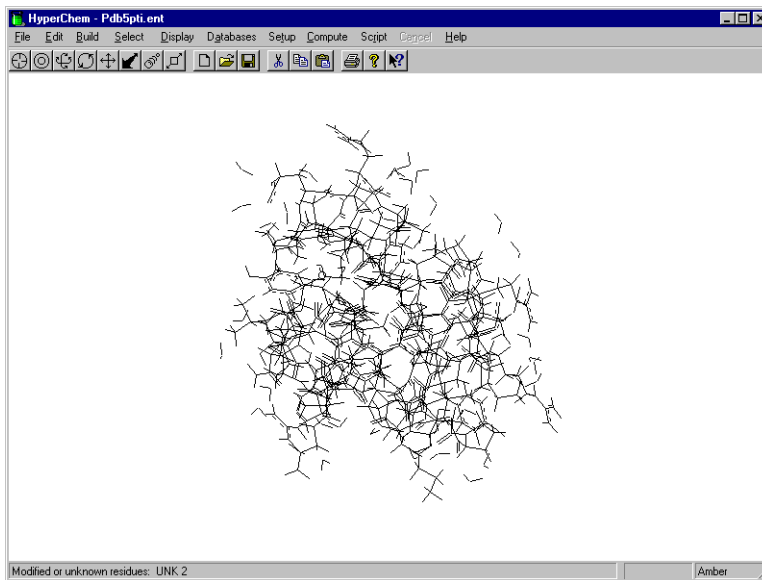
The Open File dialog box appears.

3. Using the Directories list box, change to the directory /usr/lib/HyperChem/samples.
4. Choose PDB as the file format.
5. L-click on pdb5pti.ent in the list of files.

When you select a PDB file, the Comments box in the dialog box shows the HEADER, COMPND, SOURCE, AUTHOR, REVDAT, JRNL, and REMARK lines of the Brookhaven PDB file.

6. Choose Open.

After a few moments, the molecule appears in the workspace:



When you open a PDB file, HyperChem reads an accompanying template file (*.tpl* extension) that contains information on connectivity, hydrogen atoms, atomic charges, and atom types.

7. If the structure appears with labels, use the Labels dialog box to remove them.
8. Make sure Show Hydrogens is turned off on the Display menu.

## Removing the Water Molecules

To simplify this lesson, remove the water molecules surrounding BPTI.

To remove the water molecules:

1. Set the select level to Molecules.
2. L-click on the Selection tool to get into selection mode.
3. L-click on the BPTI molecule.

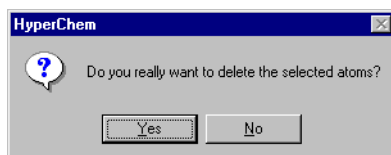
This selects the BPTI molecule.

4. Choose Complement Selection on the Select menu.

This selects only the water molecules.

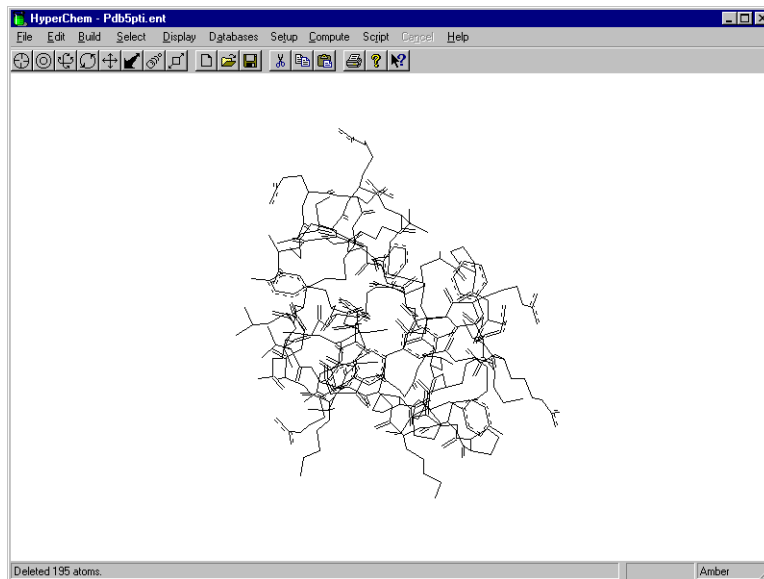
5. Choose Clear on the Edit menu.

The following dialog box opens:



6. Choose Yes.

This deletes the water molecules. The structure should look like this:



To avoid modifying the PDB file for BPTI, save the file to a different filename for this lesson.

1. Choose Save As on the File menu.
2. Using the Directories list box, change to the directory where you want to save your work.
3. Choose HIN as the file format.
4. In the Selection box, L-click after the last slash (/), enter **testbpti**, then L-click on Save.

HyperChem saves the modified structure as testbpti.hin.

## Displaying the Structure

### Displaying the Primary Structure

To display the primary structure of the molecule:

1. Make sure Show Hydrogens is turned off on the Display menu.  
This helps eliminate a cluttered display.

2. Choose Select Backbone on the Select menu.

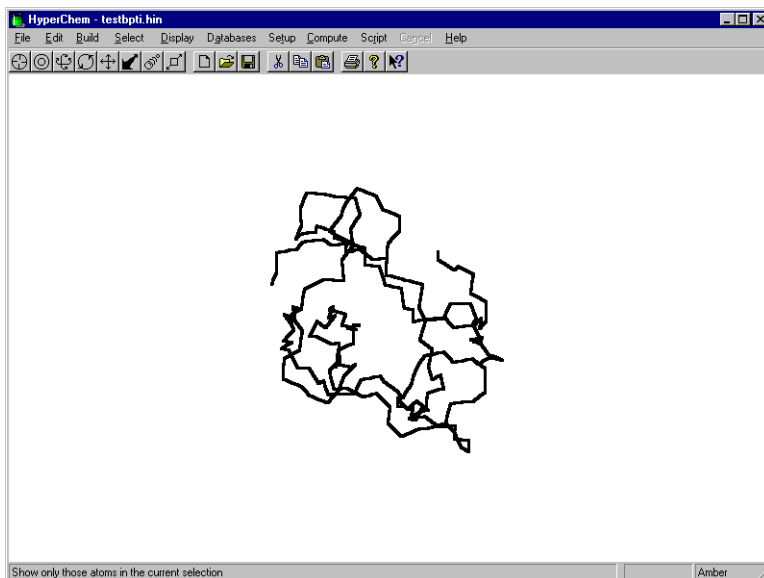
This highlights the backbone.

3. Press the space bar.

You get a closer look at the backbone. If you hit the space bar when an area is selected, you zoom in on that area.

4. Choose Show Selection Only on the Display menu.

This displays only the backbone.



5. R-click in an empty area of the workspace.

This deselects the backbone.

## Displaying the Side Chains

To redisplay the side chains:

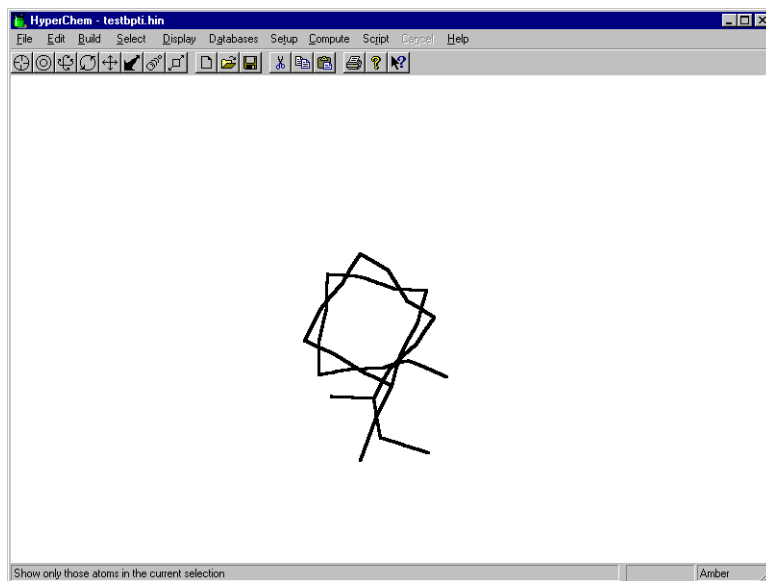
1. Choose Show All on the Display menu.

## Displaying the Alpha Helix

To display the alpha helix:

1. Make sure Multiple Selections and Residues are selected on the Select menu.
2. Choose Select on the Select menu to open the Select dialog box.
3. Make sure By number is selected, and then L-click on Residue.
4. Enter **47** in the Residue/atom text box, then choose OK.  
This selects SER 47. This is an easy way to select a specific residue in the molecule.
5. Open the Select dialog box again and specify residue number **55**.  
Residues 47 and 55 are highlighted.
6. L-drag between residues 47 and 55.  
This selects all residues between 47 and 55.
7. Choose Show Selection Only on the Display menu.  
This shows an uncluttered display of the selected residues.
8. Press the space bar to center the selection in the workspace.
9. R-click in an empty area to deselect the residues.
10. Choose Select Backbone on the Select menu, then Show Selection Only on the Display menu.  
HyperChem displays only the backbone of the alpha helix corresponding to residues 47 to 55. The rest of the backbone is selected but not displayed.

- Translate and rotate the alpha helix so it is on end. It should look like this:



- Choose Show All and Scale to Fit on the Display menu.
- Get into selection mode and R-click in an empty area of the workspace to deselect the selection.

## Displaying the Disulfide Bridges

BPTI contains cystine residues, which form covalent crosslinks along the peptide backbone. In this exercise, you zoom in on a portion of the molecule to display these crosslinks.

To zoom in on a disulfide bridge:

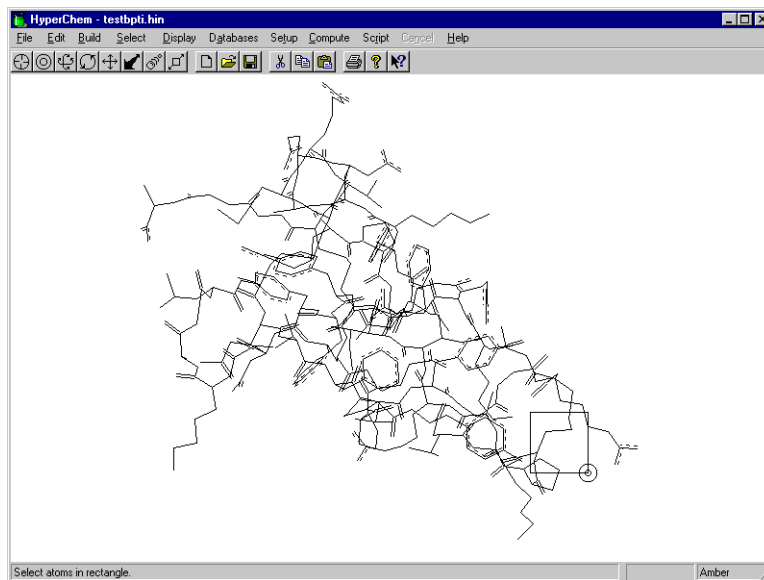
- If the whole protein is not displayed in the workspace, choose Show All on the Display menu, and hit the space bar.

This displays and centers the structure in the workspace.

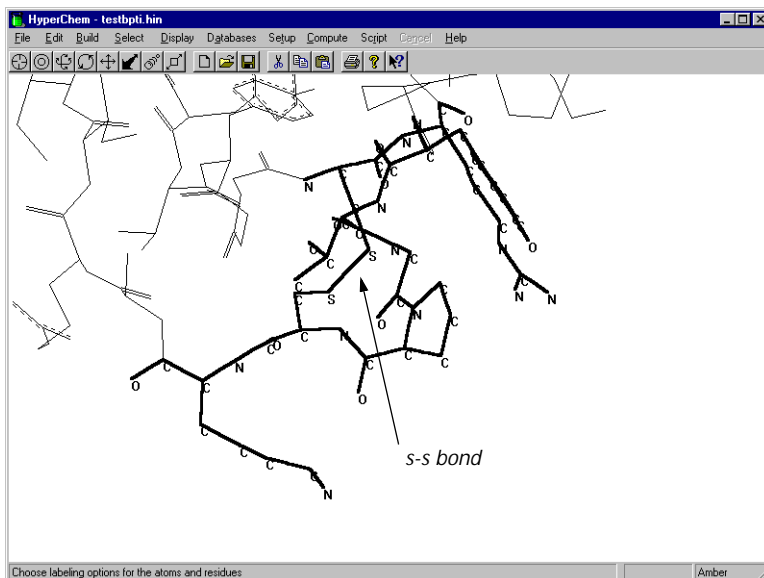
- LR-drag the Selection tool to include the cystine residue in the lower-right side of the molecule. Be sure to start in an empty area of the workspace. You may include a few neighboring residues as well.



**Note:** The cystine residue appears in yellow if you are using the default colors. Also, if Select Sphere is checkmarked on the Display menu and you do not LR-drag starting in an empty area of the workspace, you create a spherical selection, not a rectangular selection.



3. Look in the status line to make sure residue CYX 14 is selected.
4. Press the space bar to center the selection in the workspace.
5. Choose Labels on the Display menu and choose Symbol as the type of atom label.  
The selection is labeled by atom.
6. Slowly rotate and zoom in on the structure until you can easily identify the -S-S bond.



7. R-click in an empty area to clear the selection.

## Selecting the Ring Structure

You can use the ring-finding feature of HyperChem to display the residues that form a ring in the protein.

To display the ring structure:

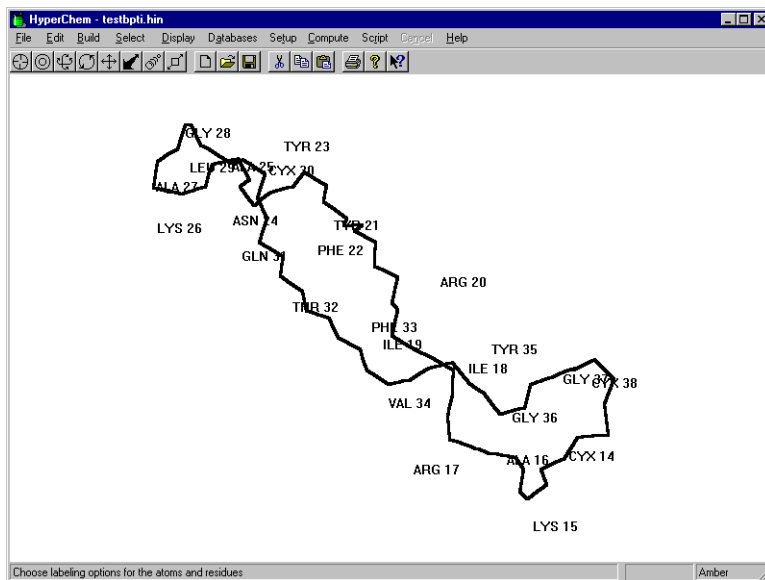
1. Change the select level to Atoms.
2. Double-click in the middle of the -S-S- bond.

This selects the ring formed via the disulfide bridges between CYX 14 and CYX 38.

3. Press the space bar to center the ring in the workspace.
4. Choose Show Selection Only.
5. Using the Labels dialog box, select Name+Seq to label the residues by name and sequence.

This displays only the ring labeled by name and sequence.

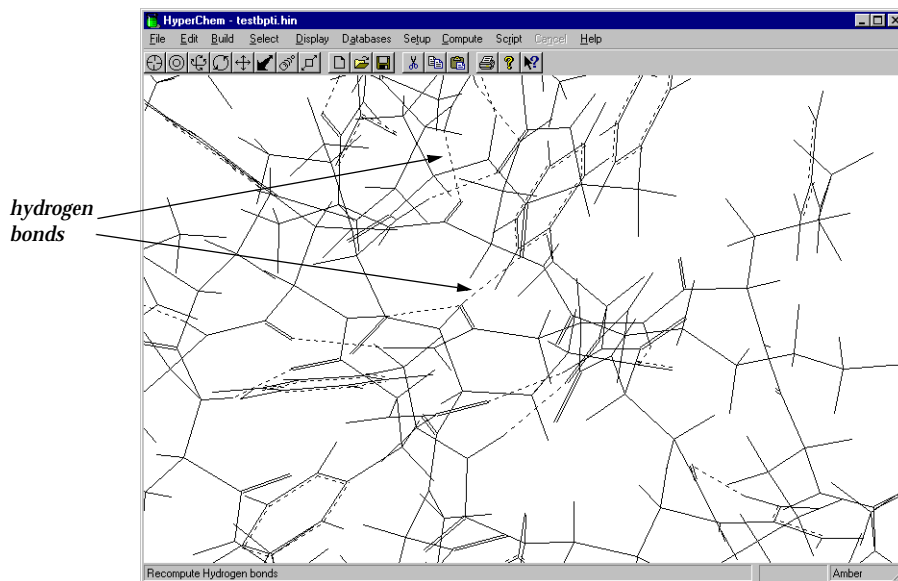
6. Rotate the ring until it looks like this in the workspace:



## Displaying Hydrogen Bonds

To calculate and display hydrogen bonds:

1. Get into selection mode and R-click in an empty area to deselect the ring.
2. On the Display menu, choose Show All, and then Scale to Fit.  
The entire molecule is displayed and centered in the workspace.
3. Remove labels from the display.
4. Turn Show Hydrogens and Show Hydrogen Bonds on.  
Show Hydrogen Bonds displays possible hydrogen bonds between selected atoms, or, with no selection, displays hydrogen bonds for all molecules in the system.
5. Choose Recompute H Bonds on the Display menu.  
HyperChem displays the hydrogen bonds as broken lines.
6. Press [PgDn] to zoom in until the hydrogen bonds are clearly visible.
7. Press the space bar to return to a scale-to-fit view.



**Note:** If you change the arrangement of molecules in the workspace (for example, by XY translation) or if HyperChem changes the conformation during a calculation, choose Recompute Hydrogens Bonds to see up-to-date hydrogen bonds.

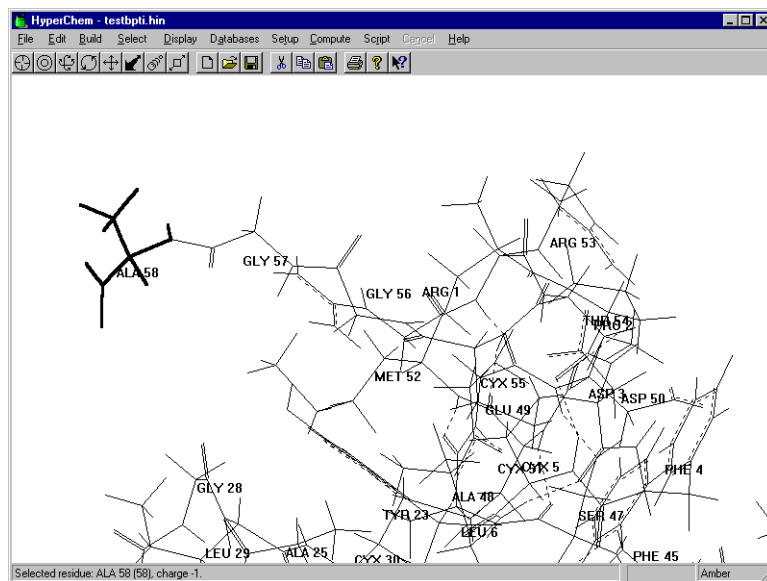
## Site-specific Mutagenesis

In Lesson 6 you performed site-specific mutagenesis on bradykinin. In this exercise, you do a series of mutations on the outside of the structure.

To mutate the structure:

1. Make sure Multiple Selections is off.
2. Label the residues by choosing Name+Seq.
3. Use the Select dialog box to select residue 58.

- Zoom in on the selection and position the structure to look like this:



- With ALA 58 still selected, choose Mutate on the Databases menu, and replace ALA 58 with VAL.
- Use the Select dialog box to select GLY 57. Replace it with PHE.
- Replace residue 56 with ALA and residue 55 with VAL.

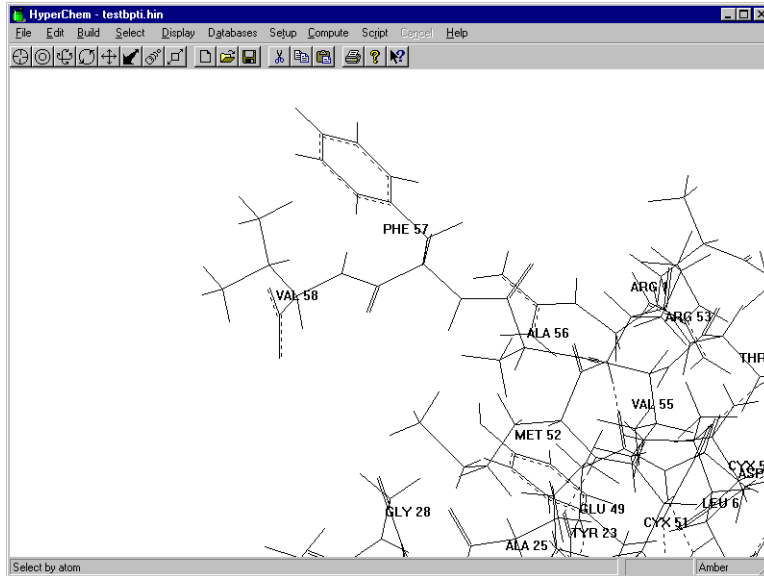
## Rotating a Side Chain

You can alter the side-chain torsion angles by using the side-chain selection feature and the internal rotation option of the Z rotation tool.

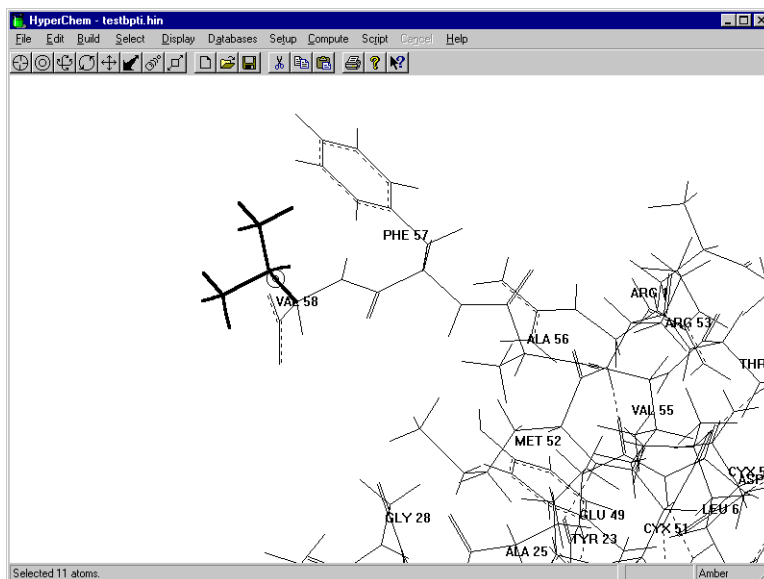
To rotate the side chain:

- R-click in an empty area to clear the selection.
- Set the select level to Atoms.

3. Position the structure to look like this.



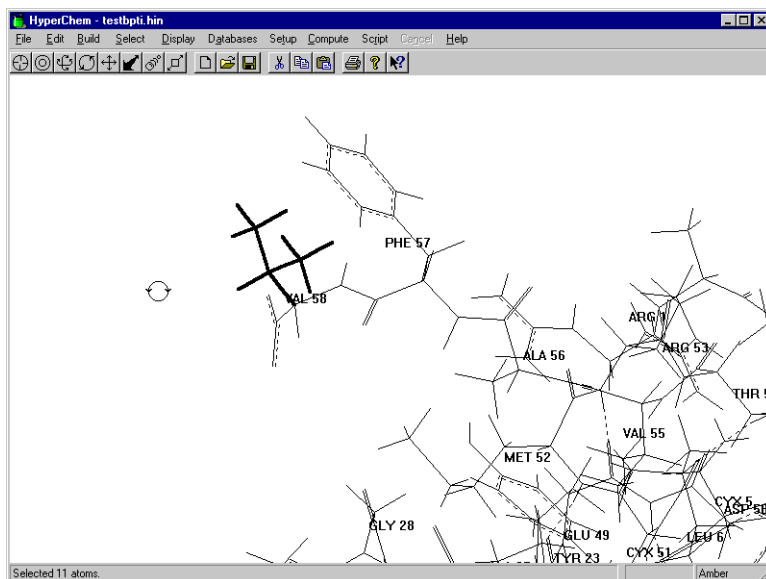
4. Double-click on the  $C^\beta$  side of the  $C^\alpha$ - $C^\beta$  bond of VAL to select the new side chain:



5. L-click on the Z rotation tool.

6. R-drag the Z rotation cursor horizontally in the workspace to rotate the valine side chain.

**Note:** To increase the rotation speed, remove part of the molecule from the display. To do this, select the part of the molecule surrounding the valine residue. Choose Show Selection Only on the Display menu and R-click in an empty area to deselect before selecting the side chain for rotation.



7. To finish this lesson, choose New on the File menu, and do not save the current changes.

## For More Information

For more information on the Brookhaven PDB files, see Chapter 9, "Protein Data Bank Files," in the *HyperChem for Windows Reference Manual*. For more information on site-specific mutagenesis, see Chapter 5, "Building Molecules," in the same manual.

# Tutorial 2

## HyperChem Calculations

Tutorial 2 focuses on the chemical calculations you can perform with HyperChem. You learn the step-by-step instructions for performing single point calculations, geometry optimization, and molecular dynamics, using both semi-empirical quantum mechanics and molecular mechanics methods. Tutorial 2 has five lessons, as described in the following table:

Lesson	Information covered	Time to complete
Minimizing the Energy of a System	Molecular mechanics geometry optimization; single point calculations; using reflection through a plane to perform a symmetry transformation; keeping a log file.	60 minutes
Simulating Dynamic and Equilibrium Behavior	Atomic charge assignment, periodic boundary conditions, superposition, and using molecular dynamics to perform simulated annealing. Using Langevin dynamics and Monte Carlo.	100 minutes
Molecular Orbital Calculations	Calculating wave functions; plotting electrostatic potential, total charge density, and molecular orbitals.	25 minutes



---

Lesson	Information covered	Time to complete
Interaction of Water with N-methylacetamide	Using builder constraints, merging structures into a single system, optimizing a subset, and characterizing the interaction between two systems.	20 minutes
Electronic Properties of Proteins	Treating part of a protein quantum mechanically.	15 minutes

---

# Lesson 9

## Minimizing the Energy of a System

### Skills Covered in This Lesson

- Performing a molecular mechanics geometry optimization
- Performing single point calculations
- Using bond torsion constraints
- Measuring and comparing structural properties of a system
- Using reflection through a defined plane for the Model Builder
- Keeping a log of your work

So far you have used HyperChem's basic building, editing, and display tools. Now you begin using HyperChem's analysis tools that will help you understand molecular behavior and interaction.

In this lesson, you minimize the energy of cyclohexane using the AMBER force field. Before you proceed with this lesson, it is important to understand some of the basic concepts of energy minimization.

Energy minimization alters molecular geometry to lower the energy of the system, and yields a more stable conformation. As the minimization progresses, it searches for a molecular structure in which the energy does not change with infinitesimal changes in geometry. This means that the derivative of the energy with respect to all Cartesian coordinates, called the *gradient*, is near zero. This is known as a *stationary point* on the potential energy surface.

If small changes in geometric parameters raise the energy of the molecule, the conformation is relatively stable, and is referred to as a *minimum*. If the energy lowers by small changes in one or more dimensions, but not in all dimensions, it is a *saddle point*.

A molecular system can have many minima. The one with the lowest energy is called the *global minimum* and the rest are referred to as *local minima*.

In this lesson, you calculate three stationary points for cyclohexane: chair, boat, and twist-boat. You perform molecular mechanics optimization on each form and compare energies to determine the global minimum energy conformation.

## Keeping a Log File

The status line at the bottom of the HyperChem window displays the most pertinent results of a calculation. You can keep these messages, and other information relating to the calculation, in a log file. This makes it easier for you to print, plot, or paste the information from the log file into a manuscript.

To start a log file:

1. If necessary, reopen HyperChem.
2. Choose Start Log on the File menu.

The Open Log menu opens and the default filename, chem.log, appears in the text box.

If the file chem.log already exists, you can append to it by checking the Append option.

3. L-click on OK.

The log file chem.log starts collecting the results of any calculations that are performed.

## Choosing the Force Field

Before you build the chair structure of cyclohexane and perform a molecular mechanics optimization, you should choose a molecular mechanics force field provided with HyperChem.

A force field contains atom types and parameters that must be assigned to the molecule before you perform a molecular mechanics calculation. For this exercise, you use the AMBER force field.

To choose the force field:

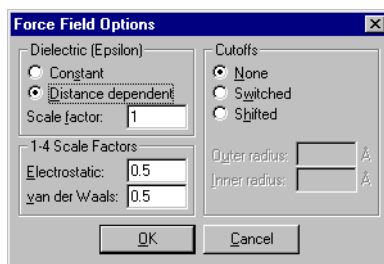
1. Choose Molecular Mechanics on the Setup menu.

2. When the dialog box appears, choose AMBER.
3. L-click on Options to open the Force Field Options dialog box.
4. Set the Dielectric to Distance Dependent.

If no explicit solvent water molecules are added to the calculation, it is a common approximation to use a distant-dependent dielectric constant in which Coulomb interactions fall off as  $1/r^2$  rather than  $1/r$  to simulate the effect of a solvent. In either case, Coulomb interactions can be scaled by an additional scale factor if you choose a value different from 1.0.

5. Set the Scale factor to **1**.
6. Set both the Electrostatic and van der Waals 1-4 scale factors to **0.5**.

These options determine the scaling of nonbonded interactions for atoms that are separated by three bonds. The AMBER parameters were derived with both scale factors set to 0.5 so you should normally use these with the AMBER force field.



7. Set Cutoffs to None.
8. Choose OK to close both dialog boxes.
9. Choose Select Parameter Set on the Setup menu.
10. When the dialog box appears, choose amber2.

Several different parameter sets are available for the AMBER force field, and users can define their own parameter sets. For more information about the differences between versions of

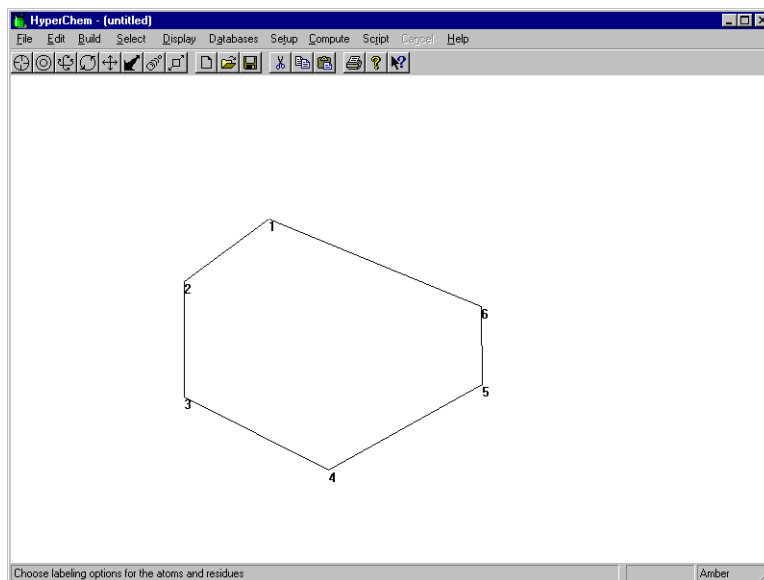
the parameter sets, please see the *HyperChem Computational Chemistry* manual.

## Building Chair Cyclohexane

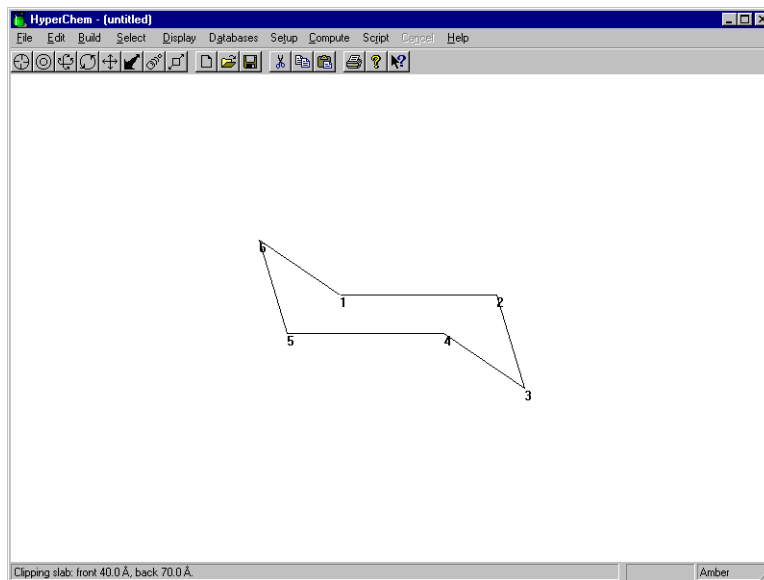
The first form of cyclohexane you build is the chair form.

To build the chair conformation:

1. Set the Default Element to carbon and get into drawing mode.
2. Set the select level to Atoms.
3. Choose Labels on the Display menu and label the atoms by number.
4. Make sure Explicit Hydrogens is turned off on the Build menu.
5. Draw the 2D structure in the order shown:



6. Choose Add H & Model Build on the Build menu.
7. Turn off Show Hydrogens on the Display menu.
8. Rotate and translate the structure until it looks like this:



The Model Builder builds the chair form of cyclohexane as the default structure. The structure is not optimized, but it has a standard set of bond lengths, angles, and torsions.

## Measuring Structural Properties of Chair Cyclohexane

Now measure the structural properties of the model-built structure. Later, you compare this with geometry measurements from the optimized structure.

To measure the geometry of the molecule:

1. Get into selection mode.
2. Set the select level to Atoms and turn Multiple Selection off.
3. Select a few bonds, angles, and torsion angles to explore the geometry of the structure.

The following values appear on the status line after you make your selections:

Bond distance: 1.54Å

Angle: 109.47°

Torsion: 60°

4. R-click in an empty area of the workspace to make sure nothing is selected.

## Performing a Single Point Calculation

Next, you perform a single point calculation to obtain the total energy of the unoptimized configuration.

To do a single point calculation:

1. Choose Single Point on the Compute menu.

The single point calculation reports the energy in kcal/mol and total root-mean-square (RMS) gradient in kcal/(mol·Ångstrom) of the current configuration of atoms. The following values appear on the status line:

Energy= 1.64

Gradient= 3.02

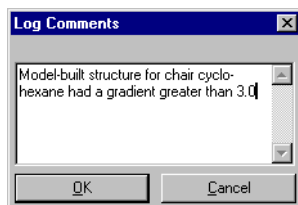
At a local minimum, the RMS gradient is close to zero. Thus the model-built structure is not a local minimum using the AMBER force field.

## Entering Comments in the Log File

All the information from the single point calculation has been saved in the log file. In addition to what appears on the status line, the log files also show the components of the energy. For future reference, you can add a comment to the log file.

To enter a comment in the log file:

1. Choose Log Comments on the File menu and enter a comment in the text box as shown:



2. L-click on OK.

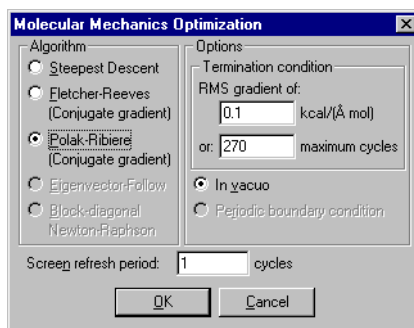
## Optimizing the Structure

The next step is to minimize the chair structure by performing a molecular mechanics optimization. First you set the minimization variables, including the type of minimizer, and then run the calculation.

### Setting Optimization Variables

1. Choose Geometry Optimization on the Compute menu.

The dialog box appears:



The dialog box options let you choose an optimization algorithm and the convergence criterion for the energy minimization you are about to perform.

2. Choose Polak-Ribiere as the minimization algorithm.

This algorithm is a good general-purpose optimizer. Both Polak-Ribiere and Fletcher-Reeves perform a series of one-dimensional searches, or cycles, in conjugate gradient directions. Different algorithms might be appropriate in different circumstances.

3. Use 0.1 for the RMS gradient condition, and the default values for the other variables.

The text boxes for RMS gradient and maximum cycles let you set conditions for ending a calculation. When the calculation reaches either of these criteria, the calculation ends.



The In vacuo option performs the calculation without periodic boundary conditions. This is the only choice unless the system has been set up in a periodic box. Periodic boundary conditions reverses the in vacuo option. This is grayed out unless you have used the Periodic Box menu option.

## Running the Calculation

You have finished setting the variables for the optimization and are ready to run the calculation.

1. L-click on OK to close the dialog box and start the calculation.

The geometry optimization begins and information about the run appears on the status line. After a few moments, the run finishes. The following values appear on the status line:

Energy: 1.33

Gradient: 0.07

The gradient is much less than the gradient measured in the model-built structure before minimization.

The number of cycles is the number of search directions used (both Fletcher-Reeves and Polak-Ribiere use a series of one-dimensional searches) and the number of points is the number of energy and gradient evaluations.

## Measuring Properties of the Minimized System

Now compare the structural properties of the minimized system with those of the model-built structure.

1. Select various bonds, angles, and torsion angles.

Values appear on the status line when you make your selections:

Bond distances: 1.53Å

Angles: 110.2°

Torsions: 58.0°

2. Compare these values with the following values from the unminimized structure:

Bond distances: 1.54Å

Angles: 109.4°

Torsions: 60°

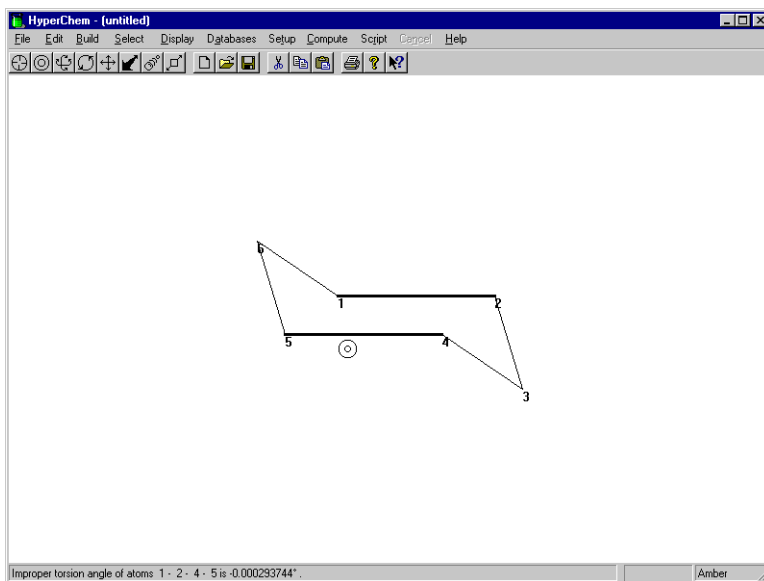
Comparing the structure properties shows that energy minimization increased the bond angles slightly from tetrahedral and decreased the torsion angle by 2 degrees.

## Transforming from Chair to Boat Cyclohexane

In this exercise, you reflect one end of the molecule to produce the boat form of cyclohexane.

To define a reflection plane:

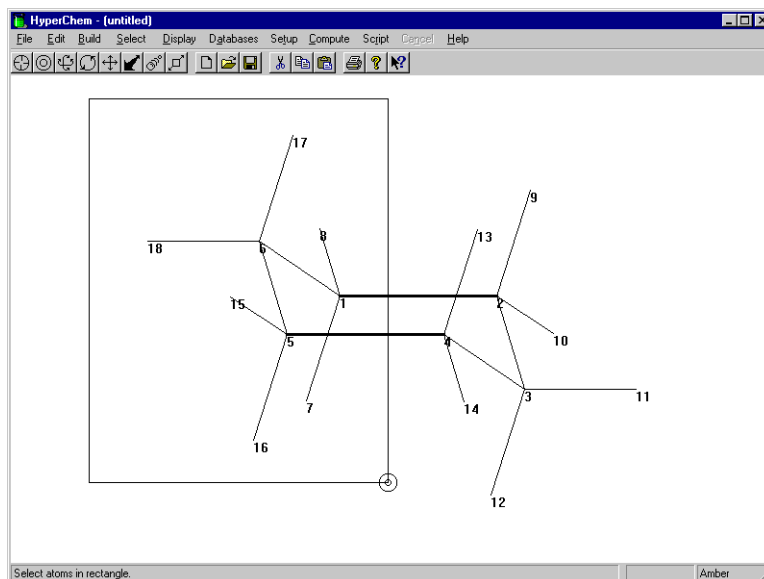
1. Turn on Multiple Selections.
2. If you are not in selection mode, L-click on the Selection tool.
3. Double-click on the Selection tool to return to the model-built structure.
4. L-click on bonds 1-2 and 4-5 to select the reflection plane shown here:



5. Choose Name Selection on the Select menu.
6. Choose PLANE, and then choose OK.

To reflect one end of the molecule:

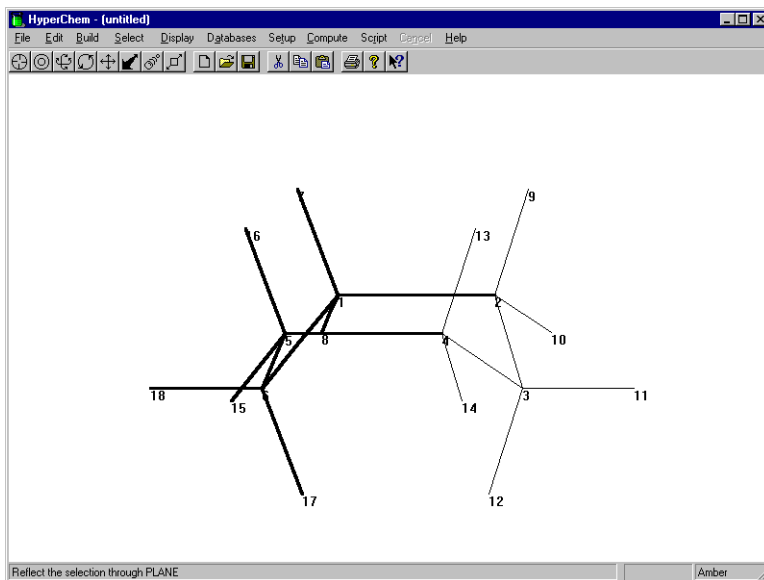
1. If necessary, Choose Show Hydrogens and use the Zoom tool to scale the molecule so that the entire molecule is in view.
2. LR-drag to extend the selection to include all atoms on one side of the initially selected bonds.



3. Choose Reflect on the Edit menu.

The selected atoms are reflected through PLANE, producing the boat transformation of cyclohexane.

The structure should now look like this:



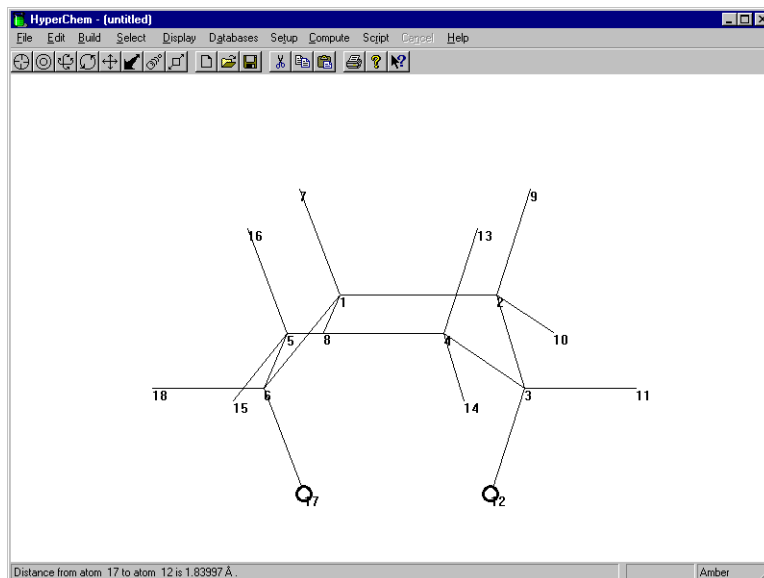
4. R-click in an empty area of the workspace to deselect the atoms.

## Measuring the Axial Hydrogens

Two axial hydrogens are fairly close together in boat structure.

To measure the distance:

1. L-click on these two atoms:



The status line reports a distance of only 1.84Å. This is quite close for atoms that are nonbonded. Optimizing the structure moves these atoms further apart to lower the energy.

## Optimizing the Boat Cyclohexane

To minimize the boat structure:

1. R-click in an empty area of the workspace to deselect the atoms.
2. Choose Geometry Optimization on the Compute menu.
3. Choose OK to begin the minimization using the previous options.

As each conformation appears in the workspace, values for the energy and gradient appear on the status line. After the minimization finishes, the status line displays these values:

Energy: 8.31

Gradient: 0.08

## Remeasuring the Axial Hydrogens

1. L-click on the two axial hydrogens.

The status line displays the new H-H distance (2.28Å). The energy minimization has slightly flattened the structure and moved the two axial hydrogens apart.

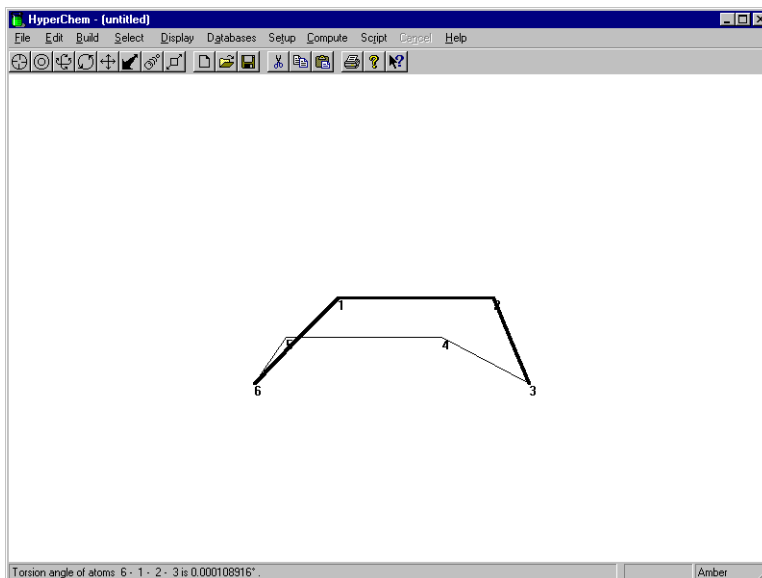
The optimized boat structure is a saddle point. The plane of symmetry in the starting structure balances all forces perpendicular to that plane. The optimizer search directions are based on these forces, and therefore, all search directions have the same symmetry plane. HyperChem finds a saddle point that is a minimum with respect to all dimensions except the symmetry plane.

## Creating Twist Boat Cyclohexane

A third form of cyclohexane, the twist boat form, is a true local minimum. A simple way to obtain this is to modify the boat form slightly by imposing a bond torsion constraint, rebuilding, and optimizing the structure.

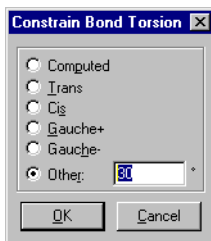
To set torsion bond constraint:

1. R-click in an empty area of the workspace to clear the selection.
2. Turn off Show Hydrogens.
3. Select a four-carbon atom torsion angle by selecting 6-1, 1-2, and 2-3 bonds in that order.



You must select the bonds in this order so that the correct angle is constrained. The Model Builder calculates geometries according to the order of selection; specifying a constraint for this particular torsion only changes the position of carbon 6.

4. Choose Constrain Bond Torsion on the Build menu, select Other, and enter 30 to set the constraint to 30 degrees. Then choose OK.



5. R-click in an empty area of the workspace.  
The torsion is deselected, but the constraint remains set.

To rebuild the molecule with the bond torsion constraint:

1. Double-click on the Selection tool to invoke the Model Builder.

HyperChem rebuilds the structure with the torsional constraint to create a canonical twist boat form of cyclohexane.

## Optimizing Twist Boat Cyclohexane

Optimize this geometry using the minimization variables from the previous optimization.

To minimize the twist boat structure:

1. Choose Geometry Optimization on the Compute menu.
2. Choose OK to begin the minimization using the previous options.

After the minimization finishes, these values appear on the status line:

Energy: 7.22

Gradient: 0.07

## Analyzing the Results

The following table summarizes the energy in kcal/mol, and gradient in kcal/(mol·Ångstrom), for the three different forms of cyclohexane after geometry optimization:

	chair	boat	twist-boat
energy	1.33	8.31	7.22
gradient	0.07	0.08	0.07



The chair and twist-boat structures have energies below the boat structure and the chair form is the global minimum. The absolute energies from these calculations are not meaningful, but you can compare relative energies with experiment:

	present	expt. estimate <sup>a</sup>
$\Delta E$ (boat-chair)	6.98	6.9
$\Delta E$ (twistboat-chair)	5.89	5.3

a. J.B. Hendrickson, *J. Am. Chem. Soc.* **83**, 4537 (1961)  
J.B. Hendrickson, *J. Am. Chem. Soc.* **89**, 7036 (1967)

## Stopping the Log File

Now that you've finished running calculations, you can stop the log file and examine it by using a text editor such as Notepad, or WordPad which are included with Microsoft Windows.

To stop logging your work:

1. Choose Stop Log on the File menu.

The logfile chem.log contains a record of the various optimization runs. You can examine it using a text editor and can also edit the log file for use with plots or documents.

To open the log file:

1. Exit HyperChem without saving your work.
2. Locate and open the text editor application.
3. Choose Open on the File menu of the text editor, e.g. Notepad.
4. Navigate to your HyperChem directory (e.g., c:\hyper7).
5. Choose "Files of type:" as "All Files". Click on chem.log or enter chem.log in the File name: text box. Choose Open.
6. Examine the contents of the file using the scroll boxes.
7. When you have finished, close the text editor without saving your changes.

Alternatively, you can edit the log file for use with plots or other documents.

## Practice Exercises

1. Repeat this lesson using the MM+ force field and compare results.

## For More Information

For more information on force fields, see appendix B, “Force Field Files,” in the *HyperChem for Windows Reference Manual*.

For more information on energy optimization and single point calculations, see Chapter 7, “Chemical Calculations,” in the same manual.



# Lesson 10

## Simulating Dynamic and Equilibrium Behavior

### Skills Covered in This Lesson

- Building and editing molecules
- Using molecular mechanics geometry optimization
- Using periodic boundary conditions
- Superimposing two molecules
- Using molecular dynamics
- Using dynamics playback and averaging
- Using Langevin dynamics and Monte Carlo

This lesson further explores HyperChem's molecular mechanics functionality, and demonstrates how to use molecular dynamics with HyperChem.

Using the AMBER force field you will optimize an alanine zwitterion in both isolation and solution to determine the effect of solvent on the optimal structure. You will also perform molecular dynamics in solution to simulate its behavior in biological systems.

This lesson has five exercises:

In exercise 1, you will create an optimized alanine zwitterion using the building and editing skills you learned in previous lessons.

In exercise 2, you will use periodic boundary conditions to solvate the zwitterion.

In exercise 3, you will use superposition to visually compare the isolated molecule and the solvated molecule.

In exercise 4, you will use molecular dynamics to anneal the system to obtain a lower energy minimum.

In exercise 5, you will use Langevin dynamics to simulate the presence of a solvent during annealing, and Monte Carlo to sample configurations from a Boltzmann-weighted distribution.

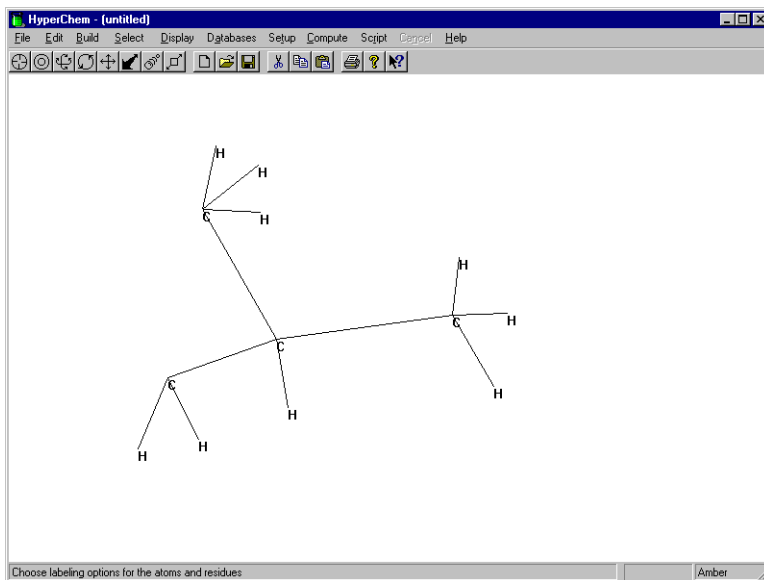
## **Exercise 1: Creating the Isolated Alanine Zwitterion**

### **Sketching the Alanine Zwitterion**

Although you could easily use the HyperChem amino acid library to display alanine, as a practice exercise you will build it from scratch.

To sketch the alanine zwitterion:

1. If necessary, reopen HyperChem.
2. Double-click on the Drawing tool to open the Default Element dialog box.
3. Turn on Allow Arbitrary Valence and Explicit hydrogens.
4. Choose carbon and then close the dialog box.
5. Turn Show Hydrogens on.
6. Set the label option to label atoms by symbol.
7. L-click on the Drawing tool to get into drawing mode.
8. Draw the following sketch.



Notice that one carbon has only two hydrogens. You will edit these hydrogens to become oxygens.

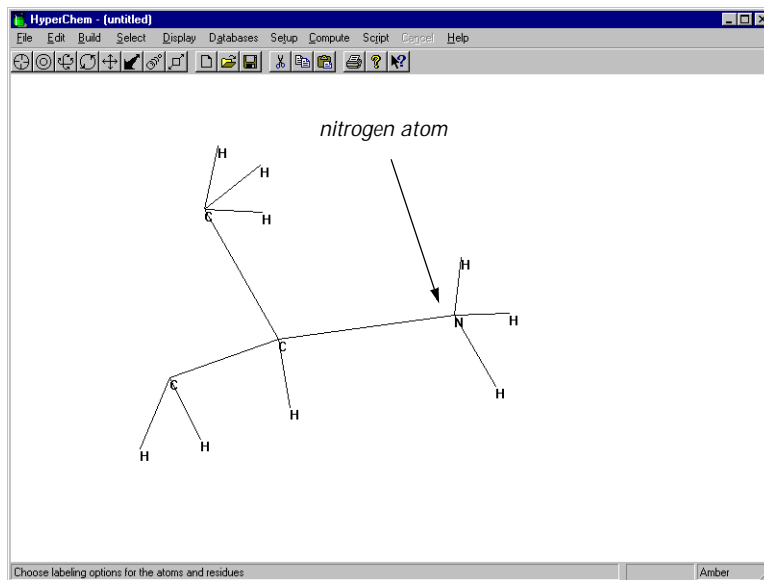
## Editing the Structure

This structure is only a hydrocarbon and you need to edit the structure to create the alanine zwitterion. Specifically, you need to change some atoms into nitrogen or oxygen and conjugate the carboxyl group.

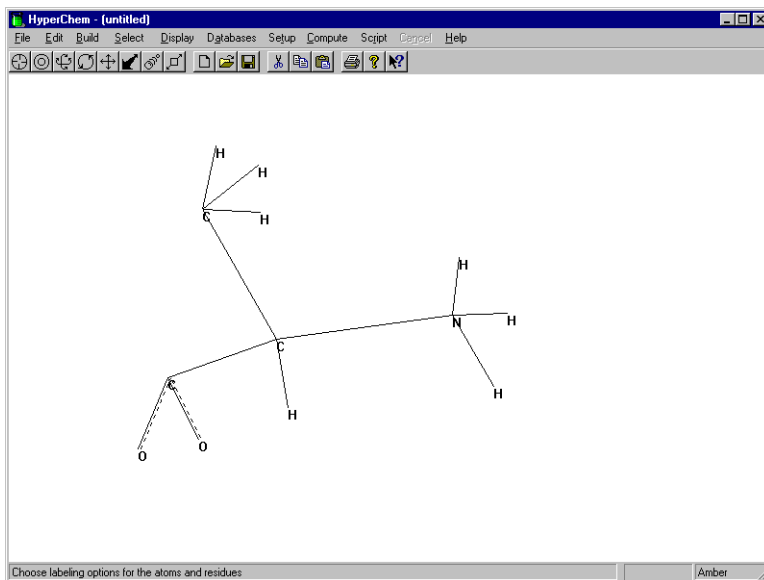
To edit the structure:

1. Open the Default Element dialog box and change the default element to nitrogen.
2. L-click on one of the methyl carbons to turn it into a nitrogen atom as shown below:

Exercise 1: Creating the Isolated Alanine Zwitterion



3. Change the default element to oxygen.
4. L-click on the two hydrogens that will become part of the carboxylate group to turn them into oxygen atoms.
5. Double-click on each of the carbon-oxygen bonds to turn them into conjugated bonds.
6. Your drawing should look like this:



## Assigning Atomic Charges

The next step is to place formal charges on some atoms to approximately represent the charge distribution in the zwitterion. You can obtain the atomic charges with HyperChem using the following methods:

- If a molecule is built from a template, such as a PDB file, the template might contain charges
- By performing a quantum mechanical calculation
- By making an approximation of the charges

In this exercise, you make a simple approximation of a formal charge of +1.0 on the nitrogen atom and charges of -0.5 on the oxygens. This results in nonzero values for these three atoms.

To assign atomic charges:

1. Get into selection mode and set the select level to atoms.
2. Make sure Multiple Selections is turned on.
3. Select the nitrogen atom.



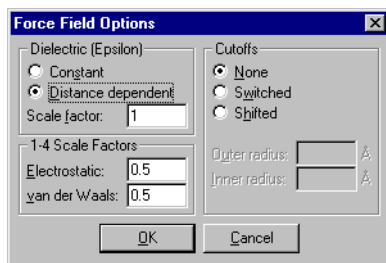
4. Choose Set Charge on the Build menu and assign a charge of **1.0**.
5. R-click in an empty area to clear the selection.
6. Select both oxygen atoms and assign a charge of **-0.5**.  
This sets a charge of -0.5 for each of the oxygen atoms.
7. R-click in an empty area to deselect the atoms.

## Choosing the Force Field

The next step is to choose a force field. You should choose a force field before you invoke the Model Builder because the Model Builder assigns atom types to each atom according to the force field that you specify. Alternatively, you could explicitly assign atom types any time by using Calculate Types on the Build menu. However, the simplest way is to assign the correct force field before you invoke the Model Builder.

To choose the force field:

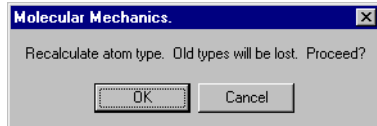
1. Select Molecular Mechanics on the Setup menu.  
The Molecular Mechanics Force Field dialog box opens.
2. Select AMBER, then L-click on Options.  
The Force Field Options dialog box opens.
3. Use the following default values, then choose OK.



Except for cutoffs and the dielectric functional form, these values are standard for the AMBER force field.

4. Choose OK to close the Force Field Options dialog box.
5. Choose OK to close the Molecular Mechanics Force Field dialog box.

6. If the following dialog box appears, choose OK.

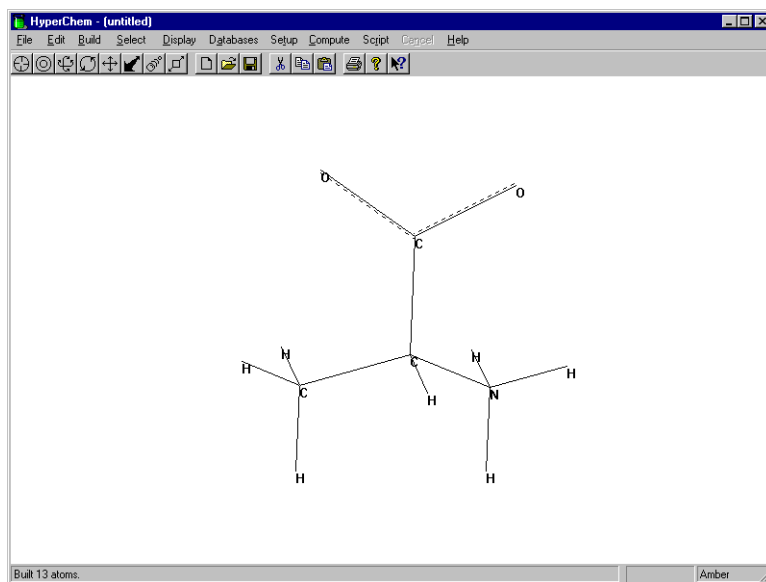


## Building and Exploring the 3D Structure

You are ready to build the 3D model for alanine and look at its conformation.

To build and orient the structure:

1. Double-click on the Selection tool to invoke the Model Builder. HyperChem builds and displays a first approximation of the alanine zwitterion.
2. Rotate and translate the structure to look at its conformation.



The model-built structure is only an approximation of the structure of the alanine zwitterion.

3. Open the Labels dialog box from the Display menu and choose Chirality for atoms. If the central carbon is labelled R rather

than S, then switch to the drawing tool, hold down the [Shift] key and click on the central carbon.

You can use [Shift]+L-click with the drawing tool to rotate two neighbors, thus changing chirality. You can also draw wedges to specify conformation or chirality; see Chapter 8 of the Reference Manual for details.

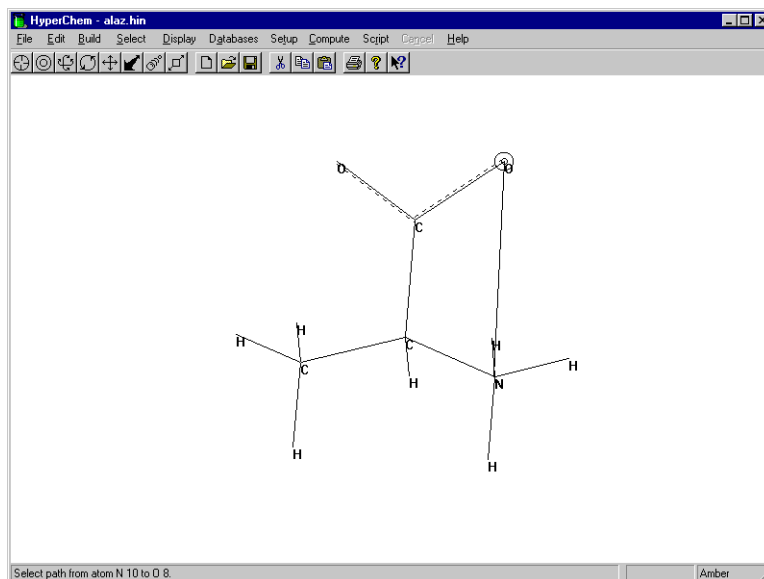
4. Change the atom labels back to Symbol and switch back to the selection tool (if you changed it to the Drawing tool in the last step).
5. Save the structure as alaz.hin.

To measure its structural properties:

1. Get into selection mode and L-drag from oxygen atom to oxygen atom.

The Model Builder built the O-C'-O angle at 120 degrees.

2. R-click in an empty area to clear the selection.
3. L-drag from the nitrogen atom to one of the oxygen atoms.

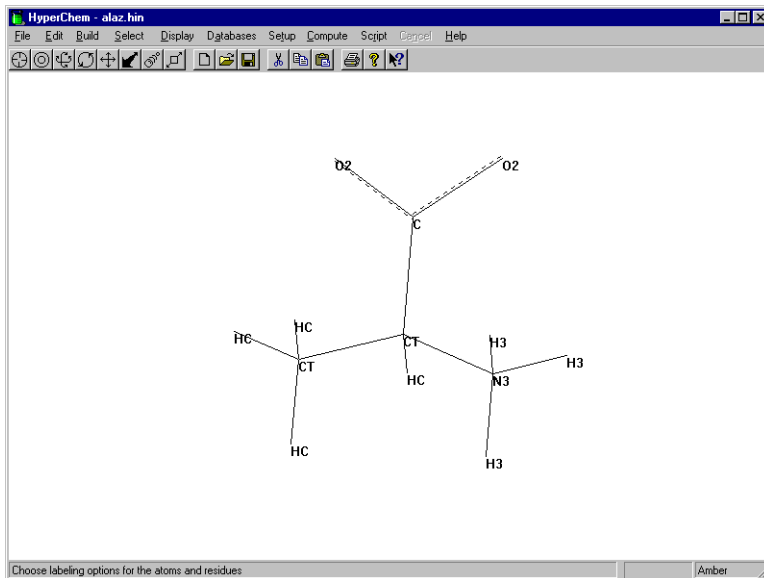


The O-C'-C<sup>α</sup>-N torsion angle is  $\pm 60$  degrees or  $\pm 120$  degrees, depending on the order that you used to draw the atoms in the structure.

4. R-click in an empty area to deselect the torsion angle.

To display the AMBER atom types:

1. Open the Label dialog box and set the atom label type to Type.



## Performing a Single Point Calculation

Now, perform a single point calculation to compute the energy and gradient.

To perform a single point calculation:

1. Choose Single Point on the Compute menu.

After a few moments, the calculation finishes and the status line reports the results:

Energy= 74.59

Gradient=95.70

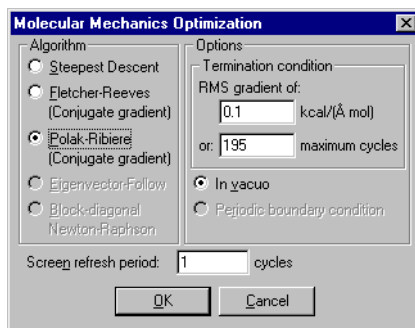
## Optimizing the Isolated Molecule

The large energy and gradient indicate that the model-built structure is far from optimal. The principal strain is due to the C-N and C-O distances.

To optimize the structure:

1. Choose Geometry Optimization on the Compute menu.

The Molecular Mechanics Optimization dialog box opens:



2. Fill in the dialog box with the values shown, and choose OK to start the calculation.

After a few moments, the minimization is completed.

The status line displays the following values:

Energy = 18.10

Gradient = 0.10

3. When the optimization is complete, remeasure the O-C'-O angle and the N-C<sup>α</sup>-C'-O torsion:

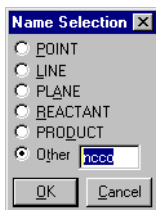
O-C'-O angle = 123.9°

N-C<sup>α</sup>-C'-O torsions = -51 and 99 degrees°

Recall that for the unoptimized structure, the O-C'-O angle measured 120 degrees and the N-C<sup>α</sup>-C'-O torsion angles measured ±60 degrees and ±120 degrees.

Later in this lesson, you use the N-C<sup>α</sup>-C'-O angle to demonstrate molecular dynamics features of HyperChem, so save this angle as a named selection.

- With the angle still selected, choose Name Selection on the Select menu.
- Select Other, enter **ncco**, then choose OK.



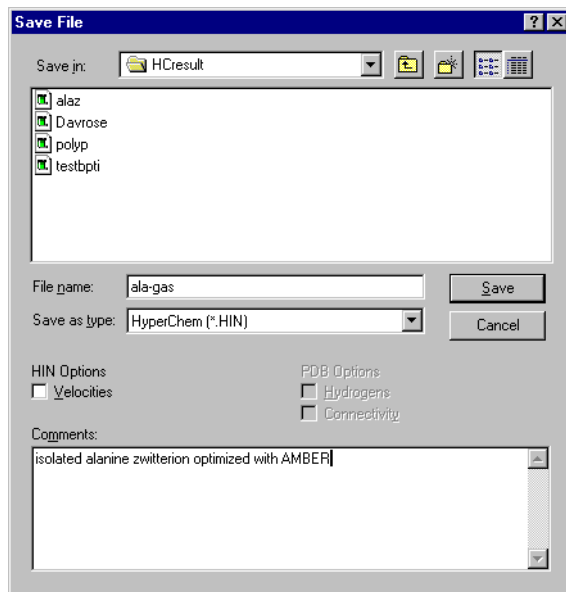
4. R-click in an empty area to deselect the angle.

## Saving the Structure

At this point you should save the optimized isolated alanine structure so that you can compare it with the solvated structure you create next.

To save the structure:

1. Open the Save File dialog box by pressing [Ctrl]+[S] as an alternative to using the menu choice.
2. Make sure HIN is selected as the File format.
3. Name the file ala-gas.hin.
4. L-click in the Comments box. As shown in the following illustration, enter comments for the file:



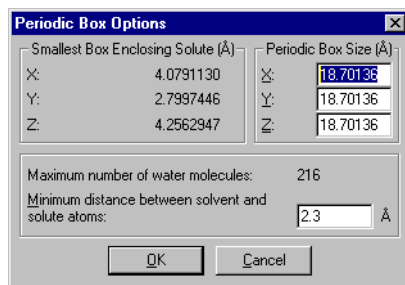
5. Choose OK.

## Exercise 2: Solvating the Structure

HyperChem lets you place a molecular system in a periodic box of water molecules to simulate behavior in aqueous solution, as in a biological system. In this exercise, you solvate the alanine zwitterion you created in Exercise 1.

Before you continue, follow these steps:

1. Remove labels from the display by using the default settings in the Labels dialog box.
2. Choose Periodic Box on the Setup menu to open the Periodic Box Options dialog box.



## Setting up Periodic Boundary Conditions

To solvate a system with HyperChem you specify a rectangular box or cube of equilibrated water molecules. You define the dimensions of the box, place the solute in the center, and define the minimum distance between the solvent and solute atoms. HyperChem eliminates all water molecules with atoms that come closer to a solute atom than the specified distance.

### Specifying the Periodic Box Size

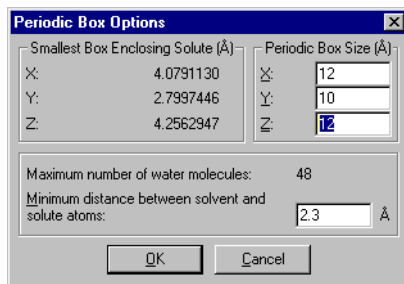
There are several factors to consider when you define the size of the periodic box. For example, it must be larger than the dimensions for the Smallest box enclosing solute that appear in the Periodic Box Options dialog box.

Box dimensions at least twice the largest solute dimension avoid solute-solute interactions when using the default cutoffs. The solute-solute interactions can also be avoided with a smaller box by lessening the cutoff radius after solvation (so that the largest solute dimension plus the cutoff radius and range is less than the smallest box dimension). The reference cube of water is formed from an equilibrated cube of  $18.70136\text{\AA}$  and its nearest images, so you shouldn't use dimensions larger than  $56.10408\text{\AA}$ . Dimensions that are multiples of  $18.70136\text{\AA}$  minimize initial bad contacts.

To choose the periodic box size:

1. Specify a box size of 12.0 by 10.0 by 12.0 Ångstroms, as shown in the following illustration.





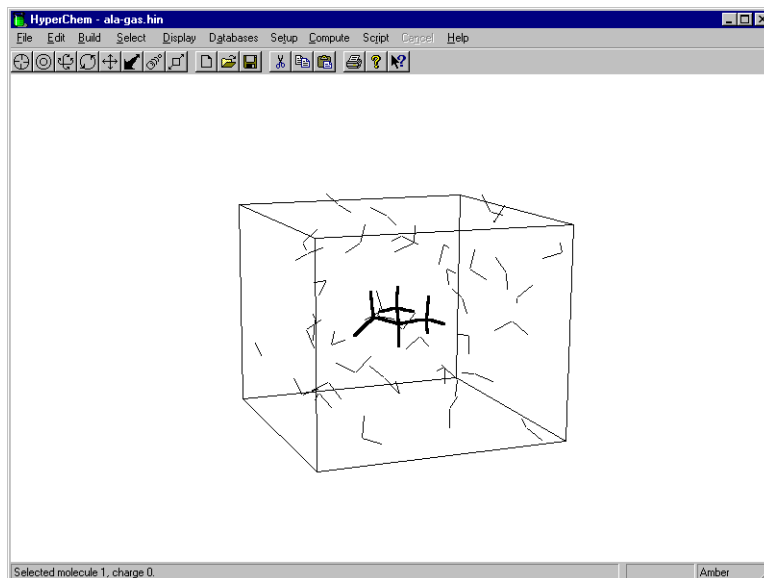
2. Choose OK.

This places the structure in the center of the box surrounded by 40 water molecules.

*Note:* Because the orientations of the water molecules in the box are not symmetrical, you may end up with fewer than 40 water molecules after the ones that are too close to your structure are deleted. If this happens, the results that you get from calculations may not be quite the same as those given in this lesson. This will depend on how your structure is oriented before you solvate it.

## Displaying the Solvated System

1. Choose Rendering on the Display menu.
2. Turn on Perspective in the Rendering/Sticks dialog box, and then choose OK.
3. Set the select level to Molecules.
4. Select a bond or atom of the alanine molecule, and rotate the whole system to look like this:

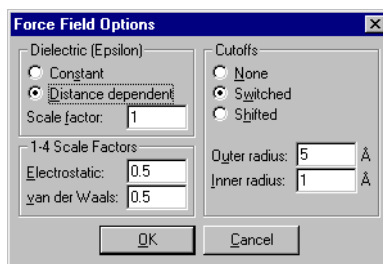


## Adjusting Cutoffs and Dielectric Options

When you use periodic boundary conditions, you need to check the options for cutoffs and dielectric in the Force Field Options dialog box.

To check the options for cutoffs and dielectric:

1. Choose Molecular Mechanics on the Setup menu.
2. Choose Options to open the Force Field Options dialog box.
3. Look at the Cutoffs options at the right of the dialog box.



When you choose the Periodic box menu option, HyperChem automatically changes the options for cutoffs to Inner and Outer options, which are more appropriate for the solvated sys-

tem with the nearest-image periodic boundary conditions. The Outer cutoff is set to one-half of the smallest box dimension, and the Inner cutoff is set to 4Å less to ensure that there are no discontinuities in the potential surface.

4. Now look at options for Dielectric, which are at the top of the dialog box.

When you use explicit solvent, you should use a constant dielectric.

5. Select Constant instead of Distance dependent.

*Note:* The dielectric constant is not automatically changed because the new setting is stored in the chem.ini file.

6. Choose OK to close the Force Field Options dialog box.
7. Choose OK to close the Molecular Mechanics Force Field dialog box.

## Optimizing the Solvated Molecule

The next step is to optimize the solvated system using periodic boundary conditions. Since the alanine molecule has an optimum geometry (at least in isolation), the optimization primarily relaxes the solvent. Changes in alanine reflect the difference between isolated and in solution optimal structures.

### To optimize the molecule in solution:

1. Get into selection mode and R-click in an empty area to deselect the alanine.

When there is an active selection, an optimization is performed on the selection only. Therefore if you started the optimization with alanine selected, only the position of the alanine atoms is allowed to change and the water molecules are constrained.

To optimize the complete system and allow reorganization of the solvent structure in the presence of alanine, you must clear the selection.

2. Choose Geometry Optimization on the Compute menu.

This opens the Molecular Mechanics Optimization dialog box.

3. Choose OK to start the optimization of the solvated alanine, including the water molecules.

The calculation takes a few minutes.

4. When the calculation finishes, the following values appear on the status line:

Energy = -1102.46

Gradient=0.09

*Note:* If you have fewer than 40 water molecules in your system, you will get different values.

The optimized structure for the solvated system might only be a local minimum. In a system with many degrees of freedom, such as this one, there might be many minima and it can be very difficult to locate the global minimum.

When there are enough degrees of freedom, it is possible that any static conformation is insignificant, and that only a statistical treatment of many low-energy conformations is appropriate.

Later in this lesson, you learn that the current structure for the solvated alanine is only a local minimum, not a global one. Nevertheless, you save this structure for later use.

5. Save the structure as ala-liq.hin.

## Exercise 3: Using Superposition

The system currently displayed is a minimized structure for the alanine zwitterion in water. It is instructive to compare the structure with the corresponding isolated optimized structure. Instead of comparing the molecules by measuring individual structural properties, you use HyperChem's superposition feature to visually compare the two structures by superimposing.

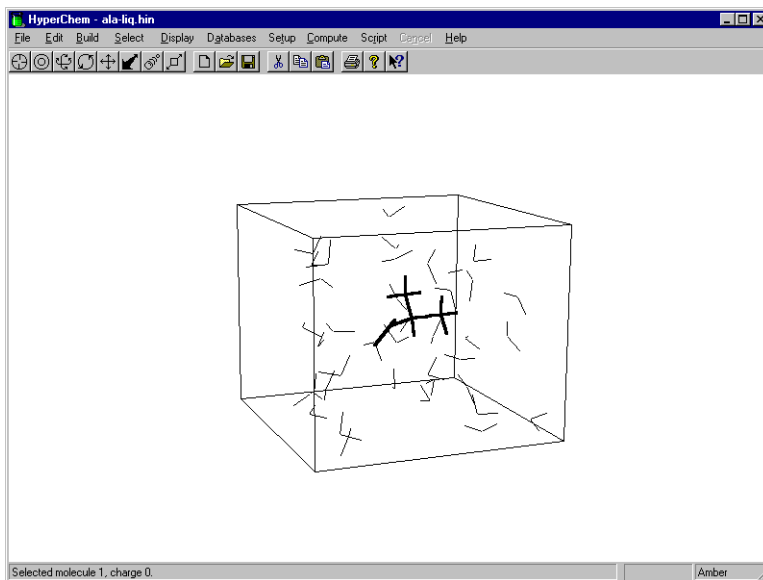
### Deleting the Water Molecules

Before superposing, delete the water molecules from the isolated structure.

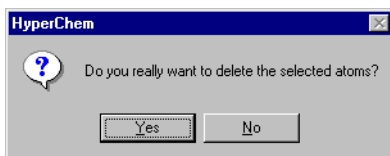
To delete the water molecules:

1. Get into selection mode and set the select level to Molecules.
2. L-click on the alanine.

This selects the alanine.



3. Choose Complement Selection.  
This selects only the water molecules.
4. Choose Clear on the Edit menu.
5. If a dialog box appears asking if you want to delete the selected atoms, choose Yes.



6. Choose Show Periodic box so that it is not set and only the alanine molecule is shown.
7. Save the file as ala-sol.hin.

## Merging the Two Systems

The next step is to merge this system with the isolated system. Before you merge the two systems, you should select the current system so that you can differentiate it on the workspace after the merge.

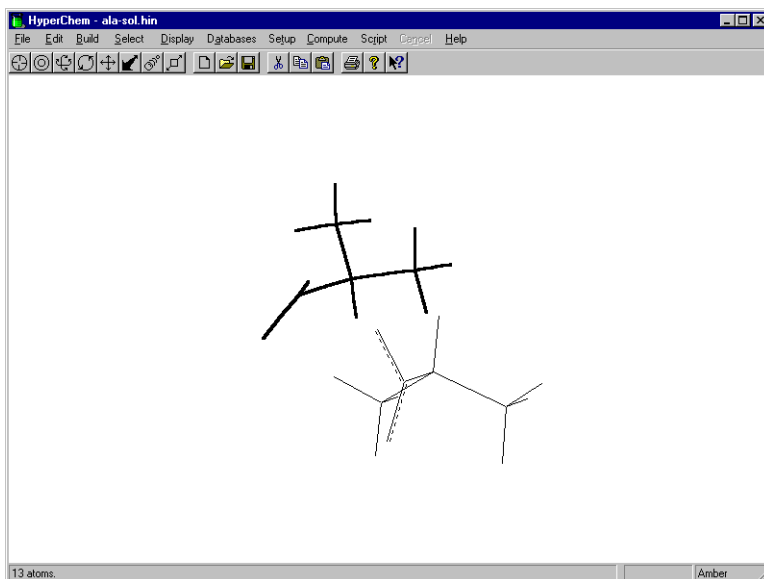
To merge the two systems:

1. Select the alanine.
2. Choose Name Selection on the Select menu and save the selection as **solvated**.

This makes it convenient for you to display this selection later if you want to recover it.

3. Choose Merge on the File menu and open ala-gas.hin.

Both the isolated and the solvated structures should now be on screen. The solvated structure should still be selected.



Merge lets you combine the current system and another system that has been stored in a file. This union becomes the current system.

4. If necessary, use the Translation tool with the right mouse button until both structures are in full view.

At this point, use labels and color to differentiate the molecules.

5. Color the selected solvated structure yellow and label the molecule by symbol.

*Note:* If Thick line is not being used as the Selection option in the Preferences dialog box, the solvated structure does not appear yellow until it is deselected.

6. Choose Complement Selection on the Select menu.

The solvated system is deselected and the isolated alanine is selected.

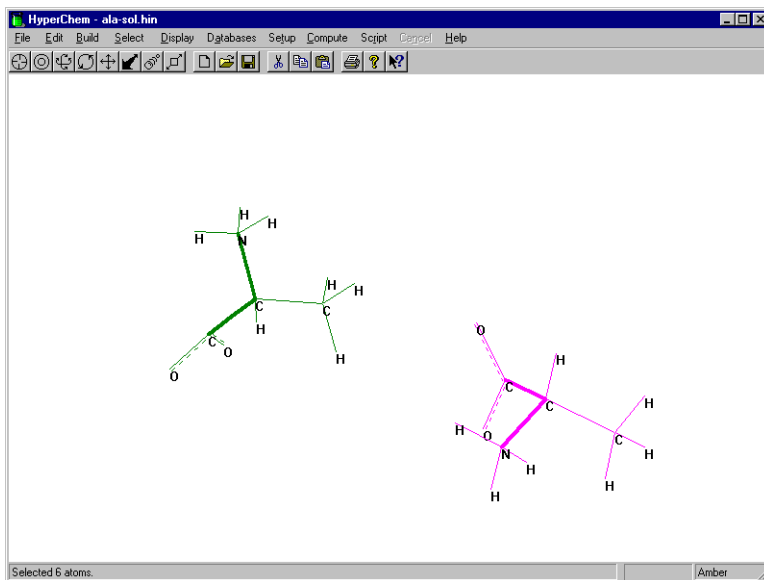
7. Color the selected isolated alanine violet and label it by symbol.
8. R-click in an empty area to deselect the isolated alanine.

## Superposing Molecules

When a system includes at least two molecules, as in this example, HyperChem can superpose the molecules. Superposition is based on the selection of three non-collinear atoms in each molecule. To begin, the first selected atoms of each molecule are made coincident. Next, the bonds between the first and second selected atoms of each molecule are made collinear. Finally, the third selected atoms of each molecule are put in the same plane.

To perform the superposition:

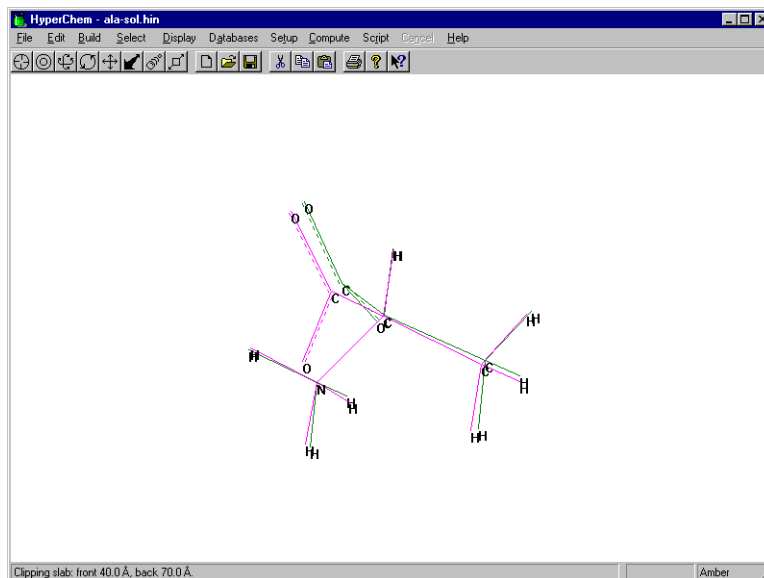
1. Get into selection mode, set the select level to atoms, and turn Multiple Selections on.
2. L-drag the N-C<sup>α</sup>-C' bond angle of each molecule. You might have to rotate the structures to do this.



3. Choose Overlay on the Display menu.
4. R-click in an empty area to clear the selection.
5. Press the space bar to center the superposed structures in the workspace.

The current system should now look similar to this:





6. Choose Save As on the File menu and save the superposed molecules as ala-sup.hin.

## Exercise 4: Simulated Annealing

Here you use molecular dynamics to anneal the system to obtain a lower energy minimum. You must first retrieve the solvated alanine zwitterion system to perform molecular dynamics with it.

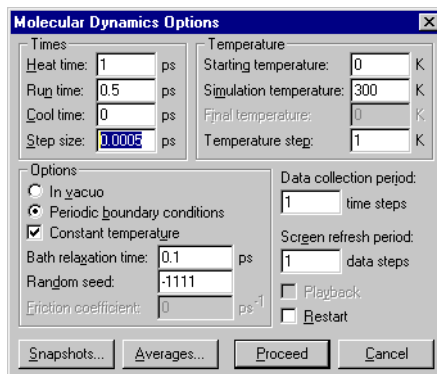
To retrieve the solvated system:

1. Open the file ala-liqsn1.hin.
2. If necessary, turn on Show Periodic Box to display the periodic box you defined earlier.
3. If necessary, remove labels from the display.

To set up the dynamics simulation:

1. Choose Molecular Dynamics on the Compute menu.

The Molecular Dynamics Options dialog box lets you set the options for a molecular dynamics calculation. Take a moment to look at the options.



A dynamics run has three optional phases: heat, run, and cool. The first phase occurs over a simulation period of heat time, using the starting temperature to set initial velocities with rescaling of velocities at temperature increments to reach the simulation temperature.

In the middle phase, velocities are rescaled only if constant temperature is selected.

The final phase occurs over a simulation period of cool time, with rescaling of velocities at temperature increments to reach the final temperature.

2. Set the Heat time to 0.1 picoseconds.
3. Set the Temperature step to 30K.

Because the gradient is very small in the starting system (corresponding to a near-zero temperature), a short heat time of 0.1 picoseconds is used to raise the temperature from a starting temperature of 100 K to the simulation temperature of 300K incrementing by a temperature step of 30K.

4. Set the Run time to 0.5 picoseconds.

The temperatures are used here to set up initial atom velocities or to adjust atom velocities. This kinetic energy might be converted into potential energy during simulation, causing the calculated temperature to drop. If the temperature eventually rises (as it does in this example), it means that potential energy is being converted into kinetic energy as the system moves to a more stable conformation.

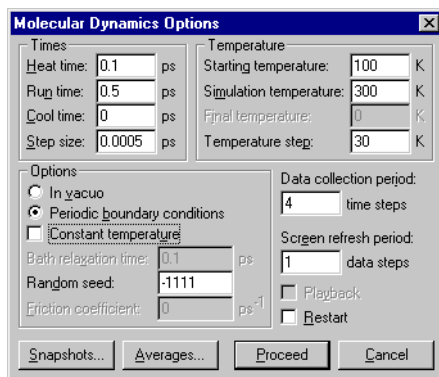
5. Make sure the Periodic boundary conditions option is on.

This option is automatically turned on by HyperChem when periodic boundary conditions are used, but it can be overridden by selecting In vacuo.

6. Set the Step Size to 0.0005 picoseconds.
7. Set the Data Collection period to 4.

For systems such as this that have explicit hydrogen atoms, a 0.5 fs step size is appropriate for accurately integrating the hydrogen stretching motion. You can update the screen at any number of time steps to display real-time molecular dynamics. Depending on the specific hardware you are using, collecting data too often can slow down the dynamics run, so the Data collection period is set to 4.

8. Turn Constant Temperature off.
9. The settings in the dialog box should look like this:



## Setting up Playback Dynamics

The next step is to set up playback dynamics. Playback dynamics is convenient because it saves time-consuming dynamics simulations for you to analyze later.

Molecular dynamics simulates the evolution of a system over time, producing a trajectory of atomic positions and velocities. These calculations can be very time consuming, and for statistically meaningful results, long simulation times corresponding to many thousands of time steps might be required.

Often you'll want to save the calculated trajectory for later playback or analysis, rather than having to repeat the simulation.

To set up playback dynamics:

1. Choose Snapshots at the bottom of the Molecular Dynamics Options dialog box.

The Molecular Dynamics Snapshots dialog box opens.

2. Enter **ala-run** as the filename.

HyperChem generates two files with the prefix ala-run.

One file, ala-run.hin, is a HIN file that contains a snapshot entry. Another file, ala-run.snp, is a binary file containing atomic coordinates and velocities.

3. Use a Snapshot period of 1 data step.
4. Choose OK to return to the Molecular Dynamics Options dialog box.

## Setting up Averaging from Molecular Dynamics

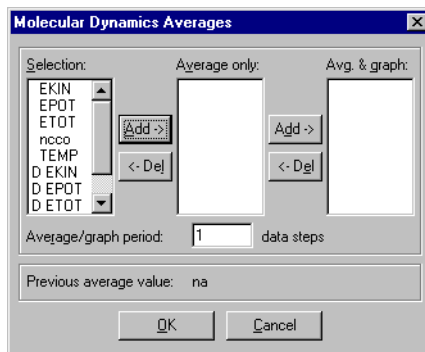
Molecular dynamics is often used to obtain macroscopic information by sampling a microscopic simulation over a long period of time. It is also useful to track energetic and geometric quantities as the simulation proceeds to see if the system has stabilized enough for sampling to be statistically valid or not.

*Note:* When you start the Averaging function, HyperChem creates a file with the extension .csv in the current working directory (see page 173). However, if you are reading a snapshot file from a CD, the current directory is on the CD; you cannot create a file there. You will need to copy the snapshot file to your hard drive first.

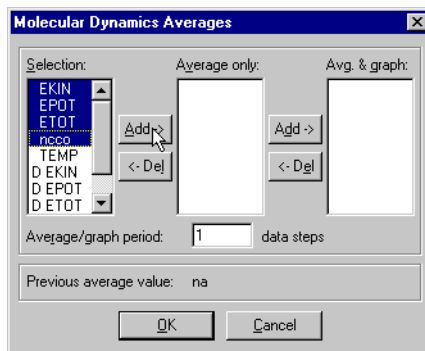
To set up averaging:

1. Choose Averages to open the Molecular Dynamics Averages dialog box.

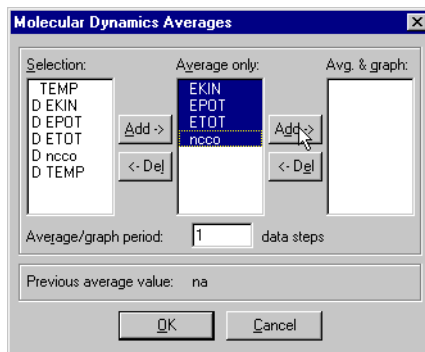
Exercise 4: Simulated Annealing



2. L-click on EKin, EPOT, ETOT, and ncco in the Selection box, then L-click on Add to move them to the Average only box.



3. L-click on EKin, EPOT, ETOT, and ncco in the Average only box, then L-click on Add to move them to the Avg. & graph box.



This specifies a plot of total energy, potential energy, kinetic energy, and the N-C<sup>α</sup>-C'-O torsion angle you saved as a named selection earlier.

HyperChem generates a file named ala-run.csv (or chem.csv if playback dynamics is not set up), which records the quantities being averaged in a format that can be easily read by plotting or spreadsheet programs for analysis.

## Proceeding with Dynamics

1. Choose OK to return to the Molecular Dynamics Options dialog box, then choose Proceed to start dynamics.

A graph titled Molecular Dynamics Results opens on the workspace.

2. Move the graph so that you can watch the simulation.

While the calculation is running, you can change the view of the system by using the rotation, translation, zoom, and clip tools. You can also use applications in other windows, but this can slow down the simulation.

3. As the simulation continues, choose Rescale to rescale the values being plotted.

Once the heating phase is finished, (when energy is being added) the total energy remains constant, and the kinetic energy mirrors the potential energy.

**Caution:** You cannot recover the plot if you L-click on Done. To recover the plot, you must use replay dynamics.

In the early part of the run after heating, the potential energy and temperature should be fairly stable, but then the system temperature rises. The potential energy during the run eventually drops significantly. The molecular dynamics at room temperature lets the system surmount energy barriers on the potential surface and find regions of lower potential energy that would not be reached by a minimization algorithm. When the dynamics run has completed, you can optimize the system to determine a new minimum.

After approximately 10 minutes, the run finishes.

## Reoptimizing the New Structure

Now that the dynamics run is finished, you can optimize the structure to determine a new local minimum.

1. Choose Geometry Optimization on the Compute menu.
2. Choose OK to perform a molecular mechanics optimization using the options from the previous calculation.

After the optimization finishes, the status line shows a structure that has lower energy than the solvated local minimum found earlier.

The results demonstrate that molecular dynamics can be used to reach thermodynamically favored conformations separated from the starting conformation by energy barriers. Thus molecular dynamics can be used to explore conformational space.

In this type of simulation, the box size and the simulation period are chosen before the simulation begins. When you want averaged results, you can use a Langevin and Monte Carlo simulation. Such sampling should begin once the system is equilibrated, often judged by kinetic and potential energy fluctuations. Monitoring about consistent values for a long period of time is necessary.

### Exercise 5: Langevin and Monte Carlo Simulations

## Exercise 5: Langevin and Monte Carlo Simulations

In this tutorial you use Langevin and Monte Carlo simulation methods to further explore conformations of the alanine zwitterion system. The Langevin Dynamics method simulates the motion of molecules subjected to random collisions and frictional forces and can be used to model solvated systems without explicitly including solvent molecules. It provides information on the time evolution of the molecular system. The Monte Carlo method is used to simulate equilibrium properties.

### Part 1: Langevin Dynamics

A Langevin Dynamics simulation is set up in much the same way as a Molecular Dynamics simulation, with small differences due to the addition of the friction coefficient.

1. Retrieve the gas phase alanine zwitterion, which you saved as ala-gas.hin (see Exercise 1 in this section).
2. Set the Molecular Dynamics Options as in the preceding Exercise.
3. Set the Friction coefficient to 0.05 ps<sup>-1</sup>.

When the friction coefficient is set to zero, HyperChem performs regular molecular dynamics, and one should use a time step that is appropriate for that method. With larger values of the friction coefficient, motions that occur over a short time are less important and larger time steps can be used.

4. Set the Time step to 0.001 picoseconds.
5. Set up Averages and Snapshots as in the preceding exercise.
6. Choose Proceed to start the Langevin Dynamics simulation.

As with the Molecular Dynamics technique, heating and cooling phases may also be added to a Langevin dynamics simulation. Equilibration of the system is not crucial for locating additional low-energy structures, but it is important when comparing simulated values with experimental properties.

## Part 2: Monte Carlo Simulation

The Monte Carlo method samples configurations from a Boltzmann-weighted distribution at a given temperature. At elevated temperatures, this technique may be used to move the molecular system of interest across potential energy barriers. In this exercise we employ the Monte Carlo method followed by geometry optimization as an additional conformational search technique.

### To set up the Monte Carlo simulation:

1. First retrieve the solvated alanine zwitterion system, which was saved as ala-liq.hin.
2. Choose Monte Carlo on the Compute menu.

The Monte Carlo Options dialog box allows you to set up the Monte Carlo simulation parameters. In this example we will run a constant temperature simulation with 1000 steps. For some systems, it may be useful to add optional heat and cool phases.



3. Set the Run Steps to 1000. Set Heat and Cool steps to 0.
4. Set the maximum step size, Max Delta, to 0.05 Å.

This sets the maximum trial atomic displacement. If it is too small only limited sampling of new configurations will occur. If it is too large, unphysical configurations may be generated. This reduces the efficiency of the simulation.

5. Set the Simulation Temperature to 300K.
6. Make sure that Periodic boundary conditions is on (✓).
7. Set the Data Collection period to 4 steps.

### Setting up Monte Carlo Playback

8. Set up a snapshot file, if desired.

Playback of the Monte Carlo “trajectory” is set up in the same way as Dynamics playback. See “Setting up Playback Dynamics” on page 170. As with Dynamics, analysis of the simulation may be conveniently carried out at playback. For example, you can stop the playback at the point in a simulation where the potential energy falls, and save the structure at that point. You can also change the graphing and averaging selections before playing back a run.

### Setting up Averaging from Monte Carlo

9. Set up Averages and Graphing as desired.

The Monte Carlo Averages setup is identical to Molecular Dynamics Averages setup. There is an additional parameter that you can monitor in Monte Carlo: the acceptance ratio. It appears as ACCR on the list of possible selections in the Monte Carlo Averages dialog box; DACCR, the rms deviation of ACCR from its mean, appears also. The acceptance ratio is a running average of the ratio of the number of accepted moves to attempted moves. Optimal values are close to 0.5. Varying the step size can have a large effect on the acceptance ratio. For a more detailed discussion, see the Monte Carlo section of the Computational Chemistry manual.

## Proceeding with Simulation and Analysis

There are no significant differences between the procedures for using HyperChem to analyze Monte Carlo simulations, and those for using it to analyze Molecular Dynamics simulations. The Monte Carlo Results window can be copied to the Windows clipboard and pasted into other applications. Average values may be conveniently retrieved by opening the Monte Carlo Averages dialog box. L-click on one of the properties previously selected for averaging; the average value is reported just below.

## Advanced Exercises

1. Follow these steps to view playback dynamics:
  - Open the file ala-run.hin.
  - Open the Molecular Dynamics Options dialog box and choose Playback. Choose Snapshots to determine which part of the simulation to replay.
  - Choose Proceed to start playback dynamics.  
The simulation will playback substantially faster than the initial dynamics calculation.
2. Create named selections for two, three, or four atoms, and use playback dynamics to plot the values.

## For More Information

For more information on molecular mechanics geometry optimization, periodic boundary conditions, and molecular dynamics, see Chapter 7, “Chemical Calculations,” in the *HyperChem for Windows Reference Manual*.



# Lesson 11

## Molecular Orbital Calculations

### Skills Covered in This Lesson

- Calculating wave functions
- Plotting the electrostatic potential, total charge density, and molecular orbitals
- Calculating atomic charges
- Using structure alignment

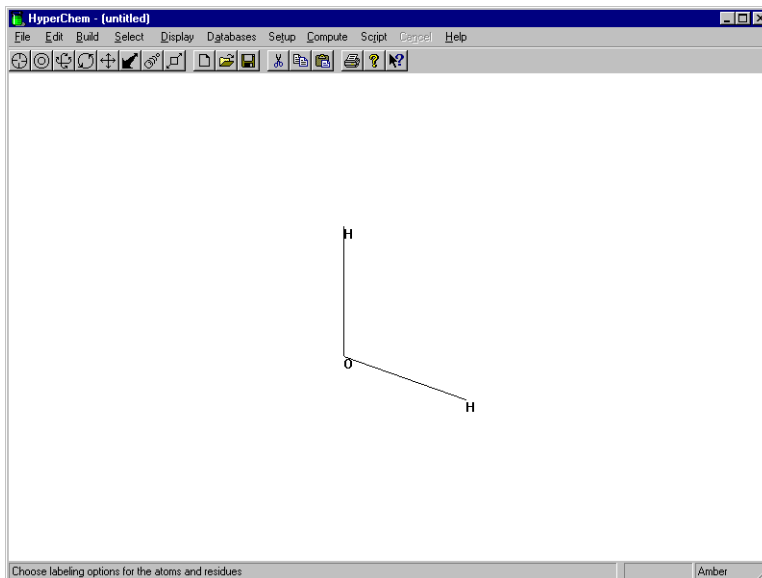
This lesson acquaints you with the various types of semi-empirical quantum mechanical calculations that HyperChem can do. The molecule you use in this lesson is water.

### Creating the Water Molecule

To create a model-built structure of water:

1. On the Display menu, make sure Show Hydrogens is turned on and Perspective is turned off in the Rendering dialog box.
2. In the Default Element dialog box, turn off Explicit Hydrogens, choose Oxygen, then choose Close.
3. L-click in the workspace with the draw cursor to draw an oxygen atom.
4. Double-click on the Selection tool to invoke the Model Builder. The Model Builder builds the water molecule and adds hydrogens.
5. Label the molecule by symbol.

Your drawing should look like this:



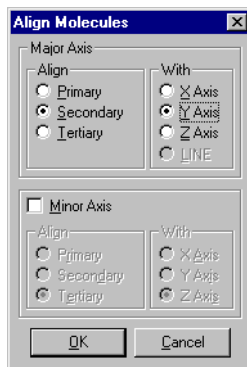
The bond angle of this model-built structure is the tetrahedral 109 degrees. You could improve this structure, but for this exercise you calculate a wave function for this model-built structure.

## Using Structure Alignment

Before you calculate the wave function, place the molecule in a standard orientation by aligning the secondary inertial axis of water (the symmetry axis in the plane of the molecule) with the y axis.

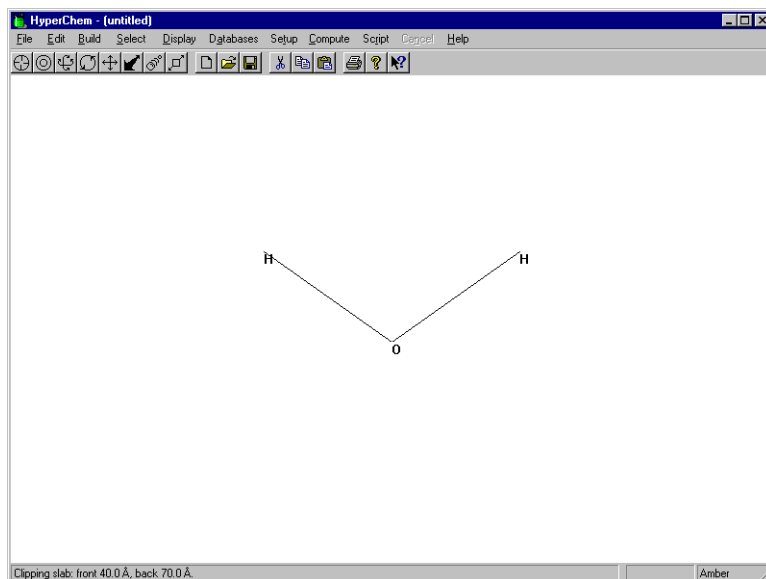
To do a structure alignment:

1. Choose Align Molecules on the Edit menu.
2. In the Align box choose Secondary, and in the With box choose Y Axis.



3. Make sure Minor is turned off; you do not want to specify a secondary alignment.
4. Choose OK.

The water molecule should be oriented in the workspace like this:



5. Save the structure as h2o.hin.

## Displaying Atomic Charges

In Lesson 10 you manually assigned atomic charges using Set Charges on the Build menu. Charges can be assigned automatically if the molecule you are using is read in from a template file, such as a PDB file.

In this lesson, you obtain the atomic charges by doing a semi-empirical quantum mechanics calculation.

First, show that no charges are currently assigned for the water molecule.

To display charges:

1. Open the Labels dialog box.
2. Choose Charge as the option for Atom, then choose OK.

The labels show that no charges are set for the molecule.

## Calculating a Wave Function

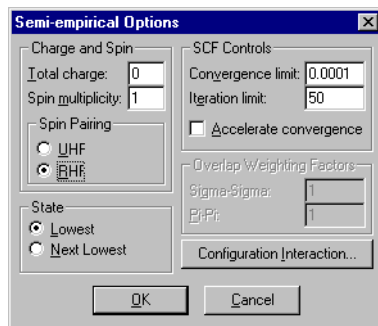
A wave function is built from molecular orbitals and describes the distribution of electrons in a molecule. In this exercise, you calculate a wave function for the entire water molecule. Later, in Lesson 12, you compute a wave function for a selected part of a structure.

To calculate a wave function:

1. Choose Semi-empirical on the Setup menu.
2. Choose CNDO as the method, then choose Options.

You could choose one of the other available methods as well, but for this example use CNDO as the computation method.

3. Use the following values in the Semi-empirical Options dialog box:



These options set up a Restricted Hartree-Fock (RHF) calculation with a convergence limit of 0.0001, which means that the calculation ends when the difference in energy after two consecutive iterations is less than 0.0001 kcal/mol.

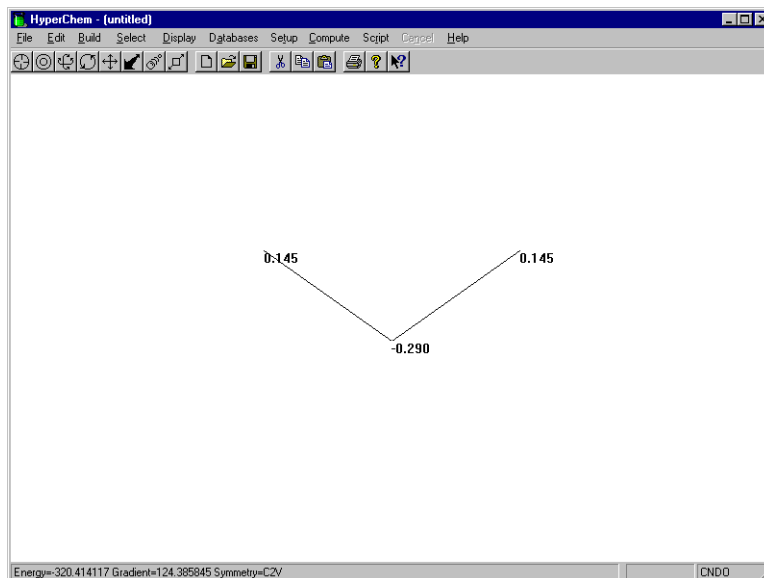
The iteration limit, which is set to 50, is the maximum number of iterations that is used to reach a self-consistent wave function. Total charge is 0 and Spin multiplicity is 1.

The calculation is performed on the lowest state without special convergence acceleration.

4. Choose OK to close the Semi-empirical options dialog box, then choose OK to close the Semi-empirical Method dialog box.
5. Choose Single Point on the Compute menu.

The resulting energy, gradient, and atomic charges should be as follows:





(Your energy and gradient values may be very slightly different.)

## Plotting the Electrostatic Potential

Now that you have computed the wave function, you can display a contour map of the electrostatic potential. Plotting the electrostatic potential might require more time than computing the quantum mechanical wave function. This is because the values must be computed at a large number of grid points to obtain the contours.

A plot of the electrostatic potential is overlaid onto the HyperChem workspace and remains until you move or change the molecule or display another plot.

The plane of the contour map is always parallel to the viewer's XY plane (the plane of the workspace) at a screen value of Z that you can set by using these guidelines for subset selection:

- If an atom is selected when a contour map is requested, the contour plane is relative to the atom. With an offset of zero, the plane passes through the atom.
- If a bond is selected, the plane is relative to the center of mass of the bond.

- In general, the plane is relative to the center of mass of the selection (or, if no selection has been made, for the whole system). The offset specifies the distance the plotting plane is from the center of mass of the selection.

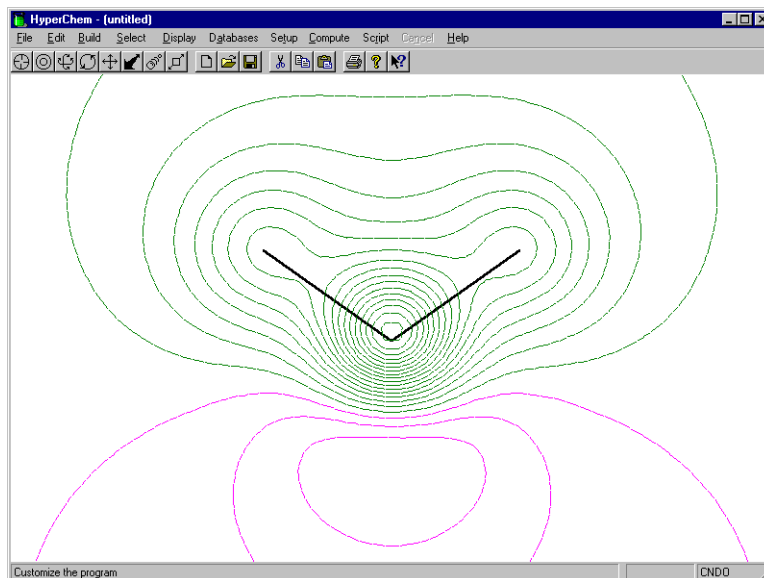
To plot the electrostatic potential:

1. Remove the labels from the display.
2. Choose Plot Molecular Properties from the Compute menu to open the Plot Molecular Properties Options dialog box.
3. Choose Electrostatic Potential as the type of graph you want to display and choose 2D Contours.

HyperChem will calculate and display the two-dimensional contour plot for the electrostatic potential.

4. Click on the Contour Grid tab to show the Contour Grid property sheet.
5. Set Horizontal grid points to **60**.
6. Set the Vertical grid points to **60**.
7. Set the Contour levels to **30**.
8. Set the Plane Offset to **0.5 Å**.
9. Choose OK to begin the calculation.
10. After a few moments, the electrostatic potential appears.

*Note:* The number of contour lines displayed will vary with the starting value and increment value used; your display may differ somewhat from the following picture.



11. Reopen the Plot Molecular Properties Options dialog box and specify 3D Isosurface instead of 2D Contours.

This will display the electrostatic potential as a 3D surface.

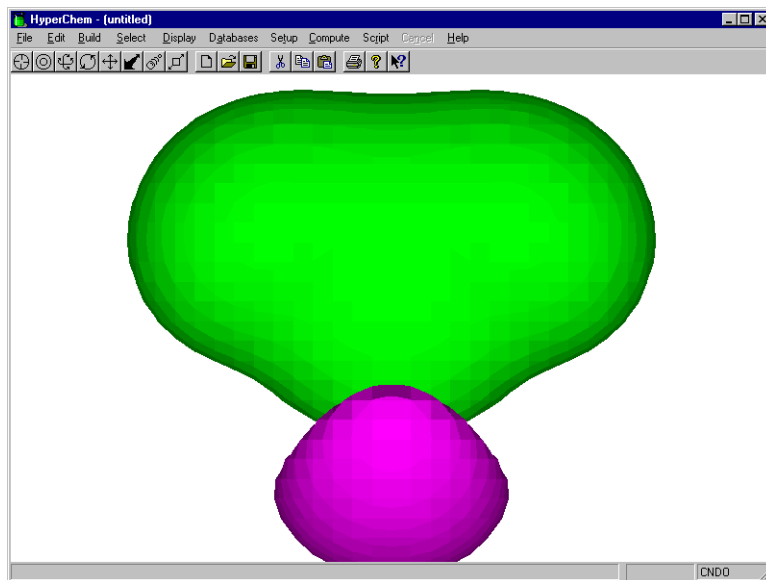
12. Click on the Isosurface Rendering tab to show the Isosurface Rendering property sheet. Specify an Electrostatic potential contour value of 0.1 in the text box, and select Shaded surface as the Rendering type.

The surface will show where in 3D space the electrostatic potential has a value of 0.1  $e/a_0$ . The surface will be shaded and solid.

13. Click on the Isosurface Grid tab to show the Isosurface Grid property sheet. Specify that the grid should be a Medium grid.

This sheet allows you to select whether the surface should be drawn with a few points (for a rough picture that is drawn quickly) or many points (for a smooth picture that takes much longer to draw).

14. Click on OK to start the calculation of the surface. This will take a few moments to complete. Then the surface is shown.



15. Reopen the Plot Molecular Properties Options dialog box and specify 3D Mapped Isosurface.

This will display a surface drawn with a value specified by Total Charge Density (see the following section) but colored according to the electrostatic potential.

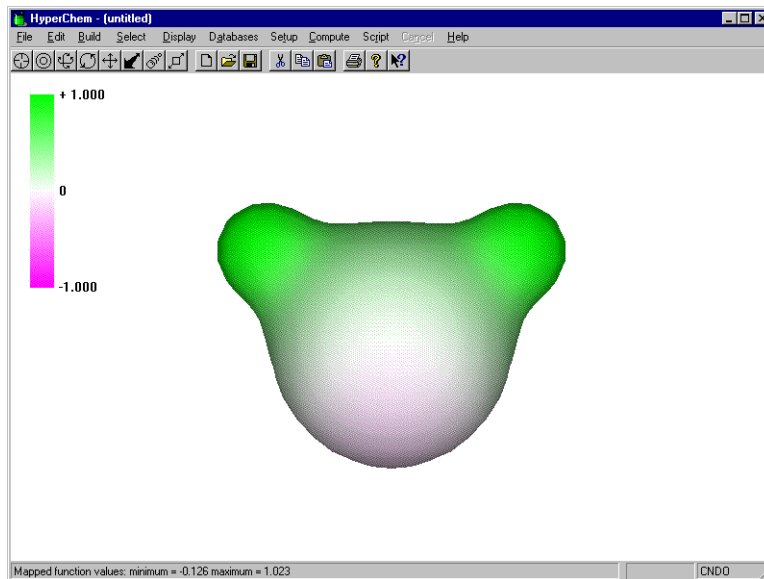
16. Click on the Isosurface Rendering tab to show the Isosurface Rendering property sheet. Specify a Total charge density contour value of 0.135 in the text box, and select Gouraud shaded surface as the Rendering type.

The surface will show where in 3D space the total charge density has a value of  $0.1 \text{ e/a}_0^3$ . The surface will be very smooth but will take some time to draw.

17. Click on the Mapped Function tab to show the Mapped Function Options property sheet. Specify a Minimum value of -1.0 and a Maximum value of 1.0. Make sure that the Display Range Legend option is on (marked with a ✓).

The surface will be colored according to the electrostatic potential, ranging from one color for a value of  $1.0 \text{ e/a}_0$  to another color for a value of  $-1.0 \text{ e/a}_0$ . A legend will be drawn, showing how the surface colors represent different values.

- Click on OK to start the calculation of the surface. This will take a few moments to complete. Then the surface is shown.



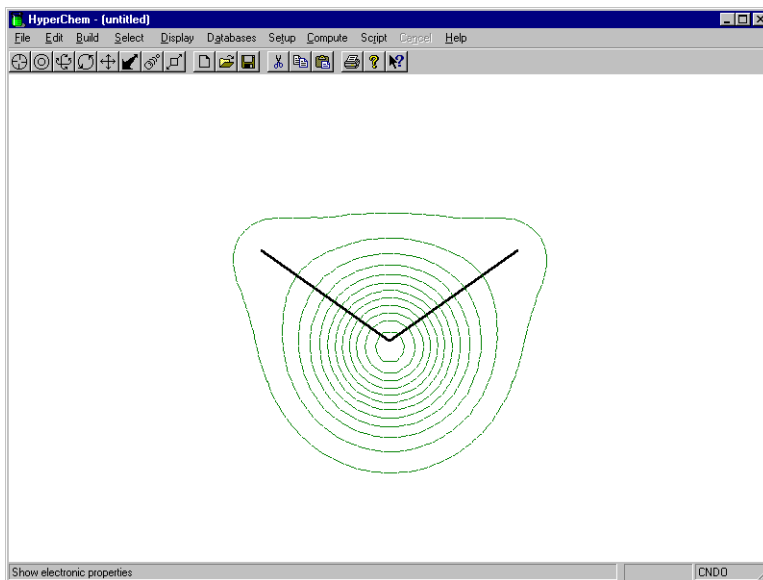
## Plotting the Total Charge Density

You can also display a contour map of the total charge (electron) density. Since CNDO and the other semi-empirical methods available with HyperChem do not include inner-shell electrons (for example, the 1s electrons of oxygen in water), the charge density shown is only the valence charge density.

To plot total charge density:

- Open the Plot Molecular Properties dialog box.
- Choose Total Charge Density as the type of plot, choose 2D Contours, and then L-click on OK to close this dialog box.

After a few moments, the total charge density appears.



The total charge density can also be drawn as a 3D Surface, but not as a Mapped 3D Surface.

## Plotting the Total Spin Density

Total spin density can be plotted in the same way as the total charge density. However, in the case of the water molecule (and all other systems in which all electrons are paired), the value of the total spin density is 0 everywhere, and there is nothing to plot. Spin density can be calculated and displayed for chemical systems with unpaired electrons.

## Plotting Individual Molecular Orbitals

You can also plot any individual molecular orbital—either the signed orbital itself or the squared value equivalent to the probability distribution—for an electron in that orbital. The orbitals are specified relative to the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO).

For this exercise, you plot the molecular orbitals in the order of increasing energy. For water, CNDO uses a basis set of six atomic

orbitals (2s, 2p on oxygen and 1s on the two hydrogens) and calculates six molecular orbitals. Four of these orbitals ( $2a_1$ ,  $1b_2$ ,  $3a_1$ , and  $1b_1$  starting at lowest energy) are occupied and two ( $4a_1$  and  $2b_2$ ) are unoccupied. The highest occupied molecular orbital (HOMO) is  $1b_1$ , and the lowest unoccupied molecular orbital (LUMO) is  $4a_1$ . The missing inner-shell 1s electrons of oxygen describe the missing  $1a_1$  orbital. Therefore the first orbital,  $2a_1$ , is HOMO-3.

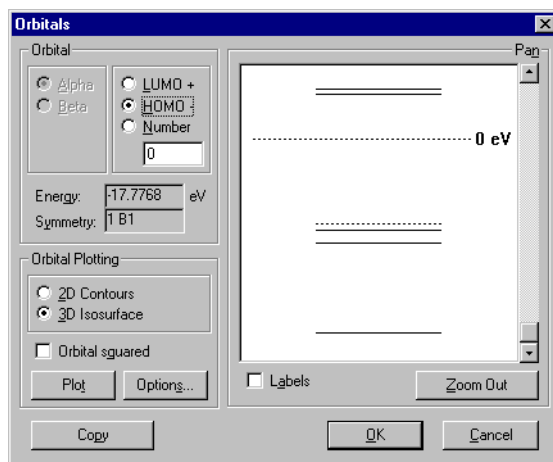
To plot individual molecular orbitals:

1. Choose the Selection Tool and R-click in an empty area to clear the workspace.
2. Open the Orbitals dialog box by choosing Orbitals on the Compute menu.
3. Choose HOMO-, then L-click in the text box for the orbital offset and set the value to 3.

This selects the orbital 3 lower in energy ranking than the HOMO. In this example, it is the lowest calculated orbital. Observe how the lowest level in the orbital energy level diagram changes color or becomes dashed. You could also have chosen the orbital by clicking on the lowest orbital energy level in the diagram.

4. Choose 3D Isosurface.
5. Make sure Orbital squared is not selected.

The Orbitals dialog box should look like this:



## 6. Choose OK.

HyperChem accumulates a grid of values for the orbital  $2a_1$  one “sheet” at a time, and displays the index of each sheet in the status line.

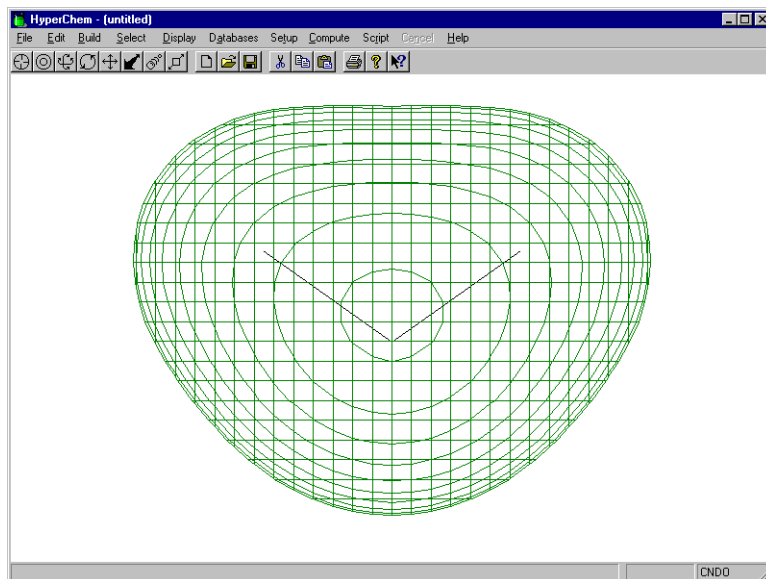
After a few moments, the 3D isosurface appears.

## 7. Open the Isosurface Options dialog box by choosing Isosurface on the Display menu, or by pressing the [F4] key.

This option allows you to change the rendering mode of an isosurface quickly and easily, without needing to recalculate the isosurface data.

## 8. Choose Wire mesh as the Rendering option, use an Orbital contour value of 0.05, and then choose OK.

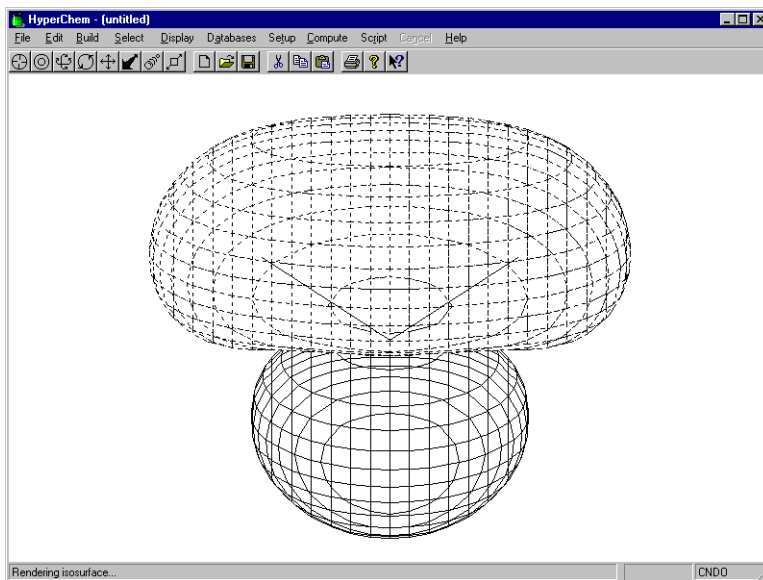
HyperChem displays the symmetric bonding  $2a_1$  orbital of water:



## 9. Reopen the Orbitals dialog box and specify a value of 1 for the HOMO- offset. Click on Options and use the same options in the Electronic Options dialog box as you did in step 7 and step 8, but change the Rendering to Jorgensen-Salem.

This displays the  $3a_1$  orbital:



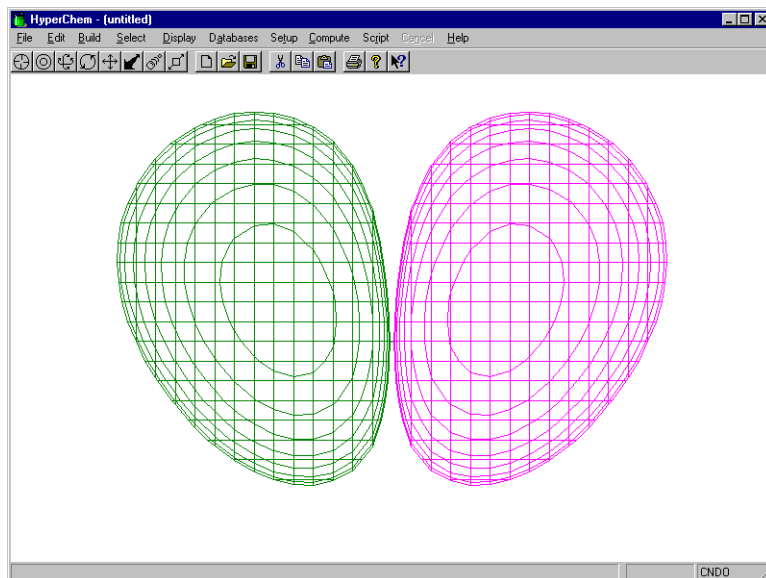


Press the [4] key a few times to rotate the structure. Notice that in this rendering mode, one lobe of an orbital can hide another lobe, but the complete molecule is always visible. If your window color is set to white and bond color is set to black, the orbitals will be drawn with solid lines for positive lobes and dashed lines for negative lobes. This rendering mode is useful for black-and-white printing.

Press the [3] key a few times to restore the original orientation of the system.

10. Repeat the calculation using a value of 2 for the HOMO- offset and choose Lines as the Rendering option in the Options dialog box.

This displays the  $1b_2$  orbital:



Note that the molecule is hidden by the orbital. If you rotate the system with the [I] or [J] key, you can see that one orbital lobe can hide another.

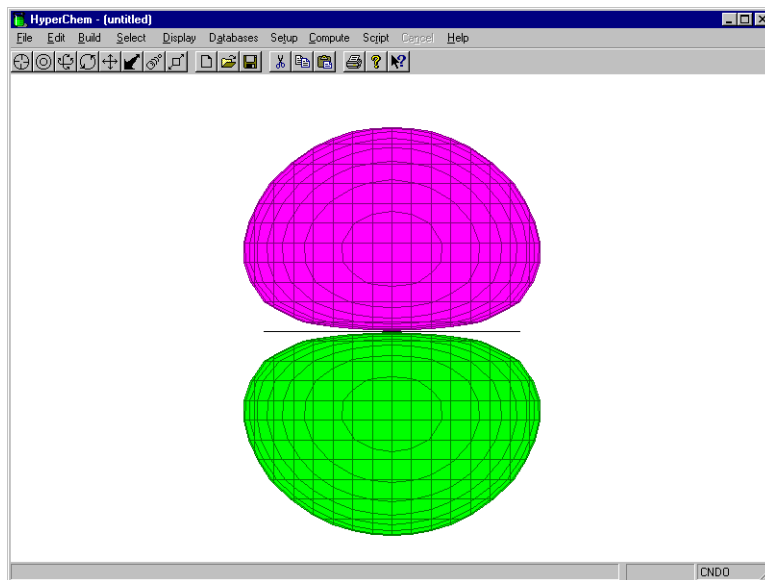
To display the HOMO orbital itself, you need to reorient the molecule so that it is in the XZ plane (the molecule seen only as a line) because the orbital has a node in the XY plane.

11. Choose Rotate on the Edit menu.
12. Choose X Axis, enter an Angle of **90**, choose Apply to Viewer, then choose OK.

Select Display/Show Isosurface, or press the [F3] key. This hides the isosurface. The molecule appears as a line in the workspace, and the orbital plot persists in storage. Pressing [F3] again will restore the plot.

13. Open the Orbitals dialog box and repeat the calculation using a value of **0** for the HOMO- offset.
14. In the Electronic Surface Options dialog box, choose Flat surface. Enter a value of **0.05**, and then choose OK.

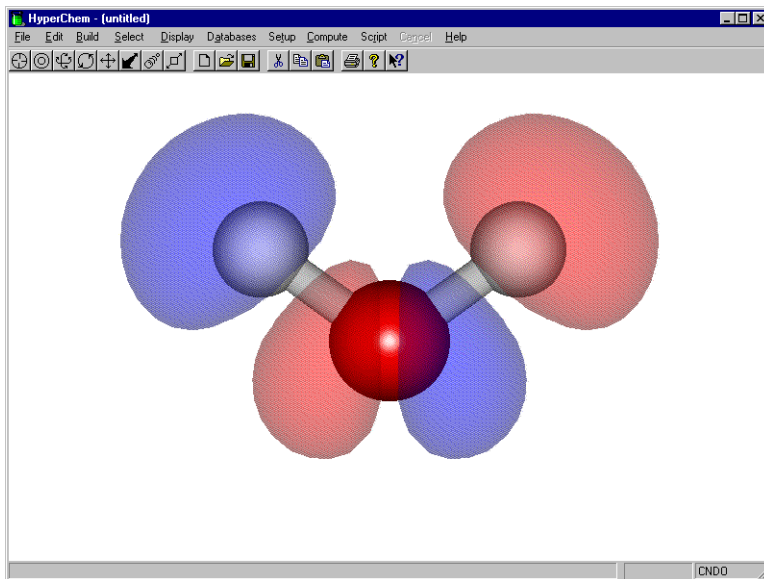
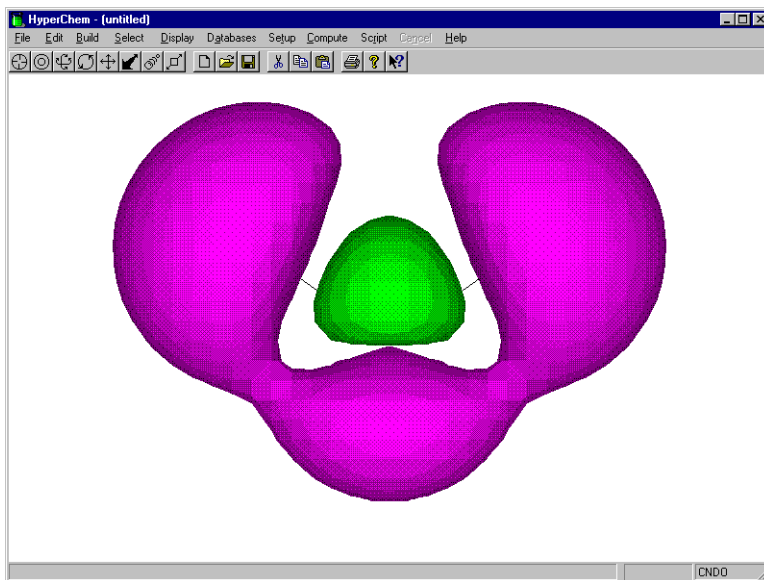
This displays the highest occupied molecular orbital of water, of symmetry  $1b_1$ :



15. Choose Rotate on the Edit menu, enter an Angle of  $-90$ , then choose OK.
16. L-click on LUMO+ and use the offset values 0 and 1 to display the unoccupied orbitals.
17. When the Options dialog box is displayed for these orbitals, choose Shaded surface as the Rendering option and use a value of 0.05 for the first.

For the second, choose Transparent surface for the isosurface rendering, and change the molecule rendering to Balls and Cylinders. Open the File/Preferences dialog box and select the Isosurface Colors property sheet; change the positive and negative colors to red and blue.

These orbitals are better viewed with a value of 0.05. As with other plots, the appearance of the display may vary with the values used.



## Saving the Molecule

You use this water molecule, with calculated charges, in the next lesson, so you need to save it.

To save the molecule:

1. Open the Save dialog box and save the molecule as water.hin.

## Advanced Exercises

1. Cyclopropane is a strained organic molecule. Build it by drawing a triangle of carbon atoms and invoking the Model Builder. The experimental bond lengths of cyclopropane are 1.510Å (CC bonds) and 1.089Å (CH bonds). Perform a RHF optimization of the model-built structure using default convergence criteria with different methods and compare the results with experiment.
2. Start a log file, then build and optimize acetone, which has an experimental heat of formation of -51.9 kcal/mol, and an experimental dipole moment of 2.88 Debyes, using AM1. Look in the log file and see the calculated heat of formation.

## For More Information

For more information on the various types of semi-empirical quantum mechanical calculations that HyperChem can do, see Chapter 7, "Chemical Calculations," in the *HyperChem for Windows Reference Manual*.

# Lesson 12

## Interaction of Water with N-methylacetamide

### Skills Covered in This Lesson

- Using bond torsion constraints
- Optimizing the geometry of a subset
- Using solvent simulations
- Performing wave function calculations to compute atomic charges
- Merging two molecules into a single system

In this lesson, you perform both molecular mechanics and semi-empirical calculations using HyperChem, and calculate atomic charges for subsequent use in molecular mechanics calculations.

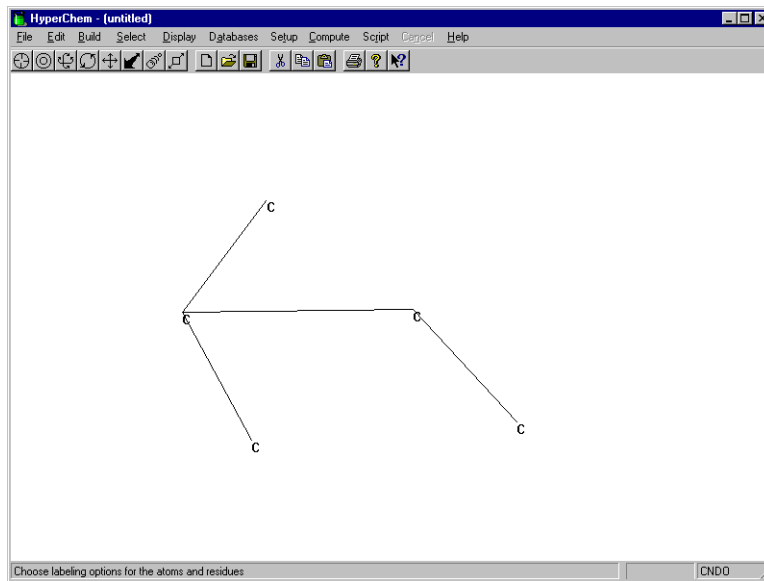
First, you create a model-built structure of N-methylacetamide (NMA), use geometry constraints to change from a *cis* form to a *trans* form, and rebuild the molecule. Next, you compute a wave function for the structure before and after you solvate the molecule with water.

### Creating the NMA Molecule

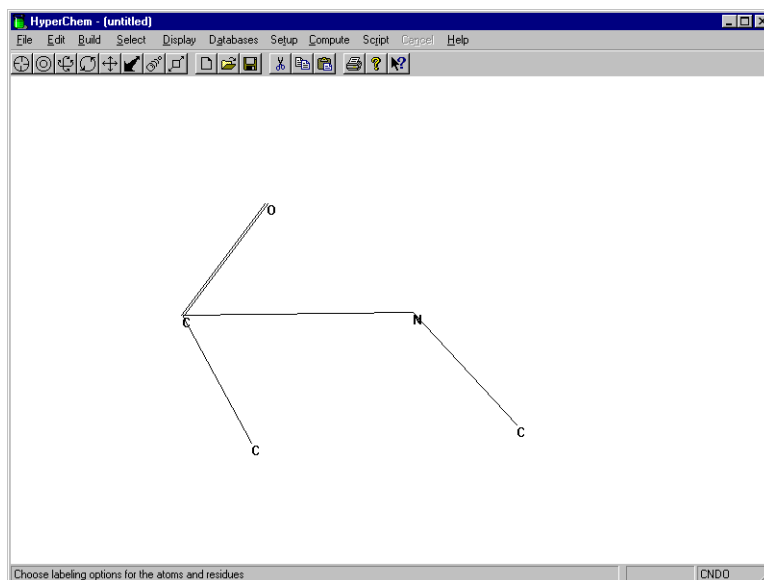
To create N-methylacetamide:

1. Choose New on the File menu to clear the workspace.
2. Open the Label dialog box and set the Atom option to Symbol.
3. In the Default Element dialog box, turn off Explicit Hydrogens, and choose Carbon as the default element.

4. Get into drawing mode, and draw the 2D sketch of the following backbone:



5. Modify the structure to look like this:



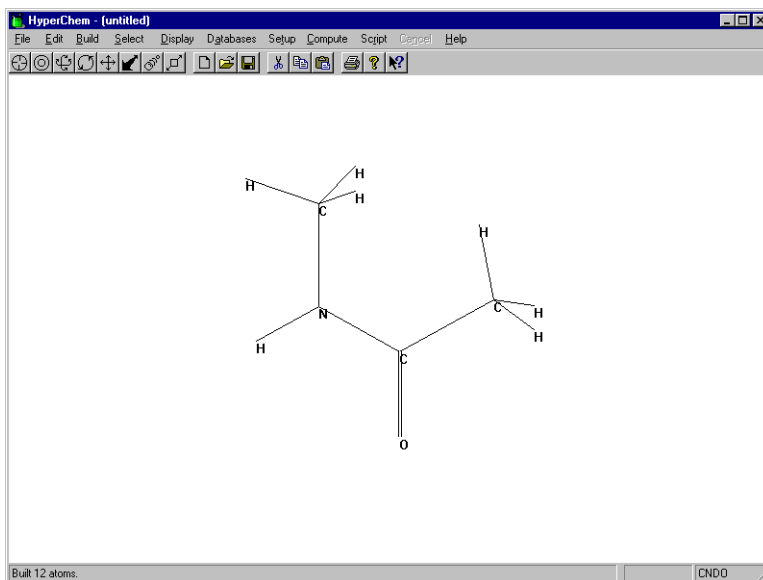
Make the C-O bond into a double bond.

6. Double-click on the Selection tool to invoke the Model Builder and create the 3D structure.

## Changing from *cis*- to *trans*-NMA

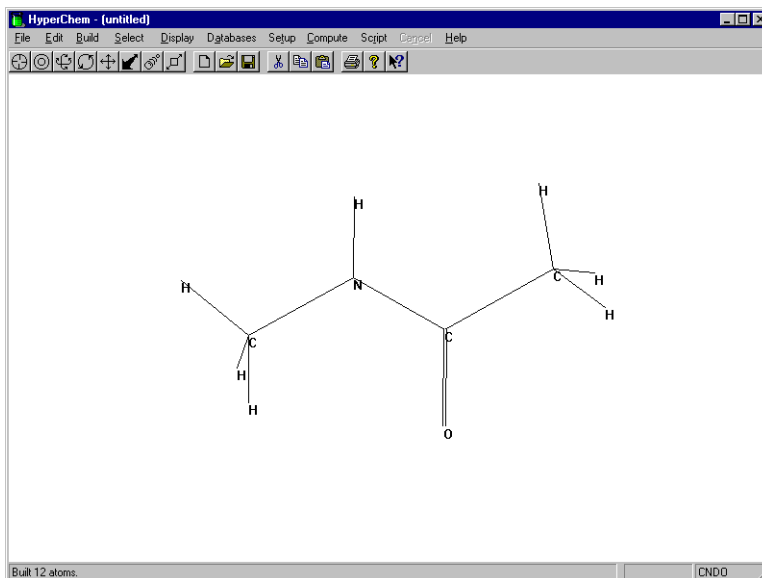
Depending on how you drew the backbone, the Model Builder produced either the *trans*-NMA, or the *cis*-NMA. The Model Builder uses *trans*-dihedral angles as the default.

If you drew the backbone in the order O-C-N-C or C-N-C-O, the vicinal atoms (O and C) would be *trans* and the result would be *cis*-NMA:



If you drew the backbone in the order C-C-N-C or C-N-C-C, the vicinal atoms (C and C) would be *trans* and the result would be *trans*-NMA:



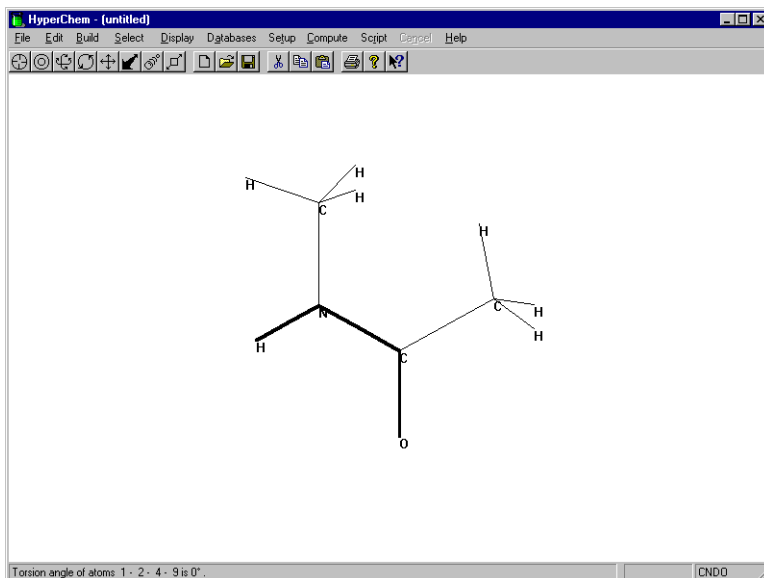


1. Rotate the model-built structure and determine if the structure is *cis* or *trans* form.

If the *cis* form shown in the first illustration is displayed, you need to change it to the *trans* form by adding a bond torsion constraint and rebuilding the structure as described in the following steps. If you built the *trans* form, skip to the next section, “Optimizing the NMA Structure.”

To change the *cis* form to the *trans* form of NMA:

1. Get into selection mode and set the select level to Atoms.
2. L-drag from the oxygen atom to the hydrogen atom attached to the nitrogen to select the four atoms of the peptide dihedral angle:



3. On the Build menu, choose Constrain Bond Torsion.
4. Set the constraint to Trans, then choose OK.
5. R-click in an empty area to deselect the angle.
6. Rebuild the molecule by double-clicking on the Selection tool.

The current system should now be *trans*-NMA.

The following calculations assume orientations of methyl groups such that a CH bond is *trans* to the CO bond, but *cis* to the NH bond. Use Set Torsion on the Edit menu to obtain the torsional conformation as necessary.

## Optimizing the NMA Structure

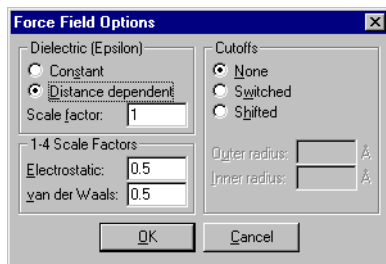
Before you compute the wave function for the molecule, optimize the structure using the AMBER force field.

To optimize the structure:

1. Choose Molecular Mechanics on the Setup menu.

If the warning message box “Recalculate atom types...” appears, choose “OK”.

2. Choose AMBER as the force field, and choose Mechanics Options.
3. Use the following default force field options:



4. Choose OK to close the Force Field Options dialog box.
5. Choose OK to close the Molecular Mechanics Force Field dialog box. You may be warned that it will be necessary to recalculate atom types; if so, choose OK.
6. Choose Select Parameter Set on Setup menu.
7. Choose amber3, then choose OK.
8. If HyperChem asks you if you want to recalculate atom types, choose OK.
9. Choose Geometry Optimization on the Compute menu.
10. In the dialog box, use the default options, but set the RMS gradient to 0.001, then choose OK to start the calculation.

After a few moments, the calculation finishes and HyperChem displays the results on the status line.

## Calculating Atomic Charges

In Lesson 11, you obtained atomic charges for the water molecule by performing wave function calculations. In the following exercise, you calculate atomic charges and use them in a subsequent molecular mechanics computation of NMA with water.

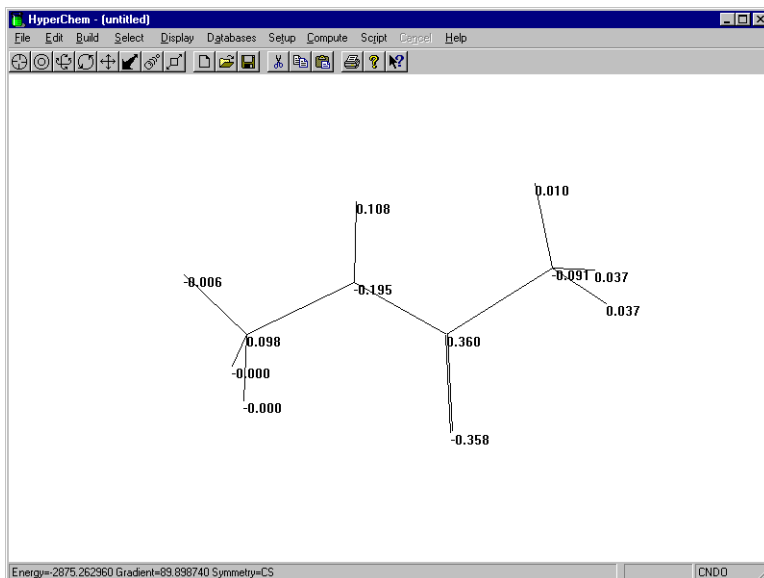
To calculate atomic charges:

1. First, label the structure by charge.

The labels show all atoms have a charge of 0.00.

2. Choose Semi-empirical on the Setup menu.
3. Choose CNDO as the type of method, then choose OK.
4. Choose Single Point on the Compute menu.

HyperChem displays the atomic charges when the calculation is complete. Your charges, energy, and gradient values may be very slightly different from the following:



5. Save this molecule as nma.hin.

## Solvating the Carboxyl Group

### Merging Two Structures

In this section, you merge the NMA molecule with the water molecule from Lesson 11 and calculate their interaction.

To display NMA and the water molecule as one system:

1. Choose Merge on the File menu.
2. Open the file water.hin.

The workspace now shows both the NMA and water molecules.

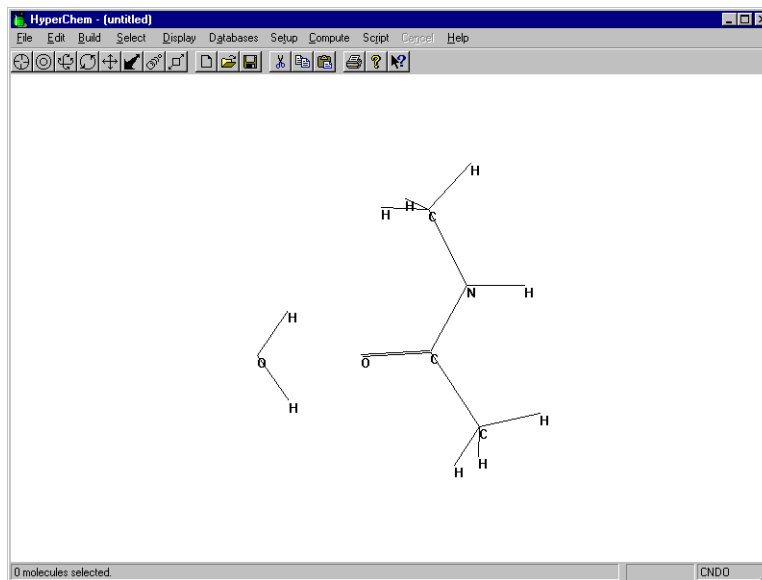
## Moving the Water Molecule

Next you move the water molecule to an initial position that approximates the formation of a hydrogen bond between the oxygen of the carboxyl group and a hydrogen atom of the water molecule. To do this, you select it and orient it relative to the rest of the system.

To move the water molecule:

1. In selection mode, R-click in an empty area of the workspace to make sure nothing is selected.
2. Label both structures by symbol.
3. Position the molecules to obtain the orientation shown in the following illustration.

**Note:** To move one molecule at a time, select the molecule you want to move, L-click on a translation or rotation tool, and R-drag. To select an entire molecule quickly, set “Molecules” in the Select menu, and L-click with the Selection tool on any part of that molecule.



## Optimizing a Subset of the System

In this exercise, you optimize the geometry of the water molecule using molecular mechanics.

When you perform a geometry optimization on a system in which subset selection is active (at least one atom is selected), the position of the selected atoms is modified during optimization. This means if you select only the water molecule when you perform the optimization, the NMA molecule remains fixed as the water moves to an optimum position. This generally is faster than allowing all degrees of freedom. When the water molecule is near an optimum position, you can optimize the whole system, if you want.

To optimize the water subset:

1. Choose Molecular Mechanics on the Setup menu.
2. Choose AMBER as the method, and choose OK to use the default force field options.
3. Select the water molecule if it is not already selected. You may receive a warning message that atom types will be recalculated; if so, choose OK.
4. Choose AMBER as the method, and choose OK to use the default force field options.
5. You may receive a warning message that atom types will be recalculated; if so, choose OK. Choose Geometry Optimization on the Compute menu.
6. Use a RMS gradient of 0.01.

Depending on the initial positioning, the optimization algorithm used, convergence, and so on, it can take a considerable number of iterations to fine-tune the position of the water molecule since intermolecular forces are small compared to intramolecular forces. Finding the correct position of the water molecule requires a finer degree of optimization than is needed for normal bonded interactions.

7. Choose OK to start the calculation.
8. After the water is optimized, deselect the water molecule, and perform a geometry optimization on both molecules.

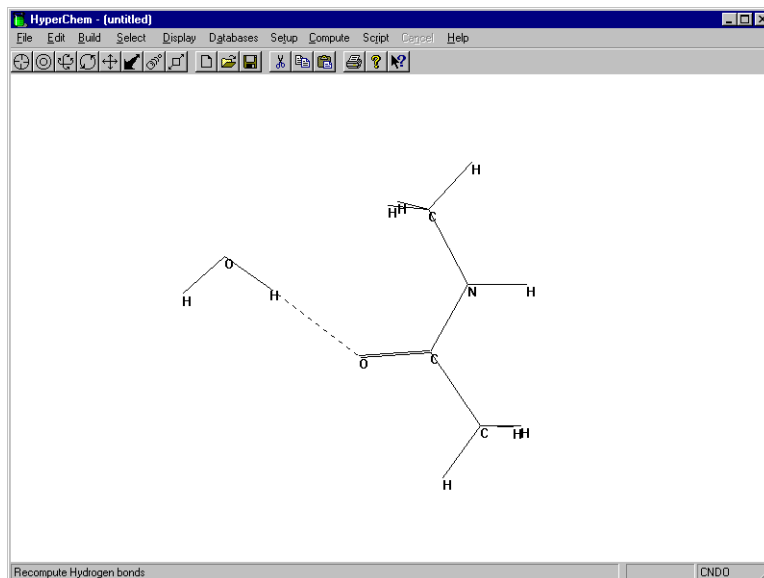
After a few moments, the calculation finishes.

## Recomputing Hydrogen Bonds

To recompute hydrogen bonds:

1. Turn on Show hydrogen bonds and Recompute H Bonds on the Display menu.

The configuration should look similar to this:



## Characterizing the Interacting System

To help characterize the solvation of NMA, you perform a quantum mechanical calculation on the interacting system. You can then compare the results with the earlier monomer calculation on water and NMA.

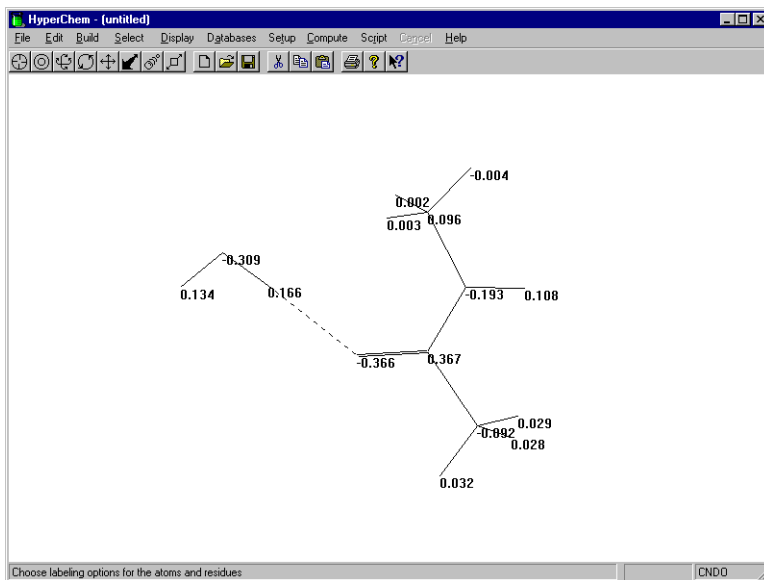
To calculate a wave function for the system:

1. Choose Semi-empirical on the Setup menu.
2. Choose CNDO as the method, then choose OK.
3. Choose Single Point on the Compute menu.

This calculates a wave function for the combined system.

- Label the system by charges to display the computed atomic charges.

This shows the charge rearrangement associated with the formation of the hydrogen bond. Your results should be similar to those shown in the following illustration:



## Advanced Exercises

- Repeat the molecular mechanics optimization with the new charges. Manually include polarization by iterating back and forth from molecular mechanics geometry optimization and semi-empirical calculations of charges. HyperChem rearranges the atomic charges based on the new configuration of atoms.
- Try plotting the charge density for various views of the interacting system.





# Lesson 13

## Electronic Properties of Proteins

### Skills Covered in This Lesson

- Calculating electron density
- Using “classical-quantum” boundaries in calculations
- Calculating wave functions

With HyperChem, you can mix classical and quantum mechanical calculations in the same molecular system. This lets you do quantum mechanical investigations on parts of large molecules, such as the active site of an enzyme, when it would be prohibitive to include the whole system.

This lesson illustrates the calculation of the electron density for the N-terminal region of the small protein bovine pancreatic trypsin inhibitor (BPTI). The type of calculation that can be done on very large proteins is restricted only by the size of the portion of the protein that you treat quantum mechanically.

The protein calculations illustrated in this lesson are done in essentially the same way as the semi-empirical quantum mechanical calculations you did in Lessons 11 and 12, except that a boundary between the “classical part” of a molecule and the “quantum part” of a molecule is specified.

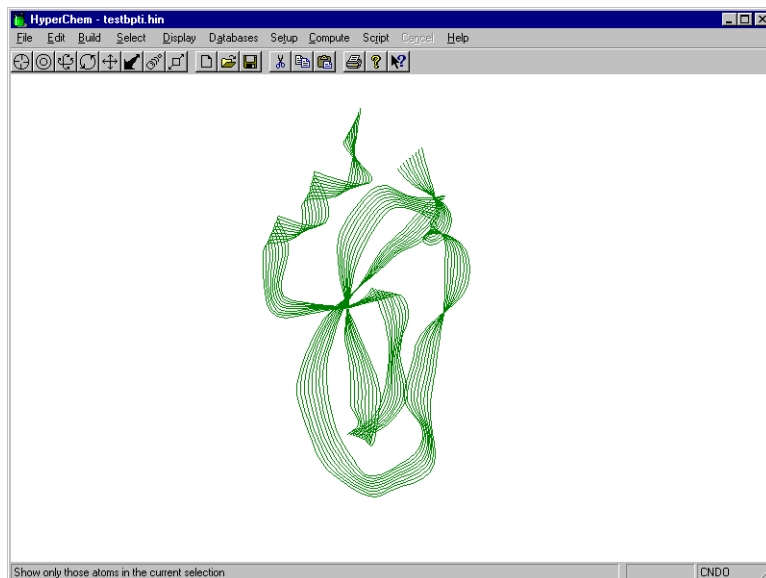
## Before You Begin

This lesson uses the file `testbpti.hin`, which you saved in Lesson 8. If you have not completed Lesson 8, follow the steps in “Reading in the PDB” and “Deleting the Water Molecules” in that lesson before you continue Lesson 13.

## Reading In and Displaying the Molecule

To read in and observe the BPTI molecule:

1. Open the Labels dialog box, and choose OK.  
This turns off the display of charges from the previous lesson.
2. Open `testbpti.hin`.  
This is the BPTI structure that you saved in Lesson 8.
3. If the structure is displayed with hydrogens, turn Show Hydrogens off on the Display menu.
4. Choose Rendering on the Display menu.
5. Display the molecule using the various options in the Rendering dialog box.  
It is useful to visualize the molecule with renderings of the structure other than the default Sticks rendering.
6. Use the menu choices on the Select and Display menus to display only the peptide backbone of the molecule.
7. Clear any selection, choose Ribbons from the Rendering/Sticks property sheet, and then choose Display/Show Selection Only.  
This shows the secondary structure as a ribbon.
8. Rotate the structure to obtain an orientation similar to this:



9. Open the Rendering/Sticks property sheet and turn Stereo on.

This displays the ribbons in stereo. If you cross your eyes to superimpose the images, or if you have stereoscopic viewers, the ribbon will appear to be three-dimensional.

## Choosing the Area of Interest

The next step is to choose which part of the system you want to describe classically and which you want to describe quantum mechanically. To accomplish this you select a subset of the system. Atoms selected at the time you perform a quantum mechanics calculation are the only atoms (with an exception described below) that HyperChem performs the quantum mechanics calculation on.

For example, you could make a spherical selection about an atom and treat every atom within a particular distance of that atom quantum mechanically. Alternatively, you could make a rectangular selection to select every atom in the left half of a molecule and treat those atoms quantum mechanically.

As you can see, BPTI has an arginine side chain sticking out of the left-top side of the molecule. In this exercise, you select this residue as the part of the molecule you treat quantum mechanically.

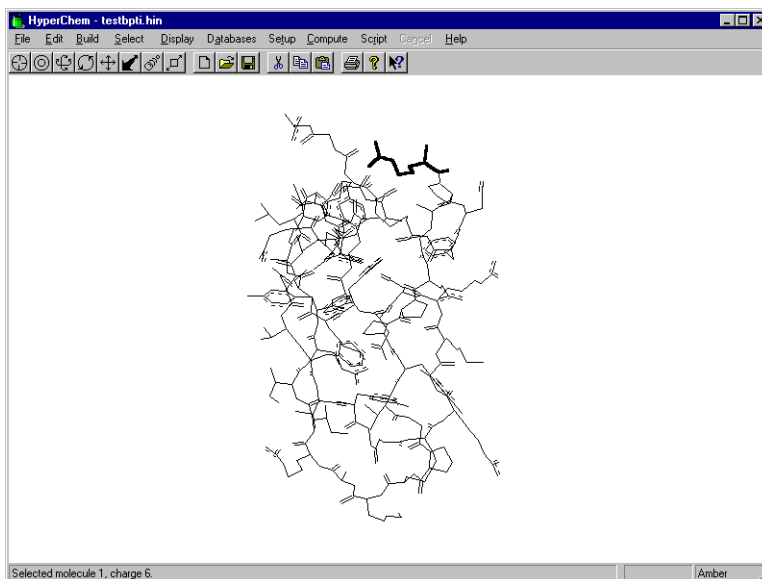
To select the arginine side chain:

1. Turn Stereo off, deselect the backbone, and display all atoms.
2. Open the Select dialog box and select residue 1.

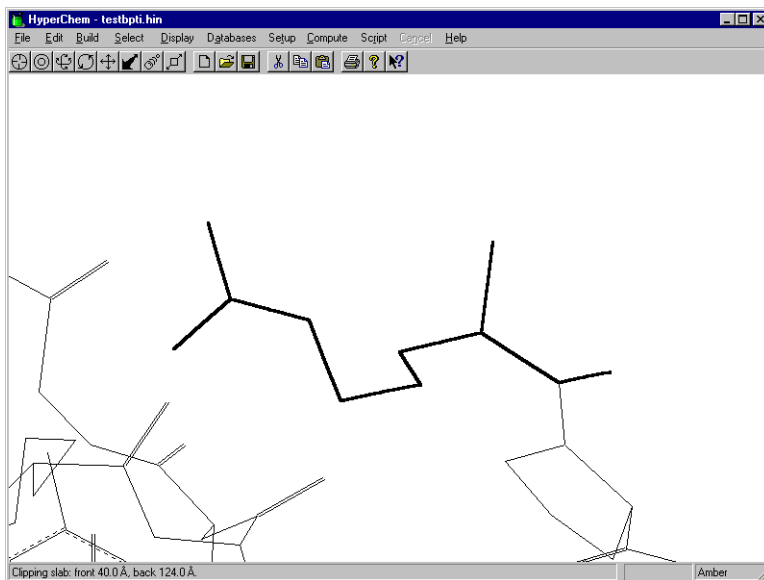
This selects ARG1.

This is the part of the molecule that you treat quantum mechanically.

3. Rotate and translate the molecule until you get an unobscured view of the selected residue:



4. Press the space bar to center the selected residue in the workspace. Position the residue so it looks like this:



## Choosing a Classical-quantum Boundary

The next step is to correctly set the boundary between the classical and quantum mechanical part of the calculation.

Briefly, this is how HyperChem solves the boundary problem: Every selected atom is rigorously treated quantum mechanically and every atom that is not selected, but is bonded to a selected atom, is treated as a boundary atom.

Boundary atoms are replaced in the quantum mechanics calculation by “pseudofluorine atoms.” These pseudo-atoms are parameterized to have as much electronegativity and as many other characteristics of the true boundary atom as possible. This closes the set of atoms that are treated quantum mechanically. The boundary occurs in the middle of the bond between a selected atom and an unselected atom, although both atoms are included in the quantum mechanics calculation; one is its original form and the other, the boundary atom, is a “pseudo-atom” analogous to a pseudopotential. The other atoms in the calculation appear as point charges to the quantum mechanical atoms.

To replace the boundary atom with a monovalent pseudo-atom, it is appropriate that the boundary be established at a single bond

between two atoms that are both sp<sup>3</sup> hybridized. We recommend that you always use such a boundary. To help, Extend to sp<sup>3</sup> on the Select menu extends the current selection in all directions until it hits the end of a molecule or finds an sp<sup>3</sup>-sp<sup>3</sup> bond. At such a bond, the interior atom is a selected quantum mechanical atom and the exterior atom is a boundary atom.

To set the classical-quantum boundary:

1. Choose Extend to sp<sup>3</sup> on the Select menu.

This extends the selection of the arginine residue to a proper boundary.

HyperChem treats carbon bonded to four atoms, nitrogen bonded to three atoms, and oxygen bonded to two atoms as being sp<sup>3</sup>. Thus, for example, a boundary might occur in the N-C<sup>α</sup> bond. If the nitrogen atom were not considered to be sp<sup>3</sup>, it would be impossible to have the boundary in a peptide chain.

## Calculating a Wave Function

Next, you calculate the wave function. First, determine if the system has a formal electronic charge, by seeing if the nitrogens are protonated.

To see if the nitrogens are protonated:

1. Display the molecule with hydrogens and label the selection by symbol.

Three hydrogens are attached to the N-terminal nitrogen atom. Also, at the end of the arginine side chain the carbon is attached only to three atoms, two of which are nitrogens that have two hydrogen atoms each; this produces a positively charged conjugated group.

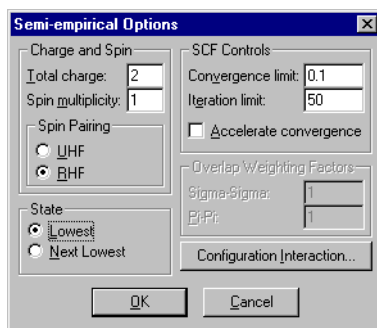
Since the N-terminal nitrogen atom has three hydrogen atoms, this nitrogen has a formal charge of +1. In addition, the side chain is positively charged (characteristic of arginine at neutral pH). Therefore the portion of the molecule you have selected for the quantum mechanical calculation is a divalent cation with a charge of +2. You must specify this to get the right number of electrons into the quantum mechanical calculation.

To calculate the wave function:

1. Remove labels.
2. Choose Semi-empirical on the Setup menu.
3. In the dialog box, choose CNDO as the method, and L-click on Options.
4. In the Options dialog box, set the Convergence Limit to **0.1**.

Since you are interested in the charges and the qualitative picture of the orbital rather than the energy, a less stringent convergence limit is used.

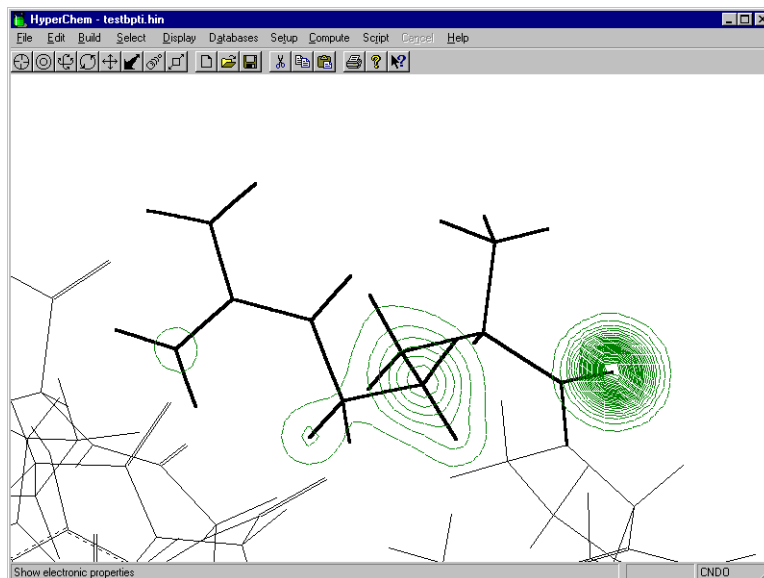
5. Set the Total charge to **2** to specify a divalent cation.
6. Set the other options as shown, then click OK:



7. Choose Single Point on the Compute menu.
8. Open the Plot Molecular Properties dialog box.
9. Choose Total Charge Density and 2D Contour in this dialog box.
10. Click on the Contour Grid tab and then set the horizontal and vertical grid points to **60** and the contour levels to **30**.
11. Choose OK in the dialog box to start plotting the density.

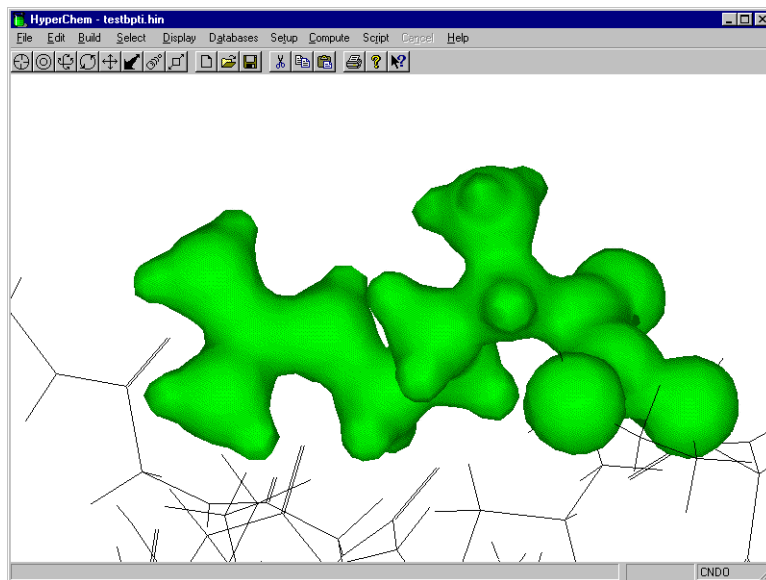
When the calculation finished, the contour plot appears:





The plane of the contour plot runs through the center of mass of the selection and is parallel to the workspace. You can obtain other views by selecting different parts of the side chain and rotating the view before replotting (with the Calculate option off). The number of contours displayed will depend on the plot options chosen.

If you plot a 3D isosurface for the calculation, using Gouraud Shading and a Total charge density contour value of 0.09, the result will look something like the following:



## Using the Results in Other Calculations

The charges of the quantum-mechanical portion are now set to the calculated values. You can use these new charges in a molecular mechanics calculation with the whole system.

## Practice Exercises

1. Plot other properties, such as the HOMO and LUMO orbitals.
2. Try using 3D Isosurface instead of 2D Contours in the Plot Molecular Properties and Orbitals dialog boxes. Use Isosurface on the Display menu (or the [F4] key) to explore the isosurfaces that result from using different threshold values.
3. Perform a quantum mechanics calculation on other parts of the system.

## **For More Information**

For more information on semi-empirical quantum mechanical calculations, see Chapter 7, “Chemical Calculations,” in the *HyperChem for Windows Reference Manual*.

# Tutorial 3

## *Ab Initio* Calculations

Tutorial 3 focuses on *ab initio* calculations you can perform with HyperChem. You learn the step-by-step instructions for performing single point calculations, geometry optimization, vibrational analysis, and electronic spectral calculations, using *ab initio* methods. While the procedures in this tutorial emphasize *ab initio* calculations, most of them can also be applied to semi-empirical methods or density functional calculations. Tutorial 3 has three lessons, as described in the following table:

Lesson	Information covered	Time to complete
Protonation of Water	Ab initio calculations with minimal basis set, and with and without MP2 correlation.	30 minutes
Vibrations of Ammonia	Geometry optimization and vibrational analysis with and without molecular mechanical restraints in terms of <i>ab initio</i> methods. Transition state searching.	50 minutes
Lowest Excited Electronic State of Ethylene	Perform singly-excited configuration interaction calculations using <i>ab initio</i> methods.	30 minutes



# Lesson 14

## Protonation of Water

### Skills Covered in This Lesson

- Creating a charged molecular system
- Selecting a basis set
- Choosing the options for *ab initio* single points and optimization
- Exploring the computation of MP2 correlation energies

This tutorial will introduce you to the basic operations needed to perform *ab initio* calculations.

### Setup Drawing

To prepare for drawing H<sub>3</sub>O with explicit hydrogens:

1. Choose Explicit Hydrogens on the Build menu.
2. Choose Allow Arbitrary Valence on the Build menu.

If you do not choose Allow Arbitrary Valence then it will be impossible to draw three bonds from the Oxygen atom since Oxygen is assumed to have the normal valence of two. Drawing and model building accommodate a formal charge on the molecular system only if you explicitly specify the formal charge on an atom using the Build/Set Formal Charge... menu item. Choosing Allow Arbitrary Valence allows you to draw as many bonds as you wish.

3. Choose Labels on the Display menu to bring up the Labels dialog box.
4. Choose Symbols for a label and dismiss the dialog box by clicking on OK.

5. Choose Default Element on the Build menu to bring up the periodic table and then double click on O (Oxygen) to choose this as the default element and dismiss the periodic table.
6. L-click on the Drawing Tool (left-most tool).

One is now prepared to draw molecules where the hydrogens are explicitly drawn rather than added automatically.  $H_3O^+$  has more hydrogens than the model builder will automatically add.

## Creating $H_3O^+$

To draw  $H_3O$ :

1. L-Click once in the working area to create an Oxygen atom.
2. Draw three bonds from the Oxygen atom to create a drawing of  $H_3O$  as shown below:

To transform the drawing into a 3D structure:

1. Double-click on the Selection tool.

Instead of selecting Model Build from the Build menu, you can invoke the model builder by double-clicking on the Selection tool as a short cut.

*Note:* With later releases of HyperChem it is possible to assign a formal charge to an atom. If a formal charge of +1 was assigned to the above oxygen atom, it would not be necessary to draw the hydrogens as the model builder would have added three hydrogens to an oxygen with a formal charge of +1.

To add the positive charge for quantum mechanical calculations:

1. Select Ab Initio on the Setup menu.
2. Click on the Options button.
3. Type in a value of 1 for the Total charge and then L-click on the OK button to dismiss the Ab Initio Options dialog box.
4. L-click on the OK button to dismiss the Ab Initio Method dialog box.

This adds a charge of +1 to the molecule for quantum mechanical calculations, essentially reducing the number of electrons

by one. The Total charge is not recognized by the model builder or most aspects of molecular mechanics but is used to determine the number of electrons to use in quantum mechanical calculations.

## Choosing a Basis Set

To choose a basis set:

1. Select Ab Initio on the Setup menu.
2. Choose Other for a Basis set.
3. Push the Assign Other Basis Set button.
4. Select 4-31G from the list and then choose OK.

This adds a 4-31G basis set to the list of basis sets that can be quickly chosen with a radio button. The basis sets that appear in the Assign Other Basis Set dialog box include all basis sets listed in the CHEM.INI file by their basis-set-specification file name (\*.BAS). Custom basis sets can be added to the list by creating a new \*.BAS text file and adding it to those named in the CHEM.INI file. Rather than use the 4-31G basis set, however, we will choose a smaller basis set, a minimal basis set.

5. Select Minimal (STO-3G) for a Basis set.
6. Choose either Apply Basis Set and then OK or just choose OK to close the Ab Initio Method dialog box.

One has now chosen a basis set for each atom of the molecule, the minimal STO-3G basis set. It is possible to specifically select atoms and use the Apply Basis Set button to place the chosen basis set only on the selected atoms with the subsequent placement of a different basis set on different selected atoms. HyperChem allows different basis sets for different atoms. To see what basis set is applied to each atom, you can display the appropriate labels.

To see the basis set applied to each atom:

1. Select Labels on the Display menu.
2. Choose Basis Set as the atom label, and then choose OK.

You will now see the basis set labels as shown below:



## Minimum Energy Structure

To calculate the optimized geometry for  $\text{H}_3\text{O}^+$ :

1. Select Ab Initio from the Setup menu.
2. Select the appropriate options for a geometry optimization by clicking on the Options button, being sure you choose Total charge = 1, Spin multiplicity = 1, Spin pairing = RHF, Convergence limit = 0.01, Iteration limit = 50, and Accelerate convergence = Yes. The Single Point only options are not used in a geometry optimization and can have any value.
3. L-Click on OK to dismiss the Ab Initio Options dialog box and then L-click on OK to dismiss the Ab Initio Method dialog box.
4. Select Geometry Optimization from the Compute menu.
5. Choose Polak-Ribiere for an optimization method and 0.1 for a terminating RMS gradient and then L-click on OK to initiate the optimization. Note the new blue HyperGauss icon while you wait for the optimization to complete.

You will find that the optimum STO-3G structure for  $\text{H}_3\text{O}^+$  is a bent one with an angle of  $113.74^\circ$  and bond lengths of 0.99 Å.

To calculate the total energy of optimized  $\text{H}_3\text{O}^+$  including correlation energy:

1. Choose Ab Initio from the setup menu, push the Options button and then choose to calculate the MP2 correlation energy, clicking on OK in the Ab Initio Options dialog box and then the Ab Initio Method dialog box to dismiss these dialog boxes.
2. Select Single Point from the Compute Menu.

You should find that the total energy of  $\text{H}_3\text{O}^+$  with an STO-3G basis set is -47270.57 kcal/mol at the SCF level and -47300.98 kcal/mol at the MP2 level when you include the -30.50 kcal/mol of computed correlation energy.

## Protonation Energy

To calculate the optimized STO-3G energy of  $\text{H}_2\text{O} + \text{H}^+$ :

1. Select the Drawing tool and R-click on one of the Hydrogens of  $\text{H}_3\text{O}^+$  to delete it and form  $\text{H}_2\text{O}$ .

The electronic energy of  $H^+$  is zero so to calculate the protonation energy of  $H_2O$  we need only to subtract the previously calculated energies of  $H_3O^+$  from the corresponding energies of  $H_2O$ .

2. Select Ab Initio from the Setup menu and push the Options button to change the Total charge from 1 to 0. Then choose OK twice to dismiss the dialog boxes.
3. Select Geometry Optimization from the compute menu and optimize  $H_2O$  as you did for  $H_3O^+$ .

You should find an optimum STO-3G bond angle of  $100.0^\circ$  and an optimum bond length of  $0.99 \text{ \AA}$ .

4. Select Single Point from the compute menu.

You should find that the total energy of  $H_2O$  with an STO-3G basis set is  $-47041.82 \text{ kcal/mol}$  at the SCF level and  $-47066.21 \text{ kcal/mol}$  at the MP2 level when you include the  $-24.45 \text{ kcal/mol}$  of computed correlation energy. These results yield a protonation energy of  $228.75 \text{ kcal/mol}$  at the SCF level or  $234.77 \text{ kcal/mol}$  at the MP2 level. The effect of correlation in this example is not large.

## Practical Exercises

1. Try these calculations with 3-21G and then 6-31G\* basis sets. You will find that the double zeta basis set gives an optimum geometry for  $H_3O^+$  that is planar. For larger basis sets, however, the molecule (at the SCF level) is predicted to be bent. The anomaly at the 3-21G level is possibly a sign of an unbalanced basis set. In fact, at the limit of s and p basis functions only, both  $H_3O^+$  and its isoelectronic  $NH_3$  will both be predicted to be planar. D-orbitals on the heavy atom are necessary to get a reasonably accurate prediction of the geometry for these molecules.
2. Explore the protonation energies of the HF,  $H_2O$ ,  $NH_3$ , and  $CH_4$  series and compare with experimental results.
3. Explore the reaction between  $H^+$  and  $H_2O$  using *ab initio* wave functions and a molecular dynamics trajectory procedure equivalent to the example included with HyperChem as the script REACT.SCR.

## **For More Information**

For more information on the various types of *ab initio* quantum mechanical calculations that HyperChem can do, see Chapter 7, “Chemical Calculations,” in the *HyperChem for Windows Reference Manual*.

# Lesson 15

## Vibrations and Transition States of Ammonia

### Skills Covered in This Lesson

- Performing vibrational analysis
- Interactively adding an extra shell of basis functions
- Using restraints to specify symmetry
- Finding transition states using both the Eigenvector Following and Synchronous Transit methods

This tutorial explores the vibrations of ammonia and the differences between the pyramidal and planar form of  $\text{NH}_3$ .

### Creating Ammonia

To draw ammonia:

1. L-click on Explicit Hydrogens of the Build menu until it is not selected.
2. Select Default Element of the Build menu to bring up the periodic table dialog box.
3. Double-click on N (Nitrogen) to select this element and simultaneously dismiss the periodic table dialog box.
4. L-click on the Drawing tool to select it.
5. L-click once in the workspace to create a single Nitrogen atom.
6. Select Add H & Model Build of the Build menu to create  $\text{NH}_3$  at the standard model builder geometry.

## Choosing a Basis Set

In this lesson you will use a 3-21G double zeta basis set augmented with d orbitals on the nitrogen atom.

To choose a basis set:

1. Select Labels from the Display menu.
2. Choose an atom label of Basis Set and then choose OK to close the Labels dialog box.

The labels on each atom should now be the basis set for that atom. At this point the labels should read, "None".

3. Select Ab Initio from the Setup menu.
4. Select Small (3-21G) as the basis set and then L-click on Apply Basis Set. Then L-click on either OK or Close.

You should now see labels on each atom that read, "3-21G".

5. Select the nitrogen atom of ammonia by L-clicking on the Selection tool and then L-clicking on the nitrogen atom. Only the nitrogen atom should be selected, as indicated by a small circle around it. To select individual atoms like this, Atoms of the Select menu must be checked and be the smallest unit of selection rather than Residues or Molecules.
6. Select Ab Initio from the Setup menu again.
7. Push the Advanced Options button and be sure Six is chosen as the number of d orbitals before clicking on OK and returning to the Ab Initio Method dialog box.
8. Push the Extra Basis Function button and choose a D shell type with an Exponent of 1.0 in the Extra Basis Function dialog box. Click on OK twice to dismiss the Extra Basis Function dialog box and the Ab Initio Method dialog box.

You have now added an extra shell of 6 primitive d-type orbitals, each with an orbital exponent of 1.0, to the nitrogen atom. The basis function label for the nitrogen should indicate this.

9. R-click in empty space to de-select the nitrogen atom.

You are now ready to perform *ab initio* calculations on ammonia with this basis set.

## Vibrational Analysis on Pyramidal Ammonia

To perform a vibrational analysis:

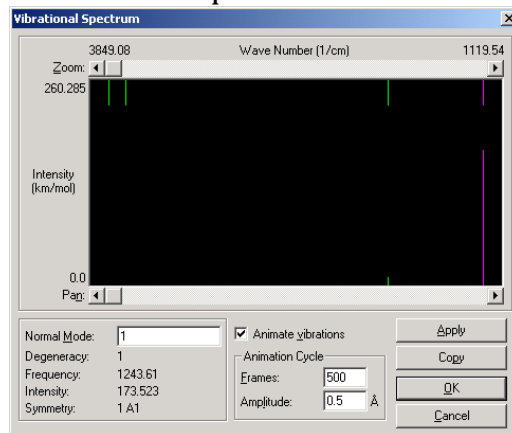
1. Select Geometry Optimization from the Compute menu and perform a Polak-Ribiere optimization to a RMS gradient of 0.1. An appropriate SCF convergence to use is 0.01. This may take a few minutes depending on the speed of your computer.

The resultant structure (H-N-H angle =  $106.5^\circ$  and N-H bond length =  $1.01 \text{ \AA}$ ) turns out to be very close to experimental results. This is a “balanced” basis set.

2. Select Single Point from the Compute menu to calculate the total energy including the MP2 correlation energy.

The total energy at the SCF level is  $-35086.49 \text{ kcal/mol}$  and the total energy including the MP2 correlation energy is  $-35191.11 \text{ kcal/mol}$ .

3. Select Vibrations from the Compute menu to perform a vibrational analysis at the optimized geometry. This calculation requires the numerical calculation of all second derivatives with respect to Cartesian displacements of the atoms and requires a few minutes.
4. Size the HyperChem window to approximately one third of the screen and place it on the right-hand-side of the screen.
5. Select Vibrational Spectrum from the Compute menu to display the vibrational spectrum as shown below:



6. Be sure that Animate Vibrations is selected as shown. Click on the lowest energy vibration at the right-hand-side of the spectrum and then push the Apply button. Move the spectrum window until you can see the animated vibration in the HyperChem workspace. If the animation is too fast, select a slower rendering (CPK spheres?) or increase the number of frames in the animation cycle. [Click on the Escape key to terminate the animation if the animation is too fast and makes it difficult to control the mouse].

This is the lowest frequency breathing motion of ammonia. It maintains  $C_{3V}$  symmetry and essentially varies the bond angle. You may wish to observe some of the other vibrations. Just click on any vibrational peak and push Apply.

The lines emanating from the top of the window show all vibrational transitions and the lines emanating from the bottom of the window show vibrational transitions with their infra-red intensities. When you click on a vibrational line, its frequency and intensity are shown in the lower left-hand-side of the window.

7. Select the lowest energy transition (right-hand-side) again. Push the Apply button Click on OK to close the Vibrational Spectrum window. Rotate the ammonia molecule with the X-Y-Rotate tool to fully inspect the lowest frequency vibration.
8. Select Cancel from the menu bar or hit the Escape key to stop the animation. Select Labels from the Display menu and click on OK to eliminate any labels that are showing. Select Renderings from the Display menu. Click on IR Vectors and then OK to display the eigenvectors of a vibrational mode. Rotate the molecule with the X-Y-Rotate tool to fully observe the IR vectors of the breathing mode as shown below:
9. Select Vibrational Spectrum from the Compute menu to display the vibrational spectrum again. Click on the breathing transition and push the Apply button to observe the breathing motion with IR Vectors. You can also observe a vibration using a spheres representation if you wish. You must first dismiss the Vibrational Spectrum window and cancel the animation before selecting the new Spheres rendering however. You may then select Vibrational Spectrum again.

*Important:* It is always important to perform a geometry optimization before a vibrational analysis calculation since the basic vibrational procedure assumes all first derivatives of the energy are zero, *i.e.*, the molecule is at an energy extremum.

## Optimizing Planar Ammonia

You will now create an optimum planar structure for ammonia. To create this particular structure you will use restraints. Restraints are extra molecular mechanics terms added to the potential to restrain bond lengths, bond angles, and bond torsions. You will use an “improper torsion” to restrain ammonia to a planar form.

To add a restraint that keeps ammonia planar:

1. Eliminate any labels or IR vectors that are still showing.
2. Be sure that Multiple Selections of the Select menu has been selected and that you are selecting Atoms rather than Residues or Molecules.
3. Select in turn, the nitrogen atom, one of the hydrogen atoms, and then the other hydrogen atoms until all four atoms are selected (*i.e.* everything is selected).
4. Select Name Selection from the Select menu, choose Other, and then type in an arbitrary name for the selection such as “**imptor**”. Close the Name Selection dialog box by choosing OK.

The named selection “**imptor**” now refers to an improper torsion angle in ammonia. You can now restrain this improper torsion to a specific angle ( $180^\circ$  = planar) by adding a restraining force for deviations from this angle. This will keep ammonia planar, provided the force is large enough.

5. R-click in empty space to de-select all atoms.
6. Select Restraints from the Setup menu. Select 4-imptor (a 4-atom selection named imptor) and push Add to add this named selection to the list of restraints. Choose Other for Restrained Value and type in a value of 180 for the torsion angle. Choose Other for the Force Constant and type in a reasonably large value, 500, to ensure that ammonia is kept planar. Choose OK to dismiss the Restraints dialog box.



7. Select Geometry Optimization from the Compute menu for an optimization to the best planar structure of ammonia. Choose Polak-Ribiere as the method and 0.1 for an RMS gradient, as before. Click on OK to initiate the optimization.
8. The optimized structure you obtain should be planar with a bond length of 0.99 Å.
9. Select Single Point from the Compute menu to calculate the energy of the planar structure including the MP2 correlation energy.

The total energy at the SCF level is -35079.16 kcal/mol and the total energy including the MP2 correlation energy is -35183.80 kcal/mol. The difference between this energy and the energy of optimized pyramidal ammonia is the inversion barrier of ammonia. The SCF value for this is 7.33 kcal/mol. The effect of correlation energy on this barrier is predicted to be negligible at .05 kcal/mol.

## Vibrational Analysis on Planar Ammonia

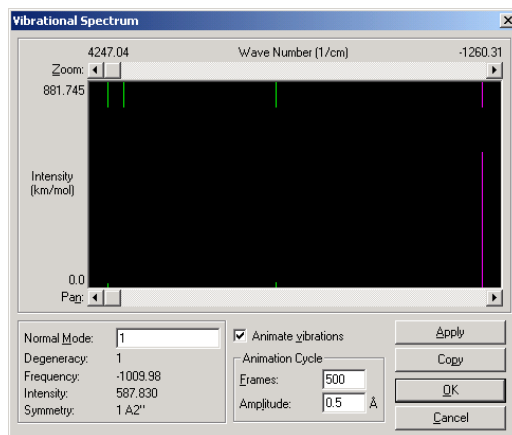
To perform a vibrational analysis:

1. Select Restraints from the Setup menu. Select the 4-imptor restraint and push Remove to move this improper torsion from the list. Click on OK to dismiss the dialog box and remove the restraint.

If you do not remove the restraint, a vibrational analysis will include second derivative components from the molecular mechanics restraining force as well as those from the straight-forward quantum mechanical *ab initio* calculation on the planar structure of ammonia.

2. Select Vibrations from the Compute menu to perform a vibrational analysis at the optimized planar geometry. This calculation requires the numerical calculation of all second derivatives with respect to Cartesian displacements of the atoms and requires a few minutes.
3. Size the HyperChem window to approximately one third of the screen and place it on the right-hand side of the screen.

4. Select Vibrational Spectrum from the Compute menu to display the vibrational spectrum as shown below:



Note that the lowest energy vibration has a negative frequency. Actually, the solution of the equations for the normal modes gives eigenvalues that are the square of the vibrational frequencies. When one of these squares of the vibrational frequency is negative, HyperChem plots it as the negative of the square root of the magnitude rather than as an imaginary number. Thus this negative value,  $-v$ , actually means there is an imaginary vibrational frequency. A single imaginary frequency with all the remaining frequencies being positive implies that this planar structure is a transition state, not a minimum energy structure, as the pyramidal structure is.

5. Click on the negative frequency transition and animate it.

You will see that this “normal mode” corresponds to the breathing mode of ammonia. This is the “reaction path” associated with the transition state. HyperChem does not directly deal with symmetry constraints and internal coordinate restraints, but rather works only with Cartesian coordinates. Nevertheless, it is still possible, via restraints, to effect many of the same results as obtained through internal coordinate constraints.

6. Click on Close to dismiss the Vibrational Spectrum window and then on Cancel to terminate the animation. Select New from the File menu to clear the workspace.

## Finding a Transition State by Eigenvector Following

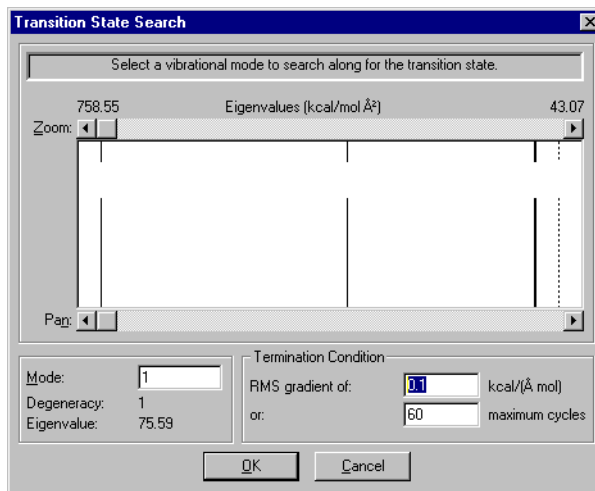
In this calculation you will use the Eigenvector Follow method of transition state searching to find the planar transition state for the inversion of the ammonia pyramid, without needing to apply extra structural restraints. This method is particularly useful if one of the natural vibrational modes of your structure tends to lead to a transition state.

1. Read the optimized pyramidal ammonia structure (which you saved in “Vibrational Analysis on Pyramidal Ammonia” on page 229) back into the workspace.
2. Double-click on the Selection Tool icon to model build the structure.

The Eigenvector Following optimizer does not work well when the starting point for the search is at a local or global minimum; even if a vibrational mode to follow is specified, the optimal atomic motions to lead from the minimum are not well defined.

3. Select Transition State... on the Compute menu.
4. In the dialog box, specify Eigenvector following, then click on OK.

The status bar shows that HyperChem is performing a vibrational calculation. When this has been completed, the Transition State Search dialog box opens.



5. Click on the lowest energy eigenvector at the right-hand side of the spectrum. As before, this corresponds to the lowest-frequency breathing mode of ammonia, which moves the atoms in directions that lead to the inversion of the pyramid.

The lowest-energy vibration is often the mode which is most appropriate for a transition state search.

6. Specify an RMS gradient limit of 0.1 kcal/(Å mol), and click on OK.

HyperChem finds a planar transition state.

7. The transition state should be planar with a bond length of 0.99 Å. Select the N atom, and then the three H atoms in any order, to see the improper torsion angle reported on the status line. It should be very close to 0° or ±180°.

## Finding a Transition State by Synchronous Transit

In this calculation you will use the Synchronous Transit method of transition state searching to find the planar transition state for the inversion of the ammonia pyramid, without needing to apply extra structural restraints. This method allows you to calculate transition states when the reactant and product systems are very different, or if you want to carefully define how the atoms are rearranged through the transition state. Reactant and product

atoms are matched, and a starting point for the search is interpolated from the initial and final configurations.

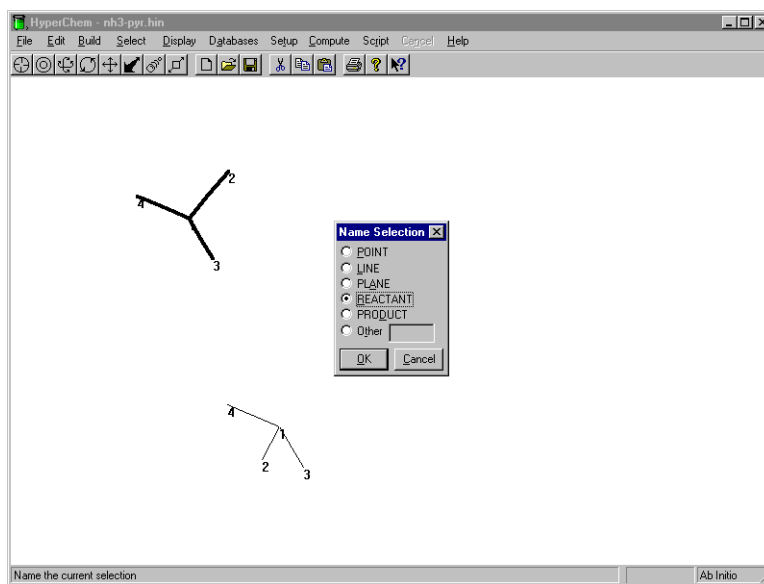
1. Read the saved ammonia structure file back into HyperChem.
2. Use Display/Labels to change the labels to Number.
3. Change to the Drawing tool by clicking on its icon. Hold the [Shift] key down and click on the N atom.

One of the H atoms moves to the “empty” position of the ammonia pyramid, reversing the “chirality” of the system with respect to the numbered atoms. Usually you would use this operation to exchange different attachments to a chiral center.

4. Optimize the geometry of the system to a gradient of 0.1 kcal/(Å mol), as before.

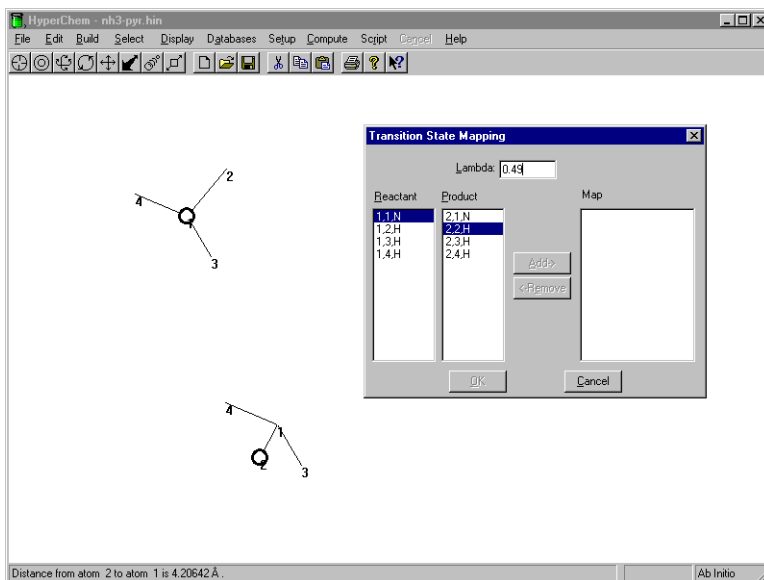
Both the reactant and product atom sets should be optimized before you find a transition state with Synchronous Transit.

5. Use File/Merge to add another copy of the saved optimized ammonia structure to the workspace.
6. Change to the Selection tool, and select one of the molecules.
7. Use Select/Name Selection to name the molecule as REACTANT.



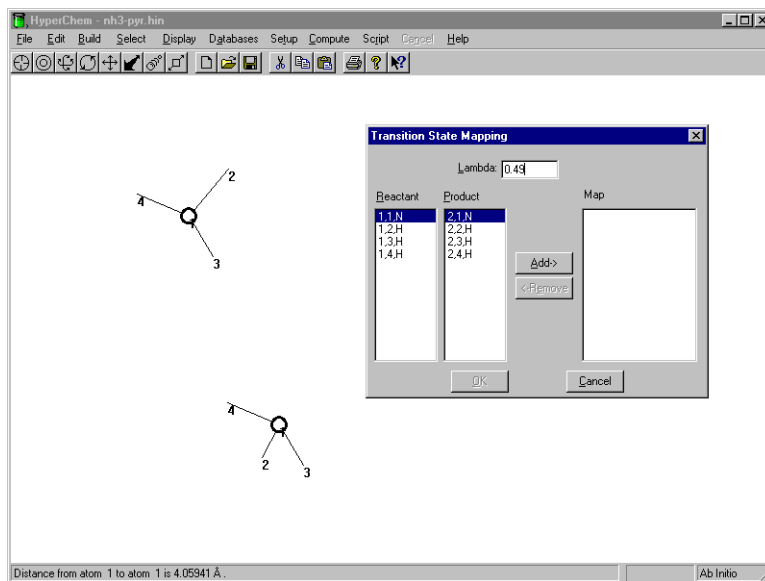
8. Use Select/Complement Selection to unselect the first molecule and select the other molecule.
9. Use Select/Name Selection to name that molecule as PRODUCT.
10. Right-click in the workspace to unselect all.
11. Click on Reaction Map in the Setup menu to open the Transition State Mapping dialog box.
12. In the dialog box, choose the N atom from the Reactant list and an H atom from the Product list. Each atom in the lists is designated by its molecule number, its atom number within the molecule, and its element.

When you select an atom from the list, it is highlighted in the workspace as a visual aid.



Because the atoms are not matched (an N atom in the reactants cannot become an H atom in the products), the Add button remains inactive.

13. Choose the N atom in both the Reactant and Product lists.  
Because the atoms are matched, the Add button is activated.



14. Click on the Add button.

The atoms are moved from the Reactant and Product lists to a matched pair in the Map list.

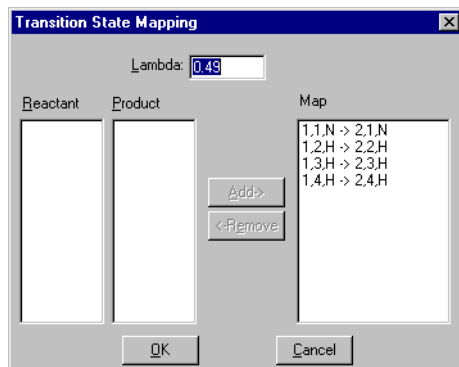
15. Select hydrogen atom #2 in the Reactant and Product lists and transfer the pair to the Map list. Do the same thing for the other two pairs of H atoms.

Because we are trying to invert the ammonia molecule, the ordering of the atoms is important.

When all of the atoms have been transferred to the Map list, the OK button becomes active.

16. Specify a value of 0.49 in the Lambda text box.

$\lambda$  (Lambda) is the interpolation parameter; its value ranges from 0 for the reactant state to 1 for the product state. Because the Reactant and Product atom sets are mirror images, a  $\lambda$  value of 0.5 would give a perfectly symmetrical starting point for the transition state search. The optimizer does not work efficiently from a perfectly symmetrical start. A  $\lambda$  value of 0.49 gives a slightly asymmetric starting point for the search.



17. Click on OK.

The atoms of the Product set are deleted, and the atoms of the Reactant set are repositioned to the coordinates of the interpolated starting point for the transition state search.

18. Press the [Spacebar] to center the structure, and rotate it so that you can see its conformation clearly.

The starting point should be almost planar.

19. Select Transition State on the Compute menu.

The Transition State Search Options dialog box opens.

20. Specify Synchronous Transit, Quadratic, and an RMS Gradient termination criterion of 0.1 kcal/(Å mol). Click on OK.

HyperChem performs a vibrational calculation, chooses its own eigenvector, and then does the transition state search. Since it is working from a good starting point for the search, it finds the transition state relatively quickly.

You could also use the interpolated starting point from the Reaction Map dialog box as the starting point for an Eigenvector Follow transition state search.

## Practical Exercises

1. Compare the geometry, barrier, and vibrational spectra of ammonia with experiment.
2. Repeat the calculations with other basis sets.
3. Compare  $\text{NH}_3$  results with isoelectronic  $\text{CH}_3^-$  and  $\text{OH}_3^+$ .



4. Compute the vibrational spectra of formaldehyde with a variety of basis sets and compare with experiment.
5. Vinyl alcohol ( $\text{CH}_2=\text{CHOH}$ ) and acetaldehyde ( $\text{CH}_3\text{COH}$ ) are tautomers. The rearrangement reaction involves an intermediate like  $[\text{CH}_2 \cdots \text{CH} \cdots \text{O}]^-$ ; depending on the environment, there may be a proton ( $\text{H}^+$ ) loosely attached to the intermediate or the proton may be solvated and distant. Investigate the rearrangement by considering the geometries and energies of the *anti* conformation of vinyl alcohol, and then the *gauche* conformation of vinyl alcohol, acetaldehyde, and the transition state between the latter two structures found by Synchronous Transit. Compare with the energy and geometry of the deprotonated intermediate. (Remember that the energy “cost” of deprotonation may be balanced by an energy gain due to the protonation of a solvent; see “Protonation Energy” on page 224.)

## For More Information

For more information on the various types of *ab initio* quantum mechanical calculations that HyperChem can do, see Chapter 7, “Chemical Calculations,” in the *HyperChem for Windows Reference Manual*.

# Lesson 16

## Lowest Excited Electronic State of Ethylene

### Skills Covered in This Lesson

- Investigating the orbital energy diagram for a molecule
- Performing singly-excited configuration interaction calculations
- Investigating the electronic spectra of molecules
- Calculating the geometry of an excited triplet state

This tutorial explores the computation of electronic excitation spectra, focusing on the triplet  $\pi \rightarrow \pi^*$  transition in ethylene.

### Optimizing the Ground State of Ethylene

You should now have little difficulty in performing simple *ab initio* calculations so this tutorial will abbreviate some of the instructions.

To create Ethylene with an STO-3G basis set:

1. Select New on the File menu to clear the workspace.
2. Make sure Explicit Hydrogens is not checked.
3. Select carbon for the Default Element and draw a carbon-carbon single bond (one line), clicking on the middle of the bond to turn it into a double bond.
4. Select Add H & Model Build from the Build menu to create ethylene.
5. Select Ab Initio from the Setup menu and choose Minimal (STO-3G) for a basis set. Be sure you have Total charge = 0, Spin

multiplicity = 1, Spin pairing = RHF, Accelerate convergence = Yes, and a reasonably tight SCF Convergence limit of, say 0.0001. Push the CI button in the Ab Initio Options dialog box and be sure None is chosen for CI Method.

To optimize the ground state of ethylene:

1. Select Geometry Optimization from the Compute menu. Choose Polak-Ribiere for a method and 0.01 for the RMS gradient. Choose OK to dismiss the dialog box and initiate the optimization.

You should find an optimized geometry for ethylene that has a C-C bond length of 1.31 Å, a C-H bond length of 1.08 Å, and an H-C-H angle of 115.7°.

To calculate the correlation energy:

1. Select Ab Initio from the Setup menu, push the Options button, and select MP2 correlation energy. Click on OK to dismiss the dialog boxes to return to the workspace.
2. Select Single Point from the Compute menu.

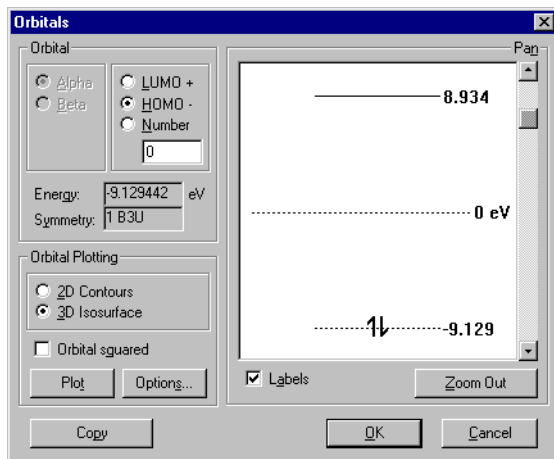
You should find an SCF energy of -48364.64 kcal/mol and a total MP2 energy of -48438.61 kcal/mol that includes -74.97 kcal/mol of correlation energy.

Small differences between the energy at optimization and the reported single point energy may sometimes be seen. These arise because the coordinates of atoms are kept as single precision values at the front end but as double precision values in the computational back ends where precision is required. Conveying coordinates between the front end and the back ends results in the retaining of only single precision values.

## Orbitals of Ground State Ethylene

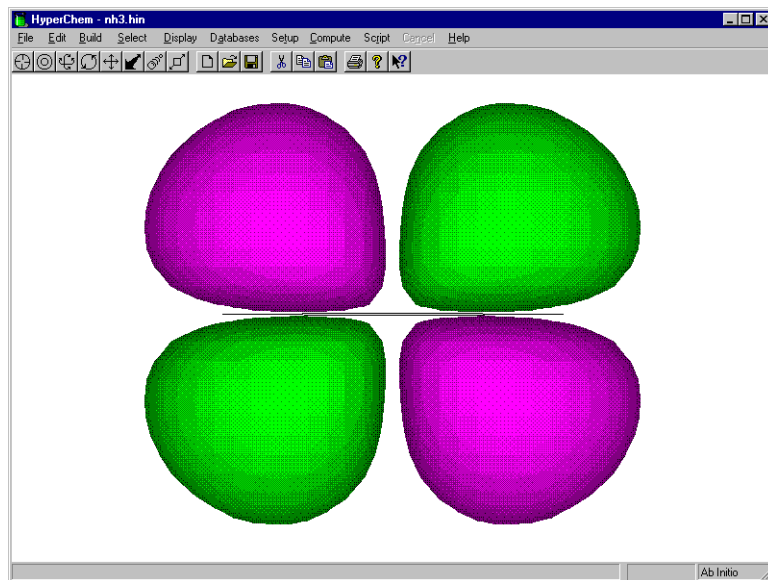
To inspect the orbital energy diagram and the orbitals:

1. Select Orbitals from the Compute menu. Drag with the mouse to zoom in on the highest occupied (HOMO) and lowest unoccupied (LUMO) orbitals as shown in the following view:



These two orbitals are referred to as the  $\pi$  (HOMO) and  $\pi^*$  (LUMO) orbitals of ethylene. The lowest excited states of ethylene correspond to removing one electron from the p orbital and placing it in the  $\pi^*$  orbital. This  $\pi \rightarrow \pi^*$  (HOMO to LUMO) transition leads to both an excited singlet state, where the electrons remain of opposite spin, and a triplet state where the electrons in the  $\pi$  and  $\pi^*$  orbitals have parallel spins.

2. Select either of these orbitals, by L-clicking on it, and plot the orbital. Orient the ethylene molecule until you get a view you like. For example, the following shows the  $\pi^*$  orbital, after one has aligned the principal axis with the x axis with Edit/Align Molecules.



## CI and the Electronic Spectrum of Ethylene

To calculate the spectrum of ethylene:

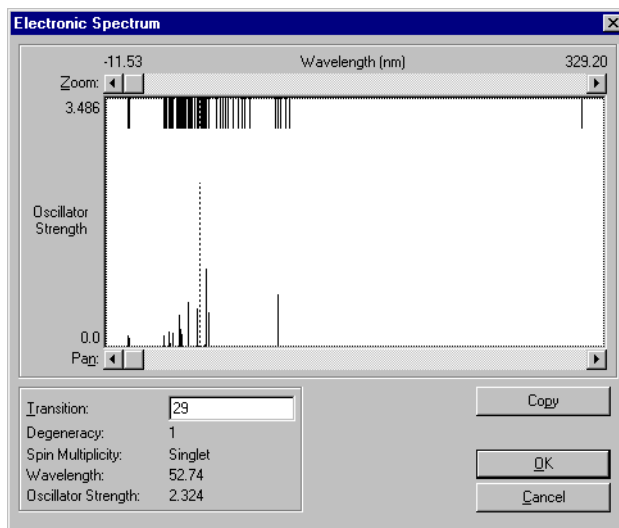
1. Select Ab Initio from the Setup menu.
2. Click on the Options button.
3. Click on the CI button.
4. Choose Singly Excited for the CI Method.
5. Choose Energy Criterion rather than Orbital Criterion. Type in a large value, 1000, for the Maximum Excitation Energy. Click on OK to dismiss dialog boxes until you have returned to the workspace.

When performing a singly-excited configuration interaction (CI) calculation, one mixes together a potentially large number of configurations (single determinants; in reality HyperChem does not form spin-adapted configurations but performs a CI with single determinants). These configurations involve the excitation of a single electron from one of the occupied orbitals to one of the unoccupied orbitals. One needs to make a decision about how many determinants should be included in the mixing. This can be done on a basis of includ-

ing all excitations involving a certain number of the lowest unoccupied and highest occupied orbitals or on the basis of an energy criterion associated with the difference in energy between the two orbitals involved.

The choice here of 1000 will result in inclusion of all singly-excited configurations in the CI.

6. Select Single Point from the Compute menu. This will initiate the CI calculation in conjunction with performing a Single Point calculation. The Compute menu item, Electronic Spectrum, will become un-grayed when the single point calculation is complete.
7. Select Electronic Spectrum from the Compute menu to display the spectrum shown below:



The strongest peak lies at 52.7 nm and is not associated with a  $\pi \rightarrow \pi^*$  transition. The lowest energy transition (longest wavelength) occurs on the far right at 313.7 nm and is the triplet  $\pi \rightarrow \pi^*$  transition. This is, of course, a forbidden transition (zero intensity) as are all singlet-to-triplet transitions (to a first approximation). The lowest energy allowed transition at 106.4 nm is the singlet  $\pi \rightarrow \pi^*$  transition.

## Geometry and Energy of Triplet Ethylene

Because the lowest excited state of ethylene has a different spin state (triplet) than the ground state (singlet), it is possible with HyperChem to easily and directly compute an optimized SCF wave function for it. This is not a simple calculation when an excited state has the same spin symmetry as the ground state and HyperChem does not currently have the capability of directly computing an excited singlet state for ethylene, apart from the information available from a CI on the ground state, such as you just performed above. The triplet state, however, is computable in a straightforward single-determinant SCF calculation.

To compute a geometry for triplet ethylene:

1. Create standard model builder ethylene in the workspace if you do not already have it there.

Since we are going to optimize a geometry, the actual starting geometry is not critical and your ethylene might be the optimized structure for the STO-3G ground state, previously obtained.

2. Select a torsion angle in planar ethylene and then select Set Bond Torsion from the Edit Menu to change the torsion angle to 30°.

HyperChem's first-derivative optimizers may not easily find a non-planar optimum (if there is one) if you start with a rigorously planar configuration for a molecule. You are helping the optimization out by starting with a general non-planar structure.

3. Select Ab Initio from the Setup menu and choose 3 (triplet) for the Spin Multiplicity and UHF for the Spin pairing. This will result in the computation of an unpaired-electron triplet state rather than the normal RHF paired-electron singlet ground state. Select 0.0001 for the SCF convergence. Be sure you are using a Minimal (STO-3G) Basis set and then click on OK to return to the workspace.
4. Select Geometry optimization from the Compute menu. Perform the optimization with an RMS gradient of 0.01.

This optimization takes longer than you want it to. The result, however, is that the excited triplet state is twisted 90° from the

ground state. The C-C double bond has greatly expanded to 1.49 Å from its ground state value of 1.31 Å showing, essentially, that the double bond has been broken and what remains is close to a single bond. The C-H distance remains at 1.08 Å and the H-C-H angle is now 117.8°.

To compute the adiabatic excitation energies:

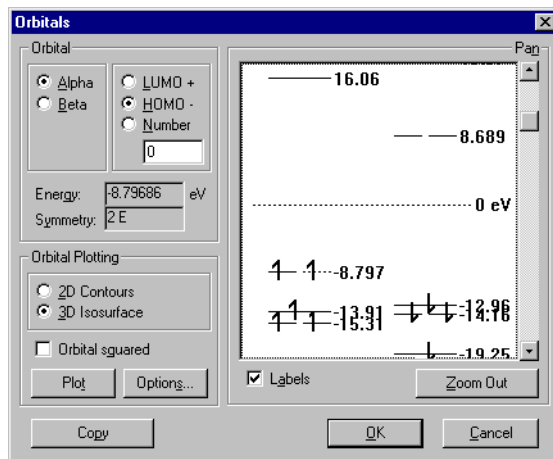
1. Select Ab Initio from the Setup menu, push the Options button, and be sure MP2 correlation energy is selected. Choose OK to dismiss the dialog boxes and exit to the workspace.
2. Select Single Point from the Compute Menu.

The SCF energy of the optimized excited triplet state is -48328.65 kcal/mol which is 35.99 kcal/mol (794.4 nm) higher than the optimized singlet ground state. This is the adiabatic excitation energy measured from the energy minimum of each state. The MP2 correlation energy in the excited triplet state (-45.02 kcal/mol) is much less than that in the ground state (-74.43 kcal/mol) so that, with the effect of correlation, the adiabatic excitation energy is 65.84 kcal/mol (434.3 nm). These numbers can be compared with the vertical excitation energies (91.14 kcal/mol or 313.7 nm) calculated by CI for this transition. A vertical excitation energy (same geometry for the excited state as for the ground state) should obviously be larger than the corresponding adiabatic excitation energy, as it is.

To explore the triplet state's orbitals:

1. Select Orbitals from the Compute menu. Zoom in on the alpha orbitals to see the following diagram:





Notice that the two occupied  $\pi$  and  $\pi^*$  orbitals have become degenerate. Click on these orbitals to select them and then plot them to explore their appearance.

## Practical Exercises

1. Explore the effects on the  $\pi \rightarrow \pi^*$  excitation energy of CI mixing with other singly-excited states. What, for example, is the excitation energy for a single ( $\pi \rightarrow \pi^*$  singlet or triplet) configuration?
2. Calculate the barrier to rotation about the C-C bond in both the singlet ground state and the excited triplet state of ethylene.
3. Repeat the ethylene calculations for the  $n \rightarrow p^*$  transitions of formaldehyde.
4. Explore the effects of geometry on the calculated singlet  $\pi \rightarrow \pi^*$  excitation energy using CI. Indirectly argue about the optimum geometry of the singlet excited state.

## For More Information

For more information on the various types of *ab initio* quantum mechanical calculations that HyperChem can do, see Chapter 7,

**“Chemical Calculations,”** in the *HyperChem for Windows Reference Manual*.



# Tutorial 4

## HyperChem & DDE

Tutorial 4 shows you how to create links with other Windows applications using the Dynamic Data Exchange (DDE) feature of Microsoft Windows. Tutorial 4 has two lessons, as described in the following table:

Lesson	Information covered	Time to Complete
Driving HyperChem from Microsoft Excel	Creating macros in Excel to run HyperChem calculations	20 minutes
Driving HyperChem from Microsoft Visual Basic	Creating a Visual Basic program for use with HyperChem	15 minutes



# Lesson 17

## Driving HyperChem from Excel

### Skills Covered in This Lesson

- Modifying macros in Excel to run HyperChem calculations
- Using worksheets in Excel to store HyperChem information
- Creating plots of HyperChem results

With HyperChem's open system design, you can create links with other Windows applications, such as Microsoft Excel, Microsoft Word<sup>®</sup> for Windows, Microsoft Visual Basic, and Q&E, by using the Dynamic Data Exchange (DDE) feature of Windows.

You can create macros and batch calculations using the built-in scripting capability, access all HyperChem modeling and analysis tools, and transfer data between Windows applications using the Clipboard.

This lesson explores the benefits of interfacing HyperChem with the spreadsheet program, Microsoft Excel. You could also use other Windows applications that have DDE macro capabilities.

This lesson has three exercises:

- In exercise 1, you run an Excel macro that invokes single point calculations on a group of molecules. Then you use Excel to plot the total energy of each molecule.
- In exercise 2, you modify and rerun the macro to rotate a benzene molecule. This demonstrates, briefly, how you can use Excel to manipulate and display molecules in HyperChem.
- In exercise 3, you modify and rerun the macro to compute the total energy as the bond length of a structure changes. You store the results in Excel and then plot them for comparison.

## Lesson Requirements

Before you begin, you must do the following:

1. Install Microsoft Excel on your computer.
2. Have the following files in your HyperChem working directory:

plot.xls (an example spreadsheet)

plot.xlm (an example Excel macro)

Anthrace.hin

Benzene.hin

Chrysene.hin

Coronene.hin

Naphthal.hin

Phenanth.hin

Pyrene.hin

These files are included with HyperChem.

## Exercise 1: Calculations on Sets of Molecules

In this lesson, you use an Excel macro to do single point calculations on seven aromatic hydrocarbons. The results are read back into Excel and then plotted for analysis. This lesson demonstrates how Excel can do batch calculations on a group of molecules. Prior to learning to drive HyperChem from Excel, it is helpful if you have a basic familiarity with Excel spreadsheet operations

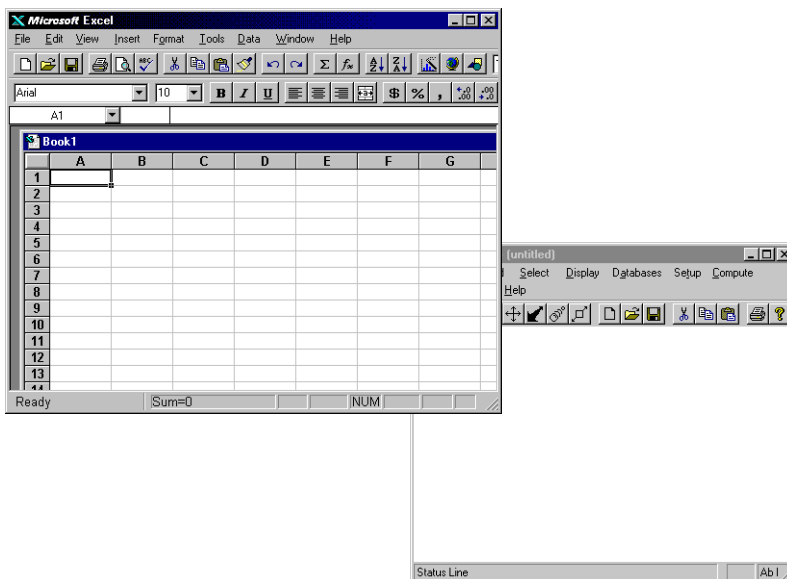
For an explanation of the macro used in this lesson, see chapter 10, "Scripts and DDE," in the *HyperChem 5 for Windows Reference Manual*.

## Opening HyperChem and Excel

To open HyperChem and Excel in Microsoft Windows:

1. Open HyperChem and make the window smaller. Place it in the lower-right corner of the screen.
2. Open Excel and place the Excel window in the upper-left corner of the screen.

Your screen should look like this:

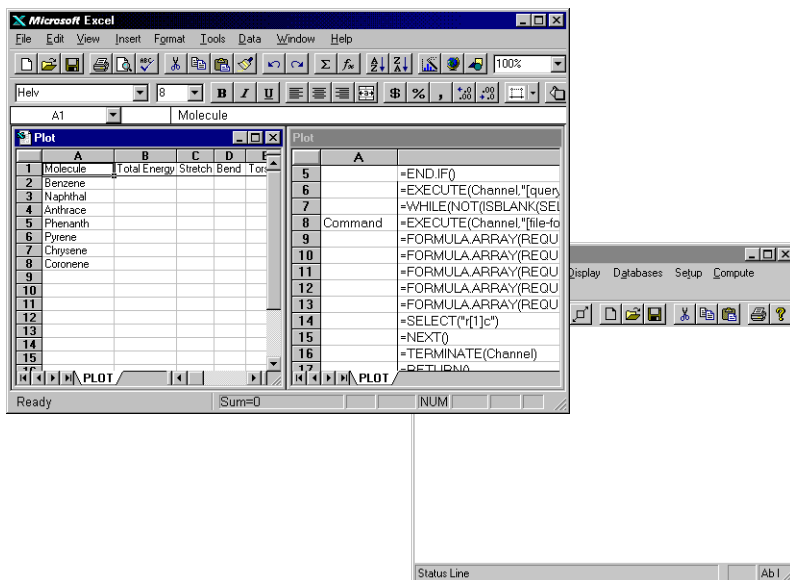


3. In Excel, choose Open on the File menu.
4. In the Directories box, double-click on [..] and change directories to \Scripts or the name of your HyperChem working directory.
5. In the Files box, open the sample spreadsheet file plot.xls and the sample macro file plot.xlm  
When you open the files, they appear as separate windows within the Excel window.
6. Choose Arrange All on the Window menu.
7. L-click on Sheet1, and then close Sheet1 using Close on the File menu.



8. Choose Arrange All again.

Your screen should look like this:



Excel spreadsheet files (with the extension *.xls*) can hold input information for HyperChem calculations and can receive output data from these calculations.

This sample spreadsheet includes the column headings of Molecule, Total Energy, Stretch, Bend, Torsion, and vdW. In the first column under the heading Molecule, a list of HIN file-names (without the *.hin* extension) appears, for which calculations will be performed.

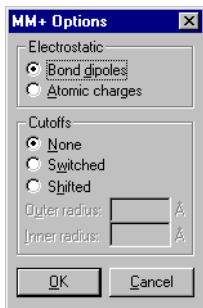
Excel provides a macro language for writing DDE messages. This language, together with HyperChem scripting commands, lets you carry out a set of calculations automatically.

The sample macro file, Plot.xml, contains HyperChem DDE commands.

## Setting up the Computation

Before you run the macro, you need to set the options for the computation.

1. L-click on the HyperChem workspace to bring it to the front of the screen.
2. Choose Molecular Mechanics on the Setup menu.
3. Choose MM+ as the Method, and then choose Options.
4. Use the following values in the Force Field Options dialog box.

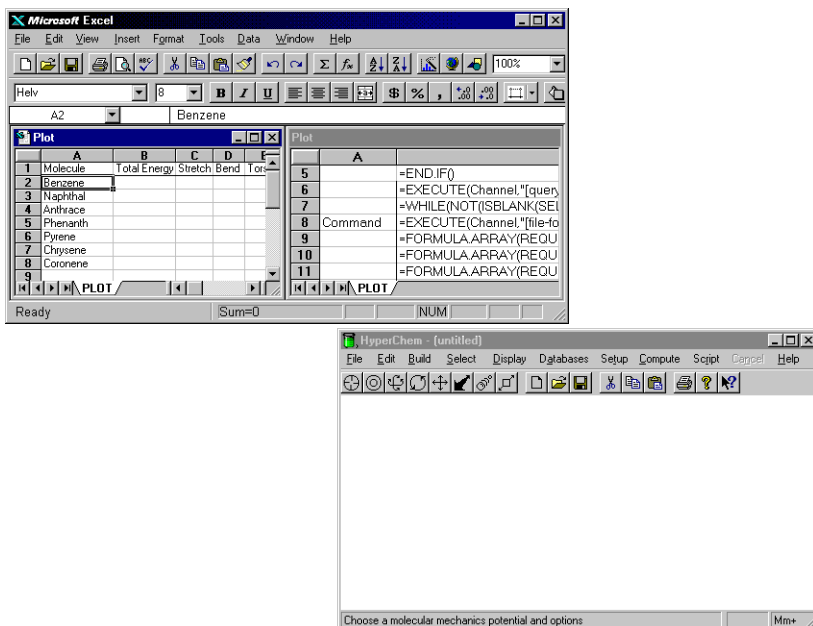


5. Choose OK.
6. Choose OK to close the Molecular Mechanics Force Field dialog box.

## Running the Macro

To run the macro:

1. In Excel, choose Arrange All on the Window menu to make the file plot.xls appear in the workspace.
2. L-click on benzene in cell A2 in plot.xls.
3. Rearrange your screen so that both the HyperChem and Excel windows appear, like this:



4. Choose Macro on the Tools menu, and double-click on PLOT.XLM!Compute.Results to run the macro.

The macro instructs HyperChem to open each HIN file, complete a single point calculation, and read the results back into Excel.

As the run progresses, the molecule is displayed in HyperChem and the values appear in the spreadsheet file.

Note: if you receive the warning message that a file cannot be opened, either the file is not there or the macro plot.xlm will

have to be edited in the OpenFile portion to reflect the correct pathname to the executable, chem.exe.

## Graphing the Results

You can now compare results for the different molecules by plotting the results in Excel. For this plot you compare the total energies.

To graph the results:

1. Increase the size of plot.xls to fill the Excel window.
2. L-click-drag from cell A2 and release on cell B8 to select the first two columns.

The screenshot shows a Microsoft Excel window titled 'Microsoft Excel - Plot'. The spreadsheet contains the following data:

	A2	Benzene									
	A	B	C	D	E	F	G	H	I	J	
1	Molecule	Total Energy	Stretch	Bend	Torsion	vdW					
2	Benzene	-1.492654	1.132	0	-5.58	2.96					
3	Naphthal	-7.422871	1.618	0	-15.4	6.36					
4	Anthrace	-13.436	2.105	0	-25.22	9.68					
5	Phenanth	-5.664224	2.105	0	-24.03	16.3					
6	Pyrene	-17.25645	2.215	0	-30.8	11.3					
7	Chrysene	-3.933764	2.592	0	-32.66	26.1					
8	Coronene	-31.05885	2.921	0	-51.78	17.8					
9											
10											
11											

The status bar at the bottom shows 'Ready', 'Sum=-80.264823', and 'NUM'.

This selects the names of the molecules and the total energies calculated by the single point calculations.

3. Choose the Chart Wizard button on the Excel toolbar. Select the Line chart type, and specify “Use First 1 Column(s) for Category (X) Axis labels”.

A bar graph displays the total energies in the file chart1. To change the appearance of the graph — for example, to move the labels above the axis instead of below it — double-click on the part of the graph that you want to change to open a dialog box for formatting.

Chrysene has the largest van der Waals (vdW) energy, because some hydrogens are very close together, producing some destabilizing strain in the molecule.

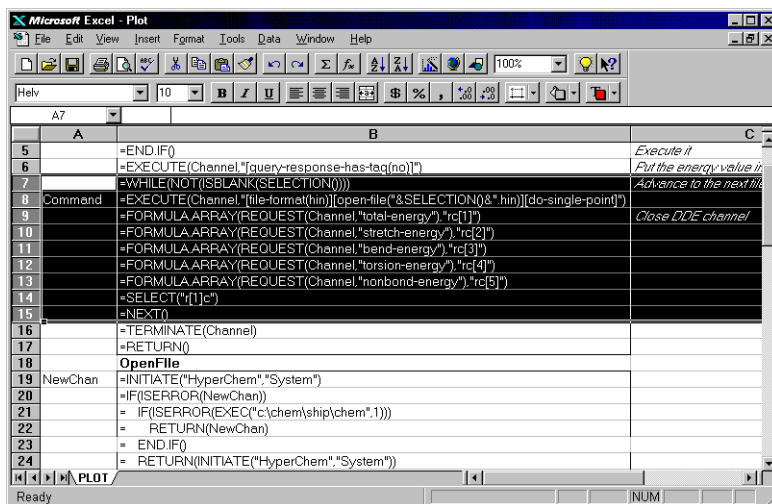
4. Choose Close on the File menu to close the plot.xls window. Do not save the changes for chart1.

## Exercise 2: Manipulating Molecules with Excel

In this lesson, you modify the Excel macro to rotate a benzene molecule.

### Modifying the Macro

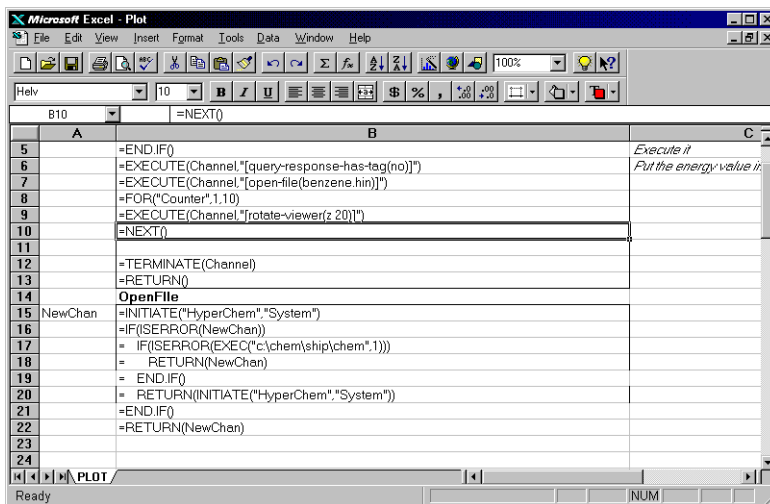
1. Choose plot.xlm on the Window menu.
2. Increase the window size of plot.xlm to full screen.
3. L-drag from A7 and release on A15.



4. Choose Delete on the Edit menu, and if a dialog box appears, choose Shift Cells Up.  
This deletes the information in the selected cells.
5. Now, to insert rows in the macro, L-click on cell 7, and choose Rows on the Insert menu.
6. Choose Repeat Insert Rows on the Edit menu four times.  
Now cells B7 through B11 are empty.
7. L-click on cell B7.

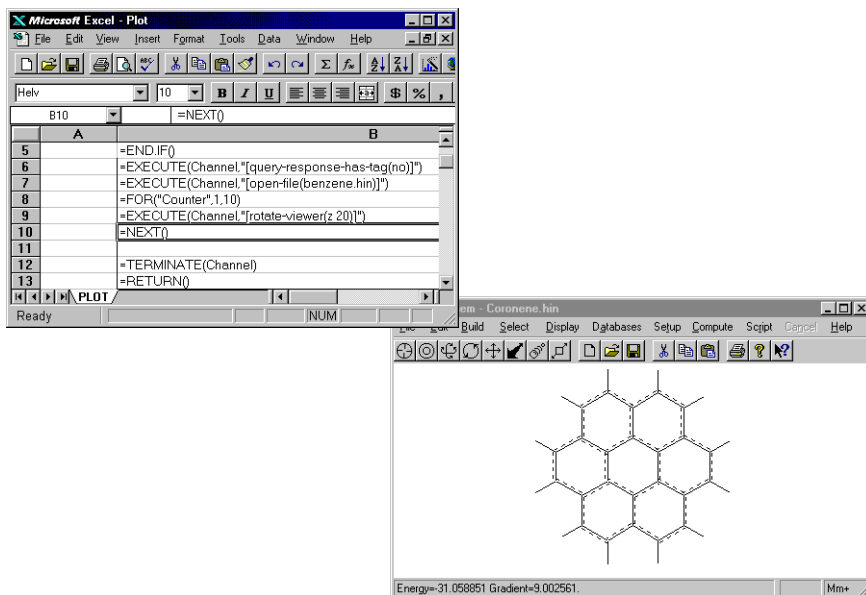
8. In cell B7, enter the following:  
=EXECUTE(Channel,"[open-file(benzene.hin)]")
9. Press [4] to move to cell B8 and enter the following:  
=FOR("Counter",1,10)
10. Enter the next two lines:  
=EXECUTE(Channel,"[rotate-viewer(z 20)]")  
=NEXT()

Your file should look like this:



## Running the Macro

1. Decrease and move the Excel window so that you can see the HyperChem window, like this:



2. In Excel, choose Macro on the Tools menu.
3. In the Run dialog box, choose PLOT.XLM!Compute.Results.  
The benzene rotates in the HyperChem workspace.

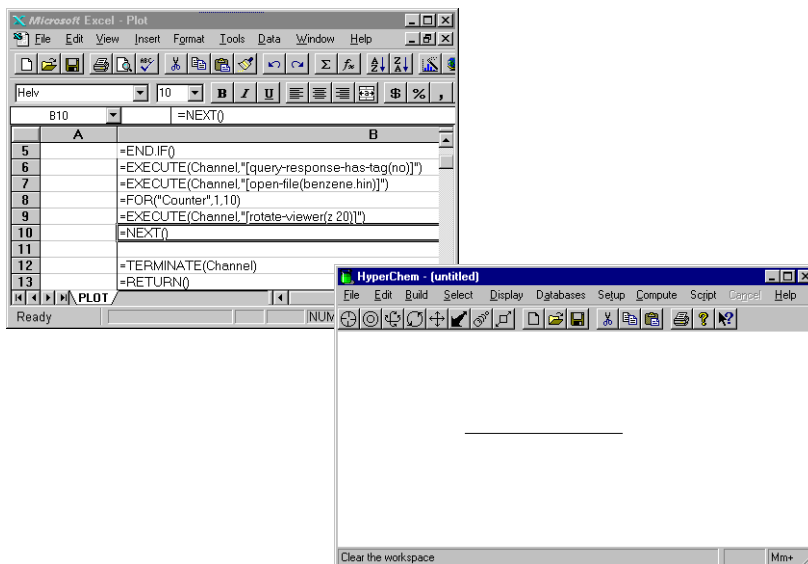
## Exercise 3: Plotting Energy vs. Torsion Angle

In this exercise, you create a model-built structure of hydrogen peroxide and modify the Excel macro from exercise 2 to change the torsion angle between the two hydrogen atoms. HyperChem reads the results back into Excel and you create a graph that plots the energy as a function of the torsion angle.

### Creating the Hydrogen Peroxide Molecule

First, create the model-built structure.

1. In HyperChem, choose New on the File menu to clear the workspace.
2. Double-click on the Drawing tool to open the Element Table dialog box and choose oxygen.
3. Draw two oxygen atoms bonded together.



4. Double-click on the Selection icon tool to invoke the Model Builder and produce the 3D model.

### Modifying the Macro

Modify the macro from exercise 2.

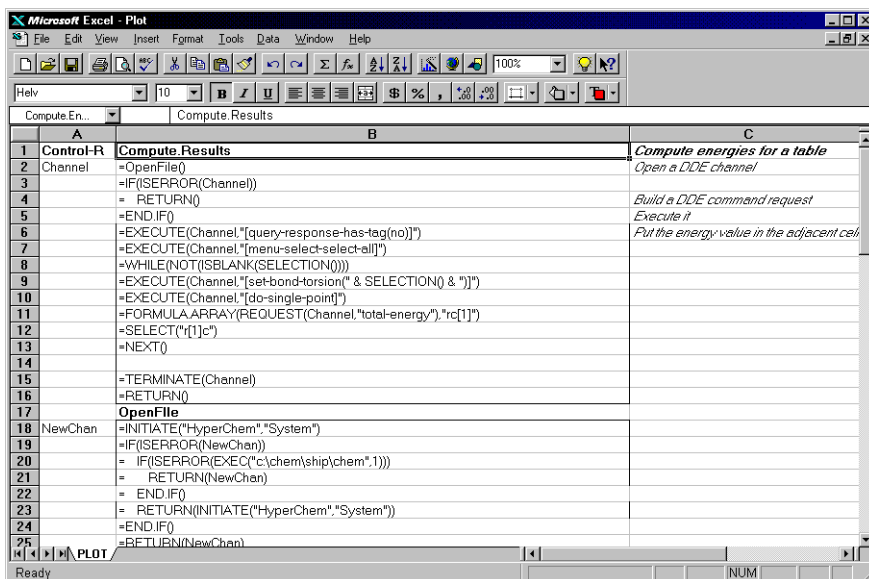


### Exercise 3: Plotting Energy vs. Torsion Angle

1. Go to Excel and increase the window for plot.xlm to full screen.
2. Delete the text in cells B7 to B9. If a Delete dialog box appears, choose Shift Cells Up.
3. Select cell B7 and choose Rows on the Insert menu.
4. Choose Repeat Insert Rows on the Edit menu 5 times.
5. Insert the following text:

```
=EXECUTE(Channel,"[menu-select-select-all]")
=WHILE(NOT(ISBLANK(SELECTION())))
=EXECUTE(Channel,"[set-bond-torsion("&SELECTION()&")]")
=EXECUTE(Channel,"[do-single-point]")
=FORMULA.ARRAY(REQUEST(Channel,"total-energy"),"rc[1]")
=SELECT("r[1]c")
```

Your file should look like this:



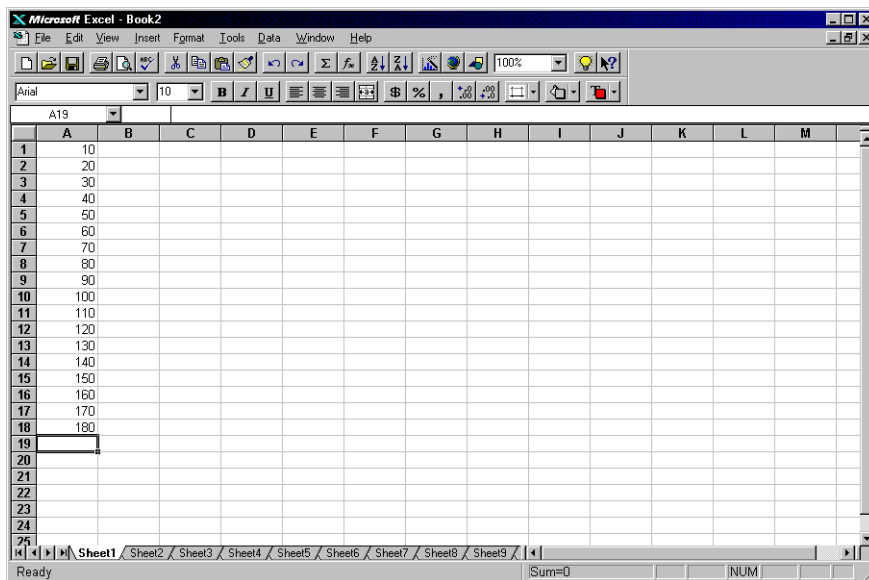
## Creating a Worksheet

Create a new worksheet and enter the values for the torsion angle.

1. Choose New on the File menu and choose Workbook.

A new worksheet, Sheet2, opens on screen.

2. In cells A1 through A18, enter values of 10–180 in increments of 10. This corresponds to values of the torsion angle in degrees. When you've finished, the worksheet should look like this:



3. L-click on cell A1.

When you run the macro, this cell is read first.

## Setting up the Computation

Before you run the macro, you need to set the options for the computation.

1. In HyperChem, choose Molecular Mechanics on the Setup menu.
2. Choose MM+ as the Method.

## Running the Macro

To rerun the macro:

1. Arrange the windows so that you can watch HyperChem and Excel simultaneously.
2. Choose Macro on the Tool menu and double-click on

### **PLOT.XLM!Compute.Results**

In the HyperChem workspace, the torsion angle between the two hydrogens changes, as instructed by the macro, and the corresponding total energies are stored in the new worksheet.

## Graphing the Results

To plot the results in a graph:

1. Increase the size of Sheet2 and select cells A1 to B18.
2. Choose New on the File menu, and choose the Chart Wizard button on the toolbar. Select the Line chart type. Specify "Use First 1 Column(s) for Category (X) Axis Labels".
3. In the First Column Contains dialog box, choose X-values for XY-Chart.

Excel displays a graph that plots the energy as a function of the torsion angle. The minimum energy occurs at approximately 110 degrees. The experimental value for the angle is 111 degrees.

*Note:* You need to further modify the macro used in this example if you want to calculate the energy as a function of the torsion angle for a molecule with more than four atoms. The macro used in this example instructs HyperChem to calculate the energy of the four selected atoms only. To do more than

four atoms, use named selections and then select the multiple-atom portion by name in the macro.

4. To finish this lesson, close HyperChem and Excel. Do not save your changes.

## For More Information

For information on HyperChem scripts and macros, see chapter 10, “Scripts and DDE,” in the *HyperChem for Windows Reference Manual*.

For information on Excel, see any number of *Microsoft Excel* books.

For more information on Dynamic Data Exchange, see for example, *Programming Windows*, by Charles Petzold.



# Lesson 18

## Driving HyperChem from Visual Basic

### Skills Covered in This Lesson

- Creating a Visual Basic program for use with HyperChem

This lesson further demonstrates how HyperChem can be driven by other Window applications through DDE.

Microsoft Visual Basic is a development system used to build a graphical user interface (GUI) for Windows applications.

In this lesson, you use Visual Basic to create a window, called a *form*. On the form you create a command button that instructs HyperChem to perform a single point calculation on benzene using the MM+ force field.

### Lesson Requirements

Before you begin, you must install Microsoft Visual Basic on your computer.

### Opening HyperChem and Visual Basic

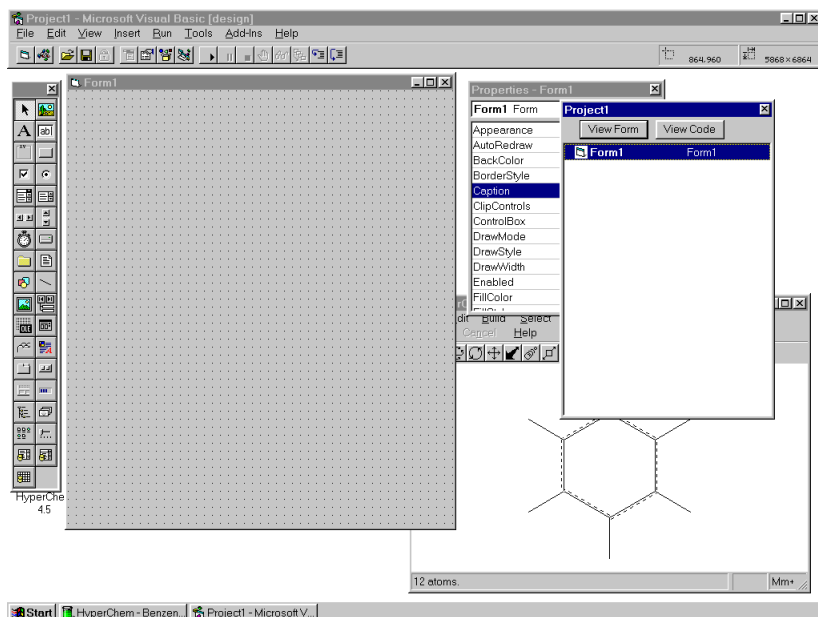
To begin, you open HyperChem, set the options for the type of calculation you want to perform, and then open Visual Basic.

1. Open HyperChem and decrease the size of the window. Place it in the lower-right corner of the screen.
2. Open the file benzene.hin.

This is the molecule you perform the single point calculation on.

3. Choose Molecular Mechanics on the Setup menu.
4. Choose MM+ as the Method, then choose OK.
5. If a dialog box appears and asks if you want to recalculate atom types, choose OK.
6. Open Visual Basic in the Program Manager.

Your screen will look something like this:

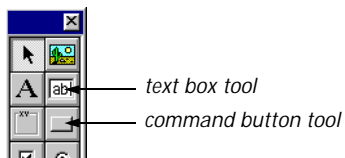


## Creating a Form

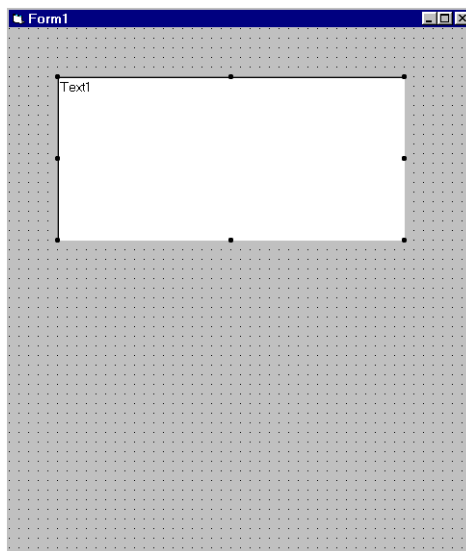
In Visual Basic, a *form* is a window that you create and customize. On forms, you draw graphical objects called *controls*, which are used to get user input and to display output. Controls include text boxes, command buttons, and list boxes.

To create a form:

1. On the toolbox, L-click on the Text box tool.



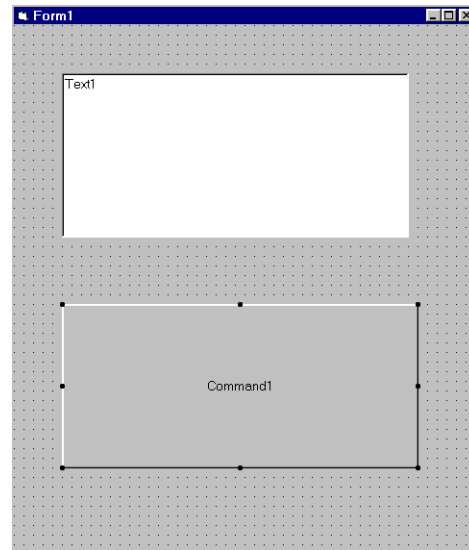
2. Move the pointer to Form1 and L-drag to draw a text box.



3. L-click on the Command button tool.
4. Move the pointer to Form1 and L-drag to draw a command button.

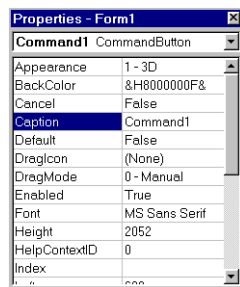
Form1 should look like this:





## Customizing the Interface

Now that you've created the form, you can customize the appearance and behavior of the controls using the properties box.



Visual Basic gives the text box the default name and caption Text1 and the command button the default name and caption Command1. For this exercise, you rename the caption of the command button to Report Calculation.

To customize the interface:

1. Highlight Command1 by L-dragging across the text in the properties bar.

2. Enter the following:

Report Calculation

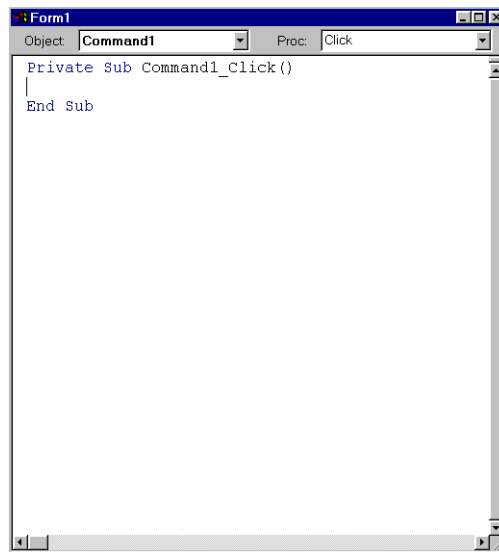
Then press [↵].

The control now has the caption Report Calculation.

## Entering Visual Basic Code

The next step is to enter the Visual Basic code that instructs HyperChem to do the single point calculation.

1. Double-click on the Report Calculation command button.



This opens the Code window named Form1.frm. The Code window is where you write, display, and edit Visual Basic code.

2. Enter the following text so the Code window looks like this:

```

Private Sub Command1_Click()
    Const COLD = 2, NONE = 0

    If Text1.LinkMode = NONE Then
        Text1.LinkTopic = "HyperChem\System"
        Text1.LinkMode = COLD
        Text1.LinkTimeout = -1
    End If

    Text1.LinkExecute " [do-single-poin]"

    Text1.LinkItem = "total-energy"
    Text1.LinkRequest
    MsgBox (Text1.Text)
End Sub

```

Notice that we intentionally left out the last “t” in “do-single-point.”

The Text1.LinkItem, which is “total-energy” here, could be any one of many possible HyperChem messages; see Chapter 10 of the Reference Manual for a list of HyperChem messages.

3. Choose Start on the Run menu.

Form1 appears.

4. L-click on the Report Calculation command button.

The following error message appears:

Ignoring bad text message: “do-single-poin”.

5. L-click OK.

A dialog box asks if you want to exit HyperChem.

6. Choose No.

The following error message is reported:

Foreign Application won't perform DDE method or operation.

7. Choose DEBUG or OK (depending on the version of Visual Basic you have).

Visual Basic shows you the line that includes the error.

8. Choose End on the Run menu.
9. Correct the error by adding a “t” at the end of “do-single-point.”
10. Choose Start on the Run menu.

Form1 appears.

11. L-click on the Report Calculation command button.

This starts the single point calculation. After about 1 minute, a message box titled Project1 reports the total energy of the benzene molecule.

12. Choose OK.
13. Choose End on the Run menu.

## Creating the Executable File

To create a stand-alone executable file:

1. Choose Make EXE file on the File menu.
2. Enter the following filename:

test.exe

Now you’ve created a program called test.exe. You can run this file when you want to do a single point calculation and report the total energy of a structure in HyperChem.

## Saving the Visual Basic Files

To save your work and exit Visual Basic:

1. Choose Save File As on the File menu.
2. Save the file as test.frm, then choose OK.
3. Choose Save Project As on the File menu.
4. Save the file as test.mak, then choose OK.

## **Adding test.exe to the Start... menu**

To add test.exe to a program group:

1. Click on Start on the Taskbar.
2. Click on Settings, then on Taskbar... .

The Taskbar Properties dialog box appears:

3. Choose the Start Menu Programs property sheet. Click on Add... to open the Create Shortcut dialog box.
4. Locate the file test.exe by browsing, then choose OK.

This brings you back to the Program Item Properties dialog box.

5. Specify a folder for the shortcut, a name (TEST) and an icon.

Then choose OK in the Taskbar Properties dialog box.

An icon called TEST appears in the new folder in the Taskbar. To open TEST, double-click on the icon.

## **Creating Other Programs**

You can use Visual Basic to create similar auxiliary programs to interact with HyperChem. For example, you could write a customized geometry optimizer in Visual Basic and use the molecular mechanics or quantum mechanics facilities of HyperChem via the DDE interface.

## **For More Information**

For more information on Visual Basic, see the Microsoft Visual Basic documentation.

# Tutorial 5

## Some Newer Capabilities

Tutorial 5 teaches you about some recent things that have been added to HyperChem such as annotations, density functional theory, and TNDO. Tutorial 5 has three lessons, as described in the following table:

Lesson	Information covered	Time to Complete
Polarizability of H <sub>2</sub> using DFT	Aspects of density functional theory (DFT) computations. Applying an electric field and computing polarizabilities.	20 minutes
Adding annotations to the workspace	Adding, positioning, editing, and sizing annotations.	20 minutes
Modifying TNDO parameters	Basic aspects of TNDO. The editing of parameters and creating parameter files.	20 minutes



# Lesson 19

## DFT Polarizability of H<sub>2</sub>

### Skills Covered in This Lesson

- Performing density functional calculations
- Picking an exchange-correlation functional
- Adding an electric field and computing polarizabilities

This tutorial will introduce you to the basic operations needed to perform DFT calculations.

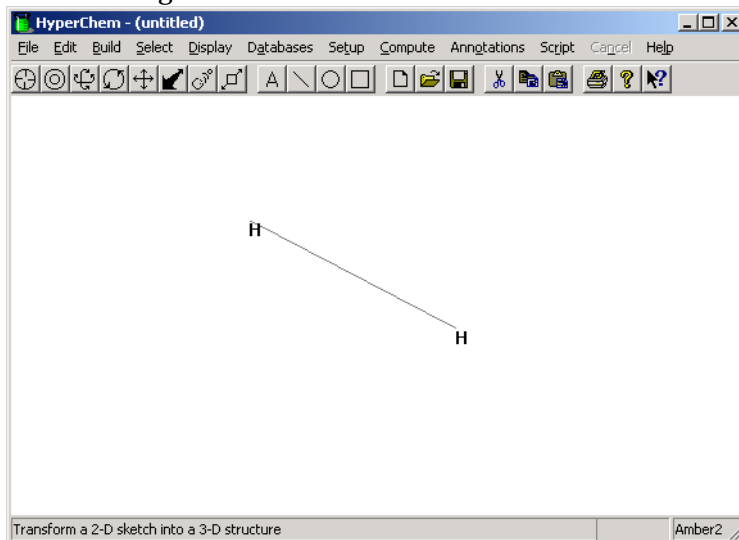
### Creating H<sub>2</sub>

To draw and create a Hydrogen molecule:

1. Choose Explicit Hydrogens on the Build menu.
2. Choose Labels... on the Display menu and select Symbols.
3. With the mouse and the drawing icon (left-most icon) draw a single line on the screen by dragging with the left button down. If you make a mistake, click with the right button to



delete atoms and bonds. You should have a screen resembling the following:



4. Click on Model Build on the Build menu to create the Hydrogen molecule with a bond length of 0.74 Angstroms oriented along the Y axis.

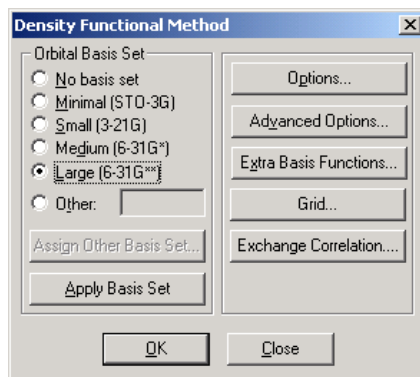
You now have a molecule with which to begin computations. We will perform calculations at the model builder geometry of 0.74 Angstroms although we could certainly optimize the geometry if we wished (see previous tutorials).

## Choosing an Exchange-Correlation Potential

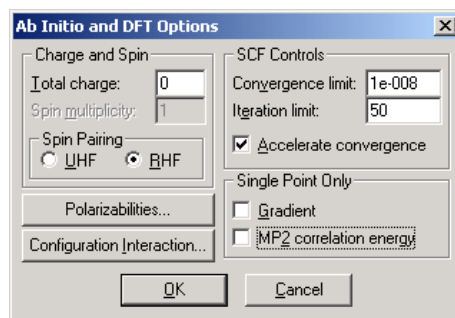
To indicate the type of DFT calculation you want to perform:

1. Click on Density Functional... on the Setup menu.

- Choose the 6-31G\*\* basis set as shown below. This basis set adds p-type polarization functions to the double zeta s-type 6-31G basis set of H<sub>2</sub>.

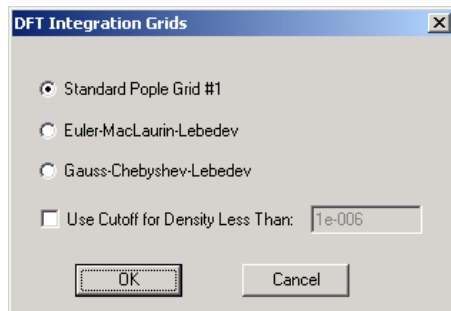


- Click on Options to select default options as shown below:



In particular, the convergence of the SCF (1.0e-008) should be set higher than normal (giving more accurate SCF results) as you are going to be numerically differentiating your results.

- Click on Grid... of the Density Functional Method dialog box and choose Standard Grid #1 as defined by Pople and collabo-

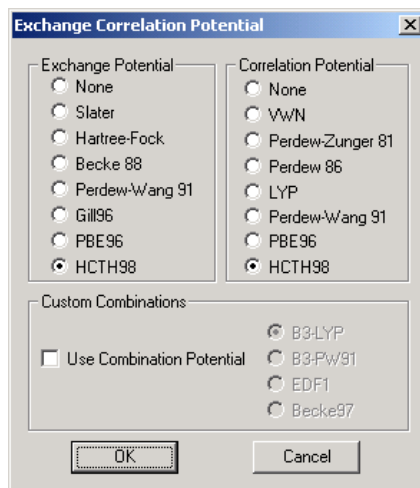


rators.

Density Functional Theory (DFT) requires you to evaluate matrix elements of the Hamiltonian just as in any quantum mechanical method. In the DFT case, terms in the Hamiltonian (the exchange and correlation potentials) have a functional form that is too complicated to allow one to develop analytic formulas for the appropriate integrals. DFT generally requires numerical integrations to compute these integrals and at this point you have to choose a grid for performing these numerical integrations.

The Standard Grid #1 is a compromise between accuracy and complexity (number of points on the grid). More points would give more accurate integrals but would require more computer time. The Standard Grid #1 grid gives results that are comparable to an Euler-MacLaurin-Lebedev grid with 50 radial points and 194 angular points. To complete the DFT setup, you next have to choose a specific density functional.

5. Click on Exchange Correlation... of the Density Functional Method dialog box to select both an exchange functional and a correlation functional. Select the HCTH98 method for both.



You can select any exchange functional in combination with any correlation functional or you can select a specific combination of exchange and correlation as optimized by various authors, e.g. B3-LYP. Many of the functionals were developed with the assumption that they would be used in specific pairs although HyperChem doesn't require this. HyperChem allows an independent choice for the exchange and an independent choice for the correlation. In our case we will choose both an exchange functional and a correlation functional (HCTH98) developed by the Handy group at Cambridge University. Little experience is probably available using only one of these pair with a different exchange (or correlation) functional. Whether one can develop an optimum exchange potential in itself without considering correlation (or vice versa) is open for discussion.

The choice "None" above is perfectly sensible for the correlation potential as Hartree-Fock calculations contain no correlation. However, the choice "None" for exchange is questionable although it corresponds to a *Hartree calculation* as performed in the early days of quantum chemistry.

## Single Point DFT Calculations in an Electric Field

To explore the application of an electric field to the workspace:

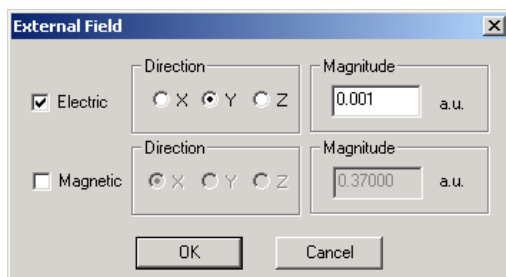
1. Click on Single Point to first perform a DFT calculation in the absence of an electric field at 0.74 Angstroms.

You should obtain an energy of -743.850760 kcal/mol.

2. Click on Show Dipole Moment on the Display menu.

The dipole moment is zero (or nearly so with single precision accuracy) as expected.

3. Click on Set Finite Field... of the Setup menu and add an electric field of 0.001 atomic units as shown below.



4. Click on OK to add the field.
5. Click on Single Point of the Compute menu to repeat the single point calculation but this time with an applied electric field.

You should obtain an energy this time of -743.852747 kcal/mol which is lower than without the electric field. The dipole moment in the electric field will no longer be zero.

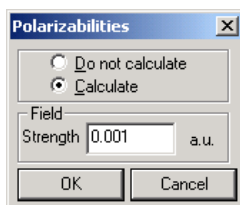
6. Click on Show Dipole Moment on the Display menu.

The induced dipole moment along the y axis is 0.16096 Debyes or  $0.16096/2.541765 = 0.00633$  a.u. The polarizability is the induced dipole moment per unit of applied field. Thus, the polarizability along the y axis is  $0.00633/0.001 = 6.33$  atomic units. You have performed a finite perturbation calculation of the polarizability of H<sub>2</sub> with the HCTH98 DFT functional. HyperChem, however allows you to directly calculate polarizabilities as you will now see.

## Polarizability Tensor Calculations

To calculate polarizabilities:

1. Click on Set Finite Field... of the Setup menu and delete the electric field you have previously applied (uncheck the check box for electric field).
2. Click on Density Functional... of the Setup Menu. Click on Options... and then Click on Polarizabilities... You can now request the computation of polarizabilities during single point calculations as per the following dialog box:.

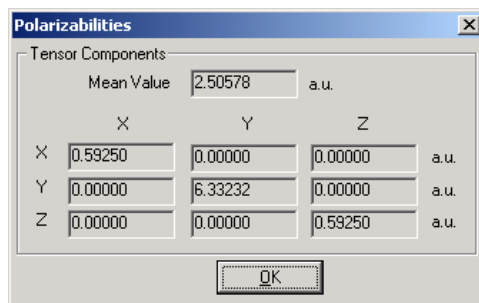


Use the same value of 0.001 which we used before rather than the default 0.0005. The default value is a compromise between precision of the change that can be observed and a value that is rigorously correct only in the limit of zero applied field.

3. Click on Single Point on the Compute menu to repeat the single point calculation but this time with the additional request for a polarizability calculation.

The computation will require 6 total single point calculations to apply the electric field twice in each of the X, Y, and Z directions.

To see the results after the calculations are complete, click on Properties... on the Compute menu. Click on Details to see the full tensor.



You will notice that the YY component is just the 6.33 value that we had previously computed manually.

# Lesson 20

## Annotations

### Skills Covered in This Lesson

- Adding annotations to a molecule
- Selecting, editing, and moving annotations
- Z-ordering the annotation layer and annotations within the annotation layer

This tutorial will introduce you to the four basic annotations - text, lines, ellipses (circles), and rectangles. It will show you how to create annotations and modify their, size, color, and other properties. HyperChem contains, in addition to its molecular modeling capability, a simple drawing program designed to create annotations for molecular systems.

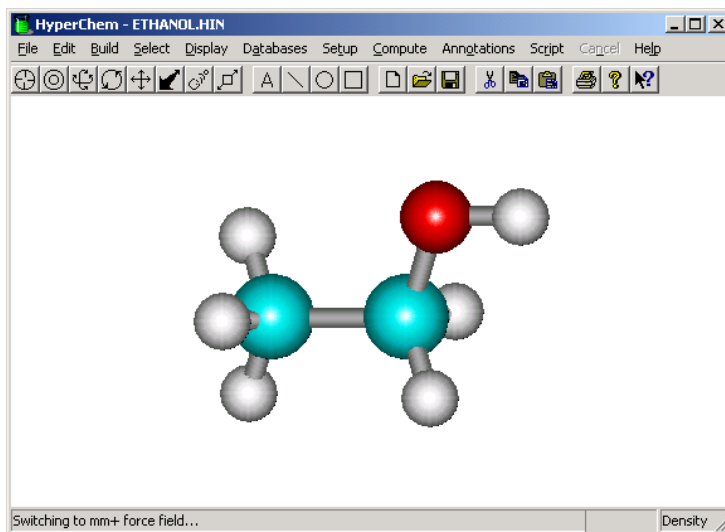
### Create a Molecular System

To create a sample molecular system:

1. Choose Open on the File menu.
2. Navigate to the Samples folder and bring in ethanol.hin.  
Choose a ball and cylinder rendering from the Display menu



to obtain a sample molecular system such as the following (alternatively, use your own molecular system).

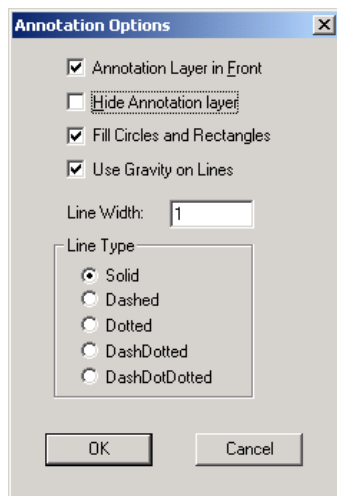


## Rectangle Annotations

Rectangle and ellipse annotations behave essentially identically. This tutorial only illustrates rectangles. You might try equivalent operations on ellipses yourself.

To explore creating and manipulating a rectangle annotation:

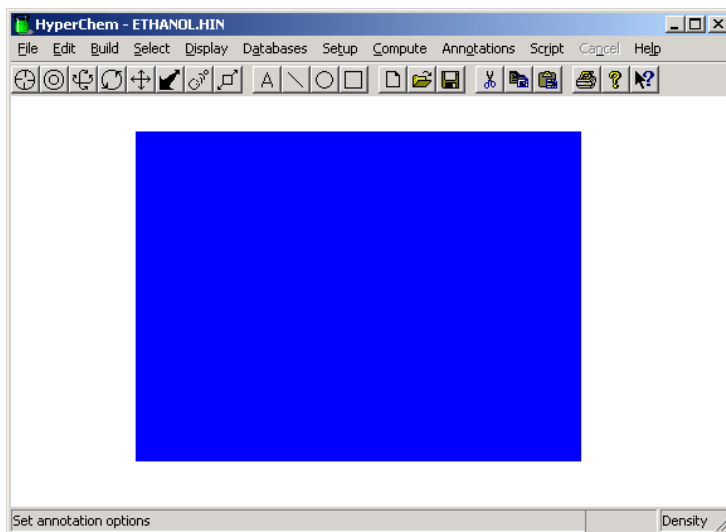
1. Choose Options... on the Annotations menu and select options that reflect those in the following dialog box:



You have selected the annotation layer to be positioned in front of the molecular layer, the annotation layer not to be hidden from view, ellipses and rectangles to be filled rather than just outlined, and “gravity” to be applied to line drawings.

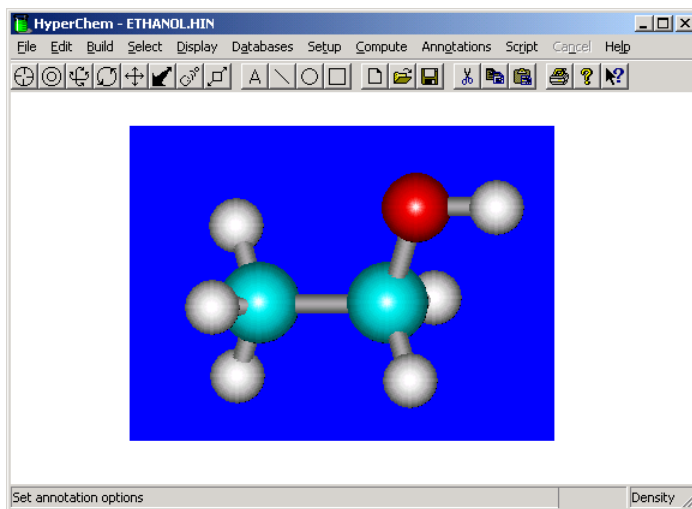
2. Choose Color... on the Annotations menu and select a blue color. This will be the color of any line, ellipse, or rectangle annotation.
3. Click on the toolbar icon that looks like a square. Notice that when the mouse pointer is in the workspace it has the shape of a square.
4. Position the mouse pointer near the top, left-hand corner of the workspace and with the left mouse button hold down, drag the mouse towards the lower, right-hand corner and then

release the mouse button. A rectangle will be created as follows:



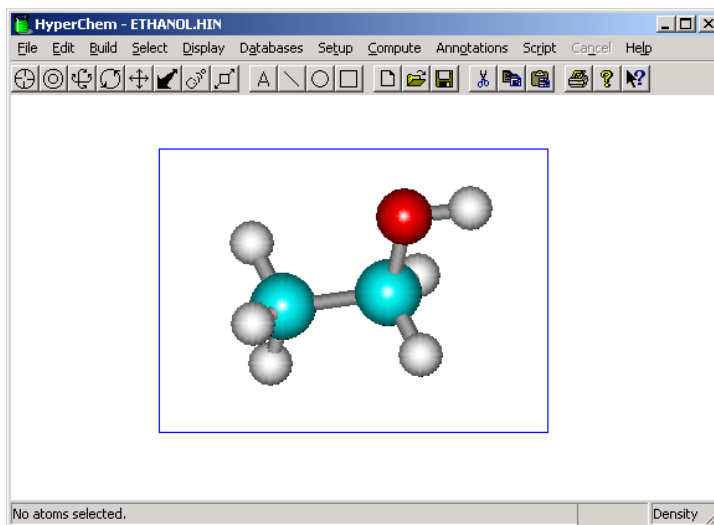
All molecules lie in a molecular layer while all annotations lie in a disjoint annotation layer. In the above situation, the annotation layer is in front of the molecular layer so that the molecule is hidden from view.

- Choose Options on the Annotations menu and deselect Annotation Layer in Front to place the annotation layer in back and create a display as follows:



- Choose the select tool and L-click in the rectangle away from the molecule to select the rectangle annotation. The selected rectangle should have a border that is the selection color (default=green). If you need to deselect, R-click in empty space. Practice selecting and deselecting either the rectangle or a portion of the molecule
- Explore translation and zooming of the selected rectangle using the translation and zoom tools. Notice that if the molecule is selected it has precedence over any annotation and only the molecule is affected by the tools. If only annotations are selected, then those annotations are acted upon by the translate and zoom tools; the other tools are inactive for annotations.
- Select only the rectangle.
- Choose Options on the Annotations menu and uncheck Fill Circles and Rectangles. Choose OK.

10. Deselect the rectangle. You should see the fill disappear and the rectangle look as follows:



11. Select the rectangle again and choose Options on the Annotations menu. Choose line type to be "Dashed". Deselect the rectangle. You should see the solid rectangle above become a dashed rectangle.

## Line Annotations

Line annotations behave like ellipses and rectangles apart from not having a fill. They differ also in possibly having "gravity" to cause lines to join.

*Note:* The line annotation has nothing in common with a bond drawn as a line in the molecular layer using the drawing tool (left-most tool).

To explore drawing lines:

1. Select the line annotation tool from the tool bar. Draw a few lines and explore selecting these, coloring them, moving and zooming them. Explore the use of gravity associated with the check box in the Options dialog box on the Annotations menu. Note that only solid lines can be wide.

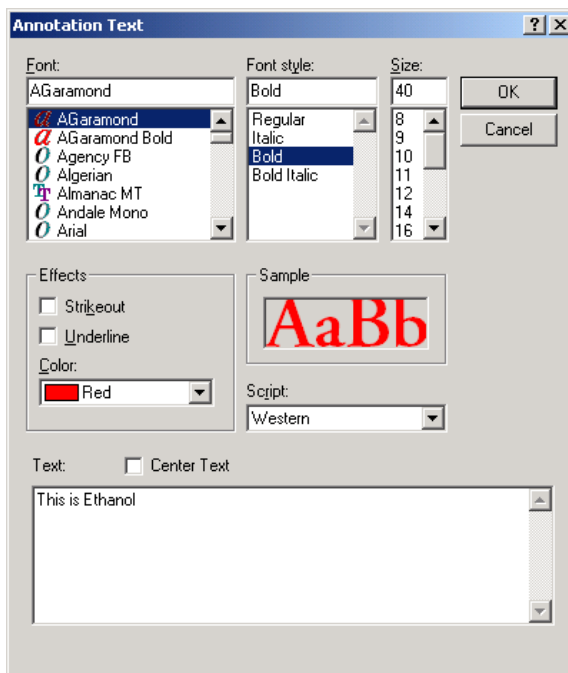
## Text Annotations

Text annotations and symbol annotations are identical except that a symbol is a single text character and has a different interface for its creation. Once a symbol is created, it is just a text annotation like any other text annotation.

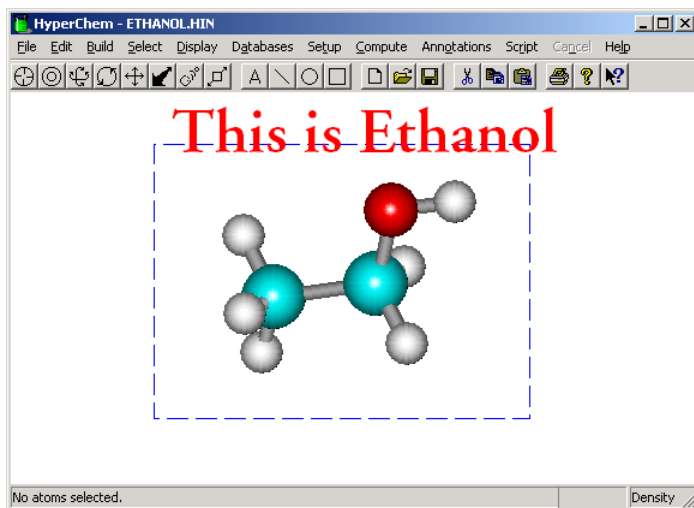
Text annotations differ in their color properties from lines, ellipses, and rectangles. The color is chosen from among only 16 colors. They also have unique properties of their own such as font, point size, centering, underlining, italics, etc.

To explore creating and manipulating text annotations:

1. Click on the text-annotation icon in the tool bar to select the text tool (resembling the letter A)
2. L-Click above the top left of the previous rectangle. This will bring up the Annotation Text dialog box,



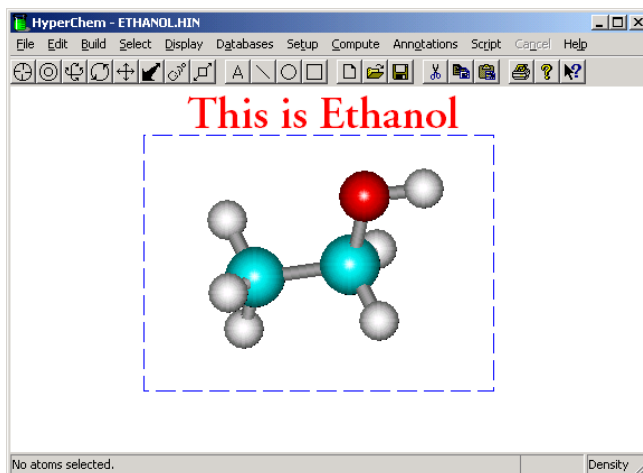
3. Choose the text options as above to create a title in the workspace, as shown below:



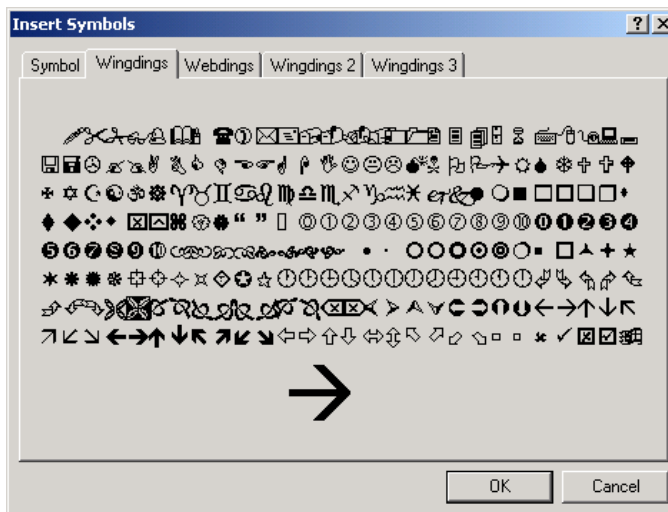
The text may need to be edited or positioned as in the above example.

4. Select the text annotation by clicking on it appropriately with the selection tool. It will become enclosed in a selection box of the selection color.
5. Choose Edit Text... from the Annotations menu and change the text size from 40 to 30 points.

6. Use the translation tool and position the text properly over the rectangle as shown below:



7. Choose Symbols from the Annotation menu and pick an arrow to use such as that below. Click on OK.



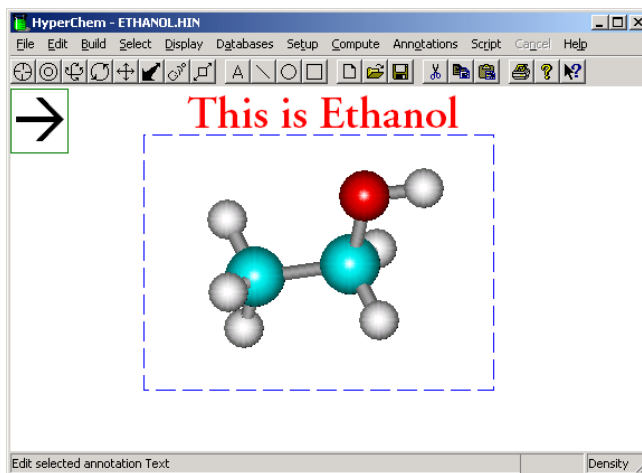
Depending on the fonts installed on your machine, you may not have all the options shown above.

The default color of a symbol is white (the default color of the HyperChem background is black) and it may not be visible if your background is white. A symbol is always initially placed



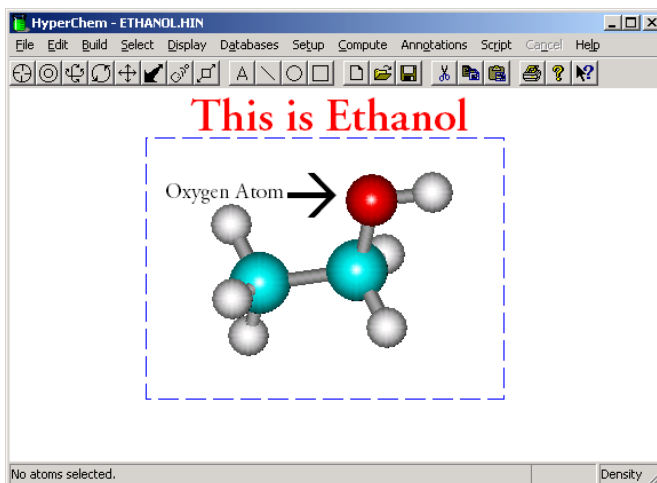
at the upper left corner of the HyperChem workspace. It is selected and ready to be translated to the correct position on the screen.

8. Choose Edit Text... from the Annotations menu and select the appropriate color for your symbol. Notice that the symbol is listed as a single character in the appropriate font. Choose OK.



9. Translate the arrow to point to the oxygen atom. Deselect the arrow. You can deselect everything by an R-Click in empty space with the select tool.

10. Create another text annotation using the text-annotation tool on the tool bar using the text, "Oxygen Atom" and position it as shown below:



If you have trouble selecting a particular annotation you can use the Select Next and Select Previous items on the Annotations menu. You can only use these when a single annotation is selected. You can only edit annotation text when a single *text annotation* is selected.

## Saving Annotations

Annotations have defined records in HIN files and hence annotations can be saved with or without a molecular system.

To explore annotations in HIN files:

1. Save the above molecule and annotations by choosing Save on the File menu.
2. After saving the HIN file, choose New on the File Menu.
3. Choose Open on the File menu and read in the file that you just wrote. The annotations were obviously included in the HIN file.



# Lesson 21

## TNDO Parameter Files

### Skills Covered in This Lesson

- Performing TNDO calculations
- Editing TNDO (and other) parameters directly
- Creating new parameters for specific types

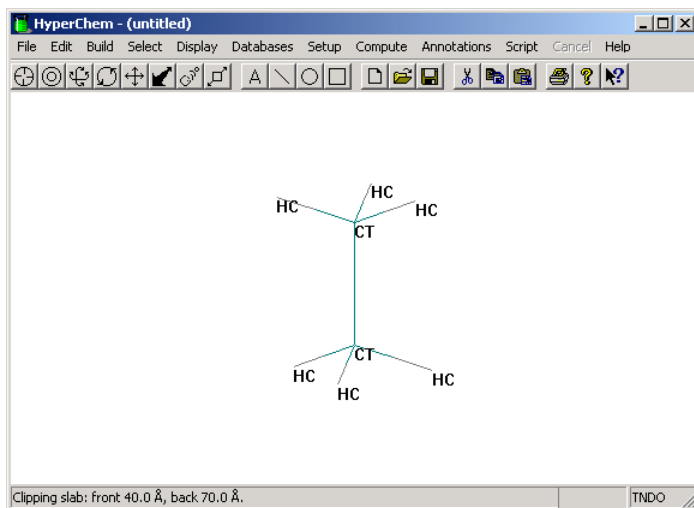
This tutorial will introduce you to the basic manipulations involved in Typed Neglect of Differential Overlap (TNDO) calculations. You will explore the basic idea of atom types in semi-empirical calculations and develop new parameters for the TNDO method.

With a default set of parameters, TNDO is just CNDO (or INDO). We will explore how CNDO specifically predicts the C-C bond lengths in ethane and ethene and develop new parameters that are specific to  $sp^2$  and  $sp^3$  systems.

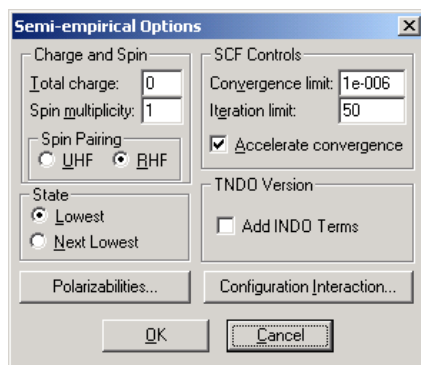
### Default CNDO/INDO Parameters

To explore the CNDO bond lengths of ethane and ethene:

1. Create an ethane molecule in the workspace and apply labels that are the *atom type*.



2. Choose Semi-empirical on the setup menu and chooses TNDO as the semi-empirical method to use. Click on Options and choose default values as shown below:



In particular, do not choose to Add INDO Terms. Without this checked, TNDO is equivalent to CNDO (with or without special parameters).

3. Choose Select Parameter Set... on the Setup menu and choose the "default" parameter set. Click on OK.

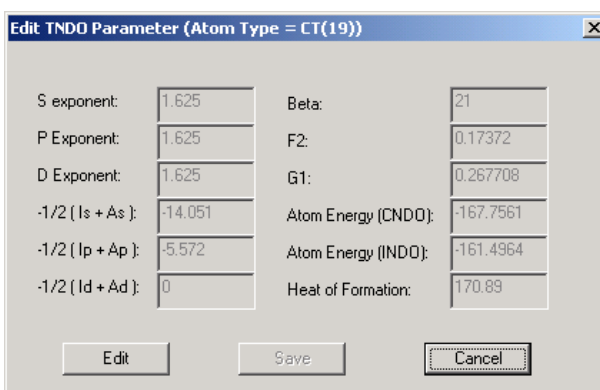


4. Use Polak-Ribiere to optimize the structure of Ethane to a convergence of 0.01 in the RMS gradient. You should find that the optimized bond length is about 1.457 Angstroms. You might like to repeat the calculation with the CNDO method to assure yourself you get the identical answer. Repeat for Ethene to obtain an optimized bond length of about 1.310 Angstroms.

## Inspecting TNDO Parameters

To inspect the TNDO parameters:

1. Select a single carbon atom of ethane and choose Edit Parameters... on the Setup menu. The following dialog box shows the default TNDO parameters which are identical to the CNDO/INDO parameters.



In TNDO these parameters are associated with a particular atom type not atomic number. In this situation the type is CT

as per the typing rules for Amber (which are borrowed for TNDO).

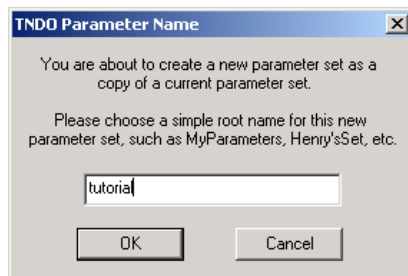
Note particularly the value of beta which is 21. This beta quantity is the parameter which is basic to the bond strength and off-diagonal elements of the Fock matrix. A larger value of beta should give stronger bonding while a smaller value of beta ought to give weaker bonding. Before we edit these default values we ought to create a new parameter file to experiment with.

*Note:* You can also inspect parameters for other semi-empirical methods or molecular mechanics methods. With molecular mechanics, you will see different parameters depending upon whether you select an atom, a bond, an angle, or a torsion.

## Copying TNDO Parameters to a New Set

To copy the “default” parameter set to a set named “tutorial” to be used for experimentation:

1. Choose Select Parameter Set... on the Setup menu as before to see the TNDO parameter sets. Select the default set and click on the Copy button. HyperChem will now ask you to choose a name for the copy:



2. Name the new parameter set by typing “tutorial” in the edit box. Hit OK. You will now be able to see this new parameter set described in the Registry if you wish.

## Editing TNDO Parameters

To edit the tutorial set :

1. Choose Select Parameter Set... on the Setup menu and select your new tutorial parameter set as the set to use (this set can now be experimented with without affecting other parameter sets).
2. Select a single carbon atom of ethane and choose Edit Parameters... on the Setup menu.
3. Push the edit button and type in a new value for beta. Click on OK. Experiment with different values of beta and their effect on the optimized C-C bond length.

You should try values for beta that are less than 21 to see if you can improve the C-C bond length from the 1.457 Angstrom value to something close to 1.54 Angstroms. A CT value for beta of about 16 will give you a much better C-C bond length.

4. Experiment with Ethene. You will find that the default value for beta of 21 leads to a C=C bond length of 1.310 Angstroms much smaller than the 1.34 Angstroms characteristic of C=C double bonds. A new CX beta parameter of about 19 will give a much better bond length for ethene.

The result of this lesson is that you will have created a new TNDO parameter file, tutorial.abp, which contains new beta parameters for C-C  $sp^3$ - $sp^3$  single bonds (type CT-CT) and for C=C  $sp^2$ - $sp^2$  double bonds (type CX-CX). You might now try these on propene to see how well they work in a new situation.





# **HyperChem<sup>®</sup> Modules**

**Part 1  
HyperChem Data**

**Part 2  
HyperNMR**

# Copyright © 2002 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperTNDO, HyperDFT, HyperGauss, HyperNMR, HyperChem Data and ChemPlus are trademarks of Hypercube, Inc.

## Third Party Trademarks

MM2 is a trademark of Dr. Norman Allinger, University of Georgia.

CHARMm is a registered trademark of Accelrys Inc.

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows, Windows 95, Windows 98, Windows ME, Windows NT, and Windows XP are trademarks of Microsoft Corporation.

IBM is a registered trademark of International Business Machines, Inc.

ChemDraw is a trademark of CambridgeSoft Corporation.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

## Part 1: HyperChem Data

<b>Chapter 1 Introduction to HyperChem Data</b>	<b>3</b>
A Database Module . . . . .	3
Databases in HyperChem Data . . . . .	3
Databases . . . . .	3
Tables . . . . .	4
The Molecular Database . . . . .	4
Naming Molecules . . . . .	4
HDF String . . . . .	5
HIN String . . . . .	6
Properties . . . . .	6
Basic Window . . . . .	6
Basic Operations . . . . .	8
Drawing Molecules . . . . .	8
Database Reading . . . . .	9
Database Creation . . . . .	9
Basic Entity is Table . . . . .	9
Standard Fields . . . . .	10
Custom Fields . . . . .	10
Adding Records . . . . .	10
Database Editing . . . . .	10
Database Querying . . . . .	10
Simple SQL Queries . . . . .	11
2D Substructure Searches . . . . .	11
3D Substructure Searches . . . . .	12
Generic Searches . . . . .	12
HyperChem-Related Operations . . . . .	12

## Chapter 2 Tutorials 15

Overview . . . . .	15
Summary of HyperChem Data . . . . .	15
Tutorials. . . . .	16
1. Starting HyperChem Data . . . . .	16
2. Opening a Database . . . . .	17
3. Querying a Database (SQL GUI Query) . . . . .	21
4. Querying a Database (Substructure Query) . . . . .	23
5. Querying a Database (Generic Query). . . . .	27
Overview . . . . .	27
How It Works . . . . .	27
Example . . . . .	28
6. Adding New Fields to a Table . . . . .	29
HSV Fields . . . . .	31
7. Adding and Removing Records From a Table . . . . .	32
Adding Records . . . . .	32
Deleting A Record . . . . .	35
8. Editing an Existing Record . . . . .	35
9. Drawing a Molecule . . . . .	36
Background . . . . .	36
10. Saving and Retrieving a Drawing. . . . .	41
11. Creating a New Database . . . . .	42
Format of Databases . . . . .	42
New Database Fields . . . . .	42
Example . . . . .	43
12. Using HyperChem with HyperChem Data. . . . .	48
Computing Molecular Properties with HyperChem Hcl Scripts . . . . .	49
Run Script... . . . .	49
Output to Log file: . . . . .	49
Output to HyperChem Data Table: . . . . .	50
Run Script and Retrieve HIN...: . . . . .	52
13. Importing HyperChem Data Records into MS Excel. . . . .	52
Importing Data into MS Query and Excel. . . . .	53
14. Saving, Compacting and Repairing Databases . . . . .	53
Creeping Database Size . . . . .	53
Closing a Database. . . . .	54
Compacting a Database . . . . .	54
Repairing a Database . . . . .	54
15. Getting a Molecule from the Database into HyperChem . . . . .	55

## Chapter 3 Reference 59

Overview . . . . .	59
Menu Items . . . . .	59
File Menu . . . . .	59
New Database... . . . .	59
Open Database . . . . .	59
Close Database . . . . .	59
Tables . . . . .	59
Close Table . . . . .	59
Compact Database . . . . .	60
Repair Database . . . . .	60
Close Query . . . . .	60
Save Query As . . . . .	60
Import Picture . . . . .	60
Save Picture As. . . . .	60
Get Picture Set. . . . .	60
Edit Menu . . . . .	60
Copy. . . . .	60
Delete . . . . .	60
Align Label . . . . .	61
Select All . . . . .	61
Select None. . . . .	61
Add Record. . . . .	61
Edit Record. . . . .	61
Discard Changes . . . . .	61
Update Record. . . . .	61
Delete Record . . . . .	61
Query Database . . . . .	61
Display Menu . . . . .	62
Refresh Screen . . . . .	62
Clear Screen . . . . .	62
Center Picture . . . . .	62
Tool Menu . . . . .	62
Select. . . . .	62
Single Bond . . . . .	62
Double Bond . . . . .	62
Right-Side Double Bond. . . . .	62
Left-Side Double Bond . . . . .	63
Triple bond. . . . .	63
Text . . . . .	63
Rings. . . . .	63
Text Colors . . . . .	63
Draw Colors . . . . .	63

HyperChem Menu . . . . .	63
Get Hinfile . . . . .	64
Put Hinfile . . . . .	64
Delete Hinfile . . . . .	64
Run Script . . . . .	64
Run Script and Retrieve HIN. . . . .	65
Run Script and Delete Records . . . . .	65
Bond Menu . . . . .	65
Constrained . . . . .	65
Unconstrained . . . . .	65
Help Menu . . . . .	65
Drawing Tools . . . . .	66

## **Chapter 4 HDF Files 69**

HDF Files . . . . .	69
PRIMITIVE Records . . . . .	70
Atom Records . . . . .	71
Label Records . . . . .	71

# Part 2: HyperNMR

## **Chapter 5 HyperNMR 75**

Overview . . . . .	75
--------------------	----

## **Chapter 6 Installation Guide 77**

Introduction . . . . .	77
System Requirements . . . . .	77
Installing HyperNMR . . . . .	78
To install HyperNMR:. . . . .	78
WIN.INI settings . . . . .	78
Interaction of HyperNMR and HyperChem . . . . .	79

## **Chapter 7 Introduction to HyperNMR 81**

Introduction . . . . .	81
------------------------	----

Some Basic Ideas . . . . .	81
Stand-alone or Combined . . . . .	81
Quantum Atoms . . . . .	82
NMR Atoms . . . . .	82
Labelled Atoms . . . . .	82
Equivalent Atoms . . . . .	83
Using HyperNMR for calculations . . . . .	83

## **Chapter 8 Tutorials 85**

Overview . . . . .	85
Tutorial I: HyperNMR Basics . . . . .	85
Starting HyperNMR . . . . .	85
To start HyperNMR: . . . . .	85
HyperNMR Window. . . . .	86
Title Bar . . . . .	87
Menu Bar . . . . .	87
Tool Bar . . . . .	87
Workspace . . . . .	87
Status Line . . . . .	87
Help . . . . .	87
Control-Menu Button . . . . .	87
Maximize/Minimize Buttons . . . . .	88
Using the Mouse . . . . .	88
Keyboard Alternatives . . . . .	88
To open a menu: . . . . .	88
To close a menu: . . . . .	88
To choose a menu item: . . . . .	89
To rotate a molecular system: . . . . .	89
To translate a molecular system: . . . . .	89
Bringing a Molecule into HyperNMR . . . . .	89
Opening a Sample File . . . . .	89
To open a HIN file: . . . . .	90
To change directory: . . . . .	91
To change file type: . . . . .	91
To scroll through the list: . . . . .	92
To select a file: . . . . .	92
Bringing a Molecule in from HyperChem . . . . .	94
Getting a molecule from HyperChem: . . . . .	94
Changing Renderings . . . . .	94
To open rendering dialog box: . . . . .	95
Changing Atom Renderings . . . . .	96
To display labels: . . . . .	96



To display disks and spheres: . . . . .	97
Changing Bond Renderings: . . . . .	99
To display bonds: . . . . .	99
To display cylinders: . . . . .	99
Translating, Rotating, and Scaling Molecules . . . . .	100
XY translation: . . . . .	100
Zoom: . . . . .	101
Scale to fit: . . . . .	101
XY rotation: . . . . .	101
Z rotation: . . . . .	102
Selecting and Deselecting . . . . .	102
To select and deselect an individual atom: . . . . .	102
To select and deselect multiple atoms: . . . . .	103
To select and deselect the molecular system: . . . . .	103
Copying a Molecular Image to the Clipboard . . . . .	103
To copy a molecular image to the Clipboard: . . . . .	103
Exiting HyperNMR . . . . .	104
To exit HyperNMR: . . . . .	104
Tutorial II: Calculation of NMR Shielding & Coupling . . . . .	104
To reopen HyperNMR: . . . . .	104
To load a molecular system: . . . . .	104
Setting Atom Types . . . . .	105
To check atom types: . . . . .	105
Setting Up Options . . . . .	105
Opening the Shielding & Coupling Options dialog box: . . . . .	105
Selecting quantum atoms: . . . . .	107
Selecting NMR atoms: . . . . .	108
Setting up equivalent atoms: . . . . .	110
Selecting labelled atoms: . . . . .	111
Setting up options for the SCF calculation: . . . . .	112
Computing Shielding and Coupling Constants . . . . .	113
To start a log file: . . . . .	113
To compute shielding and coupling: . . . . .	114
Viewing Shielding and Coupling Constants . . . . .	115
To view shielding constants: . . . . .	115
To view coupling constants: . . . . .	117
Saving Molecules and Calculations . . . . .	118
To save results: . . . . .	118
Tutorial III: Calculation of NMR Spectra . . . . .	118
Setting up Options . . . . .	118
To set up the reference shielding: . . . . .	119
To set up type of spectrum: . . . . .	119
To set up the spectrometer frequency: . . . . .	120

Computing an NMR Spectrum . . . . .	121
To compute an NMR spectrum: . . . . .	121
To stop logging to a log file: . . . . .	122
Viewing NMR Spectra . . . . .	122
Line View of an NMR spectrum . . . . .	122
To view peaks of an NMR spectrum as lines: . . . . .	122
Envelope View of an NMR Spectrum. . . . .	123
To view an NMR spectrum as an envelope: . . . . .	124
To change the line width function: . . . . .	124
To change line widths: . . . . .	125
Investigating Peaks in an NMR Spectrum . . . . .	126
To zoom in on a region of the spectrum: . . . . .	126
To select and integrate peaks: . . . . .	127
To view a table of peak information: . . . . .	129
Restoring Expert Options . . . . .	130
To restore Expert Options: . . . . .	130
Tutorial IV: Protein NMR . . . . .	130
Making Various Selections. . . . .	131
Opening a protein molecule: . . . . .	131
Selecting quantum atoms with HyperNMR: . . . . .	132
Selecting quantum atoms with HyperChem: . . . . .	133
Selecting NMR atoms with HyperNMR: . . . . .	133
Selecting NMR atoms with HyperChem: . . . . .	134
Selecting equivalent atoms with HyperNMR: . . . . .	134
Selecting equivalent atoms with HyperChem: . . . . .	135
Getting the protein from HyperChem . . . . .	136
Computing NMR Shielding, Coupling and Spectrum . . . . .	136
Setting up NMR shielding and coupling options: . . . . .	136
Setting up NMR spectral options: . . . . .	136
Computing NMR shielding, coupling and spectrum: . . . . .	137
Displaying and Copying NMR Results . . . . .	137
Displaying, plotting and copying shielding constants: . . . . .	137
Displaying and copying coupling constants: . . . . .	138
Displaying and copying NMR spectrum: . . . . .	139

## Chapter 9 Reference 141

Overview . . . . .	141
What is HyperNMR . . . . .	141
Starting HyperNMR . . . . .	142
To start HyperNMR: . . . . .	142
HyperNMR Window. . . . .	142
Molecular System View . . . . .	144

	Shielding Constant and Chemical Shift View . . . . .	145
	Spin-Spin Coupling Constant View . . . . .	147
	Spectrum View . . . . .	148
	Using the Mouse . . . . .	149
	Using the Keyboard . . . . .	151
	HyperNMR Menus. . . . .	151
	HyperNMR Coordinate Systems . . . . .	152
	Viewer's Coordinate System:. . . . .	152
	Molecular Coordinate System:. . . . .	152
Menus . . . . .		153
	File Menu . . . . .	153
	Edit Menu . . . . .	154
	View Menu . . . . .	155
	Tool Menu . . . . .	157
	Setup Menu . . . . .	157
	Compute Menu . . . . .	158
	Help Menu . . . . .	159
Tools . . . . .		159
	Tool Buttons . . . . .	159
	Manipulating Molecules with the Mouse . . . . .	161
	Using the mouse to rotate the molecule . . . . .	161
	Using the mouse for selection . . . . .	161
	Using the mouse for zooming and translation . . . . .	161
	Keyboard Equivalents for Mouse Actions . . . . .	162
Dialog Boxes . . . . .		162
	Open File Dialog Box . . . . .	163
	Save As Dialog Box . . . . .	164
	Start Log Dialog Box . . . . .	165
	Preferences Dialog Box . . . . .	165
	Replace Selection Dialog Box . . . . .	167
	Rendering Dialog Box . . . . .	167
	Shielding Tensor Dialog Box . . . . .	168
	Spectral Values Dialog Box . . . . .	169
	NMR Graph Dialog Box . . . . .	170
	Shielding & Coupling Options Dialog Box . . . . .	171
	Quantum Atom Selection Dialog Box . . . . .	173
	NMR Atom Selection Dialog Box . . . . .	175
	Equivalent Atom Selection Dialog Box . . . . .	176
	Labelled Atom Selection Dialog Box . . . . .	178
	Select NMR Atoms Dialog Box . . . . .	180
	SCF Wavefunction Options Dialog Box . . . . .	181
	Spectral Options Dialog Box . . . . .	182
	Reference Shielding Dialog Box. . . . .	184

Solvation Effects . . . . .	185
Files. . . . .	186
Program Files . . . . .	186
HyperNMR Default Settings . . . . .	187
[nmr]. . . . .	187
[ReferenceShieldings] . . . . .	192
Atom Type Files . . . . .	194
Atom Typing Rules . . . . .	194
Syntax . . . . .	195
forcefield. . . . .	195
<element-symbol>. . . . .	195
;<comment> . . . . .	196
<clause> . . . . .	196
<test>? . . . . .	196
“Connected to” Expressions . . . . .	197
Side Groups . . . . .	197
Rings . . . . .	197
<Assignment-statement>. . . . .	198
Semi-empirical Parameter Files . . . . .	198
Help Text . . . . .	199
Communication Files . . . . .	199
HyperNMR Input Files . . . . .	200
Z-Matrix Files . . . . .	200
NMR Files . . . . .	201
Molecular System: . . . . .	203
Syntax: . . . . .	203
Description: . . . . .	204
; comment . . . . .	204
forcefield <force-field-name> . . . . .	205
sys <temperature>. . . . .	205
view <view-distance> <view-scale> <slab-distance>	
<slab-thickness> <view-transform> <view-transform>	
205	
box <x> <y> <z> . . . . .	206
mol <mol#> <mol-name> . . . . .	206
res <res#> <res-name> <PDB#> <previous-res#>	
<next-res#> . . . . .	206
atom <at#> <atom-name> <element> <type> <flags>	
<at-charge><x><y><z><cn><nbor#nbor-bond>	
206	
vel <at#> <x> <y> <z> . . . . .	207
mass <at#> <mass> . . . . .	207
basisset <at#> <basis set> . . . . .	207

endres <res#> . . . . .	207
endmol <mol#> . . . . .	208
Selections: . . . . .	208
Options: . . . . .	208
NMR Parameter . . . . .	209
NMR Spectrum . . . . .	209
Log Files . . . . .	210

## **Chapter 10 Scientific Background 213**

Overview . . . . .	213
Principles of Nuclear Magnetic Resonance . . . . .	214
General Principles . . . . .	214
Magnetic Shielding . . . . .	216
Nuclear Spin Coupling . . . . .	222
Typed Neglect of Differential Overlap Method . . . . .	225
Conventional Semi-Empirical Methods . . . . .	226
NDO . . . . .	226
NDDO . . . . .	228
Typed Semi-Empirical Methods . . . . .	228
TNDO/1 . . . . .	229
TNDO/2 . . . . .	230
TNDO Perturbation Theory of Magnetic Shielding Constants . . . . .	233
TNDO Perturbation Theory of Nuclear Spin Coupling Constants . . . . .	236
Self-Consistent Perturbation Theory . . . . .	237
Parameterization . . . . .	240
Calculation of Chemical Shifts . . . . .	242
Nuclear Spin States and Transitions . . . . .	247
Spin Wave Functions . . . . .	247
Stationary-state Wave Functions . . . . .	248
Selection Rules and Intensities of Transitions . . . . .	249
Weak Coupling Approximation . . . . .	250
Parameter Sets . . . . .	251
Bibliography . . . . .	252

# HyperChem Modules<sup>®</sup>

## Part 1 HyperChem Data



# Chapter 1

## Introduction to HyperChem Data

This chapter contains an overview of the basic properties and functionality of HyperChem Data. It principally explains the reasons behind the existence of the product rather than details of how to use it.

### A Database Module

HyperChem Data is a module that has been integrated into HyperChem. It augments HyperChem. It communicates with HyperChem to provide input for HyperChem and to store results computed by HyperChem.

HyperChem Data is a Database Module in that it adds database functionality to HyperChem. It comes with a large database of molecules that can be transferred to HyperChem for modeling calculations. Output from HyperChem can be transferred to HyperChem Data and as such it provides a database for storing the results of molecular modeling calculations.

New databases can also be created with custom fields. Thus HyperChem Data provides a general purpose database capability for HyperChem.

### Databases in HyperChem Data

#### Databases

Databases used by HyperChem data are simply Access 97 databases in a form that can be directly read by Microsoft Access. They can be read either by Access 97 (in their native format) or by Access



2000 (converted as they are read into the newer format). If they are converted into the Access 2000 form, they cannot be read by HyperChem Data.

The reason for keeping them in Access 97 form rather than the newer Access 2000 form is that such databases use only one byte per text character (Ascii) rather than 2 bytes (Unicode). This keeps the HyperChem Databases at approximately one half the size they would be if converted. They can always be converted, if necessary, (by reading them into Access 2000) for use in other contexts.

## Tables

HyperChem Data uses individual tables within an Access database to hold molecular data. That is, all the information about a set of molecules is in a table. A \*.mdb database file can hold multiple tables.

## The Molecular Database

The molecular databases associated with HyperChem Data consist fundamentally of records describing isolated molecules and their properties. HyperChem Data will complain if individual records attempt to describe a molecular system that consists of multiple molecules. Single molecules with names and properties are the basic "stuff" of HyperChem data. While many properties can be associated with a molecule, each individual molecule is described by three fundamental things:

- Name of the molecule
- HDF string describing 2D structure
- HIN string describing 3D structure

## Naming Molecules

A Molecule in HyperChem Data has three potential names:

- Name - the string you use to describe that molecule.
- Alias - a second optional name for the same molecule.
- Root - a contraction of the name used as an index in the database

The "Root" is a string that is automatically generated from the name and provides an index for database lookup and ordering.

The "Name" is the name that you give to a molecule. This could, for example, be:

1,2-Dimethoxybenzene

or

Benzene, 1,2-dimethoxy

The root is generated from alphabetic characters prior to numeric characters. The second choice above is to be preferred because of the root it generates. The corresponding two roots are:

dimethoxybenzene1,2

and

benzenedimethoxy1,2

The second of these will cause this molecule to be indexed with the rest of the substituted benzenes rather than being stored with "dimethoxy" systems.

The "Alias" is an optional second name that in this instance is "Veratrole". The Alias is often a generic rather than a systematic name. It is an optional string that can be used to help search the database.

## HDF String

Molecules in a HyperChem Data database are described by "pictures" or 2D drawings of the relevant molecule. These pictures are encoded by a \*.hdf file containing an HDF string that encodes the picture. That is, the HDF string or file contains records (lines) that describe the basic primitives of the 2D drawing.

For example, the HDF string describing Ethane (a single line primitive drawn to describe the carbon-carbon bond) is:

```
"PRIMITIVE",1,100,#FALSE#,0,-2147483640,1,2,1995,1845,2395,1845
```

```
"ATOM",1,"C",1995,1845,1,2,1
```

```
"ATOM",2,"C",2395,1845,1,1,1
```

The first record describes the single-line primitive that was drawn. The last two records describe the two atoms inferred from the drawn line. A complete description of an HDF file or string is described in Appendix 1.

A 2D structural search of the database searches the HDF string of each database record for the 2D substructure in question.

## HIN String

A 2D HDF string describing a 2D drawing of a molecule can be converted by HyperChem to a full 3D description of the molecule. This full 3D description of molecules is contained in a \*.hin file or equivalently a HIN string.

HyperChem Data databases have a field for the HIN string but this string is often the null string implying the database contains only a 2D description of a molecule (the HDF string). However, it is possible to have HyperChem Data use HyperChem to create a HIN string and then imbed that string in the appropriate field of the database. Such a database contains an optional 3D description of a molecule as well as the required 2D description.

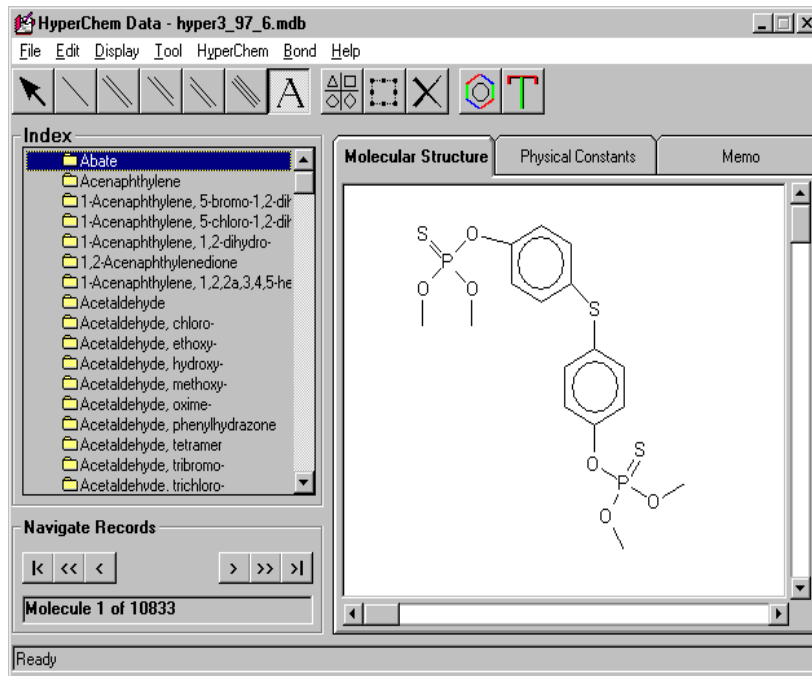
## Properties

The fields of a HyperChem Data table include a few standard required fields plus any custom fields. The standard fields include properties such as the name of the molecule, and others, such as the HDF string discussed above, plus other fields that are automatically generated such as the molecular formula and weight. If other properties of a molecule such as its dipole moment, spectral frequencies, etc. are desired to be included in a database, then these fields must be explicitly added to the definition of a table. The record values of these custom fields may have to be manually added to the database.

Certain properties can be related to HyperChem State Variables (HSV's). These properties can be include in a table without manual entry by having HyperChem compute the property with subsequent automatic entry into the database.

## Basic Window

The basic window defining the product is shown below.



The window consists of a number of areas:

- **Menu Bar** - the menu bar contains menu items, in standard form, for operating the program, setting parameters, etc.
- **Tool Bar** - the tool bar below the menu bar contains tools primarily associated with creating a drawing of a molecule.
- **Index** - this area shows the full name of each entry in the database. The entries are sequenced according to their Root name. Clicking on an entry makes it the current record and displays it below and to the right of the index area.
- **Navigate** - this area has buttons for navigating through the database and indicates the particular record that is the current one.
- **Status Line** - the status line at the bottom shows information relevant to the previous or current operation.
- **Molecular Structure** - this area shows the drawing associated with the current record.

- Physical Constants - this area show the values of the fields in the database, apart from the memo fields (HDF and HIN).
- Memo - this area essentially displays information associated with the two memo fields in a database, the HDF string and the HIN string. One can display a picture of the Windows metafile (picture) created from the HDF string or else shows the HDF string itself. A HIN string can also be displayed if the database record contains one. If not, the first approximation to a HIN file derived from the HDF file is shown. In either case it is the HIN string that would be sent to HyperChem when requested.

## Basic Operations

HyperChem Data has only a few basic operations that the user needs to understand. These can be used in isolation or can be combined.

### Drawing Molecules

One can use HyperChem Data just to draw a picture of a molecule. These pictures can, of course, be imbedded in a database but they need not be.

HyperChem has traditionally been used for molecular modeling and not as a drawing package. The drawing capability of HyperChem Data allows you to create a drawing in the form that is readily recognized as appropriate for publication. Later versions of HyperChem come closer to allowing it to be used as a simple drawing program but if you wish to just create a 2D drawing of a molecule, HyperChem Data is preferred.

A drawing or picture can be saved in one of three formats:

- HDF file - the HDF file is proprietary to Hypercube, Inc. It is a simple Ascii text file that describes a basic molecular drawing. It is not intended to be a universal drawing format but is primarily associated with HyperChem Data.
- Windows Metafile - an HDF string can be used to create a Windows metafile, i.e. a Windows drawing which is portable to different devices such as screens, printers, etc. Unlike a bitmap, a metafile creates lines on a printed page that have the resolution of the printer. A metafile is appropriate for incorporating

a chemical drawing into a manuscript. For example, Microsoft Word can import metafiles and can print chemical drawings at full resolution, unlike bitmaps.

- Windows Bitmap File - an HDF string can also be converted to a bitmap. The \*.bmp bitmap can be used anywhere that it is appropriate to use bitmaps such as on web pages. A bitmap, unlike a metafile, will have only the resolution of the original device, usually a computer screen.

A picture can also be converted to a HIN file by HyperChem Data rather than be directly transferred to HyperChem. Such HIN files do not yet contain any real 3D information since they have not yet been run through the model builder of HyperChem. That is, all the z coordinates are zero. Such a HIN file might be referred to as a 2D HIN file although it has exactly the same form as any other HIN file; the z coordinates are just all zero.

## Database Reading

HyperChem Data can read databases such as those provided with the program (Hyper3.mdb). Only databases specifically associated with HyperChem Data can be read. Such databases, however, are just specific examples of Microsoft Access databases and can be used with Microsoft Access and other database programs, when required.

When a database is read, one needs to open a specific table to obtain access to molecular data. The table in Hyper3.mdb that holds the data is referred to as the "Default" table.

When a database is read, the Index area is filled with all the names of the molecules and a picture of the molecule associated with the current record is displayed. The Physical Constants area shows all the fields of the database apart from the HDF and HIN strings of the memo fields.

## Database Creation

### Basic Entity is Table

HyperChem Data allows you to create a whole new Database with any number of tables or just to create a new table of the existing

database. Again, a table is the basic entity for holding molecular information.

## Standard Fields

When you create a new table, you are expected to define the fields of that table. A standard database has only the standard default fields. These include the HDF string or drawing that represents the molecule.

## Custom Fields

You can define any new fields that you like, including fields that are related to variables in HyperChem that can be filled in automatically from HyperChem calculations.

## Adding Records

After a table is created, records can be added. This generally requires selecting "Add Record" from the Edit menu, drawing the molecule, giving it a name, and specifying its properties associated with the standard or custom fields of the table. The record must then be "Updated". Certain fields such as Formula and Molecular Weight are computed automatically by HyperChem Data and need not be added manually.

The principal information associated with a record in a table of HyperChem data is the drawing of the molecule. Other information may be calculated automatically but the drawing is the fundamental information associated with each record. The drawing plus a name defines the molecule.

## Database Editing

Any record in a table can be edited (and then updated). This edit operation might include modifying the structure (via the drawing of the molecule), modifying the name, the properties, etc.

## Database Querying

An important operation on any database is querying. In particular, HyperChem Data comes with a database of over 10,000 molecules and it is important to be able to search that database and

extract a subset (the query) that reflects a set of molecules of particular interest (for molecular calculation). A common reason for performing a query is that you wish to have HyperChem perform calculations on this described subset. Other times you may just want to explore the data in the database or look for specific values. HyperChem Data includes quite generic database searching capabilities, particularly when combined with HyperChem.

## Simple SQL Queries

The normal default fields of a HyperChem Database table are:

- Name of the molecule
- Alias name
- Molecular Formula
- Molecular Weight
- Number of Hydrogen atoms
- Number of Carbon atoms
- Number of Nitrogen atoms
- Number of Oxygen atoms

An SQL query can be created using any of these fields. A simple graphical interface allows easy generation of SQL queries or they can be explicitly defined by text written by the user.

## 2D Substructure Searches

Substructure searches require searching through the HDF string associated with a record. Such searches can be combined with first performing an SQL search or can be performed independent of any SQL search. These 2D substructure searches look for atoms that are connected to specific other types of atoms. A simple graphical interface allows easy generation of these queries.

These searches essentially are looking for specific connections in the connection table describing the molecular graph. One looks for rows of the connection table having specific properties. For example, by combining these, one might search for organic acids that are aromatic and have a Chlorine on the aromatic ring.

HyperChem Data cannot look directly for a completely arbitrary substructure buried within a molecular system. It operates by



looking for specific types of atoms with specific types of bonds. Nevertheless, in combination or repetition, one should be able to narrow a search down so as to be able to find the types of molecules that one wishes.

### 3D Substructure Searches

In HyperChem Data it is possible to query in three dimensions as well as in 2D. That is, one might want to find all molecules where a Nitrogen is within 3 Angstroms of two other Nitrogen atoms. Such queries are important in drug design and are possible using the generic query capability described next.

### Generic Searches

HyperChem Data has the ability to execute a HyperChem script over a set (table) of molecules. The "Run Script and Delete Records" menu command allows completely generic queries to be made.

For all entries of a table a script is run. A new table initially holding all the original entries is created. If the script returns a specified variable as "True" then that entry is kept. Otherwise that entry is deleted from the new table. The final table (query) contains only those entries that have passed the script successfully.

In this way it is possible to query all molecules that have a PM3 dipole moment of more than 2 Debyes, all molecules that have an atom with an atomic charge greater than 0.4, etc. Completely generic queries can be issued that depend on arbitrary molecular structure and energetics.

## HyperChem-Related Operations

Since HyperChem Data works in close conjunction with HyperChem, it is possible to perform sophisticated operations with single records or sets of records. The related HyperChem operations are:

- Get HIN file - get a structure from HyperChem and put it in the HIN string of current database record.
- Put HIN file - put structure from current database record into HyperChem. This structure may come either from the HIN string if one is present or from a converted HDF string if no

HIN string is present in the database record. If a structure in HyperChem comes from an HDF string, it is usually assumed that it will be subsequently modelbuilt in HyperChem to give a 3D structure starting point.

- **Run Script** - run a HyperChem script over the current set of records of a current table or query. This essentially performs batch molecular modeling calculations on a table or query from HyperChem Data.
- **Run Script and Retrieve HIN** - Like the previous operation this runs a script over a set of molecules from HyperChem Data but subsequently imports the resulting HIN string back into each record. This would allow one, for example, to create a new table that had the optimized PM3 structures of over 10,000 molecules by just pushing one button and waiting the appropriate period of time.
- **Run Script and Delete Records** - this generic operation essentially implements a powerful query capability as described above.



# Chapter 2

## Tutorials

### Overview

This chapter includes a number of tutorials showing how to use HyperChem Data. It is recommended that a new user go through these tutorials to learn how to use HyperChem Data effectively.

### Summary of HyperChem Data

HyperChem Data is a database program for storing and managing chemical information. In particular, it can be used for searching and querying tables of chemical structures and chemical properties stored in a database. It uses databases of the standard Microsoft Access form.

The data in the database can be inputted or queried quite independently of HyperChem. Alternatively, a close relationship with HyperChem can be established such that HyperChem can have its molecular systems and calculated properties placed into the database in a completely automated fashion.

HyperChem Data includes three main functions:

- It provides a set of drawing tools that can be used to create any molecular system in the HyperChem Data workspace and save it in the Windows Metafile format.
- HyperChem Data can be used to store molecular systems with their properties. Users can search a database to find a set of molecules that satisfy given search conditions and save the results of the search conveniently.
- HyperChem Data can be used to invoke HyperChem and to ask HyperChem to compute molecular properties for a given

set of molecules (possibly a query) stored within HyperChem Data.

HyperChem Data can be used as a stand-alone program or in conjunction with HyperChem. If you do not have a copy of HyperChem installed on your computer, you must use HyperChem Data in stand-alone mode. In stand-alone mode, HyperChem Data can be used for the first two main functions described above. If you have a copy of HyperChem installed on your computer, you may find it convenient to automate the computing of molecular properties for a set of molecules from HyperChem Data.

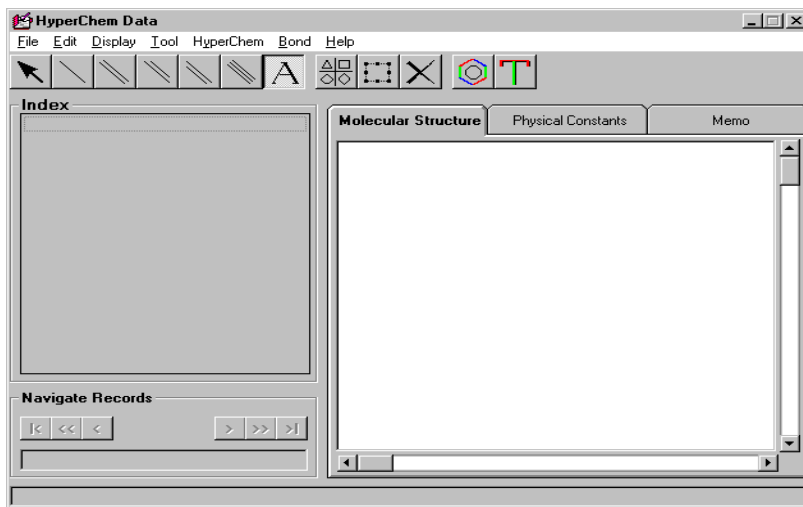
In addition to its database aspects, HyperChem Data is also a 2D drawing program and standard drawings of molecules can be created for inclusion in a scientific paper (via the clipboard or \*.wmf files in the Windows metafile format). The 2D drawing can also be passed (put) to HyperChem as a HIN input file and HyperChem can use its model builder to convert the 2D drawing into a 3D structure. It may be convenient for some users of HyperChem to draw their input structures in HyperChem Data rather than in HyperChem itself.

## Tutorials

These tutorials contain information on starting HyperChem Data, opening and querying databases, editing records, adding fields, creating new databases and tables, drawing molecular structures, and using HyperChem with HyperChem Data.

### 1. Starting HyperChem Data

From the Windows Taskbar Start button, choose Programs, then HyperChem Data. This brings up the main HyperChem Data window.



When there is no database open, not all of the functions described below are available.

The main HyperChem Data window consists of a menu bar, a tool bar for drawing and editing sketches of molecules, and the areas described below:

<b>Index</b>	Shows the indices of the molecular systems in the current table. All molecular systems in the current table are indexed by their root name, in alphanumeric order.
<b>Navigate Records</b>	Use to advance or to go back by one (>) or ten (>>) records, or to go to the last (> ) or first ( <) record.
<b>Molecular Structure</b>	Shows the current molecular structure, or functions as workspace for drawing new structures.
<b>Physical Constants</b>	Displays current values for properties stored as fields; values may be edited in this window.
<b>Memo</b>	Displays information related to the memo fields (HDF string and HIN string) of current record.

## 2. Opening a Database

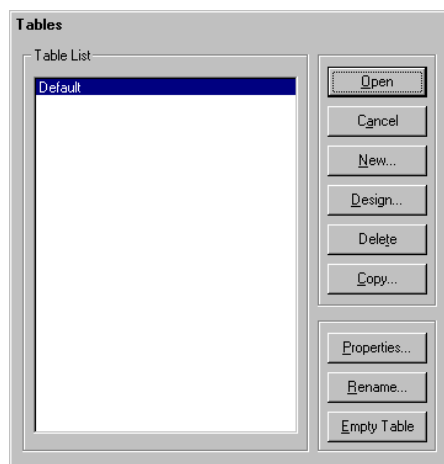
A HyperChem Data database with an extension of MDB consists of a set of tables. Each table can have as many fields as you like. Examples of fields are: molecular structure, molecular formula,

melting point, dipole moment, HOMO energy, etc. You can define fields with different field types and sizes to optimize storage and search efficiency.

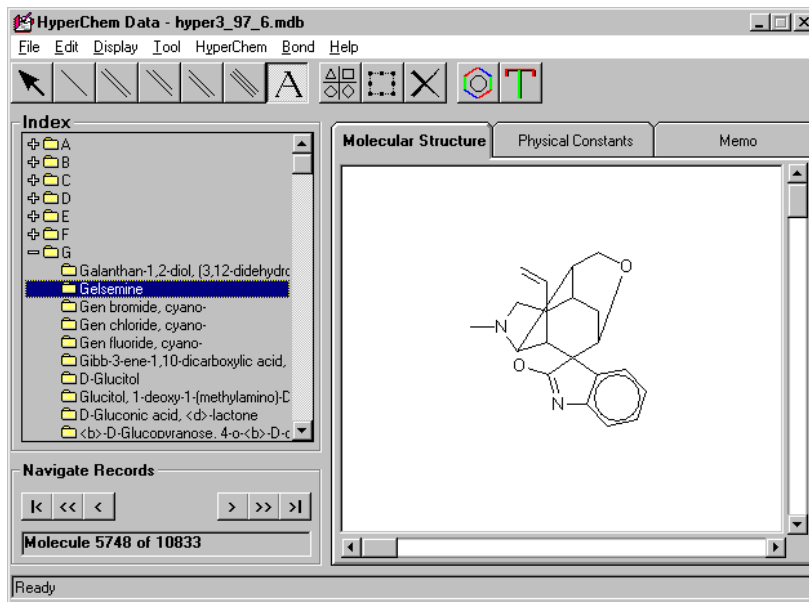
HyperChem Data comes with a sample database of more than 10,000 molecules representing common organic compounds. This database is a *read-only* Access file, **D:\hcddata\hyper3.mdb** on your CD-ROM, assuming that D is the letter assigned to your CD-ROM drive.

You may read the data in hyper3.mdb directly from your CD-ROM into HyperChem Data. Alternatively, you may copy the database file to your hard disk, setting its attributes not to be read-only, if you want to modify and use the database from your hard disk.

1. Assuming D: is your CD-ROM drive, copy D:\hcddata\hyper3.mdb to a suitable location on your computer's hard drive. The recommended location is C:\Hyper6\Data\hyper3.mdb.
2. To change the attributes of hyper3.mdb, right-click on it and choose Properties. At the bottom of the Properties dialog box, click in the box next to Read Only so that it is un-checked. Then click on OK to close the Properties dialog box.
3. To open the database that comes with HyperChem Data, go to the File menu and choose Open Database. Go to the folder into which you copied the database file, click on hyper3.MDB, then choose Open. To open any other MDB database, follow the same procedure, substituting the path and name of that database.
4. In order to use the database, you must open a table. A table consists of a set of records, each made up of a molecular structure, molecule name, and the fields that have been defined for that table.



5. In the Tables dialog box, choose Default under Table List; then click on Open to access the Default Table.



Notice that some of the folders in the Index window are marked with a “+”; these are folders that contain records. Those marked



with a “-” are folders that are already open. Unmarked folders are empty. Records are stored in *alphanumeric* order in the Index folders.

- Click on a name in the Index box to see the corresponding molecule displayed in the Molecular Structure window. Scroll up and down the Index window. Click on the “+” next to one of the marked folders to view its contents; the Molecule\_Name field for each record in the folder is displayed in the Index box. Notice that the Record Number is displayed below the Navigate Records buttons. The Record Number is based upon the index order of records in the database.
- To advance to the next record, use the button in the Navigate Window. Alternatively, to choose the next molecule from the index window, simply click on it.

At any time you can view the properties or notes stored with a structure by choosing the Physical Constants or Memo Windows. When you choose the Molecular Structure Window, an image of the molecule is displayed. Click on Physical Constants. This shows the current value for each field (property) defined in the current table.

The screenshot shows the HyperChem Data software interface. The title bar reads "HyperChem Data - hyper3\_97\_6.mdb". The menu bar includes File, Edit, Display, Tool, HyperChem, Bond, and Help. The toolbar contains various icons for navigation and editing.

The **Index** window on the left shows a tree view of folders. The "Gelsemine" folder is selected and expanded, showing a list of sub-folders including "Galanthan-1,2-diol, (3,12-didehydr...", "Gen bromide, cyano-", "Gen chloride, cyano-", "Gen fluoride, cyano-", "Gibb-3-ene-1,10-dicarboxylic acid", "D-Glucitol", "Glucitol, 1-deoxy-1-(methylamino)-D", "D-Gluconic acid, <d>-lactone", and "<b>-D-Glucopyranose, 4-a-<b>-D-c".

The **Navigate Records** window below the Index shows navigation buttons (K, <<, <, >, >>, I) and the text "Molecule 5748 of 10833".

The **Physical Constants** window on the right displays the following fields and values:

Fields:	Value:
Formula:	C <sub>20</sub> H <sub>22</sub> O <sub>2</sub> N <sub>2</sub>
Molecule_Name:	Gelsemine
Alias:	Gelsemine
Mol_Weight:	322.405975
CAS_Number:	
Melting_Point:	
Boiling_Point:	
Refraction_Index:	

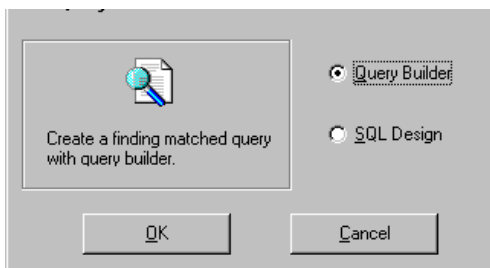
The status bar at the bottom of the window shows "Ready".

### 3. Querying a Database (SQL GUI Query)

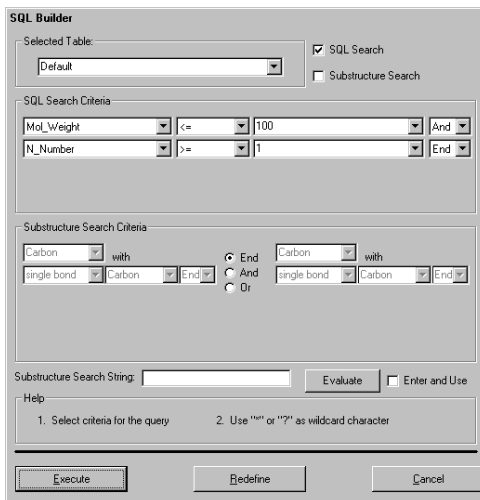
A number of different ways can be used to make a query. The first way described here is via a simple SQL query using the graphical user interface to generate the SQL query.

#### Setting up a query:

1. Go to the Edit menu and choose Query Database.



2. In the New Query dialog box, choose Query Builder and then choose OK. The SQL builder comes up.



3. Enter the name of the table to be searched in the “Selected Table” Window.
4. Check only SQL Search at the top right. Other tutorials will discuss Substructure Search.

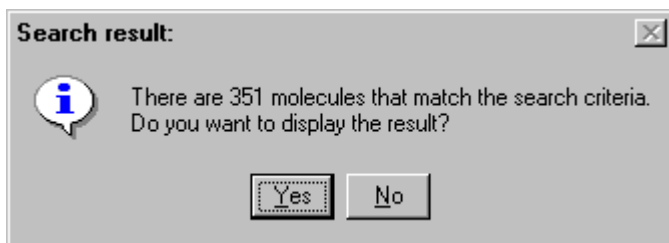
5. Choose the desired SQL Search Criteria, using one or more field names as descriptors as shown above.

For example, to search the database for all entries with Molecular Weight under 100 and one or more Nitrogens, choose Mol\_Weight, then  $\leq$ , then type in 100. In the last box, choose And, then choose N\_Number, then  $\geq$ , then type in 1. Use an End to complete the definition of search criteria.

**Executing the search:**

6. To search the current database and table and display a new table representing the query, choose Execute.

The Search Result window reports the number of molecules that match the search criteria and gives you the option of displaying the results.



If you choose to display the results, they appear in the HyperChem Data (main) window, where you can view and navigate using the usual tools.

7. Click Yes in the above dialog box to view the results of the query.

**Saving a Query:** If you decide that you would like to save these results, you can save them as a new table in the current database or as a table in a completely new database. To save a query, choose Save Query As under the File menu and choose appropriate names for the table and the new database, if one is created.

If you decide not to save these results, choose Close Query under the File menu.

## 4. Querying a Database (Substructure Query)

A second way to make a query is to search for 2D substructure. In these searches one is looking for a specific topology in the 2D molecular graph describing a molecule. Ideally, one would just draw a portion of a molecular graph and the program would search all molecules in the database looking for that subgraph in the graph of the molecule. This is somewhat too ambitious.

The 2D description of molecules within molecular modeling programs is usually described via a "connection table" that describes each atom and the connections it has to other atoms. The HDF string as well as the HIN string basically contain connection tables as part of their semantics.

Substructure search in HyperChem Data basically looks for rows of these connection tables that satisfy user constraints. Thus one can look for a specific kind of atom that has specific kinds of connections to other atoms. For example, one could look for a Carbon that has an aromatic bond to another Carbon, plus a single bond to a Nitrogen. This is essentially just requires matching to a row of a connection table. In addition, one can combine two of these searches via an AND/OR clause.

If necessary such a query can be saved and subsequent queries made on the first query.

Substructure queries can be combined with an SQL query performed first (as described in the previous tutorial) by choosing to ask for both SQL and Substructure queries in the appropriate check box. SQL queries are more efficient than substructure queries because they do not need to search a memo field such as the HDF string. It is efficient to narrow your search with an SQL search prior to a substructure search, if this is possible.

### Setting up a query:

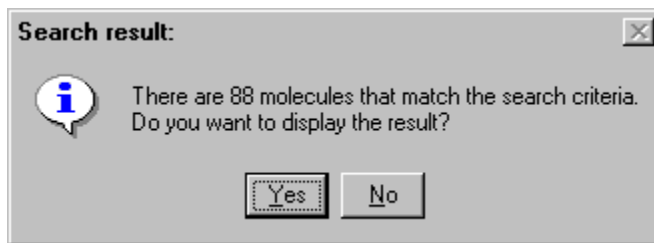
1. Go to the Edit menu and choose Query Database as in the last tutorial.
2. In the New Query dialog box, choose Query Builder and then choose OK. The Query builder comes up.

3. As shown above, we search for an acid (second part) having a Nitrogen attached to an aromatic structure. That is, we can combine two searches (with AND or OR) or just do one search (END). The first search above essentially looks for an aromatic structure (ring?) by searching for a C-C aromatic bond with a Nitrogen attached by a single bond to one of the Carbons. The second part of the search looks for a carboxyl group (Carbon attached to two Oxygens via single and double bonds).

#### Executing the search:

4. To search the current database just click on Execute.

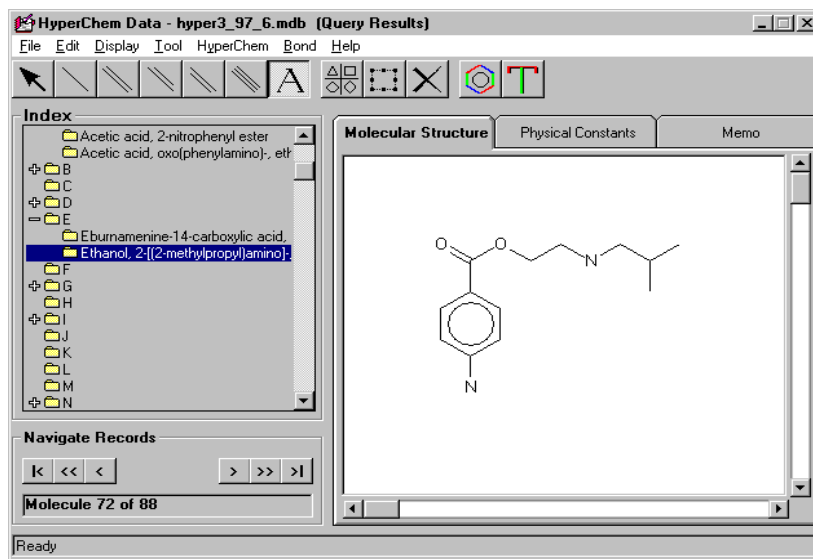
The Search Result window reports the number of molecules that match the search criteria and gives you the option of displaying the results.



Depending on the size of the database and the type of substructure query, this search may take some time. The search must look at every HDF string and parse that string to look for the appropriate information on connections. Such a search is considerably more time consuming than an SQL search which needs only to match field values. If at all possible, you should try to combine the substructure search with an SQL search that will decrease the number of HDF strings that need to be parsed.

To perform substructure queries, a new temporary table called "Query" is created that is a copy of the original table. Entries are deleted from the copy as the query proceeds. This same name is used for every substructure query and this temporary table must be deleted prior to a subsequent substructure query. This generally happens automatically but there may be situations in which this "Query" table becomes visible.

5. Say "Yes" to display the results of the query. The query can be closed, saved, or implicitly closed by requesting a new query.



**Saving a Query:** If you decide that you would like to save these results, you can save them as a new table in the current database or as a table in a completely new database. To save a query, choose Save Query As under the File menu and choose appropriate names for the table and the new database, if one is created.

If you decide not to save these results, choose Close Query under the File menu.

#### **Optionally Entering the Query String Directly:**

As illustrated above, a substructure query can be generated via an easy to use GUI (graphical user interface). The GUI in reality generates a substructure search string that the user need not be concerned with.

However, it is possible to enter a specific substructure search string and bypass the GUI if necessary. This might be necessary if, for example, one is looking for specific bonding to a Germanium atom which is not one of the atom choices in the GUI. To enter your own substructure search string:

1. Click on the "Evaluate" button to see the current substructure search string. You will be modifying it.
2. Click to say yes to the checkbox, "Enter and Use".

3. Enter a substructure search string with no blank characters. For example, the string, "c-ge1-se2" would ask for a search of records where a Carbon is connected to a Germanium atom by a single bond and also connected to a Selenium atom by a double bond.
4. Click on Execute to initiate the search as you would if the substructure string had been created by the GUI.

## 5. Querying a Database (Generic Query)

The most generalized way of querying a database in HyperChem Data is referred to as Generic Querying.

### Overview

By a Generic Query we mean that for every record in the database, one runs a generic HyperChem script on the molecule from the database. This script can compute anything associated with the molecule. The script must finally make a decision, in code, on whether the molecule satisfies the query criterion.

The script that defines a generic query must ultimately include a Tcl script (\*.tcl). This is because the script must have conditionals so as to decide whether the query is satisfied or not.

### How It Works

HyperChem Data when asked to "Run Script and Delete Records" creates a new Hyperchem State Variable (HSV) in HyperChem. This HSV is called "hc-fail". This variable is set by HyperChem Data to have an initial value 1 or "true" before running the requisite script on every record of the database. The script must set this value to 0 or "false" if the query criterion is satisfied by the script and the record is to be kept (not deleted) as a successful record in the query. If the script does nothing about "hc-fail" or sets it to 1 or "true" then the query fails on that record and the record is deleted from the table.

Since the query script can calculate a host of molecular modeling properties (anything HyperChem has to offer), the query is a very generalized one. For example, the script could optimize the structure using a particular method or basis set and then perform a quantum mechanical calculation of the ionization potential. With code to check the result, the query could return only those



molecules that have an ionization potential less than, say, 5 Electron Volts.

## Example

This example queries a database for all molecules that have a quantum mechanically calculated dipole moment greater than 1.0 Debyes.

### Setting up the script:

1. Use notepad or your favorite editor to create a new script called `bigdipole.scr`. This script will have only one line in it reading `read-tcl-script bigdipole.tcl`

2. Use notepad or your favorite editor to create a new script called `bigdipole.tcl`. This script should look as follows:

```
TclOnly
hcExec "query-response-has-tag false"
hcExec "menu-build-model-build"
hcExec "do-single-point"
set dipole [hcQuery dipole-moment]
if {$dipole > 1.0} then {
hcExec "hc-fail = 0" } else {
hcExec "hc-fail = 1" }
Exit
```

This script assumes one has set the defaults in HyperChem to reflect the type of calculation, e.g., PM3, that one is going to do when the Single Point calculation is requested via "do-single-point". The script would be more robust if commands like

```
calculation-method semi-empirical
semi-empirical-method pm3
```

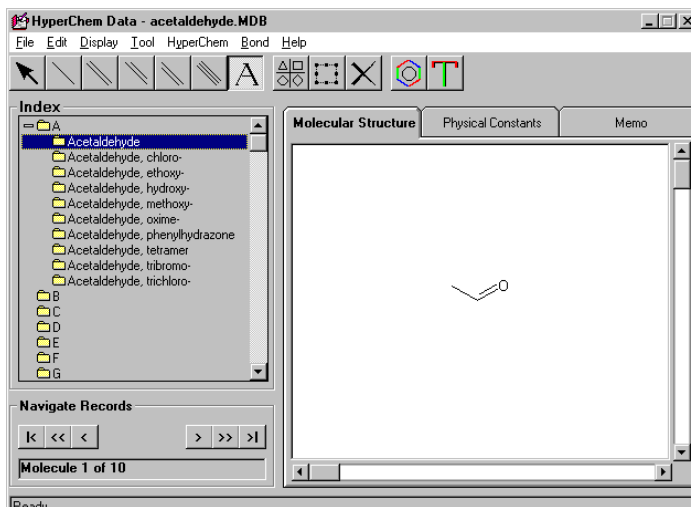
were added, depending on the type of calculation required.

The main Tcl script first ensures that the value that will be returned for dipole moment has only the value without a tag. It then builds a 3D structure from the 2D drawing sent to it by HyperChem Data, performs a single point calculation and then queries the value of the calculated dipole moment. If the value is larger than 1.0 Debyes, then the query will succeed.

## Opening a Database and Table

For this tutorial, we recommend using a very small test table, created in any convenient way, such as from an earlier query.

3. Open a small database table to explore Generic Querying.



### Requesting the Query:

4. Click on the menu item <HyperChem/Run Script and Delete Records...> and select bigdipole.scr. The script will be run and a query will be created as per other simpler query operations. Depending on the script complexity and the size of the database that is used, this query could take significant time (even days) to complete. We recommend exploring query scripts on only small test databases until you are familiar with the general procedure.

### Saving the Query:

A Generic Query can be saved or closed like any other simple SQL query.

## 6. Adding New Fields to a Table

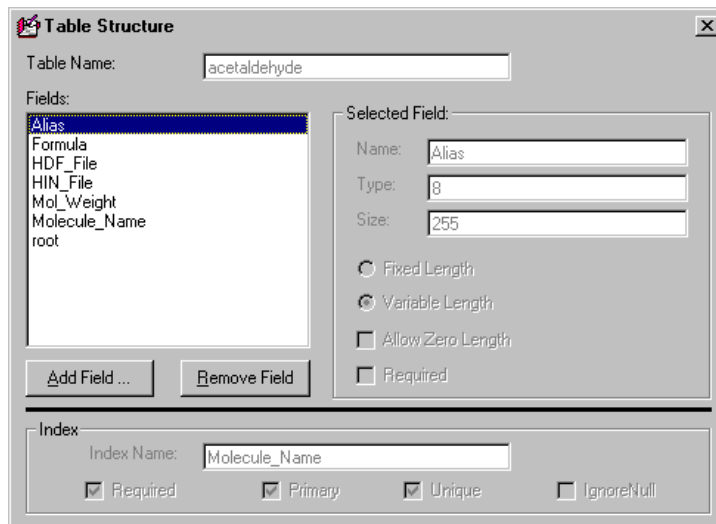
Once you're familiar with exploring and querying a database, you may wish to know how to add an additional property (field) to the table. This procedure is described below.

## Adding New Fields to a Table:

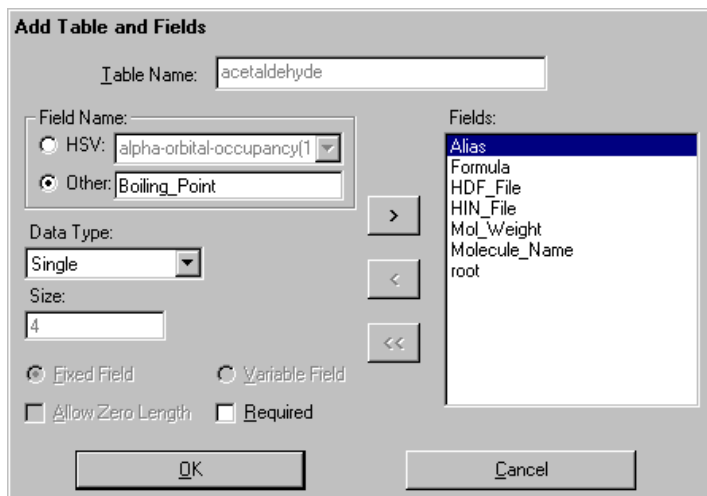
1. Select a table to open from the Table List.



2. Click on Design.



3. Click on the Add Field button.



4. Enter the Field Name. Choose the desired data type, and specify fixed or variable field length. You may also specify whether to allow zero length, and designate the field as required, if desired.

**Example.** You may want to enter the boiling points of the molecules in your table. To do this, enter “Boiling\_Point” in the Field Name box. Choose “Single” as the Data type. Accept the default Size, and then click on the right arrow button to add this field name to the displayed set of fields.

**Removing a Field:** In the Table Structure window, choose Remove Field. Note that the default fields (Alias, Formula, HDF\_File, HIN\_File, Mol\_Weight, Molecule\_Name and root) cannot be removed.

## HSV Fields

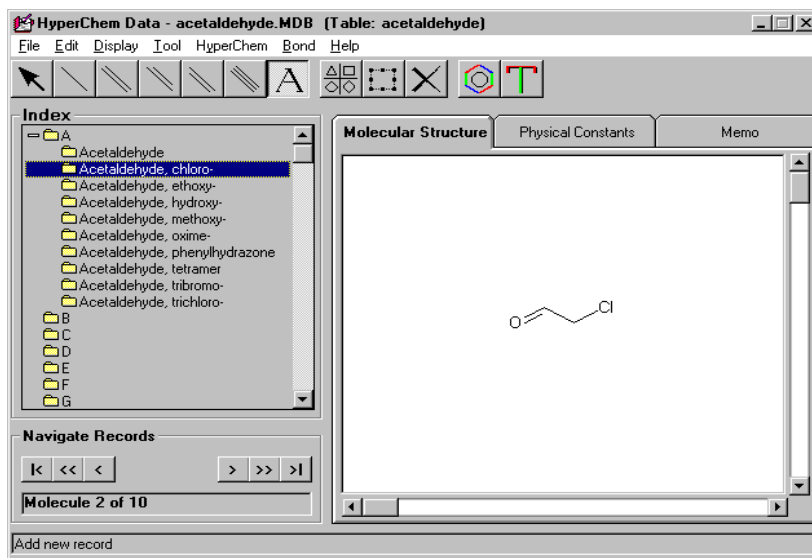
In addition to custom fields such as Boiling Point described above, it is possible to add fields that are tied to HyperChem State Variables (HSV's). An HSV is a variable in HyperChem that may contain an energy, an ultraviolet frequency, an atom charge, etc. There are hundreds of these HSV in HyperChem describing variables of the molecular modeling program.

If a field in a database is tied to an HSV, then it is possible via running a script to have these fields of the database filled in automatically by HyperChem in cooperation with HyperChem Data. These fields have pre-specified names that begin with "h\_" followed by the name of the HSV. A later tutorial describes how to use a script to fill in a database with properties computed by HyperChem.

## 7. Adding and Removing Records From a Table

### Adding Records

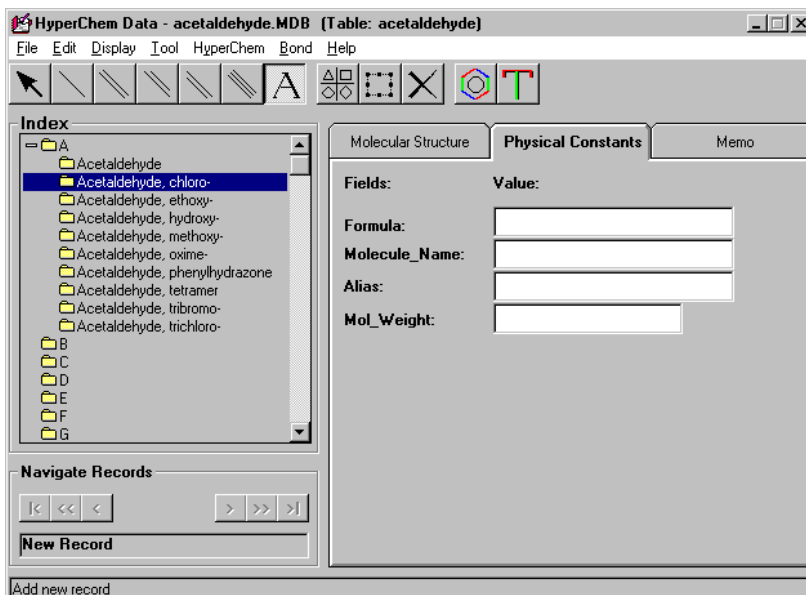
A fundamental operation with HyperChem Data is the creation of a new molecule record. We assume you have an open table such as the following:



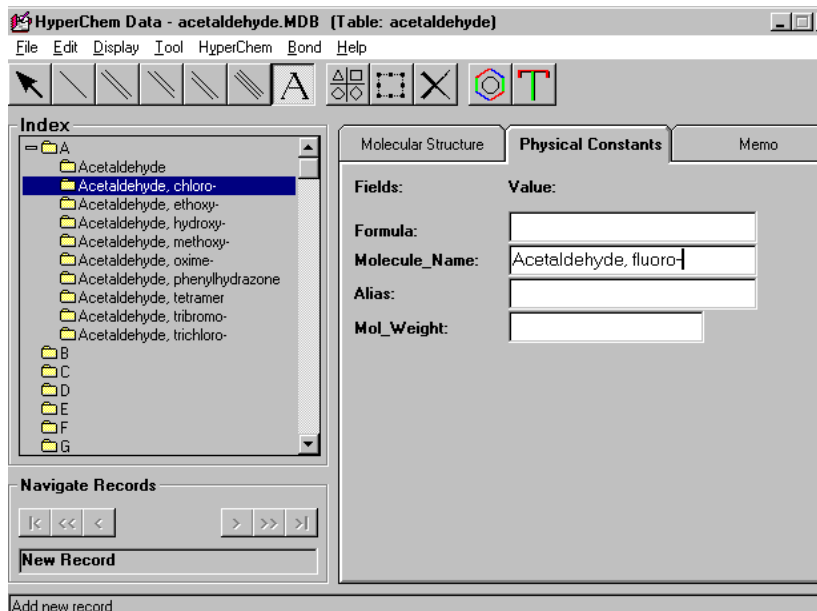
The addition of a new molecule occurs as follows:

1. Under Edit choose Add Record.

The Molecular Structure area will be cleared and the Physical Constants section will be blank.

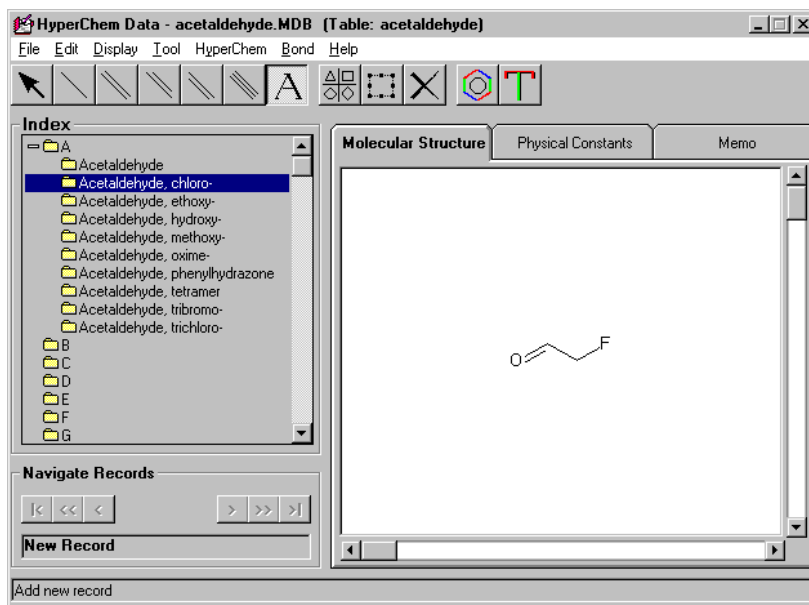


- Fill in the Molecule Name and Alias, if it has one. The other two fields (Formula and Molecular Weight will get filled in automatically).



- Switch to the Molecular Structure Panel and draw the new molecule.

If you are not facile at drawing yet, go through the tutorial on Drawing Molecules.



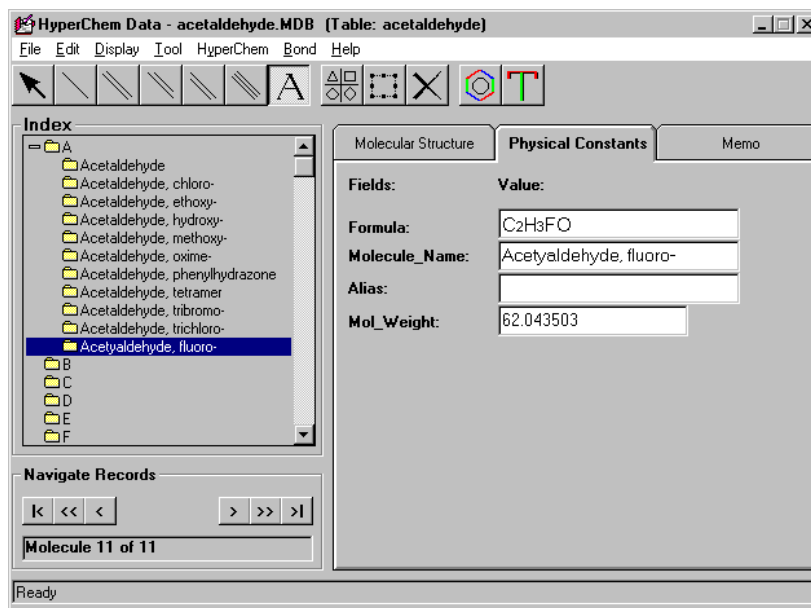
- Choose Edit/Update Record to permanently add the new record.

You will be asked if you want to save this new record. If you do not want to save this new record but want to start over or have changed your mind about this new record, you can say No to the Update and use Edit/Discard Changes to cancel the Edit/Add Record selection you made earlier.

- Click on Yes to save the new record.

If you have not given a name to the molecule, you will also be asked to do so at this time since a name is mandatory.

The new record's Physical Constants Panel looks as follows:



The memo panel will show the metafile and hin file (preliminary 2D form) for the new molecule.

## Deleting A Record

To delete a record from a table is straight forward.

1. Click on Edit/Delete Record to delete the current record.

## 8. Editing an Existing Record

Editing an existing record is similar to adding a new record. You select Edit, modify the appropriate information or drawing and then select Update.

1. To enter Edit mode, choose Edit Record from the Edit menu.

*Note:* Note: If you make changes to any field (other than Molecular Structure; see below) without first entering Edit mode, you must still enter Edit mode before saving changes.

2. Make the desired changes to the molecule structure or fields.
3. Choose Edit/Update Record to save changes.



## 9. Drawing a Molecule

### Background

A molecular drawing can be included as part of a database record or drawn completely independently of a database record as an isolated drawing for inclusion, for example, in a manuscript.

Molecular drawings in HyperChem Data are described via an HDF string or HDF file (\*.hdf) that contains the multi-line string). The HyperChem Drawing File (HDF) is an Ascii file (delineated in Appendix 1) that describes what you have drawn so that the drawing can be reproduced, edited, saved, etc.

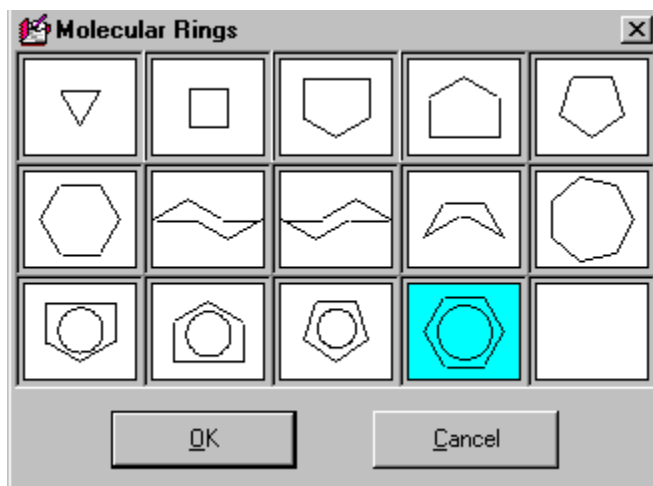
An HDF file can be converted into two other types of files. The first is a Windows Metafile (\*.wmf). This is the Windows format for a line drawing. It is an appropriate format for including, for example, a molecular drawing into a Microsoft Word document. HyperChem Data can create a Metafile from an HDF file but not vice versa.

Secondly, HDF files can be converted into HIN files, the standard file for input to HyperChem. Since HyperChem HIN files primarily describe 3D molecular structure not drawings, this may be less relevant with regard to simple drawing than the creation of Metafiles.

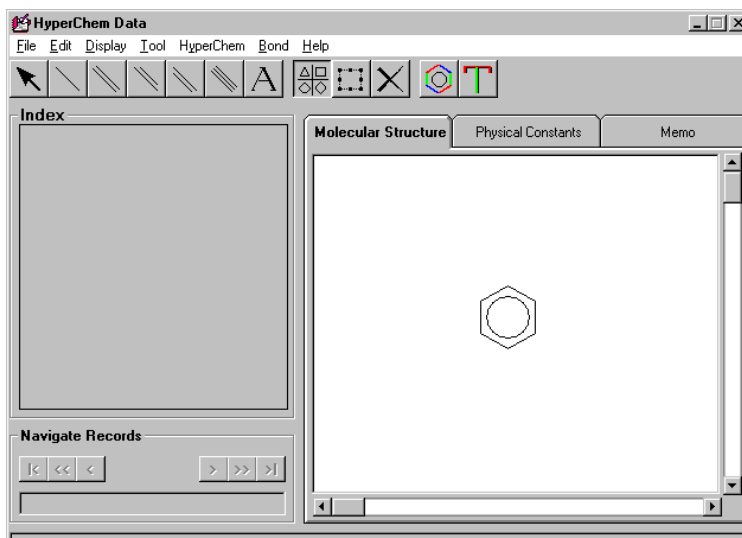
[The latest Releases of HyperChem have new drawing capabilities relevant to HIN files.]

### Drawing a Substituted Naphthalene:

1. Close any open tables or databases in HyperChem Data.
2. Click on Ring Tools of the toolbar to bring up the dialog box for choosing a ring structure.

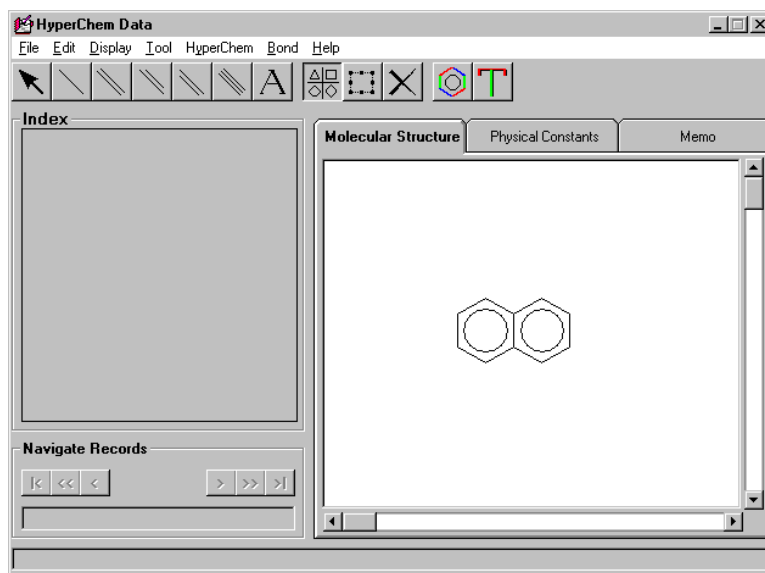


3. Click on the Benzene ring and then on OK. Your mouse cursor should now resemble the toolbar button.
4. Click the left mouse button in the drawing area and hold the button down. Drag the mouse around to get a feeling for the way one can rotate the benzene ring on the screen. When you have an orientation such as that shown below, release the left mouse button.



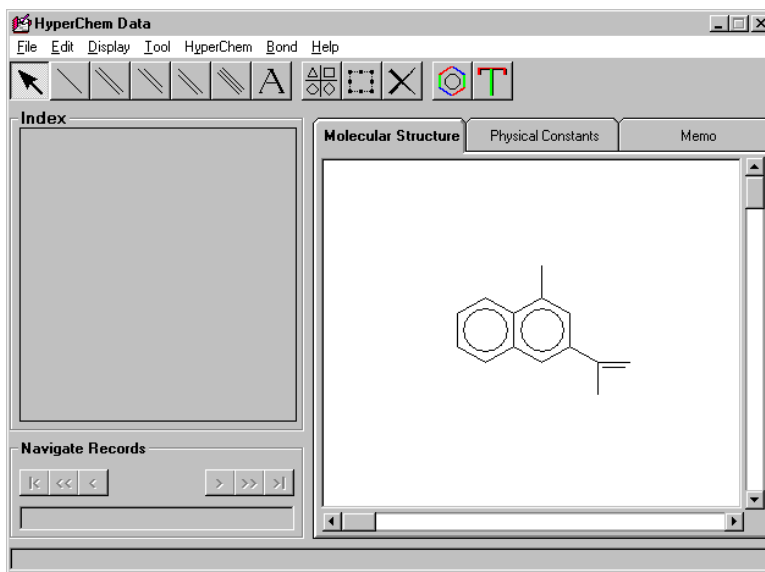
If you have trouble, you can start over in a number of ways:

- (a) Use Display/Clear Screen to clear the drawing area.
  - (b) Use the selection cursor (left-most button on the toolbar) to drag a selection rectangle around the portion to be deleted to select it. Alternatively, select individual components of the drawing. Then hit the delete button or Edit/Delete.
  - (c) Use the Select all toolbar button (9th from the left) or the Edit/Select all menu item to select the whole drawing. Then hit the delete button or Edit/Delete.
5. Using the same Ring cursor, click on the North East carbon atom of the benzene ring with the left mouse button and hold the button down. Drag the mouse around to join the second ring to the first until your drawing looks as follows:



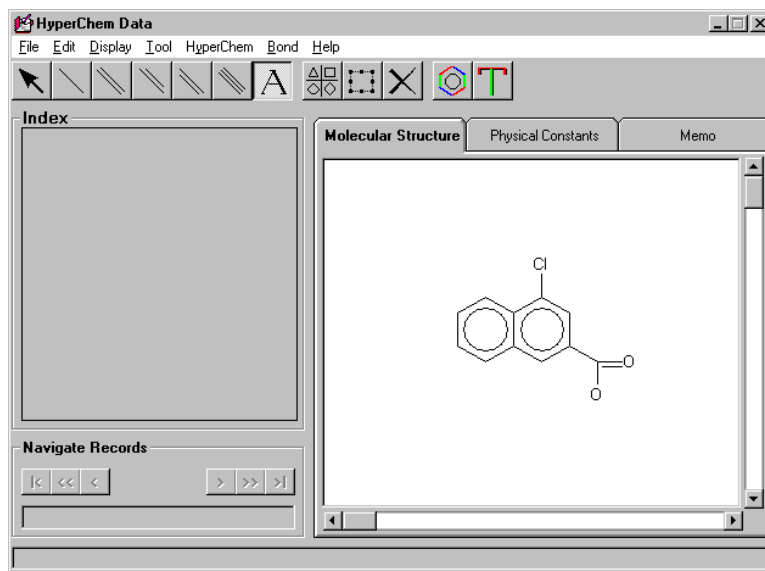
You have now created Naphthalene. We will now add two substituents.

6. Use the single bond cursor (second toolbar button from the left) and one of the double bond cursors to draw something equivalent to that shown below. All three of the double bonds are chemically equivalent and which one you use depends only upon how you prefer your drawing to look.



At this point all atoms are Carbon Atoms. The drawing consists of a set of vertices, each of which, by default, is a Carbon atom. However, we can modify any vertex to change the Carbon atom into any other Element.

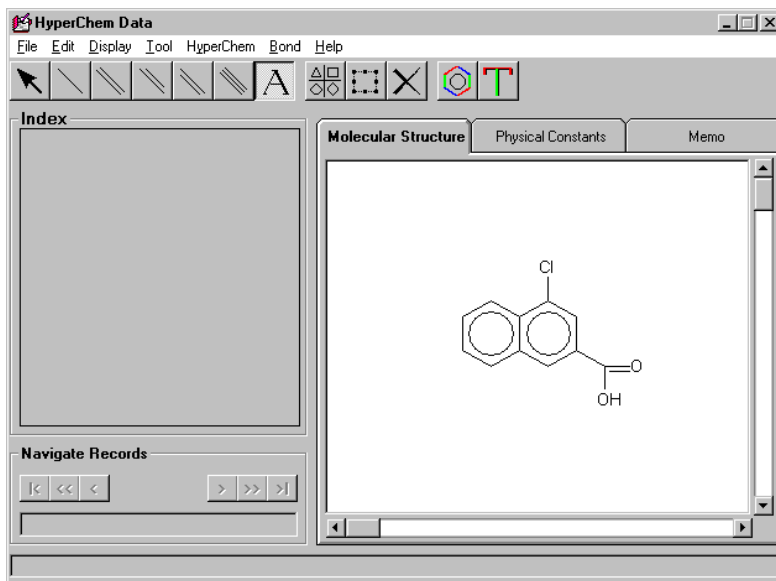
7. Click on the Ascii text button of the toolbar (7th from the left) to select the text cursor. Use the cross-hairs of the cursor to click on the atom (end of the line) at the top of the drawing (top end of the single bond you have just drawn). That is, click on a vertex to change that default Carbon atom at that vertex into a new Element .
8. After you click, you will see a flashing cursor indicating you can type characters. Type the Characters "Cl" followed by hitting "Enter". This will change the default Carbon Atom into a Chlorine atom.
9. Do the same with the other two open-ended bonds but type an "O" (rather than "Cl") followed by hitting "enter" each time. Your drawing should now look as follows:



Your drawing is now complete and you have created a meta-substituted Naphthalene. The appropriate Hydrogens will be added by HyperChem as necessary.

In drawing the carboxylic acid group we have not drawn the "H" attached to one of the "O"s. Whether we do so or not is not important to HyperChem. We could have typed the text "OH" rather than "O" if we had wanted the drawing to show an -OH group explicitly. HyperChem Data will recognize any string as the label to be attached to a vertex. However, when passed to HyperChem to create a 3D molecular structure, only certain strings are recognized -- all the element symbols plus certain others such as "OH", "NH", "CH", "NH2", "CH2", and "CH3".

Thus the above drawing and the following drawing are different drawings..



However, when passed to HyperChem to create a 3D structure, they are identical.

Use the next tutorial (item 2. particularly) to save your drawing.

## 10. Saving and Retrieving a Drawing

You can edit an existing drawing of a molecule, or make a new one in the Molecular Structure area and save the results without first entering Edit mode or even having a database open.

1. Open HyperChem Data and draw the desired structure; alternatively, edit an existing molecule drawing.
2. To save the drawing, choose File/Save Picture As...and enter a name for the resulting HDF (HyperChem Draw File).
3. To retrieve the structure, choose Import Picture from the File menu.

## 11. Creating a New Database

### Format of Databases

HyperChem Data uses databases (\*.mdb) that are Microsoft Access databases consistent with Access 97. They are kept in Access 97 format because of their much smaller size than corresponding databases consistent with newer formats such as that used by Access 2000. Access 2000, however, will convert these databases into a form directly usable within Access 2000, or equivalent Database programs. In doing so, however, the databases will approximately double in size because of the conversion from Ascii data (Access 97) to Unicode data (Access 2000).

HyperChem Data comes with a database of over 10,000 molecules. It also allows you to create molecular databases of your own choosing.

### New Database Fields

New HyperChem Data databases have a set of default fields that cannot be removed. On the other hand, you can create as many new fields as you like. The default fields are what make the database a molecular database for HyperChem Data. The default fields are:

- **Molecular\_Name** - a name for a molecule. It should be unique.
- **root** - the index field automatically generated from Molecular\_Name
- **Alias** - an alternative name for a molecule. The field is required but can be left blank.
- **Formula** - automatically generated from the drawing of a molecule.
- **Mol\_Weight** - automatically generated. It includes the implicit Hydrogen atoms.
- **HDF string** - an Ascii string that describes a required 2D drawing of a molecule. The string is generated from the drawing and the drawing can be regenerated from the string.
- **HIN string** - an Ascii string that potentially describes the 3D structure of a molecule. A 2D version (all z coordinates set to zero) can be generated from the HDF string. The full 3D ver-

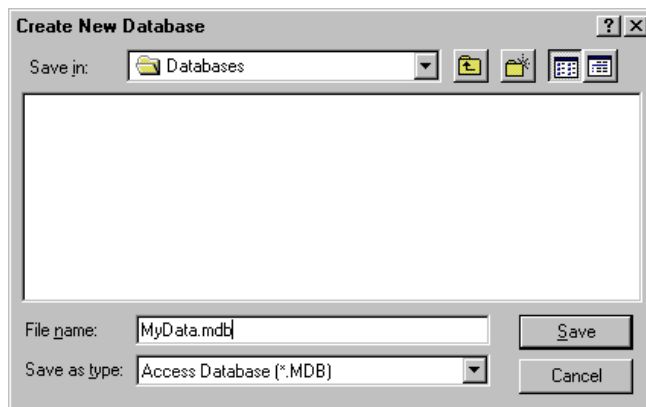
sion (retrieved from HyperChem) allows the database to hold full 3D structures. By default the HIN string is a NULL string.

A new database can have in addition to the above fields any fields that Access databases allow. These new fields fall into two types - those that are just normal Access fields such as an Integer, Single or Double floating point values, Text strings, etc. The second new types are fields that are tied to HyperChem State Variables (HSV's). These are fields that can be filled in automatically by HyperChem from data created in molecular modeling calculations.

## Example

The example below creates a whole new database rather than just a new table for an existing database. The two are basically the same procedure, however.

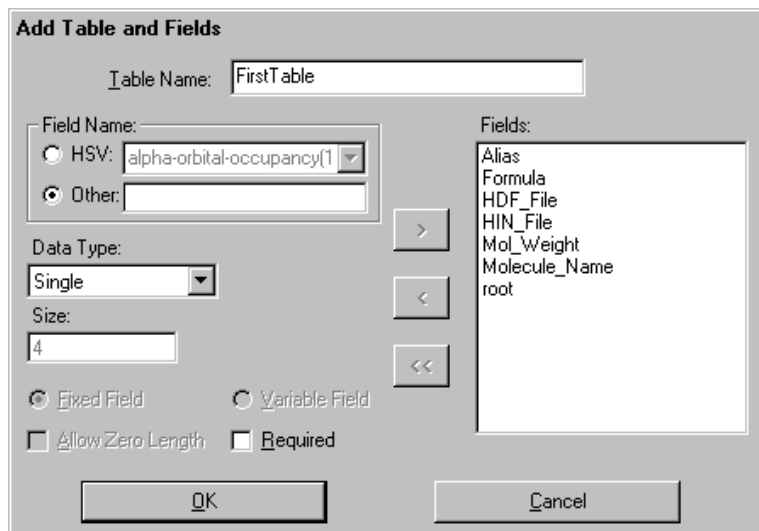
1. Under File, choose New Database. Enter the name for the new database in the dialog box. Click on Save to close the dialog box.



2. The Tables dialog box now appears. Click on New...

One now has the option to design the fields of the new database as shown below.



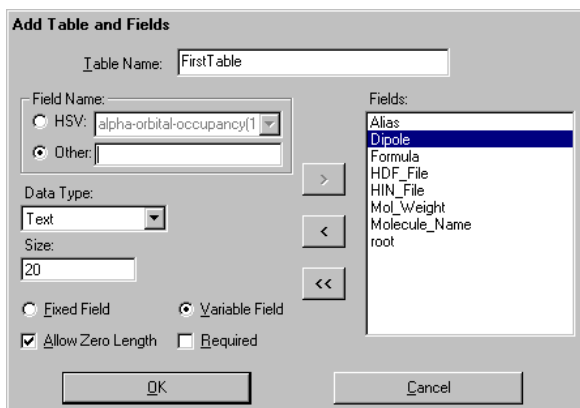


- Fill in a name for the new table in the new database. As shown above, we have call ours, "FirstTable" in the database MyData.mdb.

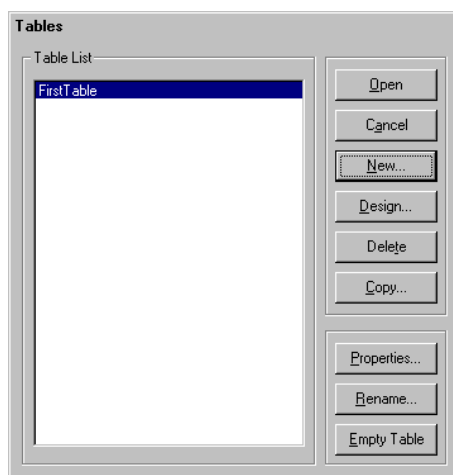
For this example, lets assume you are creating a database of experimental dipole moments for eventual comparison with calculated values.

You can choose new fields that are tied to a HyperChem State Variable (HSV). These fields will be filled in automatically when you run a HyperChem script on a table or query. Alternatively, you can choose new database fields having no specific relationship to HyperChem, using the Other option.

- Create a new field called Dipole by entering the string "Dipole" in the Other Field Name position, clicking on the ">" button to assign the field as shown below.

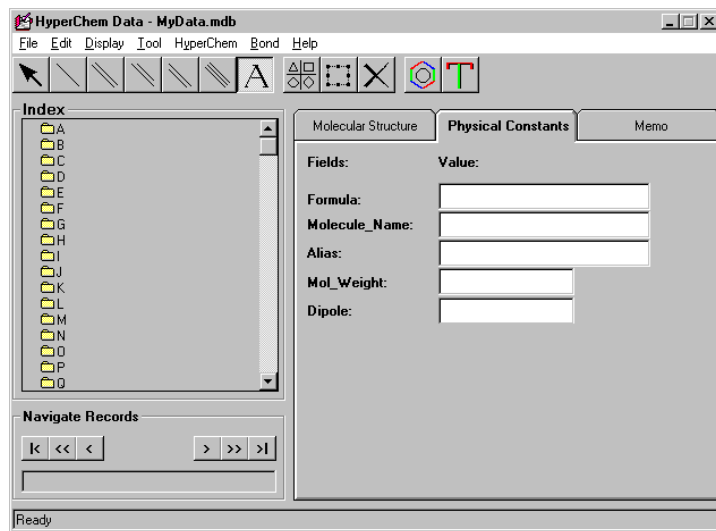


5. Click on OK to create the table and bring up the Tables dialog box,



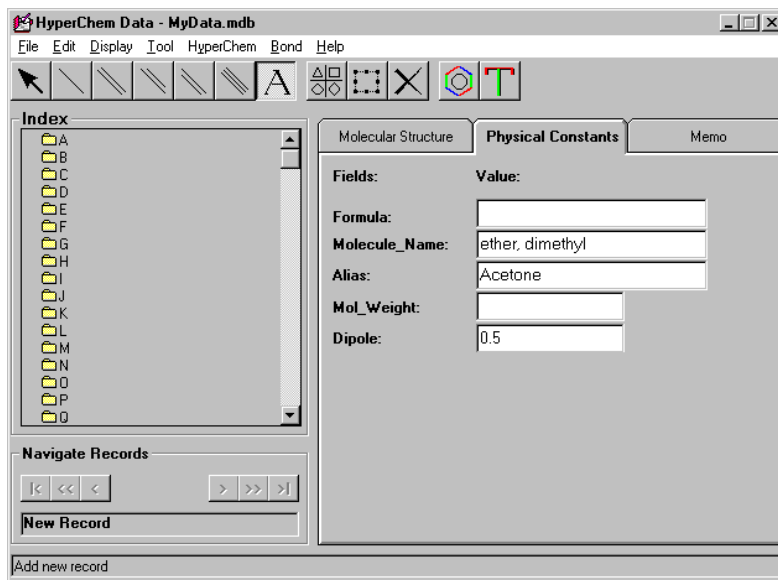
6. Click on Open to open the new table. At this point the table will be empty.

The new table has zero records. The Molecular Structure tab is empty and the Physical Constants tab looks as follows.

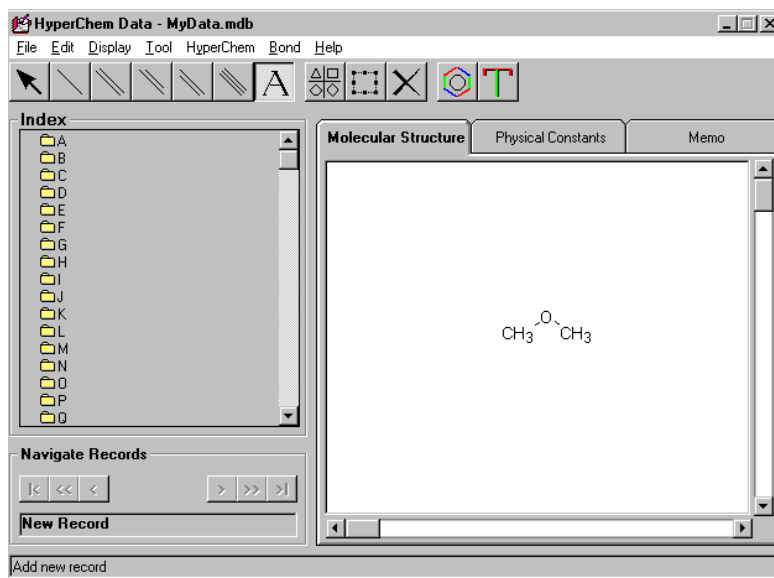


The new Dipole field of the database table is shown above. To add records to this new table one must use Edit/Add Record, fill in the data, draw the molecule, etc., and then use Edit/Update Record to complete the addition of the new record. This is repeated until you have added as many records as desired. You should then close the table and/or the database. Here we will illustrate adding a single record. The record will be for acetone and assume the dipole moment is 0.5 Debyes.

7. Click on Edit/Add Record and then fill in the Physical constants for Acetone as shown below:



8. In the Molecular Structure tab create a drawing of acetone as shown below:



This drawing has added the text label "CH3" at two of the vertices. These labels are not necessary as far as HyperChem is concerned but they may add to the clarity of the drawing, as desired.

9. Click on Edit/Update Record to complete adding this new record to your new table and new database.
10. Click on File/Close Table to close the table. Click on File/Close Database to close the database.

## 12. Using HyperChem with HyperChem Data

HyperChem Data communicates with HyperChem and the combination can be used for many calculations not feasible with either product in isolation.

Scripts written to automate calculations in HyperChem can be executed from HyperChem Data. The result is that the specified operations are carried out on all the molecules in the currently active table. This provides a very convenient method for adding molecular information to the database.

In addition, of course, the generality of Hcl and Tcl scripts allow one to perform many useful molecular modeling calculations but now perform them with a set or database of molecules. The script can write files, save results, etc. This batch computation on a database table of molecules can automate many research computations.

When a script is a Tcl script, its generality allows for essentially any calculation you might like. Any calculation could be performed on a molecule from the database, including calculations performed by the script that have nothing to do with HyperChem *per se*.

When a script is a simple Hcl script, the molecular modeling results can appear in the normal log file or be automatically placed in the database. With a simple Hcl script described below fewer options are available than with Tcl scripts.

## Computing Molecular Properties with HyperChem Hcl Scripts

### Run Script...

HyperChem Hcl scripts, which are described in detail in the HyperChem Chemist's Developer Kit manual, can be used in conjunction with HyperChem Data to perform calculations on molecules in a HyperChem Data table. The results can be stored in the table, or they can be sent to an output text (log) file.

### Output to Log file:

1. Use a text editor such as Windows Notepad to create a text document consisting of Hcl script commands. An example of a useful script follows:

```
errors-are-omsgs
warnings-are-omsgs
omsgs-to-file errors.txt
start-logging log.txt, yes
do-single-point
stop-logging
```

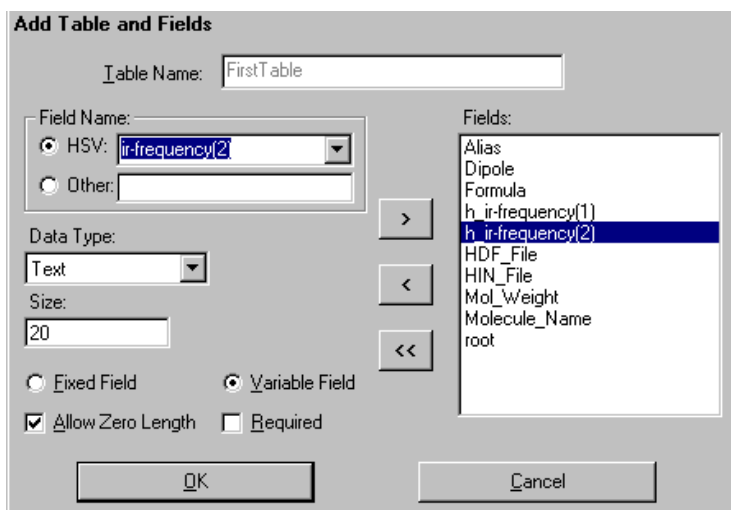
This script sets up the file errors.txt to receive possible warnings and error messages, opens a HyperChem log file named log.txt, instructs HyperChem to perform a single point calculation on the current molecule using the current setup parameters, sends the output to log.txt and then ends the log session.

2. Save the script, preferably in the Scripts sub-folder of the HyperChem folder, for example, C:\Hyper6\Scripts.
3. Open a HyperChem window and set the desired method and options for the single point calculation.
4. In HyperChem Data, open the database and the table of molecules whose properties are to be computed. To execute the script and log the results to log.txt, go to the HyperChem menu in HyperChem Data and choose Run Script. Then go to the HyperChem Scripts folder and select the desired script.

The results of the calculation will be sent to the file log.txt. This file will hold a concatenation of the results from Single Point calculations on each of the molecules of the database.

## Output to HyperChem Data Table:

1. Choose "File/Open Database" to open a database file.
2. Choose a table from the list provided, and press the "Design..." button.
3. Press the "Add Field..." button.
4. In the panel named: Field Name: check the radio button next to HSV:.
5. Choose a variable from this list, whose value you want to get from HyperChem and put into the database for each molecule.
6. If you have chosen a variable which is a vector, for example: ir-frequency(1), you can change the value of the index by typing in a new one ("1" is always the default). So, you can add the frequency of the second IR frequency for each molecule by editing "ir-frequency(1)" to "ir-frequency(2)".



7. Press the ">" button to add this variable to the list of fields for this table.
8. Repeat this process for each variable you wish to add. For example, one could add the following variables to save the first six lowest-frequency IR lines of each molecule:

```
ir-frequency(1)
ir-frequency(2)
```

```

ir-frequency(3)
ir-frequency(4)
ir-frequency(5)
ir-frequency(6)

```

to the table by clicking on the `ir-frequency(1)` six times, and changing the (1) to (2), (3), (4), etc. **before** clicking on the ">" button.

9. Press the OK button.
10. Dismiss the Table Structure dialog by clicking on the "x" (dismiss dialog) button at the very top right corner of the dialog box.
11. Click on the Open button to open the table
12. Use the Edit/Query Database command and follow the procedure for Querying a Database (*above*), if desired, to pare down the database to those molecules whose properties you wish to compute. Use the File/Save Query As... menu button to save the query result to a new table.
13. Create a HyperChem script file containing the script commands necessary to calculate the values of the variables chosen above. So, for example, if you have chosen the variables `ir-frequency(1)` and `ir-frequency(2)`, the following script commands will suffice:

```

select-none
menu-build-model-build
calculation-method semi-empirical
semi-empirical-method PM3
uhf no
configuration-interaction NoCI
do-geometry-optimization
do-vibrational-analysis

```

To create this script file, enter the commands listed above using a text editor and save the file as an SCR file. For example, you could save the commands above in a file named *ir-freq.scr* in the Hyper6/Scripts folder. Close the file before proceeding to the next step.

14. Start HyperChem if you do not have an active HyperChem window.



15. In HyperChem Data, choose Run Script from the HyperChem menu, and open the file you just saved (for example, C:\Hyper6\Scripts\ir\_freq.scr). This will cause HyperChem to run this script file on each of the molecules in the current table. The value of the selected HSV variables (for example, ir-frequency(1), etc.) will be entered into the database for each molecule after the execution of the last line of the script command.

### **Run Script and Retrieve HIN...:**

HyperChem Data includes a menu item, HyperChem/Run Script and Retrieve HIN... , that behaves very much like the Run Script... menu item just described above.

In both cases these menu items result in a HyperChem script being run on a table or query from HyperChem Data. In the above case, fields of the database corresponding to HSV's get filled in automatically as the script is run on a set of molecules.

In this second instance, the HIN field of the database (containing the 3D structure of a molecule) also gets filled in at the end of the script, in addition to the HSV fields. This allows you to capture a molecular structure into the database after an optimization, a dynamics run, a transition state search, etc.

With just Run Script..., the molecular structure in the database is left unchanged after running a script on the structure. In the second Run Script and Retrieve HIN... case, the database is modified to contain the molecular structure remaining at the end of the script run for each. Of course, if the script results in no change to a molecular structure, the two menu items give identical results.

## **13. Importing HyperChem Data Records into MS Excel**

Since HyperChem Data databases are just Microsoft Access databases, they can be viewed by a variety of database relevant software packages. A database, for example, can be read directly into Microsoft Excel. Importing data from HyperChem Data into Microsoft Excel is especially useful for investigating trends in molecular properties.

In particular, however, this tutorial shows you how to query a database using Microsoft Excel and Microsoft Excel Query. To

query in this fashion , you must obviously have both Microsoft Excel and Microsoft Excel Query installed.

## **Importing Data into MS Query and Excel**

Once MS Query has been installed, data can be transferred from a HyperChem Data table by the following procedure:

1. Start up Microsoft Excel.

The database and table to be transferred must be closed in order to transfer information to Excel.

2. If there is an open database in HyperChem Data, choose File/Close Database.

In MS Excel:

3. Use File/New to create a new workbook.
4. Choose the menu command Data/Get External Data/New Database Query...
5. In the Choose Data Source dialog, choose: MS Access Database. Click on OK.
6. The Select Database dialog appears next. Use it to find and select the database file that you created with HyperChem Data.
7. At this point, the Microsoft Query Wizard program will be visible. The upper left pane should display the name of the tables in your database. Select the table and the columns for your query and place them in the upper right pane. Click on a series of Nexts, then Finish, and OK to see your query.
8. The current sheet in Excel should now contain the data from your HyperChem Data database, ready for plots, averages, or other manipulations.

## **14. Saving, Compacting and Repairing Databases**

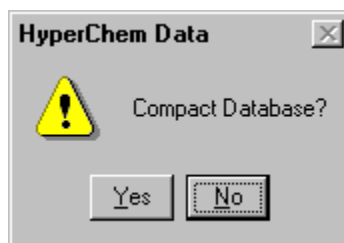
### **Creeping Database Size**

The size of a database can sometimes grow into an unmanageable size if many, many queries are performed on it. In many database operations data is replicated and in many delete operations data is just flagged as deleted rather than truly deleted. It is necessary to be able to compact a database to reduce its size after you have used

the database extensively. All database operation would still work but unless you compact a database regularly (depending somewhat upon your use of the database) it will take up much more memory than it needs to.

## Closing a Database

Every time you close a database in HyperChem Data you are given the option of compacting it.



Often you should just say no but if you think your operations have sufficiently expanded it, then you can compact it at that point.

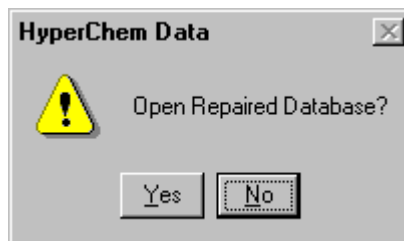
## Compacting a Database

You also have the option to compact at any time using the menu item File/Compact Database...

## Repairing a Database

If you believe your database may have been corrupted, you have a probability that Microsoft software can repair it. To attempt this just click on the menu item File/Repair Database.... and select the database to repair.

If the database is successfully repaired, you will be asked if you wish to open it.

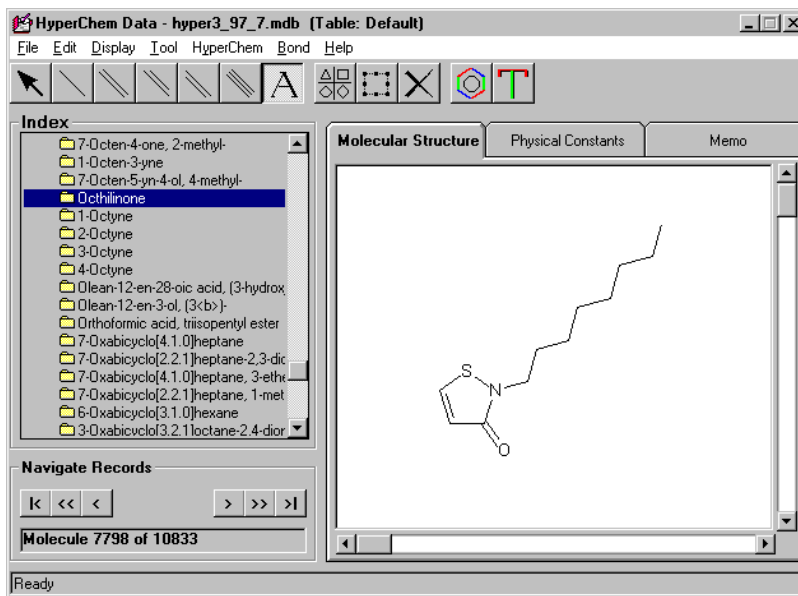


## 15. Getting a Molecule from the Database into HyperChem

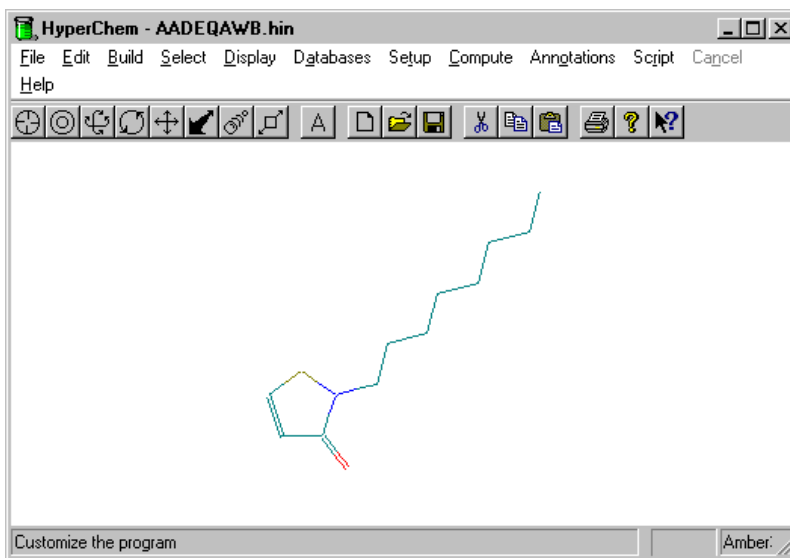
One of the simplest operations one might want to perform is to just search the database for a molecule that you want to use in HyperChem for molecular modeling calculations.

To do this:

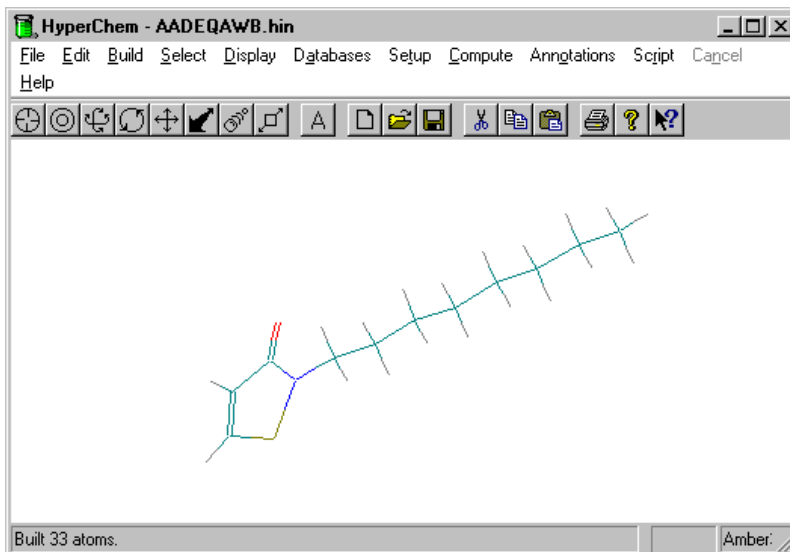
1. Query or search until the molecule you want is the current record.



- Click on the menu item HyperChem/Put HinFile to transfer the 2D structure to HyperChem.



- In HyperChem use the menu item Build/Add H & Model Build to create a 3D structure from the 2D drawing.



You now have a 3D structure with all the relevant Hydrogen atoms added. The model builder gives an excellent first approximation to the 3D structure but you might still wish to optimize the 3D structure with your favorite computational approach.



# Chapter 3

## Reference

### Overview

This chapter includes a number of reference materials on the HyperChem Product.

### Menu Items

This section describes each of the menu items in HyperChem Data.

#### File Menu

##### **New Database...**

Click on this menu item to create a new database (and ultimately a new table for the database).

##### **Open Database**

Click here for normal file-like opening of a database.

##### **Close Database**

Click here to close the current database.

##### **Tables**

Bring up a dialog box for choosing a database table along with an operation on the table.

##### **Close Table**

Close an open database table.



## **Compact Database**

Run a compact operation on the current database or a database of your choice. Compacting a large database may take some time.

## **Repair Database**

Attempt repair on a corrupted database.

## **Close Query**

Close a current query and return to original table.

## **Save Query As**

Save a current query as a table of the current database or as a table in a new database to be specified.

## **Import Picture**

Read in an HDF file as a replacement for the Molecular Structure of current record (possibly a NULL record if there is no current database table).

## **Save Picture As**

Write out (HDF, BMP, or HIN) file representing the current Molecular Structure.

## **Get Picture Set**

Read in a complete set of concatenated HDF files as the Molecular Structures for a set of molecules representing a database table.

## **Edit Menu**

### **Copy**

Copy a bitmap of the current Molecular Structure to the Clipboard

### **Delete**

Delete any selected portions of the current Molecular Structure. Equivalent to the Del key.

### **Align Label**

Align the text label describing a vertex of a Molecular structure drawing.

### **Select All**

Select all portions of the Molecular Structure Drawing.

### **Select None**

Deselect all portions of the Molecular Structure Drawing.

### **Add Record**

Place in mode where a new record can be added to a database table.

### **Edit Record**

Place in mode for editing the current record of a database table.

### **Discard Changes**

Discard any editing performed on the current record of a database table.

### **Update Record**

Make permanent any changes or edits to the current record of a database table.

### **Delete Record**

Delete the current record of the current database table.

### **Query Database**

Bring up dialog boxes for specifying the parameters of a database query. Perform the Query.

## Display Menu

### Refresh Screen

Refreshes the screen. Only needed in unknown rare circumstances.

### Clear Screen

Erases all elements of the drawing in the current Molecular Structure Panel.

### Center Picture

Centers the molecule picture in the Molecular Structure Panel.

## Tool Menu

These menu items are rigorously equivalent to clicking on one of the toolbar buttons.

A right mouse button click with a particular new type of bond changes the bond under the cursor to the new type of bond.

### Select

Used to select a portion of a picture in the Molecular Structure panel. Clicking selects the bonds under the cursor. Dragging creates a rubber banded area for selection.

### Single Bond

Used to draw single bonds.

### Double Bond

Used to draw equal length double bonds.

### Right-Side Double Bond

Used to draw double bond with shorter segment on right side.

## **Left-Side Double Bond**

Used to draw double bond with shorter segment on left side.

## **Triple bond**

Used to draw triple bonds.

## **Text**

Used to add a label at a vertex of a molecular drawing. Click on Vertex, type label, and hit Enter. Numbers are automatically subscripts.

## **Rings**

Click on to bring up a dialog box containing rings to be selected as drawing components. Rings may be aromatic or single bonded. Double bonds, etc. are created by modifying single bonds (right button click).

The rings in HyperChem Data involve the only mechanism for creating aromatic bonds.

Rings may be merged correctly by just overlapping the appropriate bonds.

## **Text Colors**

Chooses a color for subsequent Text Labels at vertices. Default color is black.

## **Draw Colors**

Chooses a color for subsequent bonds. Default color is black.

## **HyperChem Menu**

These menu items are associated with the interaction between HyperChem and HyperChem Data.

## Get Hinfile

Looks at the molecule in the workspace of HyperChem and places a corresponding HIN file describing its 3D structure into the HIN string of the current record of the HyperChem Data database table.

## Put Hinfile

If the current record of the database table of HyperChem Data has a valid HIN string in it, passes that string to HyperChem as a 3D HIN file to be read with the associated molecule being placed in the workspace of HyperChem.

If the HIN string of the current record is NULL, then converts the HDF string to a HIN string and passes the molecule to HyperChem. HyperChem will thus have a molecule with all z coordinates of zero and an appropriate next operation would be to apply the model builder or to optimize the structure.

## Delete Hinfile

This menu item gives you the option of deleting the HIN string from the current record of the current database table or from all the records of the current database table.

The HIN string potentially gets filled in by HyperChem as a result of the menu item Run Script and Retrieve HIN... At times one would perhaps like to go back to a HIN string that is NULL (but can still be generated from an HDF string). This menu item allows clearing out HIN strings from the database.

## Run Script

Runs a HyperChem script over all the molecules of the current database table of HyperChem Data. This script can perform any calculations on the molecule from HyperChem Data that the script specifies.

If the database table has fields that start with "h\_", i.e. fields which correspond to HyperChem State Variables (HSV's), then those fields will be filled in with the current value of the HSV at the end of the script.

A script must be a Hcl \*.scr script but that script might include a "read-tcl-script" call so that the script is completely generic.

## **Run Script and Retrieve HIN**

This menu item behaves identically to "Run Script" except that at the end of the script the current molecule in the workspace of HyperChem is placed in the HIN string of the current record of the database table. In this way, for example, one command could fill in a whole database with optimized structures from HyperChem.

## **Run Script and Delete Records**

This menu item essentially controls generalized querying in HyperChem Data. A script is run over all the records of a table from HyperChem data and depending on the value of an HSV variable, "hc\_fail" at the end of the script, the database record is deleted or kept as part of a query. This allows generalized querying where the success or failure of the query for a particular record depends on arbitrary code placed in the script and on the results of generalized molecular modeling calculation associated with the molecule of the database record.

## **Bond Menu**

### **Constrained**

Selects the drawing of bonds to be constrained as to both their length and angle (increments of 30 degrees). The constrained length can be short, medium, or long.

### **Unconstrained**








Selects the drawing of bonds as to be unconstrained as to length and angle. This allows, for example, the joining of distant atoms by a bond.

## **Help Menu**

HyperChem Data includes extensive on-line help.

## Drawing Tools

This section gives details on the HyperChem Data drawing tools.

Button	Corresponding Menu Command:	Action:
	Edit/Select All	Select the whole image of the system shown in the workspace
	Edit/Delete	Remove the current selection of the system from the HyperChem Data workspace
	Tool/Select	Change the mouse mode to be the selection mode. With the Selection Mode, you can use the mouse to select a part of the current image or the entire image.
	Tool/Line	Change the mouse to the Line mode. You can use this mode to draw single bonds in the HyperChem Data workspace
	Tool/Double Bond	Change the mouse to the Double Bond mode. You can use this mode to draw double bonds in the HyperChem Data workspace.
	Tool/Double Bond (Right Sideline)	Change the mouse to the Right Sideline Double Bond mode. You can use this mode to draw double bonds with the right bond shorter than the left bond.
	Tool/Double Bond (Left Sideline)	Change the mouse to the Left Sideline Double Bond mode. You can use this mode to draw double bonds with the left bond shorter than the right bond.



Tool/Triple Bond

Change the mouse to the Triple Bond mode. You can use this to draw triple bonds.



Tool/Text

Change the mouse to text mode. You can use this mode to add text to a vertex of a drawing. Position the crosshair where you want to enter text and click the left mouse button. Press **Enter** when you are finished, to anchor the text. Numbers are converted to subscripts.



Tool/Rings

Change the mouse to Rings mode. This brings up the Molecular Rings dialog box, which shows the pre-defined rings. After clicking one of these rings, and then pressing <OK>, you can use the mouse to position this ring in the HyperChem Data workspace.



Draw Color

Access the Color dialog box and specify a color for drawing.



Text Color

Access the Color dialog box and specify a color for text





# Chapter 4

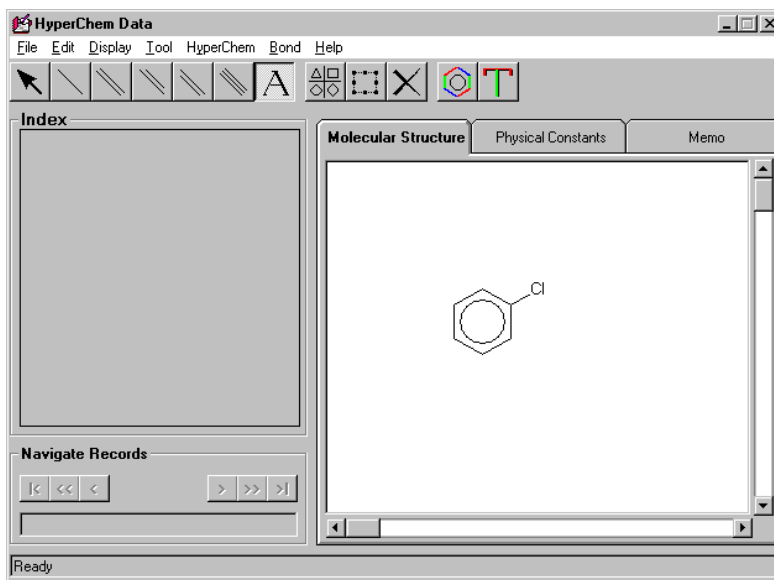
## HDF Files

### HDF Files

HDF files include information on each of the bonds in a molecular drawing as well as each of the text labels associated with particular vertices.

Each vertex in a drawing describes an atom and the file also includes at the end of the file a record explicitly describing these atoms.

If we have a Molecular Structure such as the following:



then the corresponding HDF file is,

```

"PRIMITIVE",1,100,#FALSE#,0,-2147483640,1,2,1875,1530,2221.41,1730
"PRIMITIVE",2,100,#FALSE#,0,-2147483640,2,3,2221.41,1730,2221.615,2129.645
"PRIMITIVE",3,100,#FALSE#,0,-2147483640,3,4,2221.615,2129.645,1874.91,2330.156
"PRIMITIVE",4,100,#FALSE#,0,-2147483640,4,5,1874.91,2330.156,1528.5,2130.156
"PRIMITIVE",5,100,#FALSE#,0,-2147483640,5,6,1528.5,2130.156,1528.795,1729.645
"PRIMITIVE",6,100,#FALSE#,0,-2147483640,6,1,1528.795,1729.645,1875,1530
"PRIMITIVE",7,3,-2147483640,1875.205,1929.645,2008.205,1699.282
"PRIMITIVE",8,100,#FALSE#,0,-2147483640,2,7,2221.41,1730,2567.82,1530
"PRIMITIVE",9,4,1,0,2463,1410,2673,1650,7
"ATOM",1,"C",1875,1530,2,2,4,6,4
"ATOM",2,"C",2221.41,1730,3,1,4,3,4,7,1
"ATOM",3,"C",2221.615,2129.645,2,2,4,4,4
"ATOM",4,"C",1874.91,2330.156,2,3,4,5,4
"ATOM",5,"C",1528.5,2130.156,2,4,4,6,4
"ATOM",6,"C",1528.795,1729.645,2,5,4,1,4
"ATOM",7,"Cl",2567.82,1530,1,2,1
"LABEL",1,"Cl"

```

## PRIMITIVE Records

Records in this file are labelled "PRIMITIVE" for lines (bonds), circles, and labels. Primitive records have tokens separated by commas as described below. The second token in a Primitive record is just the index of the primitive, i.e. 1,2,3,... The third token identifies the type of Primitive according to:

- 100 - single bond
- 101 - double bond
- 102 - right-sided double bond
- 103 - left-sided double bond
- 104 - triple bond
- 3 - circle
- 4 - label

The fourth token describes whether the primitive has children (#TRUE# or #FALSE#) and the fifth token describes the index of the parent primitive (generally 0). The sixth token describes the color of the primitive in a somewhat encoded fashion (such as -2147483640 for black!). For a bond primitive, the seventh and eighth tokens describe the atom indices at the start and end of the

bond (the two vertices). Finally, the last four tokens describe the coordinates of the two end points of the bond.

The circle primitive (3) differs from the bond primitives in that it has tokens only for its color and coordinates of the enclosing rectangle.

The label primitive (4) has a fourth token that describes the index of the label. The fifth token for a label primitive describes its color (a different representation than used in bonds). The 6 through 9 tokens describe the coordinates of the enclosing rectangle for the label. Finally the last token gives the index of the vertex (atom) to which the label is attached.

## Atom Records

Atom records describe the index of the atom, the element symbol for the atom, the x and y coordinates of the atom vertex, the coordination of the atom (number of other atoms it is connected to), and finally the types of bonds it makes.

The bonding information indicates the index of the atom that a given atom is bonded to and then the type of bond. This is repeated for each bond connection.

The types of bonds are:

- 1 - single bond
- 2 - double bond
- 3 - triple bond
- 4 - aromatic bond

## Label Records

The final "Label" records shown above describe the text of each label. The second token is the index of the label and the third token is the label text.



# HyperChem Modules®

## Part 2 HyperNMR



# Chapter 5

## HyperNMR

### Overview

HyperNMR™ is a software package for a *priori* prediction of one-dimensional NMR spectra. It can predict  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{15}\text{O}$ ,  $^{19}\text{F}$  or  $^{31}\text{P}$  spectra for molecular systems involving the elements hydrogen through argon.

HyperNMR displays molecular systems and NMR spectra graphically. They can be copied to other Windows applications via the Windows clipboard to produce hard copy.

HyperNMR provides several renderings of molecules. These renderings are disks and shaded spheres for atoms and sticks, unshaded cylinders, and shaded cylinders for bonds. This can yield, for instance, the popular ball and stick rendering of molecules.

HyperNMR provides other viewing options. Both the mouse and keyboard can be used to rotate the molecule, change its position, and magnify its size. The mouse can be used to select and deselect atoms in a molecular system and to select and deselect the whole molecular system.

HyperNMR can be used in one of two ways:

- As a stand-alone program.
- In conjunction with HyperChem®.

If you do not have a copy of HyperChem installed on your computer, you must use HyperNMR in stand-alone mode. This requires reading a molecule into HyperNMR from a molecular structure file, such as a HIN file, that contains the coordinates of the atoms of the molecule whose NMR spectrum is to be computed.



If you have a copy of HyperChem installed on your computer, you may find it convenient to compute the NMR spectra of a molecule obtained directly from HyperChem using the “Get HyperChem” menu item of HyperNMR. It is not essential to use HyperChem, however, even if you have it installed on your computer.

# Chapter 6

## Installation Guide

### Introduction

This guide contains information on the following:

- System and software requirements.
- Installation procedures.
- WIN.INI settings.
- The interaction of HyperNMR and HyperChem.

### System Requirements

HyperNMR™ requires the following minimum equipment:

- An Intel® 386, 486 or Pentium® compatible CPU. A 386 CPU will require an Intel 80387™ math coprocessor. A computer using an Intel 80486 SX will require an 80487 SX coprocessor.
- A minimum of 4 MB of random access memory (8 MB recommended).
- At least 8 MB of free hard disk space.
- A video display compatible with Microsoft® Windows 3.1, including IBM Video Graphics Array (VGA or Super VGA).
- A mouse or other pointing device compatible with Microsoft Windows, version 3.1.
- Microsoft Windows Version 3.1 or higher running in 386 Enhanced mode and Version 5 or later of DOS, or Windows 95 or Windows NT for PCs.

## Installing HyperNMR

The HyperNMR setup program decompresses and copies files to your hard disk. The names of compressed files end with an underscore (\_).

### To install HyperNMR:

*Caution:* When you install this software, please make sure that the write-protect tabs on the disks allow the disks to be written to. Please use only standard keyboard letters and numbers when you type in the registration information, without using the [Alt] key to get accents or special characters. We recommend that you make a backup copy of the software before you install it, and that you install from the backup rather than from the original disks.

1. Make sure you know your serial number, which is on a label on the back of Disk 1, on the back of this manual, and on your registration card.
2. Place Disk 1 of the distribution set into a floppy drive in your computer.
3. From Program Manager, choose Run from the File menu.
4. In the Command Line text box, enter  
a:\setup  
(Use a different initial letter if your floppy drive is other than a:), and choose OK.
5. Follow the instructions provided by the Setup program.

## WIN.INI settings

The HyperNMR setup program detects the possible presence of an existing copy of HyperChem by looking in the HyperChem section of your WIN.INI file (which is stored in your Windows directory). If it detects a copy of HyperChem it adds the following lines to this section, of the form:

```
NMRExePath = C:\HYPER\NMR
```

```
NMRIniPath = C:\HYPER\NMR
```

```
NMRParmPath = C:\HYPER\NMR
```

(If you installed HyperNMR in a different directory, the name of that directory appears.)

## **Interaction of HyperNMR and HyperChem**

If the HyperNMR setup program does not detect a copy of HyperChem on your machine it will install a new Group (the HyperNMR Group) into the program manager, with a HyperNMR icon in it:



If you double-click on this icon, HyperNMR (NMR.EXE) will be executed and you may perform NMR computations independently of HyperChem.

If a copy of HyperChem is detected, then the HyperNMR icon will be inserted into the HyperChem Group. If you double-click on this icon you can perform NMR calculations independently of HyperChem or interact with a single copy of HyperChem that you execute independently.



# Chapter 7

## Introduction to HyperNMR

### Introduction

Welcome to HyperNMR!

HyperNMR allows you to compute the  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{15}\text{O}$ ,  $^{19}\text{F}$  or  $^{31}\text{P}$  spectrum of a molecular system involving the elements hydrogen through argon. (The elements potassium through xenon could also be included in a non-quantum mechanical fashion, but you would need to define new atom types and certain semi-empirical parameters.) The computation is an *a priori* computation in that for many situations, you need only input the three dimensional structure of a molecule and HyperNMR will subsequently predict all the relevant chemical shifts and coupling constants for the molecule. If desired, this can be followed by a computation of the frequencies and intensities of the one-dimensional NMR absorption spectra.

### Some Basic Ideas

Some of the fundamental ideas in HyperNMR are as follows:

#### Stand-alone or Combined

HyperNMR can be used as a stand-alone program that reads a molecular structure from a file, either in HIN (HyperChem Input) or in Z-Matrix (MOPAC input) format. Alternatively, HyperNMR can be used in conjunction with HyperChem, and a molecular structure can be transferred from HyperChem to HyperNMR by Dynamic Data Exchange (DDE). In either event, the 3-dimensional Cartesian coordinates of a molecule are the starting point for an *a priori* quantum mechanical computation of the NMR spectra.

## Quantum Atoms

Calculations can be performed on systems from small organic molecules to portions of very large macromolecules. For large systems, the molecule can be divided into a quantum mechanical region whose atoms (the “quantum atoms”) are included in the quantum mechanical calculation and a classical region whose atoms are included in the calculation via their electrostatic perturbation on the quantum mechanical atoms. The boundary between the classical region and quantum mechanical region is treated by superimposing a pseudo-atom (called a *capping atom*) on top of any non-quantum atoms that are bonded to quantum atoms. HyperNMR includes parameters for a boundary that is between  $sp^3$ - $sp^3$  bonded atoms or that doesn't cross bonds. You can add additional parameters for other situations.

Quantum atoms and any associated capping atoms are restricted to the elements hydrogen through argon. The elements hydrogen through xenon can be in the classical region, as long as the capping atoms that are bonded to quantum atoms do not include the elements potassium through xenon. If you wish to use any of the elements potassium through xenon, you will need to define new atom types and parameters (see chapters 5 and 6 for information on atom types).

## NMR Atoms

For the atoms that are included in the quantum mechanical calculation, you must specify those atoms (the “NMR atoms”) that should have shielding and spin-spin coupling constants calculated. These atoms are the only ones for which chemical shifts and coupling constants are computed.

## Labelled Atoms

The NMR atoms by default will be protons,  $^{12}\text{C}$ , etc. — the isotope of highest natural abundance. If you specify that certain atoms are “labelled atoms,” then they undergo isotopic substitution to make inactive nuclei active and vice versa. Thus a *labelled* hydrogen atom will have a deuterium nucleus ( $I = 1$ , inactive), and a labelled carbon atom will have a  $^{13}\text{C}$  nucleus ( $I = 1/2$ , active), for example.

## Equivalent Atoms

Because the quantum mechanical calculations of shielding constants and spin-spin coupling constants are done at a fixed nuclear geometry, equivalent to 0 degrees Kelvin, the protons of a methyl group, for example, will not all be equivalent. HyperNMR makes it possible for the user to specify sets of atoms, “equivalent-atoms-1”, “equivalent-atoms-2”, etc. whose shielding constants and spin-spin coupling constants are to be averaged prior to producing a spectrum. This way you can ensure that all the protons of a methyl group share a single chemical shift, for example.

## Using HyperNMR for calculations

There are typically three steps required to perform a HyperNMR prediction of an NMR spectrum.

The first step is to set up a molecular system. In this step you load a molecule into HyperNMR and specify the sets of atoms described above. If you have HyperChem you can create the molecular system and named selections in HyperChem. This step is described in Tutorial I.

The second step is to calculate shielding and nuclear spin-spin coupling using semi-empirical quantum mechanical calculation. This step is described in Tutorial II and the computational chemistry methodology is described in Chapter 6.

The third step is to calculate and display a one-dimensional NMR spectrum, for the type of nucleus and simulated spectrometer frequency that you specify. This step is outlined in Tutorial III and the computational chemistry methodology is described in Chapter 6.





# Chapter 8

## Tutorials

### Overview

This chapter contains a set of tutorials for learning how to use HyperNMR. We strongly encourage you to go through the tutorials one at a time so as to become familiar with the operation of HyperNMR. Each tutorial introduces specific features of HyperNMR but leans on previously introduced features, so the tutorials should be completed in the order given.



In some cases, alternative procedures can be followed if you use HyperChem with HyperNMR. These sections begin with a HyperChem icon (the beaker at the beginning of this paragraph). Some of the optional features of the later tutorials can only be completed in conjunction with HyperChem.

### Tutorial I: HyperNMR Basics

This tutorial introduces you to the HyperNMR interface and discusses using menus and tools to open, edit, display, and manipulate molecular systems.

#### Starting HyperNMR

##### To start HyperNMR:

1. Start Windows if it is not already running.
2. In the Program Manager window, position the cursor over the HyperNMR icon:

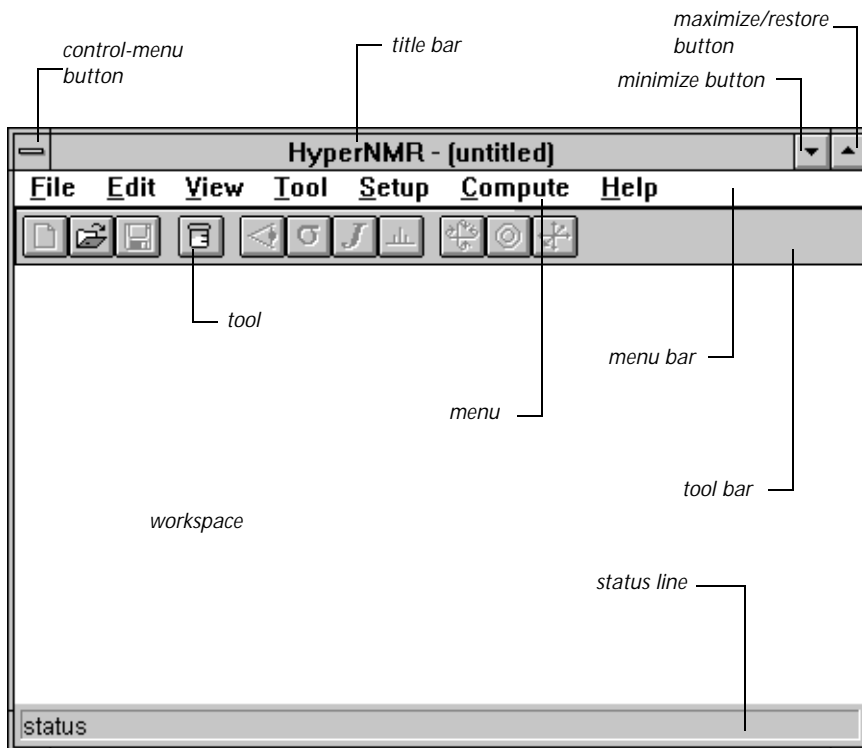


3. Double-click on the HyperNMR icon or click on it and then press the Enter key.

The HyperNMR window opens.

## HyperNMR Window

Familiarize yourself with different parts of HyperNMR window.



If you are using Windows 95 or Windows NT, the appearance of the frame and the arrangement of the control-menu, maximize/restore, and minimize buttons may be slightly different.

## **Title Bar**

The title bar shows the name of the file you are working on. The name appears as untitled if you are working on a file that has not been previously saved.

## **Menu Bar**

The menu bar contains the names of the HyperNMR menus which are: File, Edit, View, Tool, Setup, Compute, and Help.

## **Tool Bar**

The HyperNMR tool bar contains eleven tools. You can use them to load molecular systems, shielding constants, spin-spin coupling constants and NMR spectra into HyperNMR, save the molecular system and calculations, change the contents of the HyperNMR workspace, and manipulate molecular systems. The HyperNMR tools are the shortcuts for the menu items in the menu bar.

## **Workspace**

The HyperNMR workspace is used to show molecular systems, view shielding and coupling constants, and display NMR spectra.

## **Status Line**

The status line of HyperNMR shows the actions and status when you load a system to HyperNMR or a brief description of the menu item or tool button when you choose a menu item or click on a tool button of HyperNMR.

## **Help**

You can use the Help menu to get on-line help and see the version information for HyperNMR.

## **Control-Menu Button**

The Control menu contains the commands to resize, move, maximize, minimize, and close HyperNMR. It also contains the Switch command, which lets you activate another Windows application.

## Maximize/Minimize Buttons

The Maximize button expands a window to display it in a full-screen size and the Minimize button shrinks a window to an icon. When you maximize a window, the Maximize button changes to a restore button, which you can use to return the full-screen window to its previous size.

## Using the Mouse

The following terms describe using the mouse with HyperNMR.

Point	Position the mouse so that the cursor points at what you want to select in the HyperNMR window.
L-click	Press and release the left mouse button.
R-click	Press and release the right mouse button.
Double-click	Press and release the left mouse button twice quickly.
L-drag	Hold down the left mouse button and slide the cursor to a new position in the workspace, then release the mouse button.
R-drag	Hold down the right mouse button and slide the cursor to a new position in the workspace, and then release the mouse button.

## Keyboard Alternatives

HyperNMR provides keyboard alternatives to using the mouse. The following is a list of some keyboard actions.

### To open a menu:

Each menu in the menu bar has an underlined letter. When you press the [Alt] key and this letter, the corresponding menu opens. For example

1. Hold down the [Alt] key and press the [F] key.

### To close a menu:

Press the [Alt] key or [Esc] key.

### To choose a menu item:

Once a menu is open, press the key for the underlined character in the menu item that you wish to invoke. For example:

1. Hold down the [Alt] button and simultaneously press the [F] key to open the File menu.
2. Press the [O] key.

The Open File dialog box opens. This is the same as using the mouse to choose Open under the File menu.

### To rotate a molecular system:

You can use the left, right, up, and down arrow keys to rotate the molecular system, when it is shown in the HyperNMR workspace, even if HyperNMR is busy doing calculations.

### To translate a molecular system:

You can press the shift key and simultaneously press any arrow keys (left, right, up, and down arrow keys) to translate the molecular system, if a molecular system is shown in the HyperNMR workspace, even if HyperNMR is busy doing calculations.

*Note:* Chapter 5 in this manual lists the details of the keyboard alternatives.

## Bringing a Molecule into HyperNMR

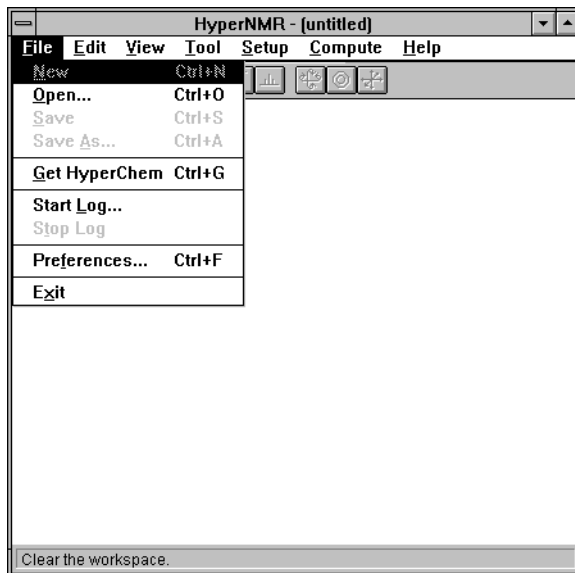
### Opening a Sample File

You can read three kinds of files into HyperNMR. They are HyperNMR files (with the extension “.NMR”) which may contain a molecular system, shielding constants, shielding tensors, coupling constants, and an NMR spectrum; HyperChem HIN files (with the extension “.HIN”) which contain a molecular system; and MOPAC Z-Matrix files (with the extension “.ZMT”) which may contain the x, y, z Cartesian coordinates or the internal coordinates of molecules. There are some sample files in your samples directory under the directory where you installed HyperNMR.

## To open a HIN file:


1. Move the cursor to File on the menu bar at the top of the HyperNMR window.
2. L-click on File.

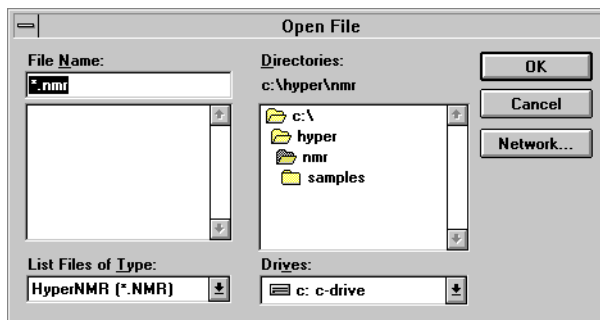
The File menu opens. An ellipsis ( . . . ) appears after some menu items. This means that a dialog box will open if you choose one of these menu items.



3. L-click on Open.

The Open File dialog box appears.

**Shortcut:** L-click the  tool button on the tool bar at the top of the HyperNMR workspace to open the Open File dialog box.



### To change directory:

If the file that you wish to open is not in the current directory, you may want to change the directory. In this case we want to locate the HyperNMR sample files. To do this:

1. If necessary, L-drag the scroll box in the Directories list box to find the name of the directory where you installed HyperNMR.
2. Double-click on the HyperNMR directory in the Directories list box.
3. Double-click on the samples directory below the HyperNMR directory in the Directories list box.

The File Name list box should now show the sample files that are distributed with HyperNMR and are in NMR format.

### To change file type:

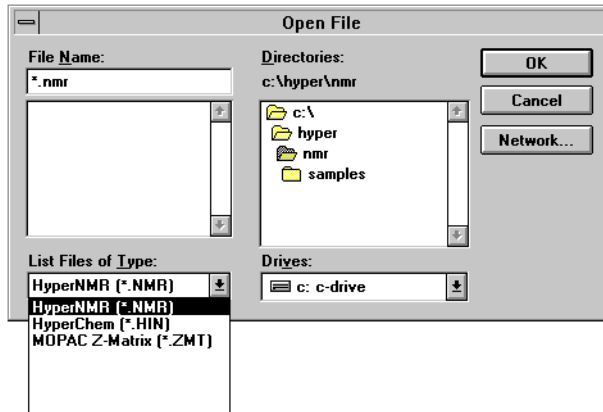
There is a list at the bottom left of this dialog box called List Files of Type. HyperNMR can open three types of files; they are HyperNMR NMR files (with the extension “.NMR”), HyperChem HIN files (with the extension “.HIN”), and MOPAC Z-Matrix files (with the extension “.ZMT”).

The default type of files in the Open File dialog box is NMR, so the File Name list box in this dialog box lists all the files with the extension NMR under the current directory. To get all the HyperChem HIN files under the current directory to be listed in the File Name list box, the file type listed in the List Files of Type box has to be changed. To do this:

1. Place the cursor on the arrow key located to the right of the List Files of Type box and L-click on it.



A menu shows all the types of files available in HyperNMR.



2. L-click on the HyperChem (\*.HIN) item.

The file type in List Files of Type box switches to HyperChem (\*.HIN) and the File Name list box shows all the files with the HIN extension under the current directory.

### To scroll through the list:

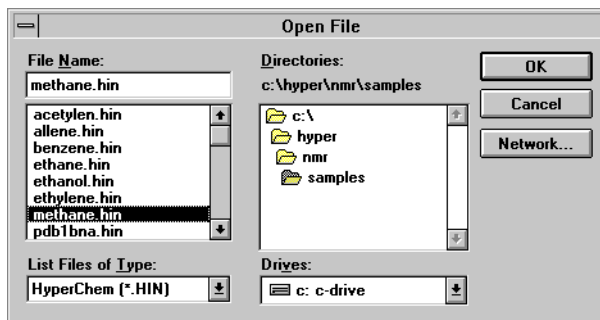
The files listed in the File Name list box are sorted in alphabetical order. You can use the scroll bar to scroll up and down the file list. To do this:

1. Position the cursor over the scroll box (the raised button in the middle of the scroll bar) and L-drag it up and down.  
HyperNMR scrolls through this list of file names.
2. L-click on the up and down arrow keys to scroll one line at a time.

### To select a file:

1. L-click on METHANE.HIN in the samples directory under the directory where HyperNMR is installed.

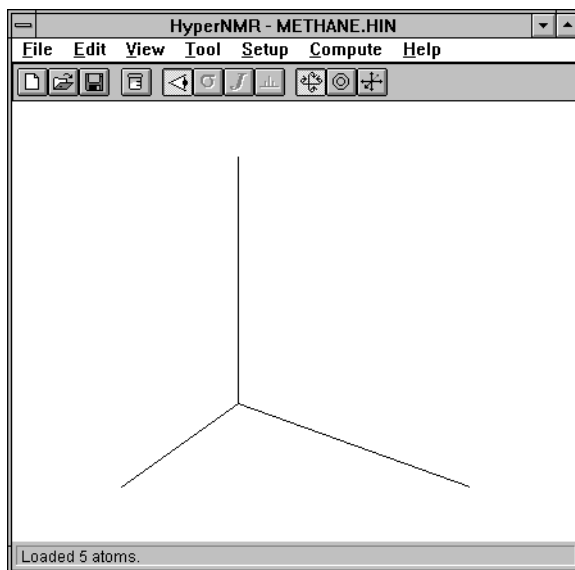
HyperNMR displays this name in the File Name text box:



You can also enter a file name by typing it in the File Name text box, instead of performing an L-click on METHANE.HIN.

2. L-click the OK button.

The dialog box disappears. The cursor turns into an hour-glass icon and all the tool buttons on the tool bar and the menus on the menu bar, except the Help menu, are grayed, because HyperNMR is currently busy reading the file and preparing to display the molecular system. As HyperNMR reads the file, the number of atoms is displayed in the Status Line at the bottom of the HyperNMR window. The title bar shows HyperNMR with the file name, METHANE.HIN in this example, after HyperNMR has read in the file.



**Shortcut:** You can also open a file by double-clicking on a file name in the File Name list box.

## Bringing a Molecule in from HyperChem

If you don't have HyperChem, please skip this section.



If you have a copy of HyperChem on your machine, you can load molecular systems directly from it to HyperNMR. If you are new to HyperChem, please consult the HyperChem manuals.

### Getting a molecule from HyperChem:

1. Start HyperChem, then draw and model build a methane molecule ( $\text{CH}_4$ ) in the HyperChem workspace.
2. Move the cursor to File on the menu bar at the top of HyperNMR window.
3. L-click on File.


The File menu opens.

4. L-click on Get HyperChem.

HyperNMR will warn you that there are no named selections (these will be set up in Tutorial II).

5. Ignore this warning by L-clicking on the OK button in the message box.

The molecule in the HyperChem workspace will be loaded in HyperNMR. The title bar shows HyperNMR with “(untitled)” because the current molecular system hasn't been saved in HyperNMR.

**Shortcut:** L-click the  tool button on the tool bar at the top of the HyperNMR workspace to get a molecule from the HyperChem workspace.

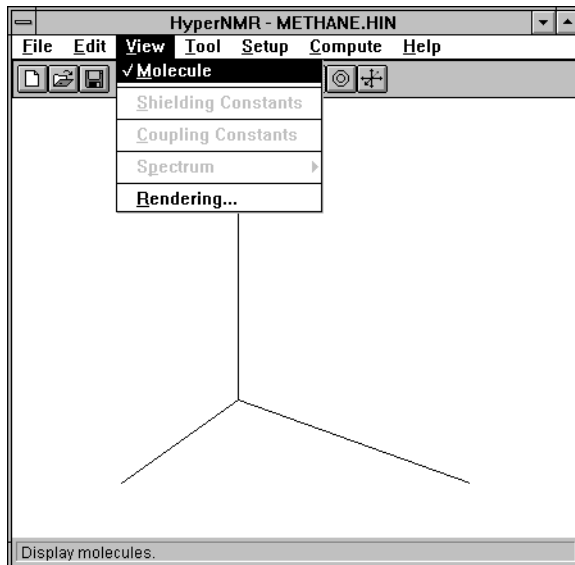
## Changing Renderings

HyperNMR automatically uses the rendering settings from the last session. You can change the rendering settings via the Rendering dialog box.

## To open rendering dialog box:

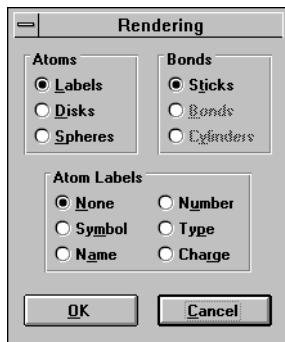
1. L-click on the View menu in HyperNMR.

The HyperNMR View menu appears as:



2. L-click on the Rendering menu item.

The HyperNMR Rendering dialog box opens:



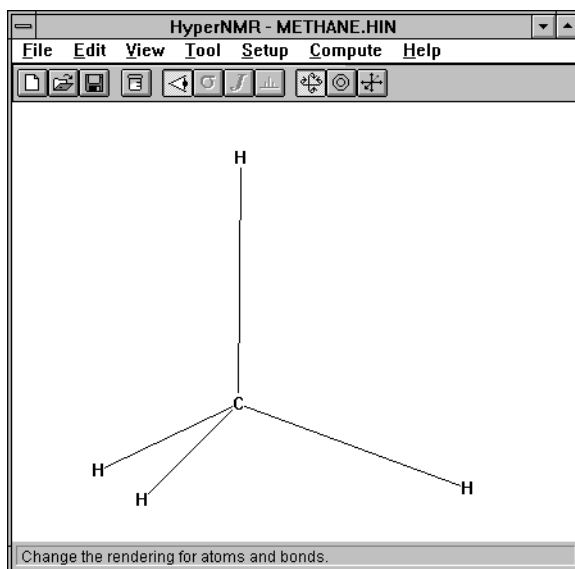
In the Rendering dialog box, there are three groups of radio buttons which are named Atoms, Bonds, and Atom Labels, respectively.

## Changing Atom Renderings

### To display labels:

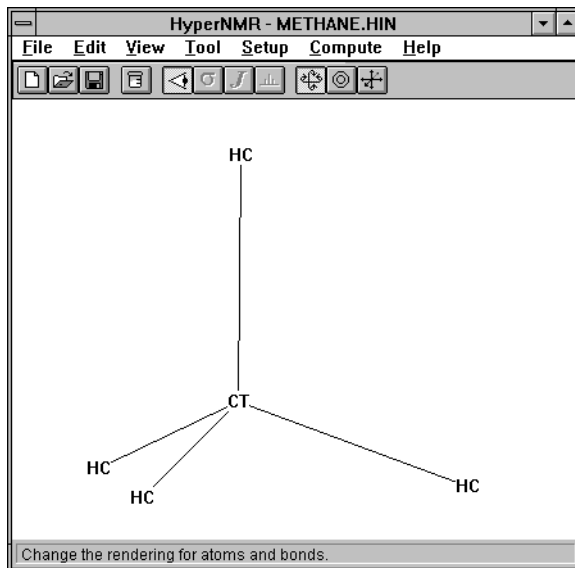
1. L-click on the Labels radio button in the Atoms group.  
This activates all the radio buttons in the Atom Labels group.
2. L-click on the Symbol radio button.
3. L-click on OK.

The HyperNMR workspace shows methane with the element symbol attached to each atom:



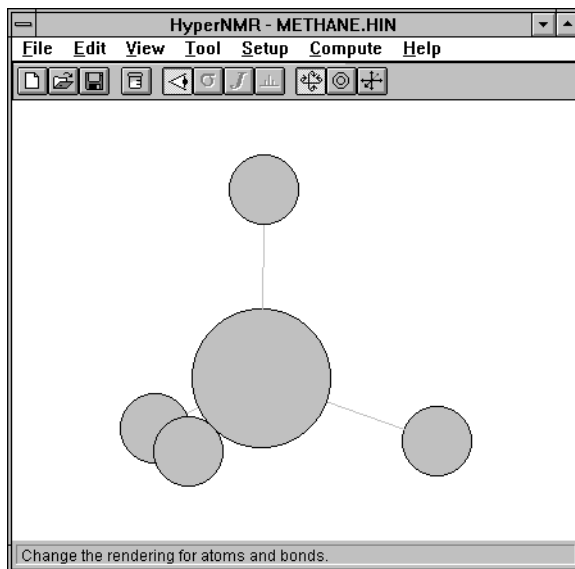
4. Open the Rendering dialog box again, L-click on the Type radio button, and then click on OK.

HyperNMR displays methane with the atom type attached to each atom:



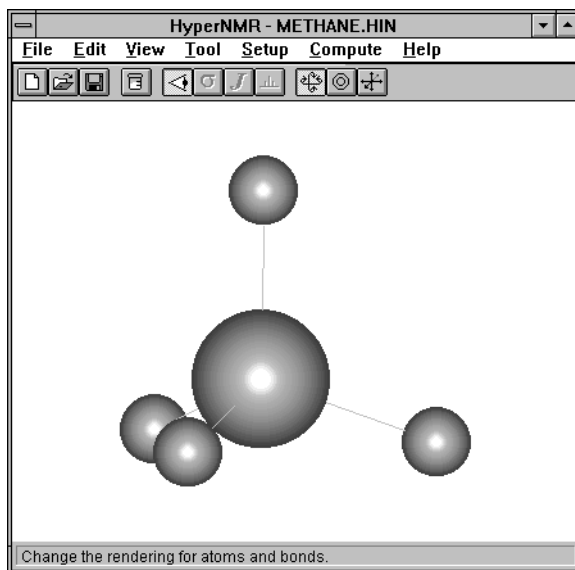
### To display disks and spheres:

1. Open the Rendering dialog box.
2. L-click on Disks.  
This disables all the radio buttons in the Atom Labels group.
3. L-click on OK.  
HyperNMR displays the atoms of methane as filled circles:



4. Repeat step 1 and L-click on the Spheres radio button in the Rendering dialog box, and then click on the OK button.

HyperNMR displays the atoms of methane as shaded spheres:



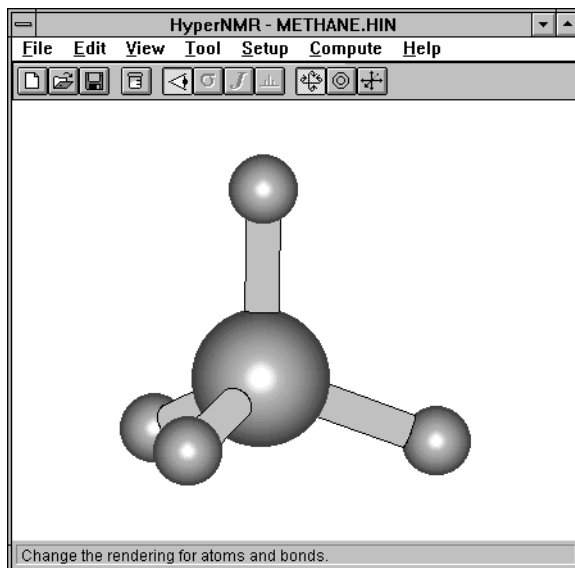
## Changing Bond Renderings:

You may have noticed that the renderings of bonds, which are drawn as sticks, have not changed at all when we changed the renderings of the atoms for methane. The following steps show how to change the renderings of bonds.

### To display bonds:

1. Open the Rendering dialog box.
2. L-click on the Bonds radio button.
3. L-click on OK in the Rendering dialog box.

HyperNMR shows the atom bonds of methane:

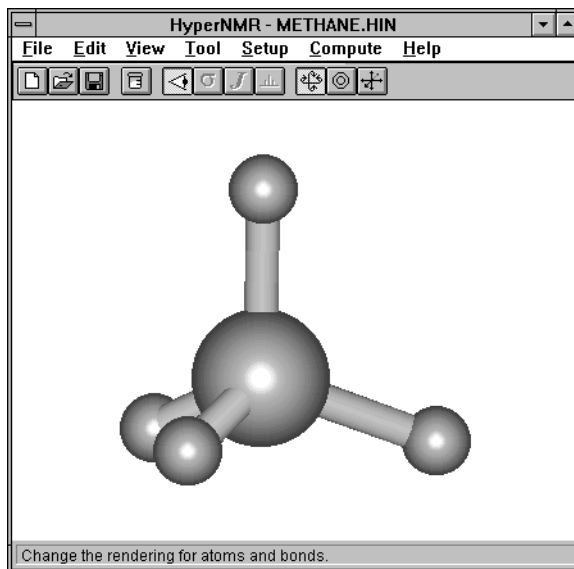


### To display cylinders:

1. Open the Rendering dialog box.
2. L-click on Cylinders and L-click on the OK button.

This dismisses the Rendering dialog box and changes the rendering of bonds to shaded cylinders.





**Note:** Some combinations of renderings of atom and bond are not available; for example, shaded bonds are available only with spheres. For details see Chapter 5 in this manual.

## Translating, Rotating, and Scaling Molecules

You can use the mouse and the keyboard to translate and rotate the molecular system and scale the molecular system to fit in the HyperNMR workspace, if the HyperNMR workspace is displaying a molecular system. The following steps illustrate using the mouse. For keyboard equivalents, see page 162.

### **XY translation:**


You can use the translation tool to move molecules along the plane of the computer screen. Before doing this, you may want to change the atom and bond renderings to be labels and sticks in order to speed up the molecular manipulation, if you have followed this tutorial.

1. L-click on Tool.
2. Select Pointer Zoom & Translate.


Now the mouse pointer is in Zoom and Translate mode.

3. Move the cursor into the workspace.
4. L-drag the mouse horizontally, vertically, and diagonally.

The molecule moves along the X axis, Y axis, and does XY translation.

*Shortcut:* L-click the  tool button on the tool bar to change the mouse mode to Zoom & Translate.

### Zoom:

Check that the  tool button is down and a molecular system is showing in the workspace (to make sure that the mouse mode is set to Pointer Zoom & Translate) and then

1. R-drag upwards.  
The molecular image gets smaller.
2. R-drag downwards.  
The molecular image gets larger.

### Scale to fit:

When a molecular system is first loaded from a file, it may be centered in the workspace in a standard size. After you have translated and zoomed it, the molecular system may not be centered anymore. You can return it to its original size and location in the workspace by using Scale to Fit menu item under the Tool menu. To do this:

1. Select Scale to Fit under the Tool menu, or alternatively, press the spacebar.  
The molecule returns to its original size and to the center of the workspace.

### XY rotation:

To rotate the molecular system in HyperNMR around the x and y axes, use the XY rotation tool. To do this:

1. Select Pointer Rotate under the Tool menu.
2. Move the cursor into the workspace.
3. L-drag horizontally.

The molecular system rotates around the y axis.

4. L-drag vertically.

The molecular system rotates around the x axis.

5. L-drag diagonally to rotate the molecular system around both the x and y axes.

*Shortcut:* L-click the  tool button on the tool bar to change the mouse mode to Rotate.

### Z rotation:

Repeat steps 1 and 2 as done for XY rotation if you have changed the mouse mode and then do the following step:

1. R-drag horizontally.

This rotates the molecule in the plane of the computer screen. Vertical R-dragging is ignored.

## Selecting and Deselecting

To look at the information about specific atoms, to select a subset of atoms to be treated quantum mechanically, or to examine the NMR properties of a subset of atoms (see Tutorial II) you can use the selection tool in HyperNMR.

In order to do selection and/or deselection, the mouse mode needs to be put in selection mode by your selecting Pointer Select under the Tool menu. Once you have changed your mouse mode, you can do any or all of the following operations.

### To select and deselect an individual atom:

1. L-click on one of the atoms in methane.

This highlights all the ends of bonds connecting to this atom and the Status Line of HyperNMR shows the atom index, element symbol, atom type, and its Cartesian coordinates.

2. R-click on this atom.


This deselects the atom.

### To select and deselect multiple atoms:

1. L-click on one of the hydrogens in methane.
2. Press and hold down the [Shift] key and L-click on all the hydrogens in methane and then release the [Shift] key.  
This selects all four hydrogens in methane.
3. R-click on any two of these four selected hydrogens.  
This reduces the number of selected atoms to two.
4. R-click on an empty area of the workspace.  
This deselects all atoms.

### To select and deselect the molecular system:

1. L-click on an empty area of the HyperNMR workspace.  
This selects the whole molecular system, methane in this example.
2. R-click on an empty area of the workspace.  
This deselects all atoms.

*Shortcut:* L-click on the  tool button on the tool bar to change the mouse mode to Pointer Select.

## Copying a Molecular Image to the Clipboard

You can use Copy under the Edit menu of HyperNMR to copy a molecular image to the Windows clipboard.

### To copy a molecular image to the Clipboard:

1. Select Copy on the Edit menu.  
This copies the image of the molecular system, methane in this example, to the Windows clipboard. The atom and bond renderings are the same as shown in the HyperNMR workspace. You can also change the renderings for atoms and bonds and copy the image again.

## Exiting HyperNMR

For now, exit HyperNMR.

### To exit HyperNMR:

1. Select Exit on the File menu of HyperNMR.

HyperNMR may bring up a message box and ask you whether or not you want to save the current molecular system. You can L-click on the No button in the message box to discard the current system or you can L-click on the Yes button to save the molecule. If you choose Yes, HyperNMR gives you the Save As dialog box and lets you choose a file name. You can simply type in CH4 in the File Name text box of this dialog box. HyperNMR will automatically add the extension .NMR to CH4, so the full name of your file is CH4.NMR. Then you can press OK. This closes the HyperNMR window.

## Tutorial II: Calculation of NMR Shielding & Coupling

This tutorial introduces you to the computation and display of NMR properties such as shielding constants, chemical shifts, and nuclear spin-spin coupling constants.

Before you can display the NMR properties for the current molecular system, the properties must be computed. HyperNMR requires various settings for computing NMR properties. This tutorial will show you how to set up the options for computing NMR properties and how to compute and display the NMR properties.

### To reopen HyperNMR:

1. Point to the HyperNMR icon in the Windows Program Manager.
2. Double-click on the HyperNMR icon.

The HyperNMR window appears.

### To load a molecular system:

Load METHANE.HIN into HyperNMR. If you need help, please read “Bringing a Molecule into HyperNMR” on page 89.

## Setting Atom Types

In order to have HyperNMR calculate shielding and spin-spin coupling constants, all atoms must have atom types assigned to them. Once you have a molecule loaded into HyperNMR, you can check if all atoms have atom types by displaying the atoms with atom type labels.

### To check atom types:

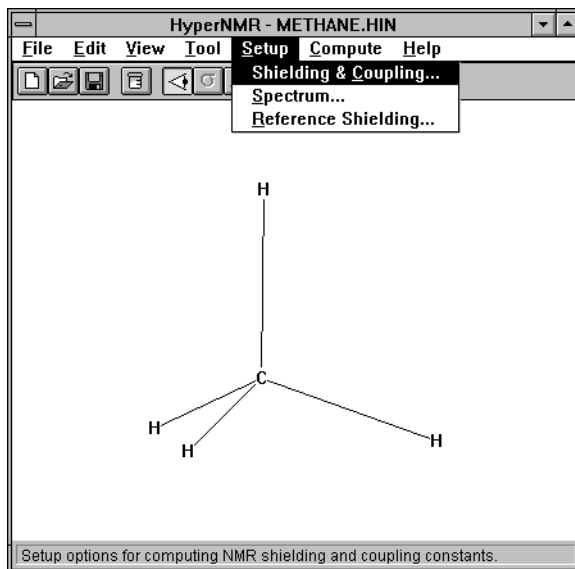
1. Use the View/Rendering menu item to open the Rendering dialog box.
2. Choose Labels in the Atoms group and Type in the Atom Labels group of the Rendering dialog box.
3. L-click on OK to close the dialog box.  
The atoms are labelled by atom type.
4. Check that none of the atoms have \*\* (unknown type) as the atom type.
5. If any of the atoms are type \*\*, use the Edit/Calculate Atom Types menu item to assign types using type rules (see Chapter 5 for details) and verify that types have been assigned.

## Setting Up Options

HyperNMR computes NMR properties based on the wave function which, in turn, is calculated using a semi-empirical quantum mechanical method. See Chapter 6 in this manual for details. Before you can start a HyperNMR computation, you need to specify various selections and options, such as quantum atoms, NMR atoms, equivalent atoms, labelled atoms, SCF wave function options and the computational method. The following steps show you how to set up these selections and options.

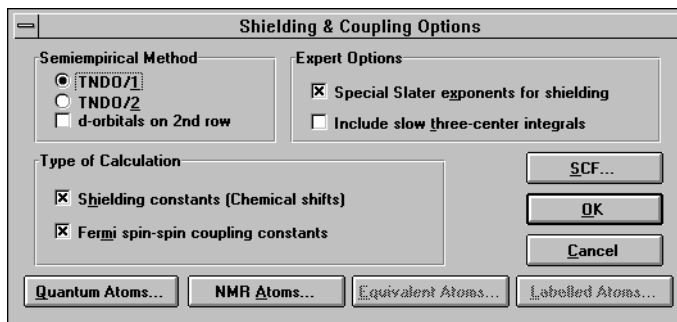
### Opening the Shielding & Coupling Options dialog box:

1. L-click on the Setup menu of HyperNMR.  
This opens a drop-down menu:



2. L-click on the Shielding & Coupling menu item.

The Shielding & Coupling Options dialog box appears:



3. L-click on the TNDO/2 radio button.

This tells HyperNMR to compute the wave function using the TNDO/2 semi-empirical method, instead of TNDO/1.

**Note:** In general, TNDO/2 gives better results than TNDO/1 and is the recommended method for most problems. The differences between TNDO/1 and TNDO/2 are described in Chapter 6.

4. Check the Shielding constants and Fermi spin-spin coupling constants check boxes in the Type of Calculation group.

This instructs HyperNMR to compute both the shielding constants and the coupling constants.

**Note:** You can uncheck Fermi spin-spin coupling to predict a decoupled NMR spectrum. To simulate a broadband decoupled  $^{13}\text{C}$  spectrum, you should uncheck this option and label all carbon atoms.

5. Uncheck the two check boxes in the Expert Options group.

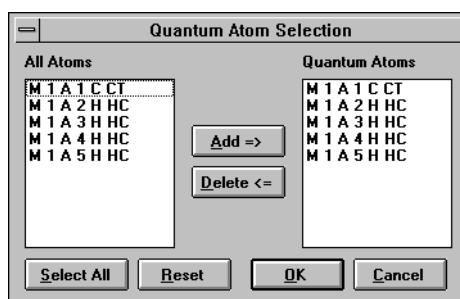
The Special Slater exponents for shielding option will be turned back on at the end of Tutorial III.

**Note:** In general we recommend using Special Slater exponents for shielding (leave this option checked) and not using the slow three center integrals (leave this option unchecked).

### Selecting quantum atoms:

HyperNMR will treat the whole molecular system quantum mechanically if there is no quantum atom selection or if the whole molecular system has been selected as quantum atoms. HyperNMR will treat the selected region quantum mechanically and the unselected region as classical point-charges if you have specified a region of quantum atoms. Specifying a selection is useful for large systems to reduce computational time and memory requirements to a tractable level, especially for a protein or other macromolecular system.

6. L-click on the Quantum Atoms button in the Shielding & Coupling Options dialog box to open the Quantum Atom Selection dialog box.



The atoms are denoted by “M” followed by the molecule number, “A” followed by the atom number within the molecule, then the element symbol and the TNDO/1 or TNDO/2 atom



type. (See Chapter 6 for a discussion of atom types and Chapter 5 for information on using type rules to assign atom types.)

**Note:** If you had any atoms selected in the HyperNMR workspace, they will automatically be added to an empty (or optionally replace a previous) Quantum Atoms list. You may prefer to choose quantum atoms with the selection tool before opening this dialog box.

7. L-click on Select All.

All the atoms in methane are selected (highlighted).

8. L-click on Add ⇒.

The selected atoms will be treated as quantum atoms

9. L-click on OK.

This accepts the current quantum atom selection and dismisses the Quantum Atom Selection dialog box.



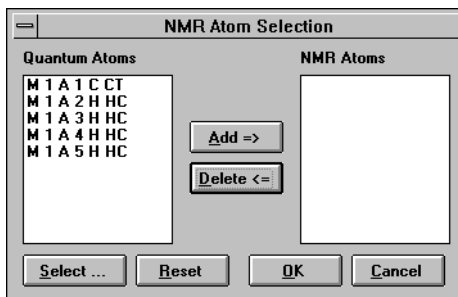
**Note:** If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To do the Quantum Atoms selection, first you select the atoms which you would like HyperNMR to treat quantum mechanically. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter ***quantum-atoms*** for the name.

### Selecting NMR atoms:

You must specify the set of atoms for which you want to calculate shielding and coupling constants. These atoms, which must also be quantum atoms, are called the NMR atoms.

10. L-click on the NMR Atoms button in the Shielding & Coupling Options dialog box.

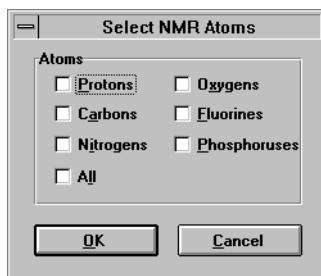
This brings up the NMR Atom Selection dialog box.



**Note:** If you had any atoms selected in the HyperNMR workspace, they will automatically be added to an empty (or optionally replace a previous) NMR Atoms list. You may prefer to choose NMR atoms with the selection tool before opening this dialog box.

11. L-click on the Select ... button.

This opens the Select NMR Atoms dialog box.



12. L-click on the All check box.

This tells HyperNMR to select all the atoms as NMR atoms.

13. L-click on the OK button in the Select NMR Atoms dialog box.

14. L-click on the Add => button in the Select NMR Atoms dialog box.

15. L-click on the OK button in the NMR Atom Selection dialog box.

This action stores the NMR selection and closes the NMR Atom Selection dialog box.



**Note:** If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To do the NMR Atoms selec-

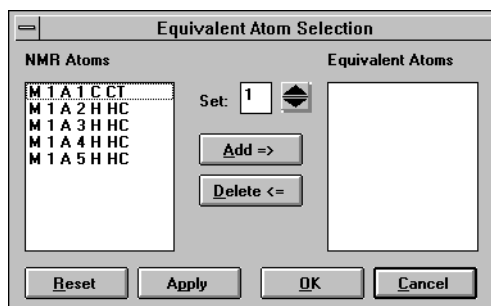
tion, first you select the atoms which you would like HyperNMR to treat as NMR atoms. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter ***nmr-atoms*** for the name.

### Setting up equivalent atoms:

You may want to specify sets of equivalent atoms within the NMR atom selection. This is often necessary because HyperNMR calculates the shielding and spin-spin coupling constants using the rigid molecular geometry of the system in the workspace. Thus atoms that would be equivalent in an experiment, due to averaging of thermal motions for many molecules, would not be automatically equivalent in a HyperNMR calculation. HyperNMR offers you an option to average the computed shielding constants and coupling constants artificially among the atoms of an equivalent atom set which you designate. For example, the four hydrogens in the current molecule may not be precisely equivalent in the fixed geometry that will be used in the calculation but would be equivalent in an experiment, so you should tell HyperNMR to average the shielding constants and coupling constants among these four hydrogens.

16. L-click on the Equivalent Atoms button in the Shielding & Coupling Options dialog box.

The Equivalent Atom Selection dialog box appears.




17. L-drag these four hydrogens and then L-click on the Add => button.

This adds these four hydrogens to the Equivalent Atoms list box.

18. L-click on the Apply button.

This tells HyperNMR to treat these four hydrogens as equivalent atom set 1.

**Note:** If you have more than one set of equivalent atoms, you can L-click on the spin up button, , to number the next set of equivalent atoms and then repeat steps 17 and 18. The maximum number of sets of equivalent atoms is 10.

19. L-click on OK.

This stores the equivalent set and dismisses the dialog box.

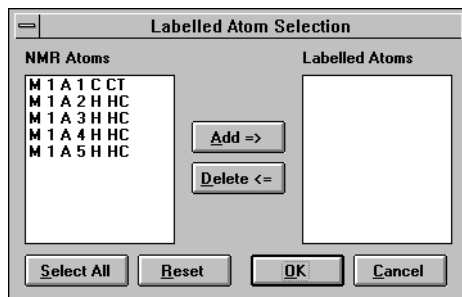


**Note:** If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To do the equivalent atom selections, first you select the atoms which you would like HyperNMR to treat as a set of equivalent atoms. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter **equivalent-atoms-1** for the name. If you have more than one set of equivalent atoms, you can repeat this procedure again but enter **equivalent-atoms-2**, **equivalent-atoms-3**, etc. for the selection names.

### Selecting labelled atoms:

Labelling replaces the most abundant isotope with the next most abundant isotope. Many nuclei, including  $^{12}\text{C}$  and  $^{16}\text{O}$ , have no nuclear spin, and thus have no spin-spin coupling with any other atoms. If you want them to have spin-spin coupling with others, you need to label them. If you want to treat some hydrogens as deuterium, and thus remove spin-spin coupling, you need to label them. To select labelled atoms:

20. L-click on the Labelled Atoms button in the Shielding & Coupling Options dialog box to open the Labelled Atom Selection dialog box.



21. L-click on the first atom, carbon, in the NMR Atoms list box and then L-click on the Add => button.

The carbon is added to the Labelled Atom list box, changing it from  $^{12}\text{C}$  to  $^{13}\text{C}$ .

22. L-click on OK.

This stores the selected labelled atoms and closes this dialog box.



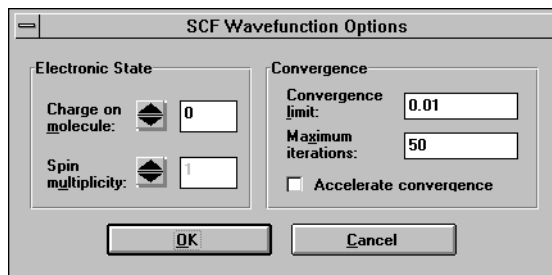
**Note:** If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To do the labelled atoms selection, first you select the atoms which you would like HyperNMR to treat as labelled atoms. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter **labelled-atoms** for the name.

### Setting up options for the SCF calculation:

The SCF options inform HyperNMR about the net charge on the molecular system, the desired accuracy of the wave function, and the maximum number of iterations allowed to reach that accuracy.

23. L-click on the SCF button in the Shielding & Coupling Options dialog box.

This shows the SCF Wavefunction Options dialog box.



24. Set the Charge on molecule to 0.
25. Enter 0.01 in the Convergence limit text box and 50 in the Maximum iterations text box.
26. L-click on OK to accept the settings and close this dialog box.
27. L-click on the OK button in the Shielding & Coupling Options dialog box.

This saves the options you have chosen and closes the Shielding & Coupling Options dialog box.

## Computing Shielding and Coupling Constants

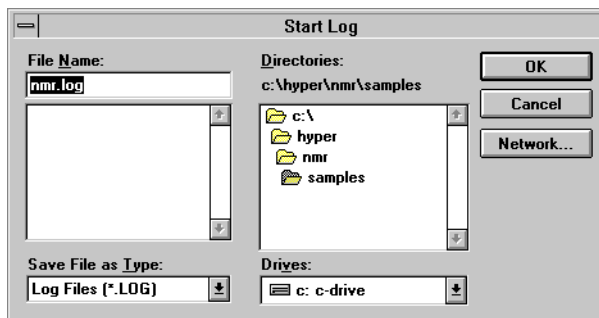
Once you have set up the selections and options, you can compute the shielding constants and spin-spin coupling constants for the current system. If you want to save details of the calculation, you can use a log file.

### To start a log file:

You can start a log file by selecting Start Log under the File menu of HyperNMR.

1. L-click on File on the HyperNMR menu bar.
2. L-click on Start Log.

The Start Log dialog box appears with the default log file name of "NMR.LOG".



3. L-click on OK to accept the log file name.

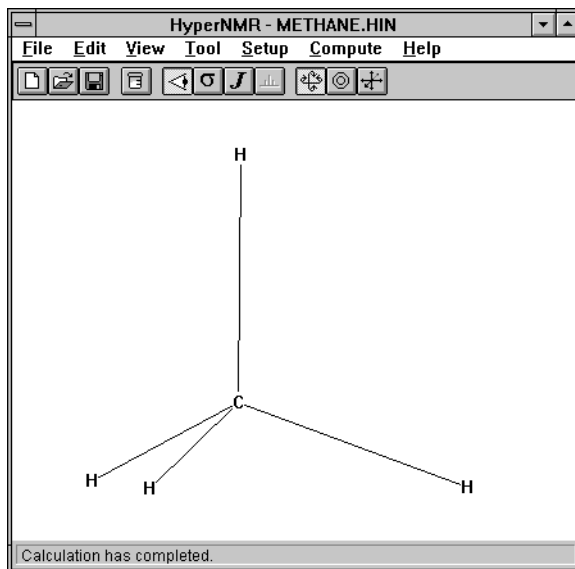
### To compute shielding and coupling:



1. L-click on the Compute menu.
2. L-click on the I - Shielding & Coupling menu item.

This inactivates the whole menu bar and the tool bar of HyperNMR, except for the Help menu, changes the cursor to an hour-glass, starts the HyperNMR backend program with the icon



and computes the shielding constants and spin-spin coupling constants. Wait until the backend icon disappears (if it did appear on your screen) and the HyperNMR status line states that the calculation has completed:



**Note:** The Shielding Constants and Coupling Constants menu items under the View menu and the respective tool buttons  and  are active once the shielding and coupling constants are available.

## Viewing Shielding and Coupling Constants

Once you have computed shielding and coupling constants, you can display them.

### To view shielding constants:

1. L-click on the View menu on the HyperNMR menu bar.
2. L-click on the Shielding Constants menu item.

This changes the contents of the HyperNMR workspace to show the shielding constant and chemical shift table. In this table there are four rows and six columns. The rows are: index, shielding constant, chemical shift, and tau value. The atom indices are comprised of the molecule number, a hyphen, the



atom number within the molecule and the element symbol in parentheses.

The screenshot shows the HyperNMR software interface for a file named METHANE.HIN. The main workspace displays a table with the following data:

Index	1-1(C)	1-2(H)	1-3(H)	1-4(H)	1-5(H)
Shielding	49.457	22.078	22.078	22.078	22.078
Shift	-0.862	1.873	1.873	1.873	1.873
Tau	10.862	8.127	8.127	8.127	8.127

At the bottom of the workspace, there is a status bar that reads "View the shielding constants." The software interface includes a menu bar (File, Edit, View, Tool, Setup, Compute, Help) and a toolbar with various icons.

**Shortcut:** L-click on the  tool button to change the contents of the workspace to display the shielding constants.

You can edit the shielding values in this table. The edited numbers will be used in any spectra that you subsequently calculate.

If the table is wider than your HyperNMR workspace, a scroll bar appears at the bottom of the workspace. You can use this scroll bar to view hidden parts of the shielding table. Similarly, a vertical scroll bar will appear at the right of the workspace if the table is taller than your workspace.

You can also select numbers (from at least two columns) and plot them in a 2D or 3D bar chart by using the View/Graph menu item. This will open an NMR Graph dialog box. Double clicking on a graph in an NMR Graph dialog box will put a bitmap image of the graph into the Windows clipboard.

HyperNMR can be used to calculate the shielding tensors for the selected NMR atoms, if the item named

CalculateShieldingTensor=

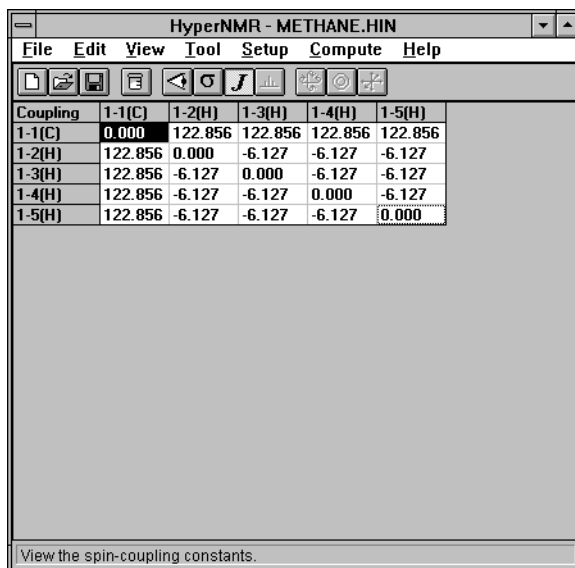
in the NMR.INI file is set to YES. You can view the calculated shielding tensors in the Shielding Tensor dialog box, which is

accessible from the View/Shielding Tensor menu item. This menu item is available only when a shielding constant table is shown in the HyperNMR workspace.

### To view coupling constants:

1. L-click on the View menu.
2. L-click on the Coupling Constants menu item.

This changes the contents of the workspace to display the coupling constant table.



Coupling	1-1(C)	1-2(H)	1-3(H)	1-4(H)	1-5(H)
1-1(C)	0.000	122.856	122.856	122.856	122.856
1-2(H)	122.856	0.000	-6.127	-6.127	-6.127
1-3(H)	122.856	-6.127	0.000	-6.127	-6.127
1-4(H)	122.856	-6.127	-6.127	0.000	-6.127
1-5(H)	122.856	-6.127	-6.127	-6.127	0.000

View the spin-coupling constants.

**Shortcut:** L-click on the  tool button to change the contents of the workspace to display the coupling constants.

You can edit the values in this table. The edited numbers will be used in any spectra that you subsequently calculate.

You can also select numbers (from at least two columns) and plot them in a 2D or 3D bar chart by using the View/Graph menu item. Double clicking on a plot will put a bitmap image of it into the Windows clipboard.

## Saving Molecules and Calculations

If you save the molecular system and its calculated NMR shielding and spin-spin coupling constants, you can retrieve it to compute its NMR spectrum later. HyperNMR offers you a menu command and tool button to save your current molecular system and calculation results.

### To save results:

1. L-click on File.
2. Select the Save As menu item.


This brings up a Save As dialog box.

3. Enter a file name of CH4 in the File Name text box.

An extension, .NMR, is automatically added to the file name, CH4. So HyperNMR saves the current molecular system, methane, with the computed shielding constants and spin-spin coupling constants to CH4.NMR.

4. L-click on OK.

The file is created and the Save As dialog box is dismissed.

*Shortcut:* You can L-click on the  tool button to open the Save As dialog box if you have a molecular system to save.

## Tutorial III: Calculation of NMR Spectra

HyperNMR computes the NMR spectrum quantum mechanically based on the calculated or imported chemical shifts and coupling constants. For details about the quantum mechanical calculation of an NMR spectrum, please see Chapter 6 in this manual.

To simulate an NMR spectrum, some options need to be set up. The following steps show you how to set up the options and how to compute the spectrum.

### Setting up Options

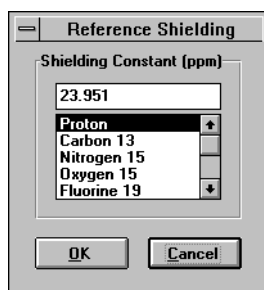
The options for computing the NMR spectrum are: the type of NMR spectrum you are interested in, the shielding constant for

this type of nucleus in the reference molecular system, and the spectrometer frequency. The reference shielding determines the origin for the chemical shift; the chemical shift (in ppm) is obtained by subtracting a computed shielding constant from the reference shielding.

### To set up the reference shielding:

1. L-click on Setup on the HyperNMR menu bar.
2. L-click on Reference Shielding under the Setup menu.

The Reference Shielding dialog box opens:



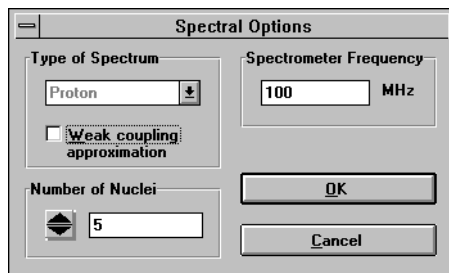
3. L-click on Proton in the Reference Shielding list box.


The Reference Shielding text box shows you the shielding constant of a proton in a reference molecule. The reference shielding of proton here is from the NMR.INI file and was computed for methane based on the current options except that Special Slater exponents for shielding was checked in the Shielding & Coupling Options dialog box. If you want to choose a value from any other reference molecular system, you may simply double-click in the Reference Shielding text box and type in the value that you prefer.

### To set up type of spectrum:

This option specifies which type of NMR spectrum you wish to compute. HyperNMR can calculate a  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{17}\text{O}$ ,  $^{19}\text{F}$  or  $^{31}\text{P}$  spectrum. To do this:

1. L-click on Setup on the HyperNMR menu bar.
2. L-click on Spectrum menu item to open the Spectral Options dialog box:



3. L-click on the arrow button, , at the right side of the text box of Type of Spectrum.

This shows a list of all the types of NMR spectrum that can be simulated with HyperNMR.

4. L-click on Proton.

### To set up the spectrometer frequency:

You can specify the spectrometer frequency to use for computing the spectrum. HyperNMR applies the corresponding magnetic field to the current system and solves the spin-Schrödinger equation to get nuclear-spin energy levels (see Chapter 6 for details).

1. Move the cursor into the Spectrometer Frequency text box.
2. Double-click the left mouse button.

The original value of the spectrometer frequency is highlighted.

3. Enter 100 in this text box.
4. L-click on the OK button in the Spectral Options dialog box.

This closes the Spectral Options dialog box and stores all the changes.

**Important:** The Number of Nuclei text box shows the number of the NMR atoms you have set up for the current system. Do not change this number; if you decrease it you will lose some of the results of your shielding and spin-spin coupling calculation. For details, please see Chapter 5.

## Computing an NMR Spectrum

You have now specified that you would like to calculate the proton NMR spectrum for the current system in a field corresponding to a proton resonance frequency of 100 MHz. Before you can display it, you will need to compute the NMR spectrum.

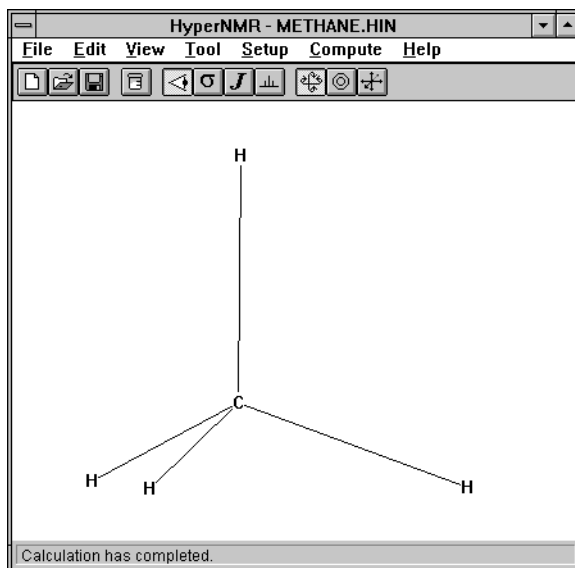
### To compute an NMR spectrum:


1. L-click on the Compute menu on the menu bar of HyperNMR.
2. L-click on the || - Spectrum menu item.

This inactivates the whole menu bar and the tool bar of HyperNMR, except for the Help menu, changes the cursor to an hour-glass, starts the HyperNMR backend program with the icon



and computes the NMR spectrum. Wait until the backend icon disappears (if it appeared on your display) and the HyperNMR status line states that the calculation has completed:



**Note:** The Spectrum menu item under the View menu and the tool button  are active once the spectrum has been calculated.

### **To stop logging to a log file:**

You can now close the log file that was started in Tutorial II. To stop it:

1. L-click on the File menu.
2. L-click on the Stop Log menu item under the File menu.

This stops logging. You can view the log file with a text editor such as Notepad or Write.

## **Viewing NMR Spectra**

After you have computed an NMR spectrum, you can display it. In HyperNMR, spectra can have peaks displayed as separate vertical lines, as an envelope, or both. The envelope is formed by applying a Gaussian or Lorentzian distribution to each peak using line widths that you can adjust.

### **Line View of an NMR spectrum**

Peaks in an NMR spectrum can be shown as separate vertical lines.

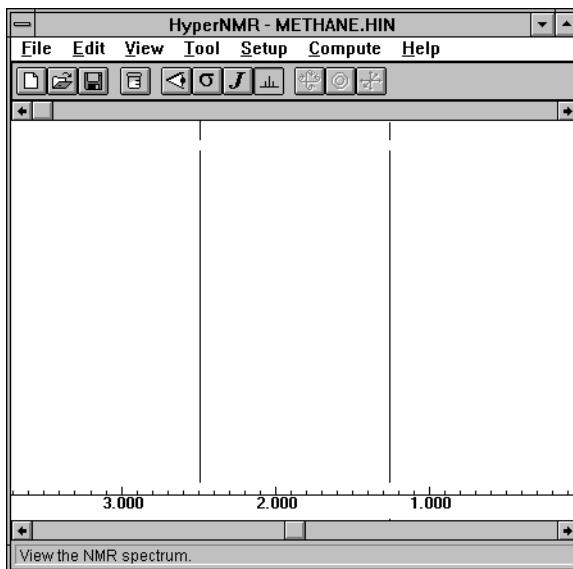
#### **To view peaks of an NMR spectrum as lines:**

1. L-click on View on the HyperNMR menu bar.
2. L-click on the Spectrum menu item.

A fly-out menu opens.

3. Select Lines from this fly-out menu.

This changes the contents of the HyperNMR workspace to show the NMR spectrum in lines.



**Shortcut:** L-click on the  tool button to change the contents of the workspace to display the NMR spectrum.

The spectrum view of the workspace has several components. The scroll bar at the top of the workspace adjusts line widths when the envelope option is active. The short vertical lines at the top of the spectrum indicate peak locations. The corresponding vertical lines at the bottom also show peak locations, but the line heights are proportional to the predicted intensity. The chemical shift scale at the bottom of the spectrum is in parts per million (ppm). The scroll bar at the bottom can be used to pan horizontally.

## Envelope View of an NMR Spectrum

Peaks in an NMR spectrum can be shown as an envelope with or without vertical lines at each peak.

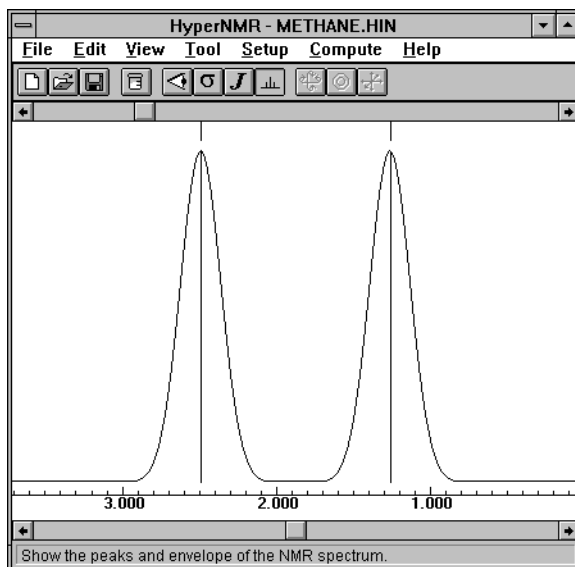
An experimental NMR spectrum does not have peaks that are perfectly narrow lines, due to relaxation and spectrometer limitations. HyperNMR can apply a distribution function to the peaks to artificially simulate this relaxation and display them with an envelope.



### To view an NMR spectrum as an envelope:

1. L-click on View on the HyperNMR menu bar.
2. L-click on the Spectrum menu item.  
This opens a fly-out menu.
3. Select Lines and Envelope from this fly-out menu.  
The initial line width setting is zero, and the spectrum does not appear to change.
4. Slide the scroll button in the top scroll bar about one-third of the way across to increase the line widths.

The proton NMR spectrum of methane changes to:



5. L-click on any one of these two peaks.  
The status line at the bottom of the HyperNMR window shows the chemical shift, frequency, intensity, degeneracy, and line width for this selected peak.

### To change the line width function:

There are two types of line width function available in HyperNMR. These are the Lorentzian distribution function and the Gaussian distribution function (see Chapter 6 for details).

1. L-click on the View menu.
2. L-click on the Line Width menu item under View.

This opens a fly-out menu.

3. Select Gaussian from this fly-out menu.

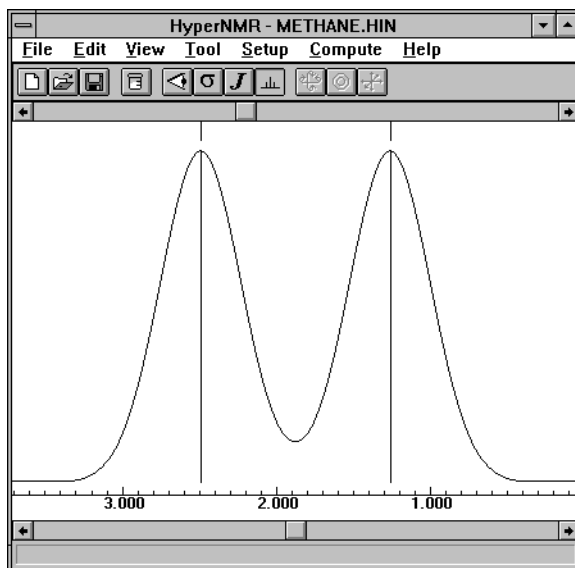
This changes the shape for each peak to that of a Gaussian distribution.

### To change line widths:

The top scroll bar is used to vary line widths. You can choose to vary all line widths at the same time or adjust them individually. The default behavior is to have the range of the scroll bar depend on the level of zoom.

1. L-click to the right of the scroll box on the top scroll bar or on the right hand button.

This increases the line width.

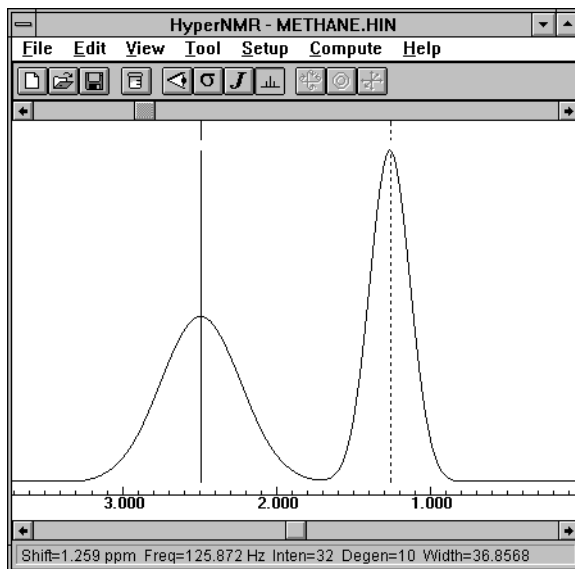


2. L-click on the View menu.
3. L-click on the Line Width menu item under View.

This opens a fly-out menu.

4. Select Multiple Line Widths from this fly-out menu to add a check mark to it.
5. L-click on the right-hand peak to select it. (Selection is discussed in more detail below.)
6. L-click to the left of the scroll box on the top scroll bar.

The line width of the selected peak decreases.



7. Use the View/Line Widths menu to clear the check mark from Multiple Line Widths fly-out menu.

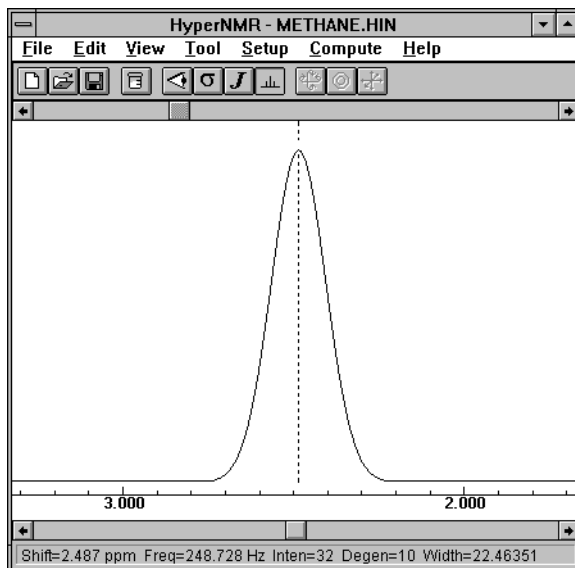
## Investigating Peaks in an NMR Spectrum

You can zoom in on specific peaks in a spectrum, select a peak to find out more about it, integrate peaks to determine how many nuclei they represent, and display the spectrum information in a table.

### To zoom in on a region of the spectrum:

1. L-drag across the horizontal region that you wish to zoom in on.

When you release the mouse button the region of the spectrum that you have selected will be drawn to fill the workspace.



2. L-click to the right of the scroll box on the top scroll bar to change the line width.

The default behavior for the line width scroll bar is to have its range decrease with increased zoom so that you can always get “thin” lines even when you are zoomed in to a very small region of the spectrum.

You can specify a scroll bar range that is independent of zoom by changing the `AbsLineWidth`, `MaxAbsLineWidth` and `MinAbsLineWidth` settings in your `NMR.INI` file (in your Windows directory), as discussed in Chapter 5.

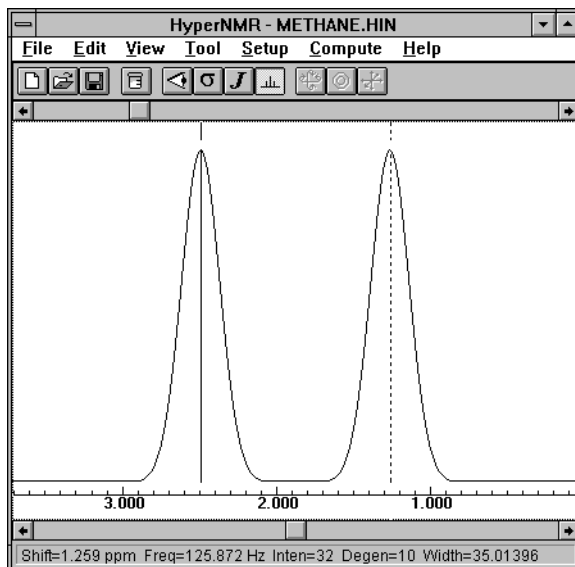
3. R-click in the workspace, L-click to the right of the scroll box on the top scroll bar and then L-click to the left of the scroll box.

This will reset the zoom and line widths to restore to the original view.

### To select and integrate peaks:

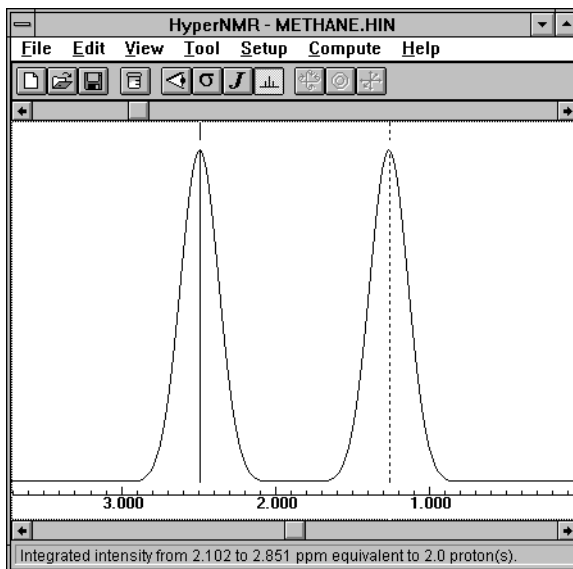
1. L-click on a peak to select it.

The color changes and the status line shows the calculated chemical shift, frequency, intensity and degeneracy of the peak followed by the envelope line width. The line width in HyperNMR is simply an envelope display option that you can adjust.



2. Hold down the Shift key and L-drag across several peaks to integrate the intensities.

The status line shows the number of nuclei (protons in this case) that yield the chosen peaks.



### To view a table of peak information:

1. Choose the Spectral Values menu item in the View menu.

This opens the Spectral Values dialog box, which shows the chemical shift, frequency, intensity, degeneracy and line width for each peak.

	1	2
Chemical Shift	1.259	2.497
Frequency	125.872	248.728
Intensity	32.0	32.0
Degeneracy	10.0	10.0
Line Width	35.014	35.014

Buttons: Plot, Copy, Close

You can use the Copy command button to copy the selected part of the table to the clipboard.

You can also select numbers (from at least two columns) and plot them in a 3D bar chart by using the Plot button. Double-

clicking on a plot will put a bitmap image of it into the Windows clipboard.

2. L-click on Close to close this dialog box.

## Restoring Expert Options

In Tutorial I the Special Slater exponents for shielding option in the Shielding & Coupling Options dialog box was turned off, so that the chemical shift calculated for methane would not be zero with the default reference shielding. You should now turn this option back on so that it will use the recommended setting for any shielding and coupling calculations that you do after completing the tutorials.

### To restore Expert Options:

1. Use the Setup/Shielding & Coupling menu item to open the Shielding & Coupling Options dialog box.
2. Change the Expert Options settings to check Special Slater exponents for shielding and leave Include slow three-center integrals unchecked.

The expert options are now set at the values that we recommend for general use.

## Tutorial IV: Protein NMR

The mixed model implemented in HyperNMR offers a tool for predicting NMR shielding, spin couplings and spectra for subsets of proteins and macromolecular systems. For these large molecular systems usually having hundreds or thousands of atoms, it may not be possible to carry out a full SCF quantum mechanical calculation for the whole system. The mixed model of HyperNMR enables you to select a part of the system to treat quantum mechanically with an electrostatic background generated by the unselected part of the molecular system. This tutorial will show you how to carry out a HyperNMR calculation for a portion of a protein.

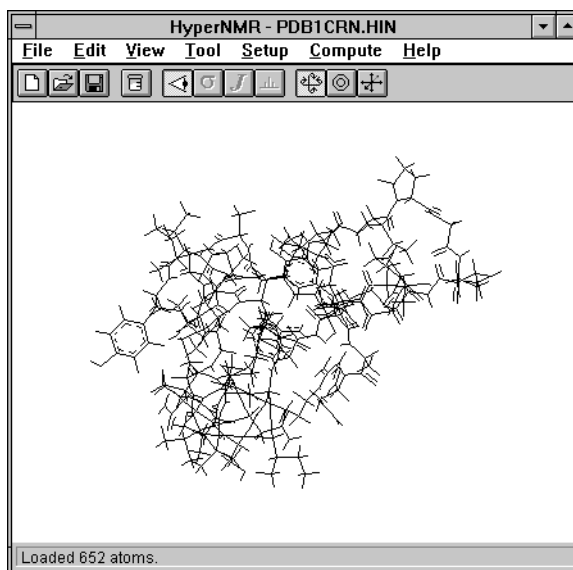
## Making Various Selections

There are several protein molecules located in the SAMPLES directory under the directory where you installed HyperNMR. These protein molecules were converted from Brookhaven Protein Data Bank files to HIN files by HyperChem.

### Opening a protein molecule:

You are expected, at this stage, to know how to open a file within HyperNMR. If you are unsure, please follow Tutorial I at the beginning of this chapter.

1. If you do not have HyperChem, use HyperNMR to open the protein molecule file named PDB1CRN.HIN from your HyperNMR sample directory. The HyperNMR window should look like:



If you have a copy of HyperChem installed on your computer, please start HyperChem and open PDB1CRN.HIN in the HyperChem workspace. You will prefer to make the various selections within HyperChem.



## Selecting quantum atoms with HyperNMR:

If you are using HyperChem skip to the next section. If you are not using HyperChem, you should follow these steps:

1. Change the mouse to be in Pointer Zoom & Translate mode by selecting the Tool/Pointer Zoom & Translate menu item.
2. R-drag the mouse downwards to magnify the molecule.
3. If necessary, L-drag the mouse horizontally, vertically, or diagonally to center the region of interest in the HyperNMR workspace.
4. Open the Rendering dialog box, select Labels for Atoms, and select Number for Atom Labels.
5. Change the mouse to Select mode.
6. Select the atoms to be treated quantum mechanically: 608, 609, 610, 620, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, and 647.

An alternative procedure is to select these atoms from the list box in the Quantum Atom Selection dialog box.

7. Choose the Setup/Shielding & Coupling menu item.

This opens the Shielding & Coupling Options dialog box.

8. L-click on the Quantum Atoms button in the dialog box.

This opens the Quantum Atom Selection dialog box, with all the currently selected atoms added to the Quantum Atoms list box.

9. Check that the correct atoms have been chosen, and then press the OK button in each of the Quantum Atom Selection and Shielding & Coupling Options dialog boxes.

*Note:* HyperNMR treats the atoms you have chosen as “quantum atoms” quantum-mechanically and the other atoms classically, except that non-quantum atoms bonded to quantum atoms are also included in the quantum mechanical SCF calculation with special “capping atom” parameters. By default, HyperNMR has capping atom parameters for carbon, nitrogen, and oxygen. If you choose a boundary that results in other elements as capping atoms, you will see an error message suggesting that you should modify parameter files.

## Selecting quantum atoms with HyperChem:



If you have the PDB1CRN.HIN molecule in the HyperChem workspace, you can do the quantum atom selection within HyperChem. To do this:

1. Choose the Display/Labels menu item in HyperChem.  
This opens the Labels dialog box.
2. Select the Name radio button and close this dialog box.
3. Change the cursor to the selection tool.
4. Use the Select/Select menu item to select residue 45.  
The residue labelled ALA 45 is selected.
5. Choose the Display/Show Selection Only menu item and then press the spacebar.  
The selected residue now fills the HyperChem workspace.
6. Choose the Select/Extend to sp<sup>3</sup> menu item.  
HyperChem will select additional atoms in all directions along bonds until the selection ends at sp<sup>3</sup>-sp<sup>3</sup> bonds or includes all bonds of selected atoms.
7. Choose the Select/Name Selection menu item.  
This opens the Name Selection dialog box in HyperChem.
8. Choose Other, type **quantum-atoms** in the text box and then close this dialog box.

## Selecting NMR atoms with HyperNMR:

If you are using HyperChem skip to the next section. If you are not using HyperChem, you should follow these steps:

1. Choose Rendering from the View menu on the menu bar of HyperNMR.  
This opens the Rendering dialog box.
2. Select Labels for Atoms with Name for Atom Labels and then close this dialog box.  
The workspace will show the protein molecule with all the atom names.

3. Clear the previous selection by R-clicking on an empty part of the HyperNMR workspace.
4. Select the hydrogens named 1HB, 2HB, 3HB, and HA.
5. Open the Shielding & Coupling Options dialog box with the Setup/Shielding & Coupling menu item.
6. Open the NMR Atom Selection dialog box by clicking on the NMR Atoms button in the Shielding & Coupling Options dialog box.
7. The currently selected atoms will appear in the NMR Atoms list box.
8. Confirm that the correct atoms have been chosen, and then close the NMR Atom Selection dialog box.

### Selecting NMR atoms with HyperChem:



If you are using HyperChem, choose the NMR atoms with the following steps:

1. Change the atom labels to atom names via the Display/Labels menu item.  
HyperChem shows all the selected atoms with their names.
2. Make sure the mouse mode is set to selection. Then choose the Select/Atoms menu item. Look at the Select menu to make sure the Multiple Selections menu item is checked.
3. Clear the selection and then select the hydrogens named 1HB, 2HB, 3HB, and HA.
4. Choose the Select/Name Selections menu item.  
This opens the Name Selection dialog box in HyperChem.
5. Choose Other, type *nmr-atoms* in the edit box and then close this dialog box.

### Selecting equivalent atoms with HyperNMR:

You may want HyperNMR to treat the three hydrogens named by 1HB, 2HB, and 3HB as equivalent atoms so that their shielding constants and coupling constants will be averaged.

If you are using HyperChem skip to the next section. If you are not using HyperChem, you should follow these steps:

1. L-click on the Equivalent Atoms button in the Shielding & Coupling Options dialog box.

This opens the Equivalent Atom Selection dialog box.

2. L-click or L-drag to select atoms 635, 636 and 637.
3. Press Add.

HyperNMR copies the selected atoms from the NMR Atom list box to the Equivalent Atom list box.

4. Press Apply to save the equivalent atoms and OK to close the dialog box.

### Selecting equivalent atoms with HyperChem:



If you are using HyperChem, define sets of equivalent atoms with the following steps:


1. Make sure the mouse is in selection mode and the Select/Atoms and Select/Multiple Selections menu items are checked.
2. Deselect all atoms by L-clicking on an empty part of the HyperChem workspace.
3. Select the atoms that you would like to be treated equivalently, in this case the 1HB, 2HB and 3HB hydrogen atoms.
4. Open the Name Selection dialog box in HyperChem via the Select/Name Selections menu item.
5. Click on the Other button and type in **equivalent-atoms-1** in the text box of the Name Selection dialog box in HyperChem and then close this dialog box.

**Note:** If you have more than one set of equivalent atoms, you can repeat steps 2, 3, and 4 and then type in **equivalent-atoms-2**, **equivalent-atoms-3**, and so on. The maximum number of sets of equivalent atoms in HyperNMR is 10 and the maximum number of atoms in each set of equivalent atoms is 20.

## Getting the protein from HyperChem



Once you have created the named selections in HyperChem you need to bring the molecular system into HyperNMR. To do this:

1. L-click on the  button on the HyperNMR tool bar.  
HyperNMR will request the molecular system and its named selections from HyperChem via DDE.

## Computing NMR Shielding, Coupling and Spectrum

At this point you should have the molecular system in the HyperNMR workspace with selections made for quantum atoms, NMR atoms and equivalent atoms. Next you need to specify the options for computing the wave function, shielding constants, and coupling constants and NMR spectrum.

### Setting up NMR shielding and coupling options:

The NMR shielding and spin-spin coupling options can be set up within the Shielding & Coupling Options dialog box. To specify these options:

1. Open the Shielding & Coupling Options dialog box by choosing the Setup/Shielding & Coupling menu item in HyperNMR.
2. Select the TNDO/2, Special Slater exponents for shielding, Shielding constants, and Fermi spin-spin coupling constants buttons within this dialog box.
3. L-click on the SCF button.  
This opens the SCF Wavefunction Options dialog box.
4. Set Charge on molecule to 0, Convergence limit to 0.01, and Maximum iterations to 50.
5. Close the SCF Wavefunction Options and Shielding & Coupling Options dialog boxes by L-clicking on their OK buttons.

### Setting up NMR spectral options:

The spectral options can be specified within the Spectral Options dialog box. To do this:

1. Open the Spectral Options dialog box by choosing the Setup/Spectrum menu item.
2. Select Proton as the Type of Spectrum and enter 100 as the Spectrometer Frequency (in MHz).
3. Close this dialog box by L-clicking on the OK button.

*Note:* You can also change the reference shielding constant for protons in the Reference Shielding dialog box, via the Setup/Reference Shielding menu item.

### Computing NMR shielding, coupling and spectrum:

There is a menu item under the Compute menu of HyperNMR for computing both the NMR shielding & coupling and the NMR spectrum. Once all the shielding, coupling and spectral options have been set up, you can use this menu item to proceed to an NMR spectrum without further intervention. To do this:

1. Select the Compute/Both: I Then II menu item.

The backend program will start to perform the NMR calculations. This calculation takes approximately two minutes on 486/50 PC with 20MB of memory.

*Note:* After you have issued the Compute command you can cancel the calculation by L-clicking on the backend icon of HyperNMR and then selecting Close to cancel the calculation.

## Displaying and Copying NMR Results

You can display and plot NMR results including shielding constants, coupling constants, and NMR spectrum. You also can copy a molecular image and NMR results to the clipboard.

### Displaying, plotting and copying shielding constants:

To display the shielding constant table:

1. L-click on the  tool button.

The HyperNMR workspace will show a table of the shielding constants.

To copy the shielding constants to the clipboard:

1. Select some cells within the shielding constant table by L-dragging the mouse across the cells.
2. Select Edit/Copy on the menu bar of HyperNMR.

This copies the contents of the selected cells, with the corresponding row and column headings, into the Windows clipboard.

To plot the shielding constants:

1. Select all the cells within the shielding constant table by L-dragging the mouse across the cells.
2. Select View/Graph/3D on the menu bar.

A new window with a three-dimensional bar graph of the results in the selected cells is displayed.

To copy the 3D graph of the shielding constants:

1. Double-click within the 3D graph.

A bitmap image of the contents of the bar graph window is placed in the Windows clipboard. You can now paste this image into another Windows application.

2. Close the graph window from its control menu button.

## Displaying and copying coupling constants:

To display the coupling constant table:

1. L-click on the  tool button on the tool bar.

HyperNMR will show the table of coupling constants on the workspace.

To copy the coupling constants to the clipboard:

1. Select the cells within the coupling constant table by L-dragging the mouse.
2. Select Edit/Copy on the menu bar of HyperNMR.

This copies the contents of the selected cells, with the corresponding row and column headings, into the Windows clipboard.

To display the coupling constants graphically:

1. Select the cells within the coupling constant table by L-dragging the mouse.
2. Select View/Graph/3D on the menu bar.

A new window with a three-dimensional bar graph of the results in the selected cells is displayed.

To copy the 3D graph of the coupling constants:

1. Double-click within the 3D graph.

A bitmap image of the contents of the bar graph window is placed in the Windows clipboard. You can now paste this image into another Windows application.

2. Close the graph window from its control menu button.

### Displaying and copying NMR spectrum:

To display the NMR spectrum:

1. L-click on the  tool button on the tool bar.

HyperNMR will show the NMR spectrum in the workspace.

To copy the NMR spectrum:

1. Select Edit/Copy on the menu bar of HyperNMR. HyperNMR will copy the current NMR spectrum from its workspace to the clipboard.

To display NMR spectrum results in a table:

1. Select View/Spectral Values on the menu bar of HyperNMR.

This will open the Spectral Values window and present the details of the NMR spectrum in this window.

To plot the NMR spectral values as 3D bar chart:

1. Select the cells within the Spectral Values window.
2. L-click on the Plot button.

This shows the selected spectral values as a 3D bar chart in the Spectral Graph window.

To copy the 3D bar chart:

1. Double click the mouse within the Spectral Graph window.



This places a copy of the 3D graph in the clipboard.

2. Close this Spectral Graph window from its control menu button.
3. Close the Spectral Values window.

# Chapter 9

## Reference

### Overview

#### What is HyperNMR

HyperNMR is a software package for simulating NMR properties and spectra. You can use it to calculate electronic shielding tensors, chemical shifts (relative values of the shielding constants), nuclear spin-spin coupling constants and NMR spectra. The size of system that can be treated depends on the memory of your computer and your patience. The electronic calculations use the novel semi-empirical methods called Typed Neglect of Differential Overlap, Version 1 (TNDO/1) or Typed Neglect of Differential Overlap, Version 2 (TNDO/2). HyperNMR includes the following functions:

- Open HyperChem HIN files or get molecules from HyperChem via Dynamic Data Exchange (DDE).
- Open MOPAC Z-Matrix files to read in three-dimensional Cartesian or internal coordinates of molecules.
- Save the input and results of NMR calculations. The data saved may include geometries of molecular systems, selections within the molecular system, and the results of calculations. You can re-open these files to display molecules and NMR results.
- Rotate, translate, zoom in on and select molecules.
- Display labels (such as atom symbols, indices, types, names and charges), disks, and spheres for the atoms in molecules and sticks, bonds, and cylinders for the bonds.

- Calculate the shielding tensors, chemical shifts and nuclear spin-spin coupling constants by TNDO/1 or TNDO/2 quantum mechanics methods.
- Compute 1-dimensional NMR spectra based on computed chemical shifts and spin-spin coupling constants or imported (user defined) chemical shifts and coupling constants
- Predict solvation effects on chemical shifts, coupling constants and NMR spectra.
- Graphically display chemical shifts, coupling constants and NMR spectra.
- Copy the images of molecular systems and NMR spectra or tabular results to other Windows applications via the Windows clipboard.

## Starting HyperNMR

### To start HyperNMR:

1. Start Windows if it is not already running.
2. In the Program Manager window, position the cursor over the HyperNMR icon:



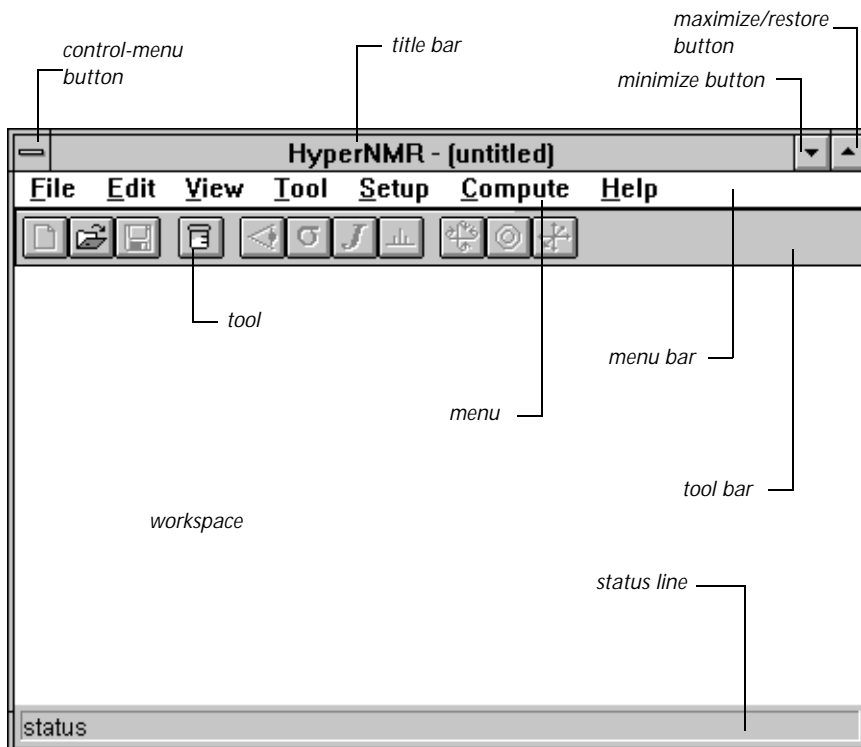
3. Double-click on the HyperNMR icon or click on it and then press the Enter key.

The HyperNMR window opens.

## HyperNMR Window

HyperNMR works with Microsoft Windows, Version 3.1 or later, or Windows 95 or Windows NT for PCs.

When you start HyperNMR, you will see an empty HyperNMR workspace:



If you are using Windows 95 or Windows NT, the appearance of the frame and the arrangement of the control-menu, maximize/restore, and minimize buttons may be slightly different.

The HyperNMR window contains the following items:

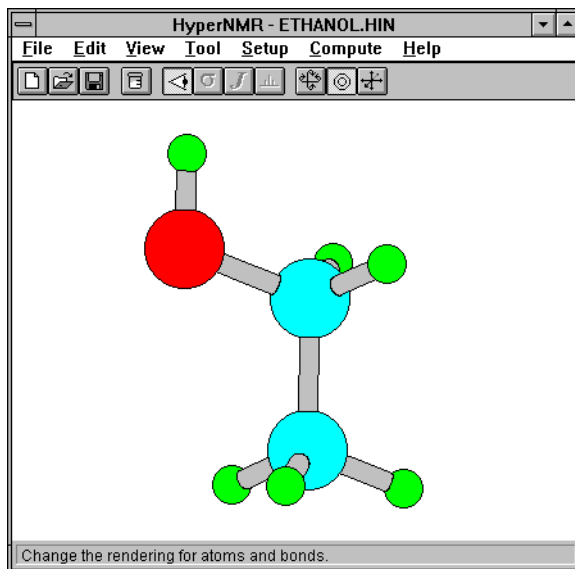
- |                            |  |
|----------------------------|--|
| <b>Control-menu button</b> | Double-click on this button to close HyperNMR. L-click on it to see the control menu that contains commands to let you move, resize or close the window.                               |
| <b>Title bar</b>           | The title bar shows the name of this application, plus the name of the file currently open or "(untitled)" if there is no file open or the current molecular system is from HyperChem. |
| <b>Minimize button</b>     | L-click to reduce the window to an icon.   |
| <b>Maximize button</b>     | L-click to enlarge the window to fill the whole screen. Once selected, this button turns into a  |

	restore button which can be used to restore the window to its original size.
<b>Restore button</b>	L-click to reduce the window from the full-screen size to its original size. This button appears only after the Maximize button has been clicked.
<b>Menu bar</b>	Shows the names of available pull-down menus. L-click on any one of these menus to pull down the corresponding menu.
<b>Tool bar</b>	Short cuts for the menu items.
<b>Workspace</b>	The Workspace shows the molecular system you are working on if you read a structure from a file or got it from HyperChem, or numerical results such as shielding constants or spin-spin coupling constants or an NMR spectrum if you have opened an NMR file or if you computed the NMR shielding, coupling and spectrum.
<b>Status bar</b>	Shows the function of menu items and status of what HyperNMR is doing
<b>Window border</b>	To change the size of the window, place the mouse pointer on the border, L-click and hold down the button, and then drag. Different parts of the border provide different types of size adjustments.

The HyperNMR workspace can be used to view the molecular system, the shielding constants and chemical shifts, the spin-spin coupling constants or the NMR spectrum.

## Molecular System View

You can use the HyperNMR workspace to display the molecular system. Several rendering options are available; the following shows a disks and bonds display:



You can copy the molecular system image to the Windows clipboard with the Edit/Copy menu item, change the rendering with the Rendering dialog box (via the View/Rendering menu item), and use the mouse to select atoms or to rotate, zoom or translate the view of the molecular system.

## Shielding Constant and Chemical Shift View

You can use the HyperNMR workspace to display shielding constants, chemical shifts, and the  $\tau$  (tau) values of the chemical shifts, via the View/Shielding Constants menu item. The HyperNMR workspace will look like:

Index	1-1(C)	1-2(C)	1-4(H)	1-5(H)	1-6(H)	1-7(H)	1-8(H)
Shielding	30.71	-9.621	22.515	22.515	22.515	21.424	21.424
Shift	17.885	58.216	1.436	1.436	1.436	2.527	2.527
Tau	-7.885	-48.216	8.564	8.564	8.564	7.473	7.473

Shielding constant table

Scroll bar

View the shielding constants.

The shielding constant table shows the atom indices, the shielding constants, chemical shifts in ppm, and  $\tau$  values for the selected NMR atoms. The atom indices include two numbers separated by a dash. These two numbers denote the index of the molecule and the atom index within the molecule, respectively. The element symbol follows these two numbers.

You can edit the shielding constants in this table and the dependent values will change automatically. The chemical shifts and  $\tau$  values are calculated from the shielding constants in this table and the reference shielding constants from the Reference Shielding dialog box, so the former values will be updated if you make changes to the latter values. The shielding constants in this table can be edited by clicking on a cell to give it the focus and then typing with the numeric keys (0 - 9), sign keys (+, -) and period key (.) to enter a new number. Most other actions are equivalent to pressing the Enter key, which will accept the change and update the dependent chemical shift and  $\tau$  value. To fix a mistake you must use the Enter key (or equivalent), select the cell and retype the number.

A scroll bar will appear at the bottom of the HyperNMR workspace when the shielding constant table is too wide to fit in the HyperNMR workspace. A vertical scroll bar will appear at the right of the

workspace if the table is too tall to fit. You can use the scroll bars to display any hidden parts of the table.

You can select cells in this table by L-dragging across them. You can copy the selected cells (and the respective headings) to the Windows clipboard with the Edit/Copy menu item and you can plot bar charts of the values in the selected cells with the View/Graph menu item.

## Spin-Spin Coupling Constant View

You can use the HyperNMR workspace to display the spin-spin coupling constants, via the View/Coupling Constants menu item. The HyperNMR workspace will look like:

Coupling	1-1(C)	1-2(C)	1-4(H)	1-5(H)	1-6(H)	1-7(H)	1-8(H)
1-1(C)	0.000	49.522	121.528	121.528	121.528	-5.297	-5.297
1-2(C)	49.528	0.000	-7.463	-7.463	-7.463	130.184	130.184
1-4(H)	121.528	-7.463	0.000	-5.660	-5.660	7.743	7.743
1-5(H)	121.528	-7.463	-5.660	0.000	-5.660	7.743	7.743
1-6(H)	121.528	-7.463	-5.660	-5.660	0.000	7.743	7.743
1-7(H)	-5.297	130.184	7.743	7.743	7.743	0.000	0.072
1-8(H)	-5.297	130.184	7.743	7.743	7.743	0.072	0.000
1-9(H)	7.009	-6.317	0.043	0.043	0.043	0.855	0.855

This table shows an N by N matrix (with N being the number of NMR atoms) of the spin-spin coupling constants for all pairs of NMR atoms. The atom indices include two numbers separated by a dash. These two numbers denote the index of the molecule and the atom index within the molecule, respectively. The element symbol follows these two numbers.

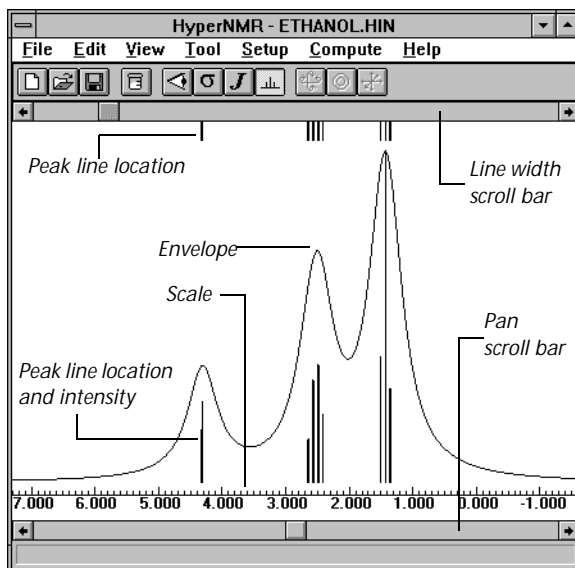
You can edit the values in the same way as you can edit the shielding constant table (see the above section). You can use the scroll bars to display any hidden parts of the table.



You can select cells in this table by L-dragging across them. You can copy the selected cells (and the respective headings) to the Windows clipboard with the Edit/Copy menu item and you can plot bar charts of the values in the selected cells with the View/Graph menu item.

## Spectrum View

You can use the HyperNMR workspace to display the NMR spectrum, via the View/Spectrum menu item. The HyperNMR workspace will look like:



This shows the line and envelope view of a spectrum. The various components of this view of the workspace are:

**Line width scroll bar** Use this scroll bar to change line widths used to create the envelope. If the View/Line Widths/Multiple Line Widths fly-out menu is checked, then this scroll bar will only affect the line widths of the visible peaks or the selected peak if a peak is selected.

By default, the range of this scroll bar varies with zoom so that you can get narrow peaks even when you are zoomed into a very small range of the spectrum. If you prefer to have a range that does not

	vary with zoom, you can set an absolute range (in ppm) in your NMR.INI file. (See “AbsLineWidth” on page 191.)
<b>Peak line location</b>	The short lines coming down from the top of the workspace show the locations of peaks, without showing their intensities. You can set a threshold intensity for showing a peak in your NMR.INI file. (See “MinimalIntensity” on page 190.)
<b>Peak line location and intensity</b>	The variable-length lines coming up from the bottom of the workspace show the locations of peaks, with the line height proportional to the peak intensity. You can set a threshold intensity for showing a peak in your NMR.INI file. (See “MinimalIntensity” on page 190.)
<b>Envelope</b>	The envelope is created by using a Lorentzian or Gaussian distribution for each peak. You can choose the distribution function with the View/Line Widths menu.
<b>Scale</b>	The scale indicates the chemical shift in ppm.
<b>Pan scroll bar</b>	Use this scroll bar to pan to different regions of the spectrum.

You can copy the spectrum image to the Windows clipboard with the Edit/Copy menu item, change the peak display with the View/Spectrum menu item, and use the mouse to select peaks, to zoom into a region of the spectrum, or to integrate intensities for a region of the spectrum. You can L-click on an unselected peak to select it and L-click on a selected peak to clear the selection. You can L-drag across a region of the spectrum to zoom in on it and R-click on the spectrum to reset it to the initial, zoomed-out view. Holding down the Shift key while you L-drag across a region of the spectrum lets you integrate peak intensities in that region.

You can also view the chemical shift, frequency, intensity, degeneracy and line width for each peak in a table with the Spectral Values dialog box. You can open this dialog box with the View/Spectral Values menu item.

## Using the Mouse

You can use a mouse, or equivalent pointing device, for these tasks:

- Translating, rotating and zooming the image of a molecular system.
- Integrating peak intensities for or zooming in on regions of a spectrum.
- Selecting and deselecting atoms, table cells, or spectral peaks.
- Choosing items from the menu bar or tool bar.
- Choosing options in dialog boxes.
- Editing text in dialog boxes.

To use the mouse with menus and dialog boxes, follow standard procedures for Microsoft Windows. You can learn about these procedures from the *Microsoft Windows User's Guide*. The mouse can also be used to manipulate the molecular system in the HyperNMR workspace.

To use the mouse with HyperNMR, first position the cursor on a molecular system or on an icon. Then use one of these procedures, depending on the task:

Procedure	How to do it	What it does
L-click R-click	Press and release the left or right mouse button.	Do this to choose menu items and options in dialog boxes, and to select atoms. L-click and R-click have opposite effects when using tools, for example L-click selects and R-click deselects atoms with the Select tool.
[Shift] + L-click [Ctrl] + L-click	Hold down the [Shift] or [Ctrl] key while you press and release the left or right mouse button	Use this with the select tool to extend a selection to more than one atom. [Shift] + L-click on each atom you would like to select.

Procedure	How to do it	What it does
L-drag R-drag	Press and <i>hold down</i> a mouse button (left or right), and then move the mouse to perform an operation; release the mouse button to complete the operation.	Use L-drag to choose menu items and text, to apply rotation or translation or to zoom in on a spectrum. Use R-drag to do in-plane rotation and to change the magnification of a molecular system.
[Shift] + L-drag	Hold down the [Shift] key while you L-drag with the mouse.	Use this to integrate peaks of a spectrum.
Double-click	Press and release the left mouse button rapidly.	This is usually a shortcut for a common operation and is also used for copying bar graphs to the Windows clipboard. In some dialog boxes, double-click on the options instead of L-clicking on the option and then on OK.

## Using the Keyboard

All menu items and dialog boxes in HyperNMR have accelerators for keyboard navigation, denoted by underlined letters. See the *Microsoft Windows User's Guide* to learn about how to use keyboard accelerators. HyperNMR has additional keyboard shortcuts for manipulating molecular systems in HyperNMR. For details, see "Keyboard Equivalents for Mouse Actions," on page 162.

## HyperNMR Menus

You can select HyperNMR commands and settings, which are on pull-down menus on the menu bar. When you highlight a menu item, a simple description of the item appears on the status line. Some menu items may be inactive. For instance, the File/New

menu item may be grayed when HyperNMR doesn't hold a molecular system, shielding constants, spin-spin coupling constants or NMR spectrum.

Use menu items to set HyperNMR conditions, to set up the options for HyperNMR calculations and to compute NMR shielding, coupling and spectra.

## HyperNMR Coordinate Systems

HyperNMR has two coordinate systems for displaying molecular systems, that is, the Viewer's Coordinate System and the Molecular Coordinate System. Each has its x, y, and z Cartesian axes.

### Viewer's Coordinate System:

The Viewer's Coordinate System defines the positions of atoms relative to the Viewer. The Viewer's Coordinate System is fixed by the screen geometry and its x axis is horizontal in the screen, the y axis is vertical in the screen, and the z axis is perpendicular to the plane of the screen.

The molecular manipulation is done in the Viewer's Coordinate System. Hence you can move, translate, and rotate molecules without changing their atomic coordinates.

When you open a HIN file or get a molecular system from HyperChem, HyperNMR will load the system with the Viewer's Coordinate System saved in the HIN file or obtained from HyperChem. When you save a molecular system via File/Save or File/Save As, HyperNMR will save the Viewer's Coordinate System in the file, so that you will have the same view if you reopen the file.

### Molecular Coordinate System:

The atomic coordinates of a molecule are defined in the Molecular Coordinate System and cannot be changed within HyperNMR. The Molecular Coordinate System is used for computing NMR properties, such as chemical shifts and spin-spin coupling constants. You can inspect the x, y, and z coordinates of an atom by selecting it. When an atom is selected, its x, y, and z coordinates, numerical index, element symbol, and atom type are shown on the status line.

# Menus

This section gives the commands and descriptions of the menus appearing on the menu bar of HyperNMR.

## File Menu

The File menu is used for reading and writing files, for getting information from HyperChem, and for setting up preferred colors. It contains the following items:

- |                                 |   |
|---------------------------------|---|
| <b><u>N</u>ew</b>               | Use New to clear the workspace. If selections and/or calculations have been performed, you will be offered the chance to save the system to a file before the workspace is cleared.   |
| <b><u>O</u>pen...</b>           | Use Open to read in a HyperNMR NMR file, a HyperChem HIN file, or a MOPAC Z-Matrix file. The HyperNMR NMR files and HyperChem HIN files may contain the view of the molecular system, the molecular indices, residue indices, atom indices, atom names, atom symbols, atom types, atom charges, atom coordinates, and the connectivity. This menu item will bring up the Open File dialog box. For details, see "Open File Dialog Box" on page 163. |
| <b><u>S</u>ave</b>              | Use the Save menu item to save a copy of the current system and results into an NMR file using the current HyperNMR file name. If there is no current file, or the current file name does not have the ".NMR" extension, then this item is the same as the Save As menu item.   |
| <b>Save <u>A</u>s...</b>        | Use Save As to save the current system into a named file. This menu item opens the Save As dialog box (see page 164) for you to use in selecting a file name.   |
| <b><u>G</u>et<br/>HyperChem</b> | Use Get HyperChem to load into HyperNMR a copy of the current system from a currently running HyperChem. Though the atoms, coordinates and viewing angle are properly loaded into HyperNMR, any non-default colors assigned to atoms by   |

- HyperChem will be lost when the molecule is loaded by HyperNMR.
- Start Log...** Use Start Log... to save the details of an NMR calculation into a log file. This item opens the Start Log dialog box for you to choose a file name. The default extension for log files is “.LOG”. For details, see “Start Log Dialog Box” on page 165.
- Stop Log** Use Stop Log to end a log file which has been started previously and may contain all the details of NMR calculations which have happened between using Start Log and Stop Log.
- Preferences...** Use this menu item to change the preferences of displaying molecular systems and NMR spectra in the HyperNMR workspace. This menu item opens the Preferences dialog box and lets you choose various display colors. For details, see “Preferences Dialog Box” on page 165.
- Exit** Use Exit to close HyperNMR. If you have done some calculations since the last time a file was saved, then you will be offered the chance to save changes.

## Edit Menu

The Edit menu is used for copying images or data to the Windows clipboard and for calculating atom types.

- Copy** Use this menu item to place a copy of the molecular image, selected cells from the shielding or coupling constant tables, or a copy of the NMR spectrum into the Windows clipboard, depending on which view is currently in the workspace. Molecular images are copied in bitmap format. A bitmap image will print with screen resolution only. Selected cells in the shielding constant table or the coupling constant table are copied to the clipboard as tab-separated text. NMR spectral images are copied to the clipboard in Windows metafile format. A metafile image will print with printer resolution.
- Replace By...** Use this menu item to replace all the selected cells in a shielding/coupling constant table by the value

specified in the Replace Selection dialog box (see page 167).

**Calculate Atom Types** Use this menu item to calculate atom types for computing the NMR shielding and spin coupling with the TNDO/1 or TNDO/2 method.

## View Menu

Use the items on the View menu to change the type of information displayed in the HyperNMR workspace.

**M**olecule Use this menu item to display molecular systems in the HyperNMR workspace.

**R**endering... Use this menu item to display the Rendering dialog box (see page 167) which you use to specify the rendering method used for atoms and bonds. This menu item is available only when a molecular system is shown in the HyperNMR workspace.

**S**hielding Constants Use this menu item to present a table which contains shielding constants (in ppm), chemical shifts (the relative values of shielding constants in the current system to the reference shielding values), and tau values. This menu item is available only when shielding constants are present in HyperNMR.

**C**oupling Constants Use this menu item to display a table which contains nuclear spin-spin coupling constants (in Hz). This menu item is available only when nuclear spin-spin coupling constants are present in HyperNMR.

**G**raph Use this menu item to plot a two or three dimensional bar graph of the selected shielding constants or coupling constants in the NMR Graph dialog box. This menu item is available only when the shielding constant table or coupling constant table is in the workspace. This menu item uses a fly-out menu:

**2D bar** Use this to choose a 2D bar graph for the NMR Graph dialog box.



<b>3D bar</b>	Use this to choose a 3D bar graph for the NMR Graph dialog box.
<b>Spectrum</b>	Use Spectrum to display an NMR spectrum in the HyperNMR workspace. This menu item is available only when an NMR spectrum is present in HyperNMR. This menu item uses a fly-out menu:
<b>L<u>ines</u></b>	Show the peaks as lines only.
<b>E<u>velope</u></b>	Show the peaks of an NMR spectrum with simulated broadening of peaks to form an envelope.
<b>L<u>ines and envelope</u></b>	Show the peaks as lines and as an envelope.
<b>Shielding Tensor....</b>	Use Shielding Tensor to display the Shielding Tensor dialog box (see page 168), which contains the details of the shielding tensors including the $\sigma_{xx}$ , $\sigma_{xy}$ , $\sigma_{yx}$ , $\sigma_{yy}$ , $\sigma_{yz}$ , $\sigma_{zx}$ , $\sigma_{zy}$ , and $\sigma_{zz}$ components of the electronic shielding tensor, the eigenvalues of the tensor, the isotropic shielding and the shielding anisotropy. This menu item is available only when the NMR shielding constant table is displayed in the workspace.
<b>Spectral Values....</b>	Use Spectral Values to display the Spectral Values dialog box (see page 169), which contains the details of the spectrum including the chemical shifts (in ppm), frequency (in MHz), intensity, degeneracy, and current line width for each peak. This menu item is available only when the NMR spectrum is displayed in the workspace.
<b>L<u>ine Widths</u></b>	Use this menu item to change the type of line width, such as multiple line widths, Lorentzian line width, and Gaussian line width. This menu item is available only when an NMR spectrum is displayed in the workspace and the spectrum is displayed in envelope or line with envelope. This menu item uses a fly-out menu:
<b>M<u>ultiple line widths</u></b>	Allow line widths to be set separately for each peak. When this is off all peaks have the same line width.

- L**orentzian     Display the NMR spectrum envelope using a Lorentzian distribution for each peak.
- G**aussian       Show the NMR spectrum envelope using a Gaussian distribution for each peak.

## Tool Menu

The Tool menu can be used to change the function of the mouse or equivalent pointing device. The items in this menu are available only when a molecular system is displayed in the HyperNMR workspace. The menu items are:

- S**cale To **E**it     Use this menu item to rescale the view so that the entire system fits in the window. This menu item rescales and centers the image. The keyboard spacebar is a shortcut for this menu item.
- P**ointer **R**otate    When this menu item is checked, the mouse can be used to rotate the molecule. See “Using the mouse to rotate the molecule” on page 161 for details.
- P**ointer **S**elect    When this menu item is checked, the mouse can be used to select atoms. See “Using the mouse for selection” on page 161 for details.
- P**ointer **Z**oom & **T**ranslate    When this item is checked, the mouse can be used to adjust display size and position of the molecule. See “Using the mouse for zooming and translation” on page 161 for details.

## Setup Menu

HyperNMR computes shielding constants, spin-spin coupling constants and NMR spectra. An NMR spectrum can be predicted from shielding constants and spin-spin coupling constants that are computed, imported, or user-defined. There are several options that you must specify before computing and simulating, and these can be set through the following menus and associated dialog boxes:

- S**hielding & **C**oupling...    Use this menu item to bring up the NMR Shielding & Coupling Options dialog box (see “Shielding & Coupling Options Dialog Box” on page 171 for details) which you can use to set up the options for

computing shielding (chemical shifts) and nuclear spin-spin coupling constants. This menu item is available only when a molecular system is present in HyperNMR and the number of nuclei shown in the Spectral Options dialog box (see below) is less than or equal to the number of atoms in the molecular system.

- Spectrum...** Use this menu item to display the Spectral Options dialog box (see “Spectral Options Dialog Box” on page 182 for details) which you can use to specify the options for calculating an NMR spectrum.
- Reference Shielding...** Use this menu item to open the Reference Shielding dialog box (see “Reference Shielding Dialog Box” on page 184 for details) which you can use to specify the reference shielding constant for each of  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{15}\text{O}$ ,  $^{19}\text{F}$  and  $^{31}\text{P}$ . The calculated shielding is subtracted from the reference shielding to give the chemical shift.

## **Compute Menu**

The Compute menu is used to compute shielding constants, spin-spin coupling constants and NMR spectra. The menu items are listed below:

- I - Shielding & Coupling** Use this menu item to compute the shielding constants and/or coupling constants for the current system. This menu item is available only when a molecular system is present in HyperNMR and the number of nuclei shown in the Spectral Options dialog box is less than or equal to the number of atoms.
- II - Spectrum** Use this menu item to compute the NMR spectrum for the given set of chemical shifts and coupling constants. This menu item is available only when HyperNMR has shielding constants and spin-spin coupling constants that have been computed, imported or entered by you.
- Both: I Then II** Use this menu item to compute the shielding constants, spin-spin coupling constants and the spectrum for the current molecular system. This menu

item is not available if there is no molecular system or if the number of nuclei in the Spectral Options dialog box is greater than the number of atoms in the current molecular system.

## Help Menu

The Help menu is used to access on-line help and the About dialog box. It has the following menu items:



<b>C</b> ontents	Lists the topics available with on-line help.
<b>U</b> sing Help	Use this menu item to view instructions for using on-line help. Start here if you are unfamiliar with using Windows on-line help.
<b>A</b> bout HyperNMR...	Displays the About dialog box, which gives version and copyright information.










## Tools

This section shows the commands and descriptions of the tool buttons on the HyperNMR tool bar. These buttons are shortcuts for menu operations: pressing one of these buttons is equivalent to selecting its corresponding menu item.

### Tool Buttons

Here are the buttons on the tool bar and the menu items corresponding to each. See the descriptions above of the corresponding menu commands for further details.

Button	Corresponding Menu Command:	Action:
	File/New	Clear the workspace.
	File/Open...	Display the Open File dialog box.

Button	Corresponding Menu Command:	Action:
	File/Save	Save the molecular system and NMR results using the current HyperNMR file name.
	File/Get HyperChem	Get the molecular system from a currently running HyperChem session.
	View/Molecule	Display the molecular system in the HyperNMR workspace. When the molecular system is displayed this button will appear depressed.
	View/Shielding Constants	Display the shielding constant table in the HyperNMR workspace. When the shielding constants are displayed in the workspace this button will appear depressed.
	View/Coupling Constants	Show the spin coupling constant table in the HyperNMR workspace. When the coupling constants are displayed in the workspace this button will appear depressed.
	View/Spectra	Show the NMR spectrum in the HyperNMR workspace. When the spectrum is displayed in the workspace this button will appear depressed.
	Window/Pointer Rotate	Use the mouse to rotate the view. When this option is active, the button will appear depressed.
	Window/Pointer Select	Use the mouse to select and deselect atoms. When this option is active, the button will appear depressed.
	Window/Pointer Zoom & Translate	Use the mouse to magnify and translate the view. When this option is active, the button will appear depressed.

## Manipulating Molecules with the Mouse

You can rotate, translate, and zoom the view of the molecular system, and select and deselect atoms by using the mouse.

### Using the mouse to rotate the molecule

Check the Pointer Rotate menu item if you want to use the mouse to rotate the molecule. The following mouse actions are available:

- L-drag for x-y rotation. While the cursor is in the workspace, drag the mouse horizontally to rotate about the y-axis. Drag vertically to rotate about the x-axis. Drag diagonally to rotate about both.
- R-drag horizontally for z rotation. Drag horizontally to rotate the molecule in the plane of the screen. Vertical cursor motion is ignored.

### Using the mouse for selection

Check the Pointer Select menu item if you want to use the mouse for selecting atoms. The following mouse actions are available:

- L-click on an atom to select that atom. Holding down the Control or Shift key at the same time will extend the previous selection to include the new atom. Without the Control or Shift key, any currently selected atoms are unselected first.
- R-click on an atom to deselect it.
- L-click on open space to select all atoms.
- R-click on open space to deselect all atoms.

### Using the mouse for zooming and translation

Check the Pointer Zoom & Translate menu item if you want to use the mouse to adjust the magnification and position of the molecule. The following mouse actions are available:

- L-drag to translate the molecule in the plane of the screen. L-drag in the direction you wish to move the molecule.
- R-drag downwards to increase magnification of the molecule; R-drag upwards to decrease the magnification.

## Keyboard Equivalents for Mouse Actions

The mouse operations for changing the view can also be performed via the keyboard by using the arrow ([*↔*], [*←*], [*→*], [*↕*]), [Home], [End], [PgUp], and [PgDn] keys. The following table summarizes the action of each key, possibly modified by either the [Shift] key or the [Control] key. For example [Shift] + [*↔*] will translate the system to the right.

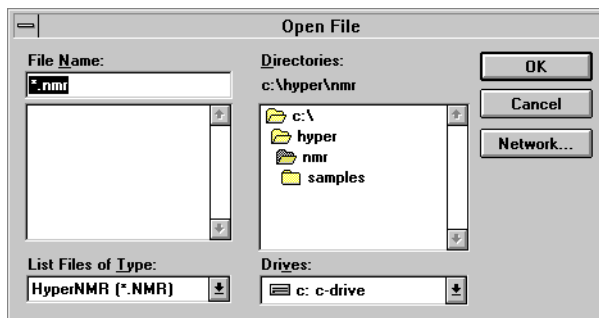
	Key	[Shift] + Key	[Ctrl] + Key
<b>Right arrow</b> ( <i>↔</i> )	Y-rotate right	Translate right	Z-rotate clockwise
<b>Left arrow</b> ( <i>←</i> )	Y-rotate left	Translate left	Z-rotate counter-clockwise
<b>Up arrow</b> ( <i>↕</i> )	X-rotate up	Translate up	Reduce size
<b>Down arrow</b> ( <i>↕</i> )	X-rotate down	Translate down	Enlarge
[Home]	Z-rotate clockwise	No effect	No effect
[End]	Z-rotate counter-clockwise	No effect	No effect
[PgUp]	Reduce size	No effect	No effect
[PgDn]	Enlarge	No effect	No effect

## Dialog Boxes

This section describes the dialog boxes available in HyperNMR, in the order of the menu items that invoke them.

## Open File Dialog Box

Use this dialog box to load a molecular system from a HyperNMR NMR file, a HyperChem HIN file or a MOPAC Z-Matrix file to the HyperNMR workspace. The dialog box provides several ways to specify a file name.



This dialog box has the following items:

- |                                  |   |
|----------------------------------|---|
| <b>File <u>N</u>ame</b>          | There are two boxes under File Name; the first one is the File Name text box and the second is the File Name list box. You can enter the path and file name in the first text box and L-click on the OK button to open this file. You can also use this text box to search for file names under the current directory (see below). The second text box contains the names of available files in the current directory (see below). You can scroll through the list if the complete list does not fit in this box. If you want to open a file from the File Name list box, you just L-click on it and L-click on the OK button. Alternatively, you can open a file from the File Name list box by double-clicking on it. |
| <b><u>D</u>irectories</b>        | This reports all the parent directories and the directories within the current directory. Double-clicking on any one of these directories will change that directory to be the current directory and the files in that directory will then be listed in the File Name list box.   |
| <b>List Files of <u>T</u>ype</b> | Choose one of the file name extensions from the drop-down list to limit the types of files that appear in the File Name list box. The file types sup-   |



ported are HyperNMR files, HyperChem HIN files, and MOPAC Z-Matrix files.

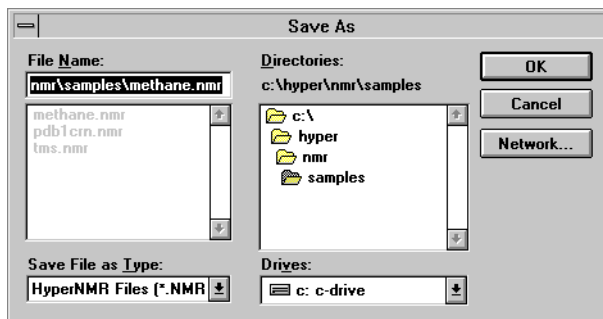
- Drives** Shows the current drive. You can change the drive by choosing from the drop-down list.
- OK** Use this button to load the contents of the file appearing in the File Name text box into HyperNMR.
- Cancel** Close the Open File dialog box without reading a file.

*Important:* If you choose an inappropriate file type for a file, errors may occur when you open the file.

## Save As Dialog Box

You may use this dialog box to save the molecular system with any named selections or calculation results.

The Save As dialog box looks like:

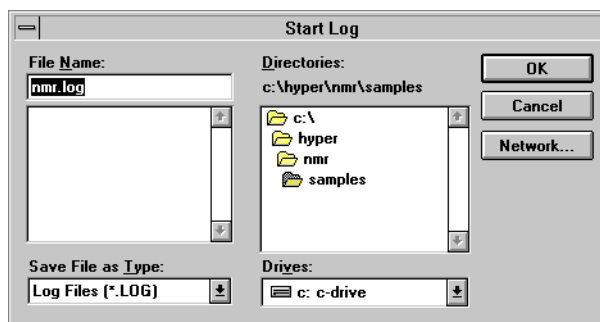


The components of this dialog box are the same as for the Open File dialog box (see above), except that here the File Name list box shows the existing files in gray. You can still choose any one of these file names from the list box by L-click on it. HyperNMR will ask you whether or not you want to overwrite the existing file. You can choose OK to overwrite the existing file or Cancel to not save the file.

Only HyperNMR files can be saved via this dialog box. If you do not specify an extension, the extension “.NMR” will be added automatically. You should avoid saving NMR files with a different extension, to simplify re-opening them at a later time.

## Start Log Dialog Box

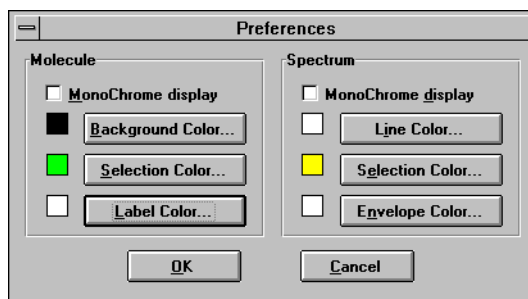
The Start Log dialog box is used to choose a file name for storing the details of computations performed by the HyperNMR backend program. Log files contain starting conditions plus results. The amount of information stored in a log file depends on the “print level” setting; for more information, see “Log Files” on page 210. The components of this dialog box are identical to those for the Open File and Save As dialog boxes (described above):



If the file you entered in the File Name text box already exists, HyperNMR asks if you want to replace the file or append to it. The log file will only contain information after you have computed NMR shielding constants, spin-spin coupling constants or an NMR spectrum.

## Preferences Dialog Box

When you choose File/Preferences, the Preferences dialog box appears. You can use this dialog box to set up the colors for displaying molecular systems and NMR spectra.



The items in this dialog box are:

**Molecule** Set up the colors for displaying molecular systems in HyperNMR.

**MonoChrome Display** Check this box to display the molecular system as black labels and selections on a white background.

**Background Color...** Use this button to set up the background color for displaying molecular systems. It opens the standard Windows Color dialog box for choosing or defining a color. The colored box next to the button shows the current color.

**Selection Color...** Use this button to set up the selection color for displaying molecular systems. It opens the standard Windows Color dialog box for choosing or defining a color. The colored box next to the button shows the current color.

**Label Color...** Use this button to set up the label color for displaying molecular systems. It opens the standard Windows Color dialog box for choosing or defining a color. The colored box next to the button shows the current color.

**Spectrum** Set up the colors for displaying the NMR spectra in HyperNMR.

**MonoChrome Display** Check this box to display the molecular system as black lines and envelope on a white background.

**Line Color...** Use this button to set up the peak line color for displaying an NMR spectrum. It opens the standard Windows Color dialog box for choosing or defining a color. The colored box next to the button shows the current color.

**Selection Color...** Use this button to set up the peak selection color for displaying an NMR spectrum. It opens the standard Windows Color dialog box for choosing or defining a color. The

colored box next to the button shows the current color.

**Envelope  
Color...**

Use this button to set up the envelope color for displaying an NMR spectrum. It opens the standard Windows Color dialog box for choosing or defining a color. The colored box next to the button shows the current color.

**OK**

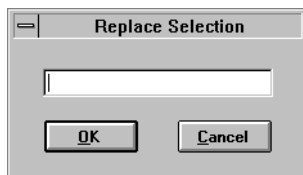
Press this button to accept all changes made in this dialog box and dismiss the dialog box.

**Cancel**

Press this button to dismiss this dialog box and ignore any changes made in the dialog box.

## Replace Selection Dialog Box

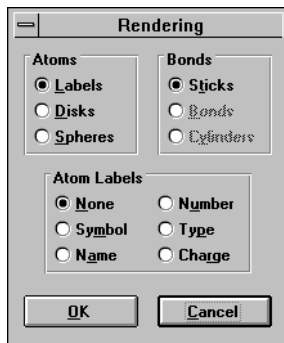
The Replace Selection dialog box looks like:



Use this dialog box to replace all the selected cells in a shielding/coupling constant table.

## Rendering Dialog Box

The Rendering dialog box looks like:



Use this dialog box to select the rendering method for atoms and bonds. It has the following items:

- |                      |  |
|----------------------|--|
| <b>Atoms</b>         | Choose one of the available rendering methods for atoms in this section. Labels displays lines along bonds with a selected label (see below) at each atom. Disks displays unshaded spheres, and Spheres displays shaded spheres.   |
| <b>Bonds</b>         | Choose one of the available renderings for bonds in this section. Sticks displays only lines along bonds, Bonds displays unshaded cylinders for bonds, and Cylinders displays shaded cylinders.  |
| <b>Labels</b>        | Choose one of the labels for atoms in this section. None shows no labels for atoms, Symbol shows the element symbol of each atom, Name shows atom names (if they have been read in from a HIN or NMR file), Number shows sequential indices within the molecule, Type shows atom types, and Charge shows partial atomic charges. This choice is available only if Labels has been selected in the Atoms group. |
| <b><u>O</u>K</b>     | Press this button to dismiss the dialog box and apply the selected rendering settings.   |
| <b><u>C</u>ancel</b> | Press this button to dismiss the dialog box without changing the rendering.  |

## Shielding Tensor Dialog Box

The Shielding Tensor dialog box can be opened with the View/ Shielding Tensor menu item when a shielding constant table is displayed in the workspace.

Shielding Tensor				
	1-1(C)			
	Sigma(x)	Sigma(y)	Sigma(z)	Sigma(x)
Sigma(x)	49.457	0.0	0.0	15.428
Sigma(y)	0.001	49.457	0.002	0.0
Sigma(z)	0.0	-0.002	49.457	0.0
Eigenvalues	49.455	49.457	49.459	15.428
Isotropic Shld	49.457			22.078
Anisotropy	0.003			19.948

+

Plot Copy Close

This dialog box displays all the components of a tensor, the eigenvalues of the tensor, isotropic shielding and anisotropy for each NMR atom. You can select cells by L-dragging across them and you can use the scroll bar to view portions of the table that do not fit in the dialog box. The available commands are:

**Plot** Use this to open the NMR Graph dialog box which will display the values of the selected cells as a 3D bar chart. Cells in at least two columns must be selected.

**Copy** Use this to copy the selected cells, and the associated headings, to the Windows clipboard as tab-separated text.

**Close** Closes the dialog box.

## Spectral Values Dialog Box

The Spectral Values dialog box can be opened with the View/Spectral Values menu item when a spectrum is displayed in the workspace.

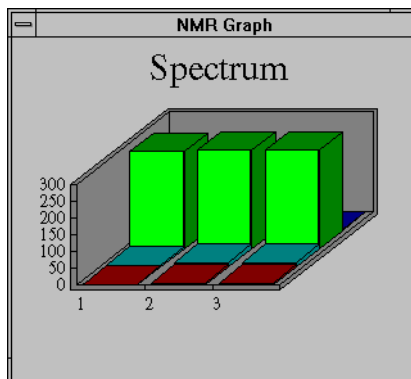
Spectral Values		
	1	2
Chemical Shift	1.259	2.487
Frequency	125.872	248.728
Intensity	32.0	32.0
Degeneracy	10.0	10.0
Line Width	35.014	35.014

This dialog box displays the chemical shift (in ppm), frequency (in MHz), intensity, degeneracy and line width for each peak in the current NMR spectrum. You can select cells by L-dragging across them and you can use the scroll bar to view portions of the table that do not fit in the dialog box. The available commands are:

- Plot**            Use this to open the NMR Graph dialog box which will display the values of the selected cells as a 3D bar chart. Cells in at least two columns must be selected.
- Copy**            Use this to copy the selected cells, and the associated headings, to the Windows clipboard as tab-separated text.
- Close**            Closes the dialog box.

## NMR Graph Dialog Box

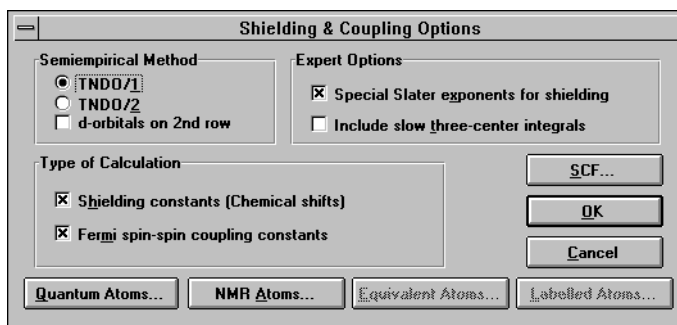
The NMR Graph dialog box is used to display tabular data as 2D or 3D bar charts. It can be opened from the View/Graph menu item when the workspace shows either shielding or coupling constants, or from the Plot button in the Spectral Values dialog box. A 3D bar chart is shown:



You can double-click on the bar chart image to copy a bitmap of it to the Windows clipboard.

## Shielding & Coupling Options Dialog Box

When you choose Setup/Shielding & Coupling, the Shielding & Coupling Options dialog box appears. You can use this dialog box to set up the options for computing the shielding constants and spin-spin coupling constants for the current molecular system. This dialog box looks like:



This dialog box contains the following items:

**Semiempirical Method** Specify a semi-empirical quantum mechanical method for computing the shielding constants and/or coupling constants.

**TNDO/1** Check this option to use the Typed Neglect of Differential Overlap semi-empirical



	method, Version 1, for a shielding constant and coupling constant calculation.
TNDO/ <u>2</u>	Check this option to use the Typed Neglect of Differential Overlap semi-empirical method, Version 2, for a shielding constant and coupling constant calculation. This is the method used for parameterization and should be used instead of TNDO/1 unless you have developed your own TNDO/1 parameters.
d-orbitals on <u>2nd row</u>	Check this option to include d-orbitals for all second row elements in the atomic basis set in the current molecular system. This option is available only with TNDO/1.
Expert Options	This group allows you to set some specialized options for the calculation of shielding constants and spin-spin coupling constants. See Chapter 6 for details.
Special Slater <u>exponents for shielding</u>	Check this option to have HyperNMR modify the Slater exponents of the atomic orbitals according to the electronic charge distribution after its SCF calculation and then compute the shielding constants. Otherwise the Slater exponents used to compute the shielding constants are the same as those used for the SCF calculation. <b>This option should always be active</b> , since the semiempirical parameters were developed using it. It does not significantly increase computation time.
Include slow <u>three-center integrals</u>	Check this option to have HyperNMR compute the contribution to the shielding constants from the three-center integrals and the non-NDO off-diagonal elements of the density matrix. Using this option can increase the time taken for the shielding and coupling calculation by an order of magnitude, so its use is not recommended.
Type of Calculation	Use this section to specify whether you want to calculate shielding constants, coupling constants or

both. At least one of these options must be checked.

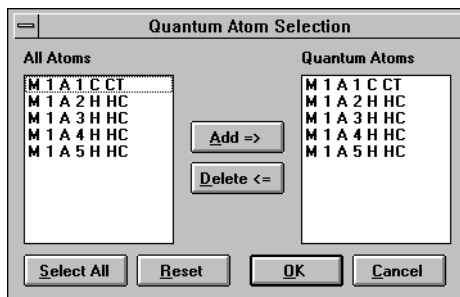
<b>Shielding constants (Chemical shifts)</b>	Check this box to compute the shielding constants.
<b>Fermi spin-spin coupling constants</b>	Check this box to compute the nuclear spin-spin coupling constants.
<b>Quantum Atoms...</b>	Use this button to open the Quantum Atom Selection dialog box (see below) for specifying the quantum atoms.
<b>NMR Atoms...</b>	Use this button to open the NMR Atom Selection dialog box (see below) for specifying the NMR atoms.
<b>Equivalent Atoms...</b>	Use this button to open the Equivalent Atom Selection dialog box (see below) for specifying sets of equivalent atoms.
<b>Labelled Atoms...</b>	Use this button to open the Labelled Atom Selection dialog box (see below) for specifying sets of labelled atoms
<b>SCF...</b>	Use this button brings up the SCF Wavefunction Options dialog box (see below) to specify the conditions for the SCF calculation.
<b>OK</b>	Press this button to accept all the changes and dismiss this dialog box.
<b>Cancel</b>	Press this button to ignore any changes and dismiss this dialog box.

*Note:* This dialog box may not be accessible if there is no molecular system in HyperNMR or if the number of nuclei in the Spectral Options dialog box (see below) is greater than the actual number of atoms in the current molecular system.

## Quantum Atom Selection Dialog Box

This dialog box lets you select the atoms which will be treated quantum mechanically when computing the SCF wave function. The rest of the atoms in the molecular system will be treated clas-

sically, except that atoms that are bonded to quantum atoms will be treated as capping atoms. The quantum atoms and resulting capping atoms are limited to the elements hydrogen through argon. If you do not specify any quantum atoms, the whole molecular system will be treated quantum mechanically.



This dialog box contains the following items:

- |                      |  |
|----------------------|--|
| <b>All Atoms</b>     | All Atoms lists all the atoms presented in the current molecular system. Each line corresponds to one atom and includes the following information: M # stands for the atom belonging to molecule #, A ## means the atom numbered ## within the molecule, and the last two “words” show the atomic symbol and the TNDO atom type. |
| <b>Quantum Atoms</b> | This lists all atoms that you have chosen to be quantum atoms, in the same format as for All Atoms.  |
| <b>Add =&gt;</b>     | Use this button to add the selected (highlighted) atoms from the All Atoms list to the Quantum Atoms list.   |
| <b>Delete &lt;=</b>  | Use this button to delete the selected set of atoms from the Quantum Atoms list.   |
| <b>Select All</b>    | Press this button to select (highlight) all the atoms listed in the All Atoms list box.  |
| <b>Reset</b>         | Delete all the atoms from the Quantum Atoms list box.  |
| <b>OK</b>            | Accept the changes made and close the dialog box.  |
| <b>Cancel</b>        | Close the dialog box without accepting the changes made.   |

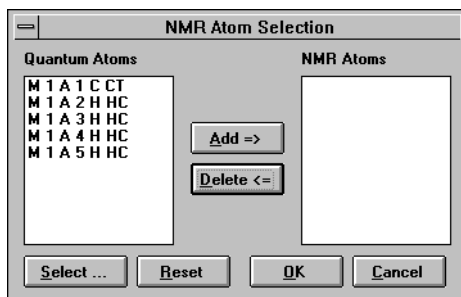
**Note:** If you had any atoms selected in the HyperNMR workspace, they will automatically be added to an empty (or optionally replace a previous) Quantum Atoms list. You may prefer to choose quantum atoms with the selection tool before opening this dialog box.



If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To create the Quantum Atoms selection, first select the atoms which you would like HyperNMR to treat quantum mechanically. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter **quantum-atoms** for the name.

## NMR Atom Selection Dialog Box

This dialog box can be used to select NMR atoms from the current molecular system if there is no quantum atom selection preceding this dialog box, or from the quantum atoms if the quantum atoms include only part of the current system, because the NMR atoms have to be treated quantum mechanically. Furthermore, the NMR atoms must be selected before running the calculation of shielding constants and spin-spin coupling constants.



This dialog box has the following items:

<b>Quantum Atoms</b>	Quantum Atoms lists all the selected quantum atoms in the current molecular system. Each line corresponds to one atom and includes the following information: M # stands for the atom belonging to molecule #, A ## means the atom numbered ## within the molecule, and the last two “words” show the atomic symbol and the TNDO atom type.
----------------------	---

<b>NMR Atoms</b>	This lists all atoms that you have chosen to be NMR atoms, in the same format as for Quantum Atoms.
<b>Add ⇒</b>	Use this button to add the selected (highlighted) atoms from the Quantum Atoms list to the NMR Atoms list.
<b>Delete ⇐</b>	Use this button to delete the selected set of atoms from the NMR Atoms list.
<b>Select...</b>	Press this button to open the Select NMR Atoms dialog box.
<b>Reset</b>	Delete all the atoms from the NMR Atoms list.
<b>OK</b>	Accept the changes made and close the dialog box.
<b>Cancel</b>	Close the dialog box without accepting the changes made.

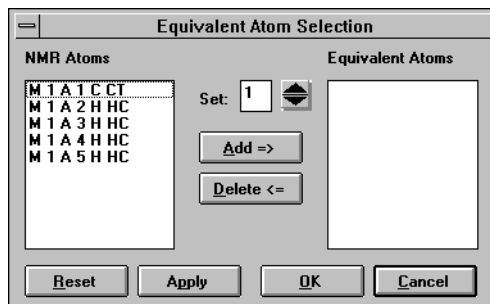
**Note:** If you had any atoms selected in the HyperNMR workspace, they will automatically be added to an empty (or optionally replace a previous) NMR Atoms list. You may prefer to choose NMR atoms with the selection tool before opening this dialog box.



If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To create the NMR Atoms selection, first select the atoms which you would like HyperNMR to treat quantum mechanically. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter ***nmr-atoms*** for the name.

## Equivalent Atom Selection Dialog Box

This dialog box can be used to set up the sets of equivalent atoms:



The items in this dialog box are:

- NMR Atoms** List all the selected NMR atoms. Each line corresponds to one atom and includes the following information: M # stands for the atom belonging to molecule #, A ## means the atom numbered ## within the molecule, and the last two “words” show the atomic symbol and the TNDO atom type.
- Set:** Use this to select the set of equivalent atoms. The number is an index starting at one. The spin buttons can be used to increment and decrement the set index. You can define a new set by increasing the index to one higher than the number of previously defined sets and then adding to the Equivalent Atoms list.
- Equivalent Atoms** This lists the equivalent atoms which belong to the equivalent set with the index shown in the Set text box. The atoms are listed in the same format as for NMR Atoms.
- Add =>** Use this button to add the selected atoms from the NMR Atoms list to the current set of equivalent atoms.
- Delete <=** Use this button to delete the selected atoms from the current set of equivalent atoms appearing in the Equivalent Atoms list.
- Reset** Delete all the atoms from the Equivalent Atoms list of the current set.
- Apply** Store the equivalent atoms shown in the Equivalent Atoms list for the current set; then you can roll up

(down) to set or reset the next set (previous set) of equivalent atoms.

- OK** Accept the changes made and close the dialog box.
- Cancel** Close the dialog box without accepting the changes made.

*Note:* This dialog box is not available before the selection of NMR atoms.



If you have HyperChem installed in your computer, you can create sets of equivalent atoms within HyperChem before getting the molecular system with HyperNMR. To create the equivalent atom selections, first select the atoms which you would like HyperNMR to treat as a set of equivalent atoms. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter ***equivalent-atoms-1*** for the name. If you have more than one set of equivalent atoms, you can repeat this procedure again but enter ***equivalent-atoms-2***, ***equivalent-atoms-3***, etc. for the selection names.

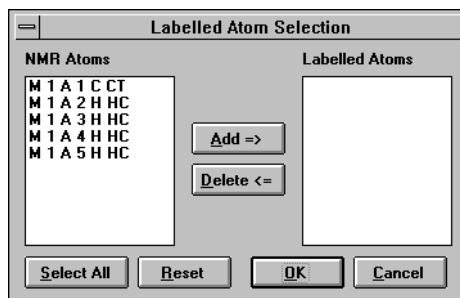
## Labelled Atom Selection Dialog Box

This dialog box lets you perform isotopic substitution by labelling atoms. In HyperNMR, labelling replaces the most abundant isotope with a less abundant isotope according to the following table:

Element	Default nucleus (unlabelled)	Labelled nucleus
H	$^1\text{H}$ (spin $\frac{1}{2}$ )	$^2\text{H}$ (spin 1)
C	$^{12}\text{C}$ (spin 0)	$^{13}\text{C}$ (spin $\frac{1}{2}$ )
N	$^{14}\text{N}$ (spin 1)	$^{15}\text{N}$ (spin $\frac{1}{2}$ )
O	$^{16}\text{O}$ (spin 0)	$^{15}\text{O}$ (spin $\frac{1}{2}$ )
F	$^{19}\text{F}$ (spin $\frac{1}{2}$ )	$^{19}\text{F}$ (spin $\frac{1}{2}$ )
P	$^{31}\text{P}$ (spin $\frac{1}{2}$ )	$^{31}\text{P}$ (spin $\frac{1}{2}$ )

Nuclei with spins other than  $\frac{1}{2}$ , such as  $^{12}\text{C}$  and  $^{14}\text{N}$ , usually do not exhibit spin-spin coupling with other nuclei. If you want them to show spin-spin coupling with others, you need to label them. If you want to treat some hydrogens as deuterium, and thus remove spin-spin coupling, you can label them.

There are no natural isotopes of F or P appropriate for NMR labeling. The naturally occurring  $^{17}\text{O}$  isotope of oxygen has a nuclear spin of  $\frac{5}{2}$  which HyperNMR cannot treat, but the artificial spin- $\frac{1}{2}$  isotope  $^{15}\text{O}$  is available in HyperNMR.



This dialog box contains the following items:

- |                       |   |
|-----------------------|---|
| <b>NMR Atoms</b>      | NMR Atoms lists all the selected NMR atoms in the current molecular system. Each line corresponds to one atom and includes the following information: M # stands for the atom belonging to molecule #, A ## means the atom numbered ## within the molecule, and the last two “words” show the atomic symbol and the TNDO atom type. |
| <b>Labelled Atoms</b> | This lists all atoms that you have chosen to be labelled atoms, in the same format as for NMR Atoms.  |
| <b>Add =&gt;</b>      | Use this button to add the selected atoms from the NMR Atoms list to the Labelled Atoms list.   |
| <b>Delete &lt;=</b>   | Use this button to delete the selected set of atoms from the Labelled Atoms list.   |
| <b>Select All</b>     | Press this button to select (highlight) all the atoms listed in the NMR Atoms list box.   |
| <b>Reset</b>          | Delete all the atoms from the Labelled Atoms list.  |
| <b>OK</b>             | Accept the changes made and close the dialog box.   |



**Cancel**            Close the dialog box without accepting the changes made.

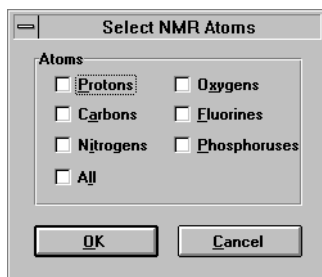
*Note:* This dialog box is not available before the selection of NMR atoms.



If you have HyperChem installed in your computer, you can create this selection within HyperChem before getting the molecular system with HyperNMR. To create the Labeled Atoms selection, first you select the atoms which you would like HyperNMR to treat as less abundant isotopes. Then open the Name Selection dialog box in HyperChem by clicking the Select/Name Selection menu item and enter **labelled-atoms** for the name.

## Select NMR Atoms Dialog Box

This dialog box can be used to select specific types of NMR atoms, such as, protons, carbons, etc. from the current molecular system if there is no quantum atom selection preceding this dialog box, or from the quantum atoms if the quantum atoms include only part of the current system, because the NMR atoms have to be treated quantum mechanically.



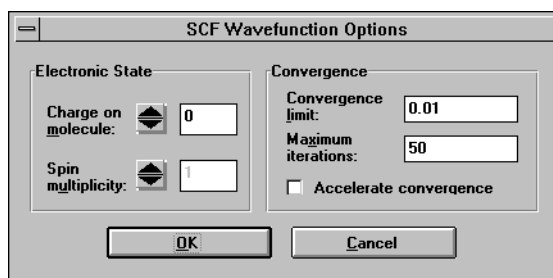
This dialog box has the following items:

- |                     |                              |
|---------------------|------------------------------|
| <b>Protons</b>      | Select all the hydrogens.    |
| <b>Carbons</b>      | Select all the carbons.      |
| <b>Nitrogens</b>    | Select all the nitrogens.    |
| <b>Oxygens</b>      | Select all the oxygens.      |
| <b>Fluorines</b>    | Select all the fluorines.    |
| <b>Phosphoruses</b> | Select all the phosphoruses. |
| <b>All</b>          | Select all the atoms.        |

<b>OK</b>	Accept the changes made and close the dialog box.
<b>Cancel</b>	Close the dialog box without accepting the changes made.

## SCF Wavefunction Options Dialog Box

The SCF Wavefunction Options dialog box lets you set up the conditions for computing the unperturbed and perturbed SCF wavefunctions which, in turn, are used for the calculation of the shielding constants and spin-spin coupling constants for the selected NMR atoms.



This dialog box has the following components:

**Electronic State** Use this area to specify the net charge for the current molecular system and electron spin-pairing information.

**Charge on molecule** This is the net excess of nuclear charge over the electronic charge. It is zero for a neutral system, a positive integer for a cation, and a negative integer for an anion.

**Spin multiplicity** This displays the electronic spin multiplicity that will be used for the current molecular system, and is always 1 (i.e., singlet state). The spin multiplicity is  $2S + 1$ , where  $S$  is the total spin quantum number.

**Convergence** This section is used to specify the desired accuracy for the SCF wave function, and the maximum number of iterations that are allowed for reaching that accuracy.

**Convergence limit** An SCF calculation ends when the difference in energy after any two consecutive

iterations is less than this amount. This is the convergence limit for total electronic energy during iterations of an SCF calculation. The default value is 0.01 kcal/mol. The practical range is 1 to  $10^{-3}$ . The calculation might not reach values less than  $10^{-3}$  because of round-off errors.

**Maximum iterations**

This sets the maximum number of iterations for an SCF calculation. The calculation ends after this number of iterations, even if it has not reached the Convergence limit. The results may not be correct if the calculation was far from convergence or the energy of the system is oscillating. The default of 50 should be correct for most cases. The practical range is 50 to 200.

**Accelerate convergence**

With this check box checked (x), HyperNMR will apply the Direct Inversion of Iterative Subspace (DIIS) procedure to speed up the SCF calculation at the expense of requiring much more memory.

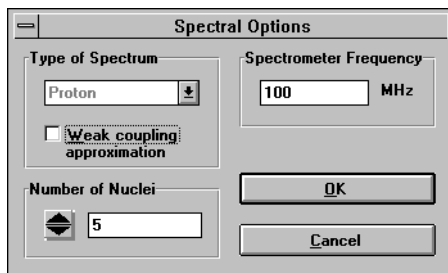
**OK** Accept the changes made and close the dialog box.

**Cancel** Close the dialog box without accepting the changes made.

*Note:* HyperNMR always uses the restricted Hartree-Fock (RHF) approximation when computing the unperturbed wave function. See Chapter 6 for details.

## Spectral Options Dialog Box

This dialog box can be used to set up the conditions for computing an NMR spectrum for the current chemical shifts and nuclear spin-spin coupling constants.



This dialog box has the following components:

**Type of Spectrum** Use this to specify the type of NMR spectrum that you wish to compute.

**Weak Coupling approximation** Check this to inform HyperNMR to use the weak coupling approximation (a perturbation method) to calculate the transition energies and the relative intensities of transitions.

**Number of Nuclei** This shows the current number of NMR atoms. It is zero if there is no molecular system in HyperNMR or if there is no NMR atom selection for the current molecular system. Decreasing this number will remove NMR atoms (and any shielding or coupling results for them) from the end of the list and increasing it will add new NMR atoms with the same nuclei as Type of Spectrum (setting their shielding and coupling constants to zero). You should not change this number unless you want to type in shielding and coupling constants for a different number of NMR atoms (see below).

**Spectrometer Frequency** Use this to specify the spectrometer frequency (in MHz) that you would like to use when you compute an NMR spectrum. Enter a real number that is greater than 0 and less than or equal to 1000000.

**OK** Accept all the changes made in this dialog box and dismiss the dialog box.

**Cancel** Ignore any changes made and close this dialog box.

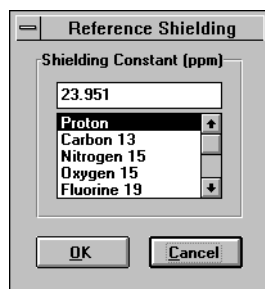
HyperNMR can be used to calculate a  $^1\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{15}\text{O}$ ,  $^{19}\text{F}$  or  $^{31}\text{P}$  spectrum. HyperNMR may only compute one type of NMR spectrum at a time, but you can calculate multiple spectra from the

same shielding and coupling results. For instance, you might be interested in both the  $^1\text{H}$  and  $^{13}\text{C}$  NMR spectra for ethanol ( $\text{CH}_3\text{CH}_2\text{OH}$ ). You can calculate all the shielding constants and spin-spin coupling constants at once, compute the  $^1\text{H}$  spectrum, change to a  $^{13}\text{C}$  spectrum in this dialog box and then compute the  $^{13}\text{C}$  NMR spectrum.

You can use the Number of Nuclei spin button to change the number of NMR atoms. HyperNMR will warn you if the number of nuclei you try to add is greater than the number of actual atoms in the current system, and ask you whether or not you would like to add a virtual molecule (without coordinates) to the current molecular system. Once the number of nuclei has been changed compared to its original value in this dialog box, HyperNMR makes the menu items View/Shielding Constants and View/Coupling Constants accessible. You can type your own chemical shifts and coupling constants into the corresponding table shown in the HyperNMR workspace and then compute the NMR spectrum using the new shielding and coupling constants. Furthermore, the menu items Setup/Shielding & Coupling, Compute/I - Shielding & Coupling, and Compute/Both: I Then II in the main window of HyperNMR will not be accessible if the number of nuclei in this dialog box is greater than the number of the actual atoms in the current molecular system, since the atoms in the virtual molecule do not have coordinates.

## Reference Shielding Dialog Box

The Reference Shielding dialog box can be used to define a reference shielding constant for each type of nucleus that will be used to calculate chemical shifts. HyperNMR computes the chemical shifts relative to the chosen reference shielding for the NMR atoms in the current system, by subtracting the computed value from the reference value.



This dialog box contains only one item:

<b>Shielding Constant (ppm)</b>	This specifies the shielding constant (in ppm) for the $^1\text{H}$ , $^{13}\text{C}$ , $^{15}\text{N}$ , $^{15}\text{O}$ , $^{19}\text{F}$ and $^{31}\text{P}$ in the reference molecular systems. The value shown corresponds to the nucleus highlighted in the list. You can specify reference shielding constants for all the nuclei. The default values for proton and $^{13}\text{C}$ have been calculated from methane with TNDO/2 using the Special Slater exponents for shielding option. For other nuclei the default value is zero.
<b>OK</b>	Accept all the changes made in this dialog box and dismiss the dialog box.
<b>Cancel</b>	Ignore any changes made and close this dialog box.

To determine a reference shielding constant you must choose a reference molecular system that includes one of the desired type of nucleus whose shielding you will use as the reference. You then set up and calculate the shielding constants of the reference system. The resulting shielding constant for the reference nucleus in the reference molecular system can then be entered in this dialog box.

*Note:* The predicted magnetic shielding constants for a given molecular system may strongly depend on the applied method. And within the same method, the shielding constants may also depend on the options. It is therefore strongly recommended that you use magnetic shielding constants generated by HyperNMR as your reference shielding values, as the errors in the absolute shielding constants will cancel out to a great extent in the chemical shifts.

## Solvation Effects

HyperNMR is not able to calculate solvent effects implicitly, as though the molecule(s) of interest were immersed in a homogeneous medium. The solvent molecules must be included explicitly. You can set up the workspace with the molecule(s) of interest surrounded by some configuration of solvent molecules, which might be included as quantum atoms (depending on the size of the system) but not as NMR atoms. Then one would calculate the shifts, coupling constants, and/or spectrum. Ideally, this would be repeated for several different configurations of solute and solvent

so as to give a statistically meaningful distribution of results. Such a series of systems could be set up with molecular dynamics calculations using a tool such as HyperChem.

## Files

HyperNMR uses these program or user-editable files:

File Type	File Names
HyperNMR program files	nmr.exe nmrcalc.exe
HyperNMR default settings	nmr.ini
Atom type file	nmrtype.rul nmrtype.txt
Semi-empirical parameters	nmr_shft.abp nmr_cupl.abp
Help text	nmr.hlp
Communication files	nmrparam.in nmrparam.out nmrspect.in nmrspect.out
HyperNMR input files	*.hin *.nmr *.zmt
Log files	*.log

## Program Files

These are the HyperNMR program files:

**nmr.exe** This program runs the HyperNMR user interface. It includes reading and writing HyperNMR input files and displaying molecular systems and the HyperNMR calculation results. This is the Hyper-

NMR “front-end”, designed to run under Windows on a PC.

`nmrcalc.exe` This program is the HyperNMR “backend” which does scientific calculations including predicting electronic shielding tensors, shielding constants, spin-spin coupling constants and NMR spectra.

These two files are not modifiable.

## HyperNMR Default Settings

The file `NMR.INI` contains all the default settings for HyperNMR. HyperNMR uses `NMR.INI` to remember your settings from the previous session. HyperNMR reads this file when you start it.

Some settings can be changed only in `NMR.INI`. You can change these settings, or any settings in this file, by using a text editor like Windows Notepad. `NMR.INI` is installed by default in your HyperChem directory, if one exists. HyperNMR gets the location of `NMR.INI` from the `NMRIniPath` record in `WIN.INI` each time it starts.

The settings are in text records, one on each line. Each record contains a keyword, an equal sign, and a value that can be Yes or No (Boolean), an integer (Int), a floating point number (Real), or one of a limited set of possibilities (Enum).

*Note:* Keywords and values in `NMR.INI` are not case-sensitive.

The `NMR.INI` file contains the following sections, each with a section title in brackets:

### [nmr]

Keyword	Value Type	Value	How to Set	Use
<code>AtomRenderType</code>	Int	1-3	View Rendering dialog box	Atom rendering type 1: Labels; 2: Disks; 3: Spheres



Keyword	Value Type	Value	How to Set	Use
BondRenderType	Int	1-3	View Rendering dialog box	Bond rendering type 1: Sticks; 2: Bonds; 3: Cylinders
BackColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Background color for displaying molecules
LabelColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Atom label color
BondColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Bond color
SelectionColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Selection color for displaying selections of atoms
SpectrumLineColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Color for displaying the lines of a spectrum
SpectrumEnvelopeColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Color for displaying the envelope of a spectrum
SpectrumSelectionColor	Enum	Color <sup>a</sup>	File Preferences dialog box	Color for selecting the line of a spectrum
DefaultColorSet	Enum	HyperChemColors CustomColors1 CustomColors2	This file <sup>b</sup>	Name of set of default atomic colors to use in rendering atoms

Keyword	Value Type	Value	How to Set	Use
DefaultRadiusSet	Enum	CPKRadii BallsSticks Radii Custom- Radii	This file <sup>b</sup>	Name of set of default atomic radii to use in rendering atoms
SemiEmpirical-Method	Enum	TNDO1 TNDO2	Setup Shielding & Coupling Options dialog box	Semi-empirical quantum mechanical method for computing shielding and coupling
TotalCharge	Int	-10 - +10	Setup SCF Options dialog box	The net excess of nuclear charge over the electronic charge
SpinMultiplicity	Int	1	Setup SCF Options dialog box	Electron spin multiplicity
SCFConvergence-Limit	Real	>0-1000	Setup SCF Options dialog box	The desired maximum difference of the energies between any two consecutive iterations before stopping SCF calculation
SCFIterationLimit	Int	>0-10000	Setup SCF Options dialog box	The maximum iterations before stopping SCF calculation
SCFAccelerate-Convergence	Bool	Yes, No	Setup SCF Options dialog box	If Yes, then uses DIIS procedure to accelerate SCF calculation
LogPrintLevel	Int	0-10	This file <sup>b</sup>	The level of information output to a log file

Keyword	Value Type	Value	How to Set	Use
MinimalIntensity	Real	>1.0e-8	This file <sup>b</sup>	The minimum intensity of a NMR spectrum to be plotted in the HyperNMR workspace and to be printed in a log file.
ModifySlater-Exponents	Bool	Yes, No	Setup Shielding & Coupling Options dialog box	If Yes, modifies the Slater exponents of the atomic orbitals for computing the shielding constants
ComputeThree-CenterTerms	Bool	Yes, No	Setup Shielding & Coupling Options dialog box	If Yes, computes the contribution to the shielding constants from the three-center integrals and the non-NDO off-diagonal elements of the density matrix
IncludeDOrbitals-ForSecondRow	Bool	Yes, No	Setup Shielding & Coupling Options dialog box	If Yes, includes d-orbitals for the second row elements in the atomic basis set if any of the second row elements exists
CalculateShielding	Bool	Yes, No	Setup Shielding & Coupling Options dialog box	If Yes, computes the shielding constants
CalculateCoupling	Bool	Yes, No	Setup Shielding & Coupling Options dialog box	If Yes, computes the nuclear spin-spin coupling constants
NMRFrequency	Int	1-32767	Setup Spectral Options dialog box	Spectrometer frequency for computing NMR spectra (in MHz)

Keyword	Value Type	Value	How to Set	Use
SpectraType	Enum	Proton Carbon-13 Nitrogen-15 Oxygen-15 Fluorine-19 Phosphorus-31	Setup Spectral Options dialog box	Type of NMR spectrum
AbsLineWidth	Bool	Yes, No	This file <sup>b</sup>	If Yes, the range of the line width scroll bar will be independent of zoom
MaxAbsLineWidth	Real	0-1000	This file <sup>b</sup>	Maximum line width value (in ppm) if AbsLineWidth is set
MinAbsLineWidth	Real	0-1000	This file <sup>b</sup>	Minimum line width (in ppm) value if AbsLineWidth is set
WeakCoupling-Approximation	Bool	Yes, No	Setup Spectral Options dialog box	If Yes, uses the weak coupling approximation
CalculateShieldingTensor	Bool	Yes, No	This file <sup>b</sup>	If Yes, computes the full shielding tensors for all the NMR atoms

a. RGB color, such as "RGB(0,255,0)". This specifies a color by giving three numbers between 0 and 255 inclusive which gives the amounts of Red, Green, and Blue components respectively in the desired color.

b. Use a text editor, like Windows Notepad, to change this record in NMR.INI.

**[ReferenceShieldings]**

Key-word	Value Type	Value	How to Set	Use
H	Real	-100000 – +100000	Setup Reference Shielding dialog box	Shielding constant in a reference molecule (ppm)
C	Real	-100000 – +100000	Setup Reference Shielding dialog box	Shielding constant in a reference molecule (ppm)
N	Real	-100000 – +100000	Setup Reference Shielding dialog box	Shielding constant in a reference molecule (ppm)
F	Real	-100000 – +100000	Setup Reference Shielding dialog box	Shielding constant in a reference molecule (ppm)
P	Real	-100000 – +100000	Setup Reference Shielding dialog box	Shielding constant in a reference molecule (ppm)

The following three sections define three sets of atomic colors and can be changed in this file (only in this file). These three sets of atomic colors are HyperChemColors, CustomColor1, and CustomColor2. Each of these sets defines a color for each atom. The default color is used for an atom if its color is not defined within the set. The default values are:

```
[HyperChemColors]
Default=RGB(255, 255, 255)
H=RGB(255, 255, 255)
C=RGB(0, 255, 255)
N=RGB(0, 0, 255)
O=RGB(255, 0, 0)
F=RGB(255, 255, 0)
Na=RGB(255, 0, 255)
S=RGB(255, 255, 0)
K=RGB(255, 0, 255)
Fe=RGB(255, 0, 0)
Co=RGB(0, 0, 255)
Cu=RGB(0, 255, 0)
Br=RGB(255, 255, 0)
I=RGB(255, 0, 0)
Au=RGB(255, 255, 0)
[CustomColors1]
```

```

Default=RGB(0, 255, 0)
H=RGB(255, 255, 255)
C=RGB(128, 128, 128)
N=RGB(128, 128, 255)
O=RGB(255, 128, 128)
[CustomColors2]
Default=RGB(0, 255, 0)
H=RGB(255, 255, 255)
C=RGB(128, 128, 128)
N=RGB(0, 0, 255)
O=RGB(255, 0, 0)

```

The following three sections define three sets of atomic radii and can be changed in this file. These three sets of atomic radii are CPKRadii, BallsSticksRadii, and CustomRadii. Each of these sets defines a radius for each atom. The default radius is used for an atom if its radius is not defined within the set. Examples of these sections are:

```

[CPKRadii]
Default=.8
Lp=0.90
H=0.83
He=0.93
Li=1.23
Be=0.90
B=0.82
C=1.20
N=1.12
O=1.05
F=0.90
...
Lr=1.65
[BallsSticksRadii]
Default=0.4
H=0.2
Lp=0.11
[CustomRadii]
Default=0.4
H=0.3
Lp=0.3

```

## Atom Type Files

HyperNMR provides two files that you need for assigning types to atoms in molecular systems.

**NMRTYPE.RUL** A text file containing a set of rules for assigning types to the atoms in a molecular system.

**NMRTYPE.TXT** List of atom types.

For a shielding and spin-spin coupling calculation, each atom in a molecular system must have an atom type. Atom types represent sets of atoms of the same element that have different chemical environments. The TNDO/1 and TNDO/2 methods in HyperNMR treat all atoms of the same type in similar way.

When you get a molecule from HyperChem or from a HyperChem HIN file, the molecular system may already contain types for all the atoms. These atom types may or may not be consistent with those defined in the `nmrtype.rul` file. However the molecular system will not have any atom types if you get it from a MOPAC Z-Matrix file. Hence computing atom types is recommended by using File/Calculate Atom Types in HyperNMR.

### Atom Typing Rules

You can add or modify the atom typing rules found in `nmrtype.rul`, which contains rules for TNDO method. The file has a hierarchical decision structure, including these levels:

- Element
- Clauses and tests
- Assignment statement

For each atom in a molecular structure, HyperNMR scans this file and makes these decisions:

- What is the element?
- For this element, does the atom satisfy the tests in the first clause? If not, in the second clause, and so on?
- When the atom satisfies the tests in a clause, HyperNMR assigns the atom type.
- HyperNMR goes to the next atom.

**Caution:** Errors in `nmrtype.rul` can cause fatal errors in HyperNMR. Make a backup copy of `nmrtype.rul` before you change it.

**Example:**

```
S:  connected to -H?
    =SH.
    =S.
```

In this example for a sulphur atom (S), the first clause contains on test: is the sulfur atom connected to a hydrogen? If true, the atom type is SH. If false, HyperNMR goes to the next clause, which has no tests, only a statement that assigns S as the atom type.

## Syntax

This is the structure of `nmrtype.rul`:

```
forcefield(tndo)
<element-symbol>:
; <comment>
<clause>
<test>?
<test>?
...
=<assignment-statement>.
<clause>
...
```

**Note:** Entries in `nmrtype.rul` are not case-sensitive, except for element symbols. HyperNMR recognizes, for example, the difference between SI and Si. Spaces between words are optional.

### forcefield

<force-field-name> is tndo.

### <element-symbol>:

This is the chemical symbol for an element, for example, S, Cl, and Na.



A colon must follow the symbol. Each element can have only one symbol. Element symbols include Lp for lone pair.

### **;<comment>**

Comments can occur anywhere in the file. A comment begins with a semicolon and ends at the end of a line.

### **<clause>**

A clause is a set of tests applied to each atom plus an atom type assignment. Each clause ends with an assignment statement.

*Note:* HyperNMR checks clauses from top to bottom, so their order is important. Place more specific clauses before general ones.

If one test in a clause fails, HyperNMR goes to the next clause. If all the tests in a clause are calculated as true, HyperNMR assigns an atom type. Some elements have a final clause without tests. This assigns an atom type if all previous clauses failed. If the force field does not have a clause without tests and if all clauses failed, HyperNMR assigns the atom type \*\*, meaning unknown.

### **<test>?**

A question mark must follow each test. A clause can obtain these tests:

Test	To pass test
Is it sp3?	True if the atom has no double, triple, or aromatic bonds.
Is it sp2?	True if the atom has one double or two or more aromatic bonds.
Is it sp?	True if the atom has one triple or two double bonds.
In 5 ring?	True if the atom is part of a five-membered ring.
In 6 ring?	True if the atom is part of a six-membered ring.
Has aromatic bond?	True if the atom has at least one aromatic bond.
Connected to <expression>?	True if the atom is bonded to atoms described in <expression>.

## “Connected to” Expressions

“Connected to” expressions include the type of connecting bond, the connected atoms, and bonds between the connected atoms. These symbols represent bonds:

-	single bond
=	double bond
#	triple bond
~	aromatic bond
(no symbol)	any bond type

“Connected to” expressions use element symbols to represent connected atoms. You can use any standard symbols from the periodic table, plus these:

Lp	Lone pair
R	Any element except H and Lp
X	Any halogen (F, Cl, Br, I)
*	Any element except Lp

*Note:* Keep “connected to” expressions as short as possible. Atom types are usually unaffected by atoms more than two or three bonds away. Also, longer expressions increase the time required to assign types.

## Side Groups

Specify side chains either by placing them in parentheses or by using numbers. For example:

-CH<sub>2</sub> or -C(H)H  
 -CH<sub>2</sub>Cl or -C(H)(H)Cl

## Rings

The atom being typed can be connected to or part of ring. Pairs of tags (@1 to @9) placed after atoms in the ring describe closure points, for example:

-C@1CCCCC@1

This describes a connection to a six-membered ring, with the tag @1 after C<sub>1</sub> and C<sub>7</sub>. C<sub>1</sub> and C<sub>7</sub> represent the same atom in the ring.

-@1CCC@1

This describes a cyclopropane ring, with the atom being typed as the first atom (C<sub>1</sub>) in the ring (@1 at the beginning of the expression modifies the atom being typed). C<sub>1</sub> and C<sub>4</sub> are the same atom.

### <Assignment-statement>.

A clause must end with an assignment statement that contains an atom type belonging to the force field. These types appear in the file **nmrtype.txt**. The assignment statement must end with a period.

## Semi-empirical Parameter Files

**nmr\_shft.abp** and **nmr\_cupl.abp** contain the semi-empirical parameters for computing the shielding constants and nuclear spin-spin coupling constants, respectively. In these two files, each line corresponds to an atom type. The entries with a minus sign, “-”, correspond to the parameters for capping atoms.

Column	Symbol	Content
1	T	Atom type. A “-” sign preceding the atom type indicates a capping atom <sup>a</sup>
2	$\zeta_s$	s orbital Slater exponent (a.u.)
3	$\zeta_p$	p orbital Slater exponent (a.u.)
4	$\zeta_d$	d orbital Slater exponent (a.u.)
5	$-(I_s + A_s)/2$	Negative of averaged ionization potential and electron affinity of s orbital (eV)
6	$-(I_p + A_p)/2$	Negative of averaged ionization potential and electron affinity of p orbital (eV)
7	$-(I_d + A_d)/2$	Negative of averaged ionization potential and electron affinity of d orbital (eV)
8	b	Bonding parameter (eV)
9	$F^2$	Slater-Condon parameter $F^2(p,p)$ (a.u.)

Column	Symbol	Content
10	$G^1$	Slater-Condon parameter $G^1(s,p)$ (a.u.)
11	$s^2(0)$	nmr_shft.abp: reserved; nmr_cupl.abp: s orbital density at nucleus
12	g	nmr_shft.abp: reserved; nmr_cupl.abp: g factor of nucleus
13	$E^{\text{atom}}$	Electron energy of isolated atom for CNDO <sup>a</sup> (eV)
14	$E^{\text{atom}}$	Electron energy of isolated atom for INDO <sup>b</sup> (eV)
15	$H^{\text{atom}}$	Heat of formation of free atom (kcal/mol)

a. Parameterized fluorine atoms that represent atoms bonded to the selected atoms for the calculation (see Chapter 6 for details).

b. Calculated from the parameters listed in these files for the corresponding ground states.

**Note:** Columns 11 and 12 are different in **nmr\_shft.abp** and **nmr\_cupl.abp**, since there are two extra parameters needed for computing the spin coupling constants: the s-orbital density at nucleus and the g factor of nucleus.

## Help Text

All Help text available on screen is in one file, NMR.HLP. You can read this text using the Help menu, or by activating the Help window with the [F1] key.

## Communication Files

The communication files of HyperNMR, nmrparam.in, nmrparam.out, nmrspect.in, and nmrspect.out, in the directory specified by the line

```
NMRIniPath=
```

in the WIN.INI file, enable the front-end and back-end communication when you want to calculate NMR properties. Hence these file names are reserved, and you may not overwrite them. If any of these files are opened by any other application, HyperNMR will complain when you want to do an NMR calculation.

## HyperNMR Input Files

HyperNMR can import a molecular system from a HyperChem HIN file or a MOPAC Z-Matrix file and get a molecular system and/or NMR results from a HyperNMR NMR file which was saved previously by HyperNMR.

### Z-Matrix Files

The MOPAC Z-Matrix file format includes the element symbol or atomic number and the internal coordinates or Cartesian coordinates of each atom in the molecular system. The following gives a sample Z-Matrix file for methane

```
Name: C:\HYPER\NMR\METHANE.ZMT
MOPAC file created on 9/2 15:16:19 1994 by HYPERCHEM
C      00000.0000 0  00000.0000 0  00000.0000 0 0 0 0
H      00001.0899 1  00000.0000 0  00000.0000 0 1 0 0
H      00001.0899 1  00109.4710 1  00000.0000 0 1 2 0
H      00001.0900 1  00109.4709 1  00119.9994 1 1 2 3
H      00001.0900 1  00109.4714 1  00239.9997 1 1 2 3
0
```

This Z-Matrix file includes four sections which are 1) Keywords or Commands in the first line which tells the MOPAC program what you want it to do. Here it is an empty line; 2) Comments (the second and third lines); 3) Internal coordinates (lines 4 to 8). Each line defines one atom and contains all the information to define the atom in the molecular system uniquely, such as the atomic symbol or atomic number (word one), bond distance (word two), to which atom it is bonded (word eight), whether or not to optimize the bond distance (word three, 0 to fix this bond distance and 1 to optimize it), bond angle (word four), to which two atoms the bond angle is formed (words eight and nine), whether or not to optimize this bond angle (word five), torsion angle (word six), to which three atoms this torsion angle is formed (words eight, nine, and ten), and whether or not to optimize this torsion angle (word seven); 4) A line with 0 (zero) or an empty line to end the molecular coordinate system.

## NMR Files

HyperNMR can store a molecular system from the workspace, plus the current settings of conditions for computing NMR data, and/or NMR results which have been computed, in an NMR file. This information can be loaded and displayed by HyperNMR later.

The NMR files are text files. You can edit these files with any text editor, like Windows Notepad or Write. If you use an editor that can impose its own format, keep the NMR file in its original text format. For example, with Windows Write, choose the No Conversion option when you open a NMR file.

An NMR file contains several sections. Each section may have several records, each on a separate line. Comment lines begin with a semicolon (;).

This is an example of an NMR file for a methane molecule with NMR results:

```

BeginHIN
forcefield amber
sys 0
view 40 0.7179181 55 15 0.9251873 -2.753075E-02 -0.3785107
0.2208629 0.8501284 0.4780181 0.3086225 -0.5258552 0.7926087
0.3136711 -0.4597336 -55
mol 1
atom 1 - C CT - 0 -0.4632772 0.4293783 -6.066945e-007 4 2 s 3
s 4 s 5 s
atom 2 - H HC - 0 -0.4632772 1.519378 -6.066945e-007 1 1 s
atom 3 - H HC - 0 0.564386 0.06604892 -6.066945e-007 1 1 s
atom 4 - H HC - 0 -0.9771005 0.06604889 0.8899868 1 1 s
atom 5 - H HC - 0 -0.9771116 0.06604096 -0.8899783 1 1 s
endmol 1
EndHIN
NMRSelectionBegin
5
mol 1 atom 1
mol 1 atom 2
mol 1 atom 3
mol 1 atom 4
mol 1 atom 5
NMRSelectionEnd
QuantumSelectionBegin
0

```

```
QuantumSelectionEnd
EquivalentAtomsBegin
EquivalentSet 1
mol 1 atom 2
mol 1 atom 3
mol 1 atom 4
mol 1 atom 5
EquivalentAtomsEnd
LabelledAtomsBegin
1
mol 1 atom 1
LabelledAtomsEnd
OptionsBegin
SemiEmpiricalMethod = TND02
TotalCharge = 0
SpinMultiplicity = 1
SCFConvergenceLimit = 0.01
SCFIterationLimit = 50
SCFAccelerateConvergence = No
LogPrintLevel = 0
ModifySlaterExponents = No
ComputeThreeCenterTerms = No
IncludeDOrbitalsForSecondRow = No
NMRFrequency = 100
CalculateShielding = Yes
CalculateShieldTensor = Yes
CalculateCoupling = Yes
OptionsEnd
NMRAtomicNumbersBegin
6 1 1 1 1
NMRAtomicNumbersEnd
ShieldingCouplingBegin
SHIELDING
49.457 22.07775 22.07775 22.07775 22.07775
TENSOR
49.4572 0.0004 0.0002 0.0008 49.4571 0.0022 0 -0.0022 49.4573
49.4554 49.4572 49.4591 49.4572 0.0027
15.4283 0 0 0.0001 35.3761 -0.0001 0 -0.0002 15.4283
15.4283 15.4283 35.3761 22.0776 19.9478
33.1597 -6.269 0 -6.2689 17.6445 0 0.0001 0 15.4282
15.4281 15.4281 35.3761 22.0775 19.9479
19.8609 3.1344 -7.6779 3.1343 17.6446 -5.429 -7.6778 -5.4291
28.7271
```

```

15.4283 15.4283 35.3761 22.0776 19.9478
19.8612 3.1346 7.6779 3.1346 17.6449 5.4293 7.6778 5.4292
28.7267
15.4283 15.4283 35.3761 22.0776 19.9478
COUPLING
0 122.8563 122.8563 122.8563 122.8563
122.8563 0 -6.127 -6.127 -6.127
122.8563 -6.127 0 -6.127 -6.127
122.8563 -6.127 -6.127 0 -6.127
122.8563 -6.127 -6.127 -6.127 0
ShieldingCouplingEnd
SpectraBegin
7
FREQUENCY INTENSITY DEGENERACY LINEWIDTH
125.872 32 10 0
248.728 32 10 0
1873054 1 1 0
1873177 4 4 0
1873300 6 6 0
1873423 4 4 0
1873546 1 1 0
SPECTRATYPE
1
REFERENCESHIELDING
23.951 48.595 0 0 0 0
NMRATOMSWITHSPINHALF
5
SpectraEnd

```

For a description of the sections and records in this example and other possible records, see the following sections.

### Molecular System:

This section begins with a keyword BeginHIN and ends with EndHIN and contains information of defining a molecular system. The following records may appear in this section. '0' indicates an optional record. Each record must begin on a separate line.

### Syntax:

```

;<comment>0
forcefield <force-field-name>

```



```

sys0 <temperature>
view0 <view-distance> <view-scale> <slab-distance> <slab-
thickness> <view-transform> ... <view-transform>
box0 <x> <y> <z>
mol <mol #> <mol-name>0
res0 <res#> <res-name> <PDB#> <previous-res#> <next-res#>
atom <at#> <atom-name> <element> <type> <flags> <at-
charge> <x> <y> <z> <cn> <nbor#-nbor-bond>1
vel0 <at#> <x> <y> <z>
mass0 <at#> <mass>
basisset0 <at#> <basis set>
atom ...
vel0 ...
mass0 ...
basisset0 ...
atom ...
vel0 ...
mass0 ...
basisset0 ...
...
endres0 <res#>
endmol <mol#>

```

## Description:

### ; comment

Each line of comments must have a semicolon at the beginning of the line. All the comment lines must be at the beginning of the molecular system (after the keyword BeginHIN).

1. This variable can be repeated up to 12 times.

**forcefield <force-field-name>**

The force-field-name must be “amber” or “tndo” in order to run a HyperNMR calculation successfully. If you get the molecular system from a HyperChem workspace via File/Get HyperChem, HyperNMR will automatically switch the force field to AMBER in HyperChem before getting the molecular system. However, if a molecular system is imported from a HyperChem HIN file, the force field in this HIN file may not be AMBER, or if the molecular system is imported from a Z-Matrix file, it does not contain any force field and atom types. Therefore, calculation or recalculation of atom types is recommended via Edit/Calculate Atom Types in HyperNMR before computing shielding and coupling.

**sys <temperature>**

This record gives the temperature of the molecular system in degrees Kelvin and is part of a HyperChem HIN file; it is ignored by HyperNMR.

**view <view-distance> <view-scale> <slab-distance> <slab-thickness> <view-transform> ... <view-transform>**

A view record gives instructions for transforming a 3D structure into 2D space.

view-distance	The distance, in angstroms, from the viewer to the origin of the molecular system.
view-scale	The magnification factor.
slab-distance	The distance, in angstroms, from the viewer to the middle of the clipping slab.
slab-thickness	The distance between the front and back clipping planes, in angstroms.
view-transform	These 12 fields describe a 3 by 4 viewing transformation matrix, with 4 rows and 3 columns. The fields appear row by row. A fourth column, normally describing perspective, is not present.

**box <x> <y> <z>**

A box record reports the presence and x, y, and z dimensions of a periodic box, in angstroms, in the molecular system.

This periodic box is an optional part of a HyperChem HIN file and is ignored by HyperNMR.

**mol <mol#> <mol-name>**

A mol record starts the description of each molecule in the molecular system. An endmol record must end each molecule.

**mol#**            The number of the molecule in the molecular system, starting with 1. The number reflects the order of adding molecules to the system. Default is 1.

**mol-name**      Up to 19 characters describing a molecule.

**res <res#> <res-name> <PDB#> <previous-res#> <next-res#>**

This record describes amino-acid or nucleotide residues. It is an optional part of a HyperChem HIN file and is ignored by HyperNMR.

**atom <at#> <atom-name> <element> <type> <flags> <at-charge> <x> <y> <z> <cn> <nbor# nbor-bond>**

An atom record starts the description of each atom in a molecule.

**at#**            The number of an atom in a molecule, starting with 1. Atom numbers are continuous through a molecule. A combination of atom number and molecule number (see mol, above) uniquely identify an atom in a molecular system.

**atom-name**    This name (up to four characters) usually comes from a Brookhaven PDB or TPL file. It cannot contain spaces or quotes.

**element**       The chemical symbol (up to three characters) for the described atom.

**type**           The atom type (up to four characters) assigned by a force field to the described atom.

<b>flags</b>	Single letter codes that describe this atom. “-” indicates no flag. Multiple flags appear as one “word”, without spaces. This field is not used by HyperNMR.
<b>at-charge</b>	The formal charge on the atom.
<b>x y z</b>	The coordinates of the atom in the Molecular Coordinate System.
<b>cn</b>	The number of atoms covalently bonded to this atom, or coordination number. cn is an integer from 0 to 12.
<b>nbor# nbor-bond</b>	This describes the bonded atoms and the type of bond for each. nbor# and nbor-bond occur as a pair of values. nbor# is the same as at# for bonded atoms. nbor-bond is a bond type: s, d, t, or a, for single, double, triple, or aromatic (conjugated) bonds. This set of values, nbor# nbor-bond, repeats to describe all bonded atoms (up to 12).

**vel <at#> <x> <y> <z>**

This record gives the x, y, and z components of velocity for the given atom # and is ignored by HyperNMR.

**mass <at#> <mass>**

This record gives the mass of the given atom number, and is ignored by HyperNMR.

**basisset <at#> <basis set>**

This record gives the basis set that has been applied to the given atom number for *ab initio* calculations in HyperChem. It is ignored by HyperNMR.

**endres <res#>**

An endres record appears after the last atom in each residue and is ignored by HyperNMR.

**endmol <mol#>**

An endmol record appears after the last atom in each molecule. <mol#> is the number of the molecule in a molecular system, starting with 1. The number reflects the order of adding molecules to the system. Default = 1.

**Selections:**

This section includes various selections: NMR atom selection, quantum atom selection, equivalent atom selection, and labelled atom selection. Each of these selections begins and ends with special keywords.

The contents within each pair of keywords (NMRSelectionBegin and NMRSelectionEnd, QuantumSelectionBegin and QuantumSelectionEnd, EquivalentAtomsBegin and EquivalentAtomsEnd, and LabelledAtomsBegin and LabelledAtomsEnd) define uniquely the atoms for the specific selection. The first record after the keyword ...Begin is the total number # of the atoms in this selection, and then the following # records contain the information on the atoms in this selection. Each of these records gives one atom with <mol #>, the molecular index, and <atom ##>, the atom index.

The records for an equivalent atom selection are slightly different from all other selections because a molecular system may contain more than one set of equivalent atoms. Therefore, the equivalent atom selection may include several sub-sections. Each of these sub-sections begins with a keyword, EquivalentSet #, and gives the information of the atoms in this equivalent set.

Furthermore, HyperNMR assumes that all the atoms in the molecular system are quantum atoms if there is no quantum atom selection (the first record after the keyword of QuantumSelectionBegin contains 0).

*Note:* Any single atom in a molecular system cannot be defined in more than one set of equivalent atoms.

**Options:**

This section begins with a keyword OptionsBegin, ends with OptionsEnd, and includes the following records.

**SemiEmpiricalMethod=**  
**TotalCharge=**

**SpinMultiplicity=**  
**SCFConvergenceLimit=**  
**SCFIterationLimit=**  
**SCFAccelerateConvergence=**  
**LogPrintLevel=**  
**ModifySlaterExponents=**  
**ComputeThreeCenterTerms=**  
**IncludeDOrbitalsForSecondRow=**  
**NMRFrequency=**  
**CalculateShielding=**  
**CalculateCoupling=**  
**WeakCouplingApproximation=**  
**CalculateShieldingTensor=**

The possible values for all the above keywords are listed in the section “HyperNMR Default Settings” on page 187.

## NMR Parameter

The NMR parameter section contains two sub-sections which are:

1. The first sub-section begins with the keyword of `NMRAtomicNumbersBegin` and ends with `NMRAtomicNumbersEnd` and contains the atomic numbers of the NMR atoms. The single record between these keywords consists of a list of space-separated integers, one for each NMR atom.
2. The sub-section beginning with the keyword `ShieldingCouplingBegin` and ending with `ShieldingCouplingEnd` gives the shielding constants and coupling constants. The first two records in this sub-section consist of the keyword `SHIELDING`, followed by a record containing all the isotropic shielding constants. The next line is the keyword `TENSOR`, and then two records for each NMR atom containing (i) the shielding tensor and (ii) the eigenvalues, the isotropic shielding, and the anisotropy. Finally comes the keyword `COUPLING`, and a record for each NMR atom containing the coupling constants.

## NMR Spectrum

The NMR spectrum section beginning with the keyword of `SpectraBegin` and ending with `SpectraEnd` contains three sub-sections:

1. The first sub-section contains the NMR transitions, beginning with an integer which shows the number of spectral lines. The second record in this sub-section is four words, `FREQUENCY`,

INTENSITY, DEGENERACY, and LINEWIDTH. The following records define the details of the NMR spectrum.

2. This sub-section has two records and defines the current type of NMR spectrum. The first record contains a word, SPECTRATYPE. The second record gives the type of the NMR spectrum and this type is expressed as an integer that is the atomic number of the nucleus.
3. The third sub-section gives the shielding constants of the nuclei in the reference molecules and contains two records. The first record in this sub-section is a word, REFERENCESHIELDING. The second records contains six values which correspond to the shielding constants of proton,  $^{13}\text{C}$ ,  $^{15}\text{N}$ ,  $^{15}\text{O}$ ,  $^{19}\text{F}$ , and  $^{31}\text{P}$  in the reference molecules, respectively.

## Log Files

HyperNMR stores the results of calculations in a log file if you choose Start Log on the File menu. In the Start Log dialog box, you can give the log file any file name and extension, but the extension .log is recommended. The amount of information shown in log file depends on the LogPrintLevel set up in the NMR.INI file (see page 189). The following table shows the information saved in the log file when you run calculations of NMR properties.

LogPrintLevel	Information saved in log file
0, or greater than 4 and less than 10	<p>start conditions, such as: Convergence limit, Iteration limit, total number of electrons, total number of orbitals, and total number of occupied orbitals</p> <p>SCF results calculated with nmr_shft.abp: iteration list, total energy, binding energy, isolated atom energy, electronic energy, core-core interaction energy, heat of formation, eigenvalues of molecular orbitals, atomic orbital electron populations, net charge and coordinates of each atom, dipole moments, perturbed SCF iteration list, shielding constants and components</p> <p>SCF results calculated with nmr_cupl.abp: iteration list, total energy, binding energy, isolated atom energy, electronic energy, core-core interaction energy, heat of formation, eigenvalues of molecular orbitals, atomic orbital electron populations, net charge and coordinates of each atom, dipole</p>
1	<p>All information when LogPrintLevel=0</p> <p>SCF results with nmr_shft.abp: final eigenvalues of molecular orbitals final eigenvectors of molecular orbitals</p> <p>SCF results with nmr_cupl.abp: final eigenvalues of molecular orbitals final eigenvectors of molecular orbitals</p>
2	<p>All information when LogPrintLevel=0</p> <p>SCF results with nmr_shft.abp final density matrix</p> <p>SCF results with nmr_cupl.abp final density matrix</p>
3	All information when LogPrintLevel is 0, 1, and 2



---

LogPrintLevel	Information saved in log file
4	All information when LogPrintLevel=0  SCF information with nmr_shft.abp core Hamiltonian matrix Fock matrix at each iteration eigenvalues and vectors at each iteration density matrix at each iteration  SCF information with nmr_cupl.abp core Hamiltonian matrix Fock matrix at each iteration eigenvalues and vectors at each iteration density matrix at each iteration

---

# Chapter 10

## Scientific Background

### Overview

Modern high field, multipulse Nuclear Magnetic Resonance (NMR) spectroscopy has proven to be an exceptionally powerful technique for solving many types of problems in chemistry and biochemistry. NMR spectroscopy, for example, allows chemists and biochemists to study the structure and conformation of molecules such as peptides and proteins, with unprecedented detail.<sup>1-3</sup> However, the problem of correct signal assignment and understanding the relationship between chemical shifts and molecular structure can be very difficult. Theoretical prediction of NMR spectroscopy can be very helpful to chemists and biochemists.

The *ab initio* calculation of magnetic shielding constants has been shown to be useful and accurate.<sup>1</sup> However, *ab initio* calculations are tedious and computationally time consuming, especially for large molecular systems. Traditional semi-empirical quantum mechanical methods have attracted much attention and the methodology of semi-empirical quantum mechanics has improved to the point that accurate and reliable predictions of chemical shifts and nuclear spin coupling constants are possible.<sup>4-13</sup>

Semi-empirical quantum mechanical methods have become a useful tool for predicting many chemical properties, such as molecular structures and conformations, heats of formation, ionization potentials and electron affinities. Semi-empirical methods are very powerful and attractive to many chemists and biochemists, because of simplicity and computational speed. In the past two decades, semi-empirical methods have attracted many chemists and biochemists for predicting chemical shifts and coupling constants for various molecular systems. However the conventional semi-empirical methods with traditional parameters, which are parameterized for giving good molecular structures and/or heats of

formation, have not been shown to give reliable chemical shifts, although they may give reasonably good predictions for nuclear spin-spin coupling constants. There are a lot of different parameter sets for various semi-empirical methods in the literature. Unfortunately most semi-empirical parameter sets are only good for predicting chemical shifts for a certain set of molecules. Different sets of parameters have to be applied to different sets of molecules.

HyperNMR introduces a new methodology within the semi-empirical quantum mechanical approach, the Typed Neglect of Differential Overlap (TNDO) method, Version 1 (TNDO/1) and Version 2 (TNDO/2) for computing chemical shifts. The semi-empirical parameters for TNDO/1 and TNDO/2 have been reparameterized primarily to give acceptable chemical shifts and coupling constants. Preliminary values are shown at the end of this chapter.

This chapter includes the following:

- A general discussion of the principles of nuclear magnetic resonance, focusing on chemical shifts and coupling constants;
- A description of the new TNDO/1 and TNDO/2 semi-empirical methods and parameterization;
- Details for computing chemical shifts and coupling constants within these two methods;
- A general introduction to the calculation of spin wave functions and transitions within spin states for a system of nuclear spins.

## Principles of Nuclear Magnetic Resonance

### General Principles

NMR is a term used to describe the interaction between an oscillating magnetic field and certain types of nuclei positioned in a static magnetic field. The static magnetic field splits the nuclear spin energy levels, and the oscillating magnetic field induces transitions between these nuclear spin energy levels and produces the NMR spectrum. The interaction can occur only if the magnetic field is oscillating at specific frequencies which depend upon the strength of the static magnetic field.

Suppose a magnetic field with a strength of  $B^0$  is applied to a nucleus. For an isolated nucleus,  $B^0$  is just the magnitude of the applied field. However for a nucleus in a molecule, the magnetic field is due not only to the applied magnetic field, but also comes from the magnetic fields produced by the electrons and other nuclei.

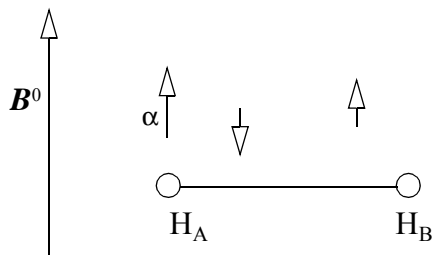
All electrons have spin angular momentum and all but s-type electrons have orbital angular momentum; magnetic dipole moments are associated with these angular momenta. However, in all but a few molecules, the electronic ground states are nondegenerate with all electron spins paired and there is no net electronic spin or orbital angular momentum. Hence, in the absence of an external magnetic field, the electrons contribute nothing to the magnetic field felt by a nucleus.

When an external magnetic field is applied, however, the electronic charge distribution is perturbed, introducing an electronic contribution to the magnetic field (induced magnetic field) at the nucleus. This induced magnetic field, at the nucleus  $A$ , is opposite in direction to the applied external magnetic field and is proportional to the magnitude of  $B^0$ , according to Lenz's law. So the magnitude of the magnetic field at nucleus  $A$  can be expressed as

$$B_A = B^0 - B_A^1 = B^0(1 - \sigma_A) \quad (6-1)$$

where the proportionality constant  $\sigma_A$  is called the shielding constant (or the screening constant) for nucleus  $A$ .

The mechanism of spin-spin coupling is slightly more complicated than the shielding. Consider the hypothetical case where two protons  $H_A$  and  $H_B$  in a molecule are separated by a single bond and are in markedly different chemical environments. In an applied magnetic field  $B^0$ , the components of the magnetic moment of the nuclei in the field direction may either be parallel ( $m_I = +\frac{1}{2}$ ) or antiparallel ( $m_I = -\frac{1}{2}$ ) to the applied field. The former case is referred to as the  $\alpha$  spin state and the latter as the  $\beta$  state. When the nucleus is in a particular spin state the electrons are polarized so that the adjacent electron spins are paired. If the proton  $H_A$  is in the  $\alpha$  spin state then the two electron spins will be polarized as follows:



Since the field associated with the electron adjacent to nucleus  $H_B$  will be in the same direction as the applied field  $B^0$ , the nucleus  $H_B$  will have an increased net field. When nucleus  $H_A$  is in the  $\beta$  spin state nucleus  $H_B$  will have a reduced net field. Conversely, depending upon whether nucleus  $H_B$  is in the  $\alpha$  or  $\beta$  spin state, nucleus  $H_A$  will experience an induced field supporting or opposing the applied field. The overall result is that there are two values for the total effective field sensed by nuclei  $H_A$  and  $H_B$ . This is the nuclear spin coupling,  $J_{AB}$ . The coupling interaction between the nuclei is a molecular electronic effect; the magnitude of the coupling, i.e., the value of  $J$ , is not, in first order, affected by the strength of the applied magnetic field  $B^0$ .

## Magnetic Shielding

The quantum mechanical treatment of magnetic shielding is discussed below.

The interaction of an external magnetic field  $B^0$  with a nuclear magnetic moment  $\mu_A$  inside a molecule, as observed in nuclear magnetic resonance, can be described by the Zeeman energy change,

$$\Delta E = -\mu_A \cdot B \quad (6-2)$$

where  $B$  is the effective magnetic field felt by the nuclear moment. This effective magnetic field arises because the electrons are influenced by the external magnetic field and give rise to an induced magnetic field at the site of the nucleus A,

$$B_j^1 = -\sigma_{ij}^A B_i^0 \quad (6-3)$$

where  $\sigma_A$  is called the shielding tensor of nucleus A. So the energy change because of the external magnetic field may be written

$$\Delta E = -\mu_A \cdot \mathbf{B}^0 + \mu_A \cdot \hat{\sigma}^A \cdot \mathbf{B}^0 \quad (6-4)$$

where the first term describes the direct interaction between the external magnetic field and the nuclear moment and the second term describes the electron coupled interaction which may be written as a second-order energy change,

$$-\sigma_{ij}^A = \left[ \frac{\partial^2 E(\mathbf{B}^0, \mu_A)}{\partial \mathbf{B}_i^0 \partial \mu_{Aj}} \right]_{\mu_A = \mathbf{B}^0 = 0} \quad (6-5)$$

arising from a consideration of the energy of a molecule in the presence of an external magnetic field and a perturbing nuclear moment  $\mu_A$ .

In the Hartree-Fock method, the electronic energy of a molecule in the presence of the external magnetic field  $\mathbf{B}^0$  and nuclear moment  $\mu_A$  is given as the expectation value of the appropriate Hamiltonian operator using a single determinant wave function. In terms of the doubly occupied molecular orbitals  $\psi_i$ , this expression is

$$E(\mathbf{B}^0, \mu_A) = \sum_{i=1}^{N/2} 2 \langle \psi_i | \hat{h} | \psi_i \rangle + \langle \psi_i | \hat{g} | \psi_i \rangle \quad (6-6)$$

where the one-electron Hamiltonian operator,  $\hat{h}$ , is

$$\hat{h} = \frac{1}{2m} \left[ -i\hbar \nabla + \frac{e}{c} \mathbf{A} \right]^2 - \sum_k^{\text{atoms}} \frac{e^2 Z_k}{|\mathbf{r}_k|} \quad (6-7)$$

and the two-electron Hamiltonian operator,  $\hat{g}$ , is

$$\hat{g} = \sum_j^{N/2} \langle \psi_j(2) \left| \frac{2 - \hat{P}_{12}}{r_{12}} \right| \psi_j(2) \rangle \quad (6-8)$$

In Equation (6-7), the vector potential  $\mathbf{A}$  is

$$\mathbf{A} = \frac{1}{2} \mathbf{B}^0 \times \mathbf{r} + \frac{\mu_A \times \mathbf{r}_A}{|\mathbf{r}_A|^3} \quad (6-9)$$

and is due to the uniform external magnetic field  $\mathbf{B}^0$  and the magnetic moment of nucleus A, i.e.,  $\mu_A$ . The number of electrons is  $N$ , and  $Z_k$  is the charge of the  $k^{\text{th}}$  nucleus. The vectors  $\mathbf{r}$  and  $\mathbf{r}_k$  are the distance vectors to the electron from the arbitrary origin of the coordinate system and nucleus  $k$ , respectively, and  $\hat{P}_{12}$  is the permutation operator.

A common difficulty in the calculation of magnetic properties is that the usual wave functions do not guarantee gauge invariance; i.e., in the simplest case, the results may depend on the position of the molecule in the Cartesian coordinate frame.

There have been several methods proposed for computing magnetic shielding constants. Of those, only the so-called Gauge-Invariant Atomic Orbitals (GIAO) method has been shown to be independent of the choice of origin of the coordinate system.

With GIAO, the molecular orbitals,  $\psi_i$ , are still expressed as a linear combination of atomic orbitals (LCAO),  $\phi_\mu$ , as in the Hartree-Fock method, that is,

$$\psi_i = \sum_{\mu}^m c_{\mu i} \phi_{\mu} \quad (6-10)$$

where  $m$  is the total number of basis functions. But the atomic orbitals (basis functions) used here are not the same as those used in the ordinary Hartree-Fock method. The relationship between the basis functions,  $\phi_{\mu}$ , used in the GIAO method and the basis functions,  $\phi_{\mu}^0$ , used in the ordinary Hartree-Fock method is

$$\phi_{\mu} = f_{\mu} \phi_{\mu}^0 \quad (6-11)$$

where  $f_{\mu}$  is the gauge factor defined as

$$f_{\mu} = \exp\left(-\frac{ie}{\hbar c} \mathbf{A}_{\mu} \cdot \mathbf{r}_{\mu}\right), \quad (6-12)$$

the vector potential  $\mathbf{A}_{\mu}$  is

$$\mathbf{A}_{\mu} = \frac{1}{2} \mathbf{B}^0 \times \mathbf{R}_{\mu}, \quad (6-13)$$

and vector  $\mathbf{R}_{\mu}$  represents the distance vector from the arbitrary origin of the molecular coordinate system to the atom on which the basis function  $\phi_{\mu}$  is located.

The total electronic energy expression of Equation (6-6) in the LCAO approximation becomes

$$E = \sum_1^{N/2} c_i^\dagger [2\mathbf{H} + \mathbf{G}] c_i \quad (6-14)$$

with the matrix elements of  $\mathbf{H}$  and  $\mathbf{G}$  being

$$H_{\mu\nu} = \langle \phi_\mu | \hat{h} | \phi_\nu \rangle \quad (6-15)$$

and

$$G_{\mu\nu} = \sum_\lambda^m \sum_\sigma^m R_{\lambda\sigma} [2(\mu\nu|\lambda\sigma) - (\mu\sigma|\lambda\nu)] , \quad (6-16)$$

the two-electron integrals are

$$(\mu\nu|\lambda\sigma) = \iint \phi_\mu^*(1)\phi_\nu(1) \frac{1}{r_{12}} \phi_\lambda^*(2)\phi_\sigma(2) d\tau_1 d\tau_2 , \quad (6-17)$$

and  $\mathbf{R}$  is the first-order density matrix with its elements given by

$$R_{\lambda\sigma} = \sum_i^{N/2} c_{\lambda i}^* c_{\sigma i} . \quad (6-18)$$

Substituting Equation (6-14) into Equation (6-5), using the Hellman-Feynman theorem, and remembering that the basis functions and two-electron terms in Equation (6-16) do not depend on the nuclear magnetic moment, the (i, j) component of the shielding tensor of atom A can be expressed as

$$\sigma_{ij}^A = \left[ \frac{\partial^2 E(\mathbf{B}^0, \boldsymbol{\mu}_A)}{\partial B_i^0 \partial \mu_{Aj}} \right]_{\boldsymbol{\mu}_A = \mathbf{B}^0 = 0} = \sigma_{ij}^d(A) + \sigma_{ij}^p(A) \quad (6-19)$$

where the diamagnetic contribution,  $\sigma_{ij}^d(A)$ , and the paramagnetic contributions,  $\sigma_{ij}^p(A)$ , are

$$\sigma_{ij}^d(A) = 2 \left[ \mathbf{R}(0) \left( \frac{\partial^2 \mathbf{H}}{\partial B_i^0 \partial \mu_{Aj}} \right)_{\boldsymbol{\mu}_A = \mathbf{B}^0 = 0} \right] \quad (6-20)$$

and



$$\sigma_{ij}^p(A) = 2 \left[ \left( \frac{\partial \mathbf{R}}{\partial B_i^0} \right)_{\mu_A = \mathbf{B}^0 = 0} \left( \frac{\partial \mathbf{H}}{\partial \mu_{Aj}} \right)_{\mu_A = \mathbf{B}^0 = 0} \right] \quad (6-21)$$

respectively, and  $\mathbf{R}(0)$  is the density matrix without the presence of the magnetic field (unperturbed density matrix). With Equations (6-7) and (6-11), the one-electron terms in Equation (6-15) are

$$H_{\mu\nu} = \exp\left(-\frac{ie}{2hc}\mathbf{B}^0 \cdot \mathbf{R}_\mu \times \mathbf{R}_\nu\right) \int \exp\left(-\frac{ie}{2hc}\mathbf{B}^0 \times \mathbf{R}_{\mu\nu} \cdot \mathbf{r}_\nu\right) \\ \times \phi_\mu^0 \left[ \frac{1}{2m} \left( -i\hbar\nabla + \frac{e}{2c}\mathbf{B}^0 \times \mathbf{r}_\nu + \frac{e\boldsymbol{\mu}_A \times \mathbf{r}_A}{c|\mathbf{r}_A|^3} \right)^2 - \sum_{\mathbf{k}}^{atoms} \frac{e^2 Z_{\mathbf{k}}}{|\mathbf{r}_{\mathbf{k}}|} \right] \phi_\nu^0 d\tau \quad (6-22)$$

where  $\mathbf{R}_{\mu\nu} = \mathbf{R}_\nu - \mathbf{R}_\mu$  and the remaining quantities have been given earlier. The origin-dependent term has been factored outside the integral.

By using the usual London approximation (in the current case this is equivalent to expanding the exponential inside the integral and keeping only the first term), Equation (6-22) can be rewritten:

$$\begin{aligned}
H_{\mu\nu} = \exp\left(-\frac{ie}{2hc}\mathbf{B}^0 \cdot \mathbf{R}_\mu \times \mathbf{R}_\nu\right) & \left\{ H_{\mu\nu}^0 \right. \\
& + \sum_{\alpha} B_{\alpha} \left\langle \phi_{\mu}^0 \left| -\frac{ieh}{2mc} (\mathbf{r}_\nu \times \nabla)_{\alpha} \right| \phi_{\nu}^0 \right\rangle \\
& + \sum_{\alpha} \mu_{A\alpha} \left\langle \phi_{\mu}^0 \left| \frac{ieh}{2mc} \frac{(\mathbf{r} \times \nabla)_{\alpha}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \\
& + \sum_{\alpha} \sum_{\beta} \mu_{A\alpha} B_{\beta} \left\langle \phi_{\mu}^0 \left| -\frac{e^2}{2mc^2} \frac{\mathbf{r}_\nu \mathbf{r}_A - r_{\nu\alpha} r_{A\beta}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \\
& \left. + \text{higher order terms} \right\} \quad (6-23)
\end{aligned}$$

with  $\alpha, \beta = x, y,$  and  $z$ . The higher order terms in the above equation don't make any contributions to the chemical shifts.

Setting the exponential appearing in front of the integral in Equation (6-23) to be one and substituting Equation (6-23) into Equations (6-20) and (6-21), the diamagnetic shielding becomes

$$\sigma_{ij}^d(A) = \frac{e^2}{mc^2} \sum_{\mu} \sum_{\nu} R_{\mu\nu}(0) \left\langle \phi_{\mu}^0 \left| \frac{\mathbf{r}_\nu \cdot \mathbf{r}_A \delta_{ij} - r_{\nu j} r_{A i}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-24)$$

and the paramagnetic shielding is

$$\sigma_{ij}^p(A) = -\frac{2ieh}{mc} \sum_{\mu} \sum_{\nu} \left( \frac{\partial R_{\mu\nu}}{\partial B_i^0} \right)_{B^0=0} \left\langle \phi_{\mu}^0 \left| \frac{(\mathbf{r} \times \nabla)_j}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-25)$$

with the summation over all the basis functions. These two equations give a full expression for the magnetic shielding tensor with the unperturbed density matrix elements,  $R_{\mu\nu}(0)$ , and the first derivative of the perturbed density matrix elements,  $(\partial R_{\mu\nu}/\partial B_i^0)$ , and with the inclusion of one-center, two-center, and three-center contributions.

## Nuclear Spin Coupling

Field-independent splittings in the nuclear magnetic resonance spectra of liquids, arising from nuclear spin coupling, were discovered about four decades ago. The interaction energy between nuclear spins was found to be expressible as

$$\Delta E = \frac{2\pi}{h} J_{AB} \mathbf{I}_A \cdot \mathbf{I}_B \quad (6-26)$$

where  $J_{AB}$  is a coupling constant (in cycles/sec) between nuclei A and B having nuclear spin (in units of  $h$ ) of  $\mathbf{I}_A$  and  $\mathbf{I}_B$ , respectively. The coupling  $J_{AB}$  is really a tensor, the experimentally observed value for liquids being the average over all orientations of the rapidly tumbling molecules, and equal to one-third the trace of the tensor  $\hat{J}_{AB}$ . It is convenient to define a reduced coupling tensor  $\hat{K}_{AB}$  such that Equation (6-26) can be rewritten as

$$\Delta E = \boldsymbol{\mu}_A \cdot \hat{K}_{AB} \cdot \boldsymbol{\mu}_B \quad (6-27)$$

where  $\hat{K}_{AB}$  is independent of the magnitude of the nuclear moments  $\boldsymbol{\mu}_A$  and  $\boldsymbol{\mu}_B$ , and the relationship between the coupling tensor and reduced coupling tensor is

$$\hat{K}_{AB} = \frac{2\pi}{h\gamma_A\gamma_B} \hat{J}_{AB} \quad (6-28)$$

Equation (6-28) doesn't include the direct dipole-dipole interaction between nuclear moments but describes the electron coupled interaction and may be represented as the second-order energy change

$$\left( \hat{K}_{AB} \right)_{ij} = \left[ \frac{\partial^2 E(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)}{\partial \mu_{Ai} \partial \mu_{Bj}} \right] \quad (6-29)$$

arising from a consideration of the energy of a molecule in the presence of two perturbing nuclear moments  $\boldsymbol{\mu}_A$  and  $\boldsymbol{\mu}_B$ .

To evaluate the nuclear-coupling energy mediated by the electronic system, it is necessary to treat those parts of the Hamiltonian which depend on  $\boldsymbol{\mu}_A$  as a perturbation on the electronic system and carry the perturbation calculation to second order (for molecules in singlet ground states with no resultant orbital angular momentum, all first-order terms vanish). In Equation (6-7), all these perturbation terms have been neglected because these terms

do not make any contributions to the magnetic shielding. They do, however, make contributions to the total energy of the molecular system and nuclear spin coupling.

The second-order terms are:

1. Spin-Orbital interaction term,  $H^{SO}$ , which involves the interaction of the electronic orbital motion with the nuclear magnetic moments.
2. Dipole-Dipole interaction term,  $H^{DD}$ .
3. Fermi-Contact term,  $H^{FC}$ , which involves the Fermi contact interaction of the electron spins.

The corresponding Hamiltonians for these three terms are

$$\hat{H}^{SO} = \frac{2\beta}{h} \sum_K \sum_A \frac{\hat{L}_{KA} \cdot \boldsymbol{\mu}_A}{r_{KA}^3} \quad (6-30)$$

$$\hat{H}^{DD} = 2\beta \sum_K \sum_A \left[ 3 \frac{(\hat{s}_K \cdot \mathbf{r}_{KA})(\boldsymbol{\mu}_A \cdot \mathbf{r}_{KA})}{r_{KA}^5} - \frac{\hat{s}_K \cdot \boldsymbol{\mu}_A}{r_{KA}^3} \right] \quad (6-31)$$

and

$$\hat{H}^{FC} = \frac{16\pi}{3} \beta \sum_K \sum_A \delta(r_{KA}) \hat{s}_K \cdot \boldsymbol{\mu}_A \quad (6-32)$$

respectively, where  $\beta$  is the Bohr magneton,  $\hat{L}_{KA}$  is the operator for the orbital angular momentum of electron K about nucleus A, and  $\hat{s}_K$  is the spin angular momentum operator for electron K. In Equations (6-30) – (6-32), the sums K and A are over all the electrons and nuclei in the molecular system, respectively.

The most important term leading to a nuclear spin coupling is the Fermi-contact Hamiltonian  $H^{FC}$ . It has been shown using the Hellman-Feynman theorem that the second-order perturbation to the electronic energy of the molecule due to the nuclear spins  $\mathbf{I}_A$  at nucleus A and  $\mathbf{I}_B$  at nucleus B can be expressed as

$$\begin{aligned}
 \mathbf{K}_{AB} &= \left[ \frac{\partial^2 E(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)}{\partial \boldsymbol{\mu}_A \partial \boldsymbol{\mu}_B} \right] \\
 &= \left[ \left( \frac{\partial}{\partial \boldsymbol{\mu}_A} \right) \langle \langle \psi(\boldsymbol{\mu}_A) | \hat{H}_B^{(1)} | \psi(\boldsymbol{\mu}_A) \rangle \rangle \right]_{\boldsymbol{\mu}_A = 0}
 \end{aligned} \tag{6-33}$$

with

$$\hat{H}_B^{(1)} = \frac{\partial \hat{H}_B^{FC}}{\partial \boldsymbol{\mu}_B} \tag{6-34}$$

The molecular wave function is written in an unrestricted form:

$$\begin{aligned}
 \psi &= (2n!)^{-1/2} \left| \psi_1^\alpha(1)\alpha(1) \cdots \psi_n^\alpha(n)\alpha(n) \right. \\
 &\quad \left. \times \psi_1^\beta(n+1)\beta(n+1) \cdots \psi_n^\beta(2n)\beta(2n) \right|
 \end{aligned} \tag{6-35}$$

Here the spatial molecular orbitals  $\psi_i^{(\alpha)}$  and  $\psi_i^{(\beta)}$  are different (because of the spin operators in the Hamiltonian), but the total number of the  $\alpha$  electrons are still equal to the total number of  $\beta$  electrons. With this wave function, the expectation value of  $\hat{H}_B^{(1)}$  in Equation (6-33) reduces to

$$\begin{aligned}
 \mathbf{K}_{AB} &= \sum_{i=1}^n \left[ \left\langle \psi_i^\alpha(1)\alpha(1) \left| \hat{H}_B^{(1)} \right| \psi_i^\alpha(1)\alpha(1) \right\rangle \right. \\
 &\quad \left. + \left\langle \psi_i^\beta(1)\beta(1) \left| \hat{H}_B^{(1)} \right| \psi_i^\beta(1)\beta(1) \right\rangle \right] \\
 &= \sum_{i=1}^n \left[ \left\langle \psi_i^\alpha(1) \left| \hat{\mathcal{H}}_B^{(1)} \right| \psi_i^\alpha(1) \right\rangle - \left\langle \psi_i^\beta(1) \left| \hat{\mathcal{H}}_B^{(1)} \right| \psi_i^\beta(1) \right\rangle \right]
 \end{aligned} \tag{6-36}$$

where

$$\hat{\mathcal{H}}_B^{(1)} = \frac{8\pi}{3} \beta \delta(r_B) \tag{6-37}$$

The molecular orbitals,  $\psi_i$ , are still expressed as a linear combination of atomic orbitals (LCAO),  $\phi_v$ , as in the ordinary Hartree-Fock method. That is,

$$\Psi_i^\alpha = \sum_{\nu}^m c_{\nu i}^\alpha \phi_\nu \quad \text{and} \quad \Psi_i^\beta = \sum_{\nu}^m c_{\nu i}^\beta \phi_\nu \quad (6-38)$$

Then Equation (6-36) becomes

$$\left\langle \Psi(\mu_A) \left| \hat{H}_B^{(1)} \right| \Psi(\mu_A) \right\rangle = \sum_{\mu\nu} (P_{\mu\nu}^\alpha - P_{\mu\nu}^\beta) \hat{\mathcal{H}}_{B\mu\nu}^{(1)} \quad (6-39)$$

where  $P_{\mu\nu}^\alpha$  and  $P_{\mu\nu}^\beta$  are the density matrix elements of the  $\alpha$  and  $\beta$  electrons respectively for the perturbed system and the Hamiltonian matrix element is

$$\hat{\mathcal{H}}_{B\mu\nu}^{(1)} = \left\langle \phi_\mu \left| \hat{\mathcal{H}}_B^{(1)} \right| \phi_\nu \right\rangle \quad (6-40)$$

Substituting Equation (6-39) into Equation (6-33), the reduced coupling tensor can be written as

$$K_{AB} = \sum_{\mu\nu} \hat{\mathcal{H}}_{B\mu\nu}^{(1)} \left[ \frac{\partial \rho_{\mu\nu}^{(1)}}{\partial \mu_A} \right]_{\mu_A=0} \quad (6-41)$$

where the reduced spin density matrix is

$$\rho_{\mu\nu}^{(1)} = P_{\mu\nu}^\alpha - P_{\mu\nu}^\beta \quad (6-42)$$

and is due to the Fermi-contact perturbation.

## Typed Neglect of Differential Overlap Method

The conventional semi-empirical quantum mechanical methods have been very well tested by many scientists and are widely used for modeling and simulating molecular systems. They have been quite satisfactory to most chemists and biochemists in many situations. However semi-empirical methods with traditional parameters have been shown to give magnetic shielding constants that compare poorly with experimental data.

To improve on calculated magnetic shielding constants and nuclear spin coupling constants, HyperNMR uses two new semi-empirical methods. Preliminary parameterization has been performed and the new methods show considerable improvement for

computing chemical shifts in particular. Parameters for spin-spin coupling have not been revised.

In order to introduce the HyperNMR semi-empirical methods, a simple overview of the conventional semi-empirical methods is given in this section, and then the new methods and parameterization are introduced and discussed.

## Conventional Semi-Empirical Methods

The conventional semi-empirical Self-Consistent-Field (SCF) Molecular Orbital (MO) procedures available for predicting the molecular energetic and structural features have attempted to provide theoretical frameworks which are simple and inexpensive from the computational point of view and at the same time are reliable and accurate enough to be of chemical utility. They are based on integral approximations which simplify drastically the LCAO SCF MO treatment for molecules and they employ adjustable parameters in order to enhance the accuracy of prediction. The literature contains two main groups of semi-empirical SCF MO procedures. They have been very well parameterized for predicting molecular energetic and structural properties and are widely used by chemists and biochemists. The first group is based on the Neglect of Differential Overlap (NDO) approximation, such as Pople's<sup>14</sup> CNDO (Complete Neglect of Differential Overlap) and INDO (Intermediate Neglect of Differential Overlap), Zerner's<sup>15</sup> ZINDO/1 (Zerner's non-spectroscopic version of the Intermediate Neglect of Differential Overlap) and ZINDO/S (Zerner's spectroscopic version of the Intermediate Neglect of Differential Overlap), Dewar's<sup>16</sup> MINDO (Modified Intermediate Neglect of Differential Overlap), and Jug's<sup>17</sup> SINDO1 (Symmetrically orthogonalized INDO). The second main group is based on the Neglect of Diatomic Differential Overlap (NDDO) approximation, such as Dewar's<sup>18,19</sup> MNDO (Modified Neglect of Diatomic Overlap) and AM1 (Austin Model 1) and Stewart's<sup>20</sup> PM3 (a reparameterization of AM1). The NDDO methods are slightly more complicated and give better energetic results than the NDO methods. The following two sections will review the NDO and NDDO mathematical approximations.

### NDO

The NDO group comes from the basic approximation

$$\begin{aligned} & \iint \phi_{\mu}^A(1) \phi_{\nu}^B(1) \frac{1}{r_{12}} \phi_{\lambda}^C(2) \phi_{\sigma}^D(2) d\tau_1 d\tau_2 \\ & \approx \delta^{AB} \delta^{CD} \iint \phi_{\mu}^A(1) \phi_{\nu}^A(1) \frac{1}{r_{12}} \phi_{\lambda}^C(2) \phi_{\sigma}^C(2) d\tau_1 d\tau_2 \end{aligned} \quad (6-43)$$

where

$$\delta^{AB} = \begin{cases} 1 & \text{if } A = B ; \\ 0 & \text{if } A \neq B . \end{cases} \quad (6-44)$$

Further approximations must be made in the NDO scheme for computing the right-hand side of Equation (6-43). However different approximations may still be made in the NDO group. For example, Equation (6-43) reduces to

$$\begin{aligned} & \iint \phi_{\mu}^A(1) \phi_{\nu}^B(1) \frac{1}{r_{12}} \phi_{\lambda}^C(2) \phi_{\sigma}^D(2) d\tau_1 d\tau_2 \\ & \approx \delta^{AB} \delta^{CD} \delta^{\mu\nu} \delta^{\lambda\sigma} \iint \phi_{\mu}^A(1) \phi_{\mu}^A(1) \frac{1}{r_{12}} \phi_{\lambda}^C(2) \phi_{\lambda}^C(2) d\tau_1 d\tau_2 \\ & \approx \delta^{AB} \delta^{CD} \delta^{\mu\nu} \delta^{\lambda\sigma} \iint [\phi_{\mu}^A(1)]^2 \frac{1}{r_{12}} [\phi_{\lambda}^C(2)]^2 d\tau_1 d\tau_2 \end{aligned} \quad (6-45)$$

in the CNDO method. The resultant Fock matrix elements in the CNDO method are

$$F_{\mu\mu} = U_{\mu\mu} + \left( P_{AA} - \frac{1}{2} P_{\mu\mu} \right) \gamma_{AA} + \sum_{B(\neq A)} (P_{BB} - Z_B) \gamma_{AB} \quad (6-46)$$

for the diagonal terms with  $\mu$  on atom A and

$$F_{\mu\nu} = \beta_{AB}^0 S_{\mu\nu} - \frac{1}{2} P_{\mu\nu} \gamma_{AB} \quad (6-47)$$

for the off-diagonal terms with  $\mu \neq \nu$  ( $\mu$  and  $\nu$  may or may not be on the same atom). In Equations (6-46) and (6-47),  $\mathbf{U}$ ,  $\mathbf{P}$ , and  $\mathbf{S}$  are the one-center one-electron part of the core Hamiltonian, the valence electron density matrix, and the overlap matrix, respectively.  $\beta_{AB}^0$  is the bonding parameter between atoms A and B. The Coulomb integrals are



$$\gamma_{AB} = \iint [\phi_s^A(1)]^2 \frac{1}{r_{12}} [\phi_s^B(2)]^2 d\tau_1 d\tau_2 \quad (6-48)$$

and involve only the two-center two-electron integrals for s-orbitals.

## NDDO

The NDDO approximation neglects differential overlap only for atomic orbitals on different atoms. The principal extra feature at this level of approximation is the retention of dipole-dipole interactions, since the integrals of type

$$\iint \phi_s^A(1) \phi_p^A(2) \frac{1}{r_{12}} \phi_s^B(1) \phi_p^B(2) d\tau_1 d\tau_2, \text{ which are roughly propor-}$$

tional to  $1/r_{12}^3$ , are included. These integrals may be computed directly from the atomic orbitals or chosen empirically. The Fock matrix elements in the NDDO approximation are

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{B(\neq A)} \sum_{\lambda,\sigma} P_{\lambda\sigma}(\mu\nu|\lambda\sigma) - \frac{1}{2} \sum_{\lambda,\sigma} P_{\lambda\sigma}(\mu\nu|\lambda\sigma) \quad (6-49)$$

with  $\mu, \nu$  on atom A and

$$F_{\mu\nu} = H_{\mu\nu} - \frac{1}{2} \sum_{\lambda}^A \sum_{\sigma}^B P_{\lambda\sigma}(\mu\nu|\lambda\sigma) \quad (6-50)$$

with  $\mu$  on atom A and  $\nu$  on atom B. In Equations (6-49) and (6-50),  $H_{\mu\nu}$  are the matrix elements of the core Hamiltonian.

## Typed Semi-Empirical Methods

As shown in the literature, the conventional semi-empirical SCF MO methods with the original parameters give poor magnetic shielding results, but reparameterization greatly improves the calculated results. However, the largest shortcoming of the conventional semi-empirical theories with these new parameters is the fact that although one set of parameters may be good for reproducing one class of chemical property, reparameterization to reproduce a second class may void the ability to apply the theory to the first. HyperNMR proposes two new semi-empirical methods,

Typed Neglect of Differential Overlap (TNDO), Version 1 (TNDO/1) and Version 2 (TNDO/2), and introduces a set of parameters specifically reproduce magnetic shielding constants and spin-spin coupling constants.

The basic idea of the TNDO/1 and TNDO/2 methods is to allow different parameters for different chemical environments by using *atom types*, instead of *elements* as in the conventional semi-empirical SCF MO methods. In the conventional semi-empirical methods, each element is treated as unique. This means that there is, for instance, only one carbon rather than many carbons, such as  $sp^3$  carbon,  $sp^2$  carbon,  $sp$  carbon, and aromatic carbon. Such classification of atom types (with different chemical environments) is used in molecular mechanics methods. The conventional semi-empirical methods try to apply Hartree-Fock SCF theory and various approximations to distinguish differences in chemical properties of elements in different environments. Actually, the conventional semi-empirical methods have generally reached this goal for energetic and structural properties of molecules, but fail in reaching this goal for calculating the magnetic properties of molecules. Hence HyperNMR uses atom types instead of atomic number to distinguish elements in different environments. HyperNMR introduces different parameters for different atom types such as  $sp^3$  and  $sp^2$  carbon atoms.

The current version of HyperNMR introduces TNDO/1 and TNDO/2, which are described below:

## TNDO/1

The TNDO/1 method comes from applying the idea of atom types to Pople's CNDO method. Standard Hartree-Fock methodology is used in TNDO/1; the conventional semi-empirical Roothaan equations are solved within the CNDO approximation,

$$F c_i = c_i \varepsilon_i \quad (6-51)$$

where  $F$  is the Fock matrix, and  $c_i$  and  $\varepsilon_i$  are the eigenvector and eigenvalue of the  $i^{\text{th}}$  molecular orbital, respectively.

The Fock matrix elements are given by

$$F_{\mu\mu} = U_{\mu\mu}^{TT} + \left( P_{AA} - \frac{1}{2} P_{\mu\mu} \right) \gamma_{AA}^{TT} + \sum_{B(\neq A)} (P_{BB} - Z_B) \gamma_{AB}^{TT'} \quad (6-52)$$

with  $\mu$  on atom A for the diagonal elements and

$$F_{\mu\nu} = \beta_{AB}^{TT'} S_{\mu\nu}^{TT'} - \frac{1}{2} P_{\mu\nu} \gamma_{AB}^{TT'} \quad (6-53)$$

with  $\mu \neq \nu$  ( $\mu$  and  $\nu$  may or may not be on the same atom) for the off-diagonal elements. In Equations (6-52) and (6-53),  $\mathbf{U}^{TT}$ ,  $\mathbf{P}$ , and  $\mathbf{S}^{TT'}$  are the one-center one-electron part of the core Hamiltonian, the valence electron density matrix, and the overlap matrix,

respectively.  $\beta_{AB}^{TT'}$  is the bond parameter between atoms  $A$  with

type  $T$  and  $B$  with type  $T'$ .  $\gamma_{AB}^{TT'}$  is the s-orbital Coulomb integral between atoms  $A$  and  $B$ . The matrix  $\mathbf{U}$  is

$$U_{\mu\mu}^{TT} = -\frac{1}{2}(I_{\mu}^T + A_{\mu}^T) - \left(Z_A - \frac{1}{2}\right) \gamma_{AB}^{TT} \quad (6-54)$$

where the ionization potential is  $I_{\mu}^T$  and electron affinity is  $A_{\mu}^T$  for orbital  $\mu$  of atom  $A$  with atom type  $T$ . The density matrix and bonding parameters are

$$P_{\mu\nu} = \sum_{i=1}^{occu} c_{\mu i} c_{\nu i} \quad (6-55)$$

$$\beta_{AB}^{TT'} = \frac{K}{2}(\beta_A^T + \beta_B^{T'}) \quad (6-56)$$

with bonding parameters  $\beta_A^T$  and  $\beta_B^{T'}$  for atom  $A$  with atom type  $T$  and atom  $B$  with type  $T'$ . The Coulomb integral matrix is

$$\gamma_{AB}^{TT'} = \iint [\phi_s^A(T, 1)]^2 \frac{1}{r_{12}} [\phi_s^B(T', 2)]^2 d\tau_1 d\tau_2 \quad (6-57)$$

In Equation (6-56), the constant  $K$  is 0.75 for the second long row elements and 1 for all other elements.

Equation (6-51) with Equations (6-54) – (6-57) is solved self-consistently until the required accuracy of the wave function is satisfied.

## TNDO/2

The TNDO/1 method discussed in the previous section introduces electron-electron interaction in the simplest manner. It does not make adequate allowance, however, for the different interactions

that actually take place between two electrons with parallel or antiparallel spins, particularly if they are on the same atom. Hence a TNDO/2 method is proposed by applying the idea of atom types to Pople's INDO method which retains monatomic differential overlap, but only in one-center integrals. This is a less approximate method than TNDO/1, but is not as accurate a theory as one which retains monatomic differential overlap completely (NDDO). TNDO/1 and TNDO/2 are very closely related. The general expressions for the unrestricted Fock matrix elements are

$$F_{\mu\mu}^{\alpha} = U_{\mu\mu}^{TT} + \sum_{\lambda}^A [P_{\lambda\lambda}(\mu\mu|\lambda\lambda) - P_{\lambda\lambda}^{\alpha}(\mu\lambda|\mu\lambda)] \quad (6-58)$$

$$+ \sum_{B(\neq A)} (P_{BB} - Z_B) \gamma_{AB}^{TT'}$$

for the diagonal elements with  $\mu$  on atom A and

$$F_{\mu\mu}^{\alpha} = (2P_{\mu\nu} - P_{\mu\nu}^{\alpha})(\mu\nu|\mu\nu) - P_{\mu\nu}^{\alpha}(\mu\mu|\nu\nu) \quad (6-59)$$

for the off-diagonal elements with  $\mu, \nu$  on atom A, but  $\mu \neq \nu$  and

$$F_{\mu\nu}^{\alpha} = \frac{K}{2}(\beta_A^T + \beta_B^{T'}) S_{\mu\nu}^{TT'} - P_{\mu\nu}^{\alpha} \gamma_{AB}^{TT'} \quad (6-60)$$

for the off-diagonal elements with  $\mu$  on atom A and  $\nu$  on atom B. Similar expressions can be developed for the  $\beta$  electrons.

Comparing the Fock matrix elements in Equations (6-58) – (6-60) in TNDO/2 with those in Equations (6-52) and (6-53) in TNDO/1, it can be seen that the extra terms in TNDO/2 are the one-center two-electron integrals. They are  $(\mu\mu|\mu\mu)$ ,  $(\mu\mu|\nu\nu)$ , and  $(\mu\nu|\mu\nu)$  with  $\mu \neq \nu$ .

Using the Slater-Condon notation, these integrals are

$$(ss|ss) = (ss|xx) = F^0 = \gamma_{AA} \quad (6-61)$$

$$(sx|sx) = \frac{1}{3}G^1 \quad (6-62)$$

$$(xy|xy) = \frac{3}{25}F^2 \quad (6-63)$$

$$(xx|xx) = F^0 + \frac{4}{25}F^2 \quad (6-64)$$

$$(xx|yy) = F^0 - \frac{2}{25}F^2 \quad (6-65)$$

and similar expressions for  $(ss|zz)$ , etc. In HyperNMR, the values for  $F^2$  and  $G^1$  are chosen empirically and the integrals  $F^0$  are calculated theoretically from Slater atomic orbitals as in INDO.  $F^0$  may thus depend on the atom types if the Slater exponents are different for different types of the same atom.

The expression for the monatomic core integrals,  $U_{\mu\mu}$ , in TNDO/2 differ from those in TNDO/1 because of the inclusion of the monatomic one-center two-electron integrals in TNDO/2. From the energy expression for the average state of X associated with the configuration  $(s^m p^n)$

$$E(X, s^m p^n) = mU_{ss} + nU_{pp} + \frac{1}{2}(m+n)(m+n-1)F^0 - \frac{1}{6}mnG^1 - \frac{1}{25}n(n-1)F^2 \quad (6-66)$$

the following relations between the orbital electronegativities and the monatomic core integrals,  $U_{\mu\mu}$ , are obtained.

Hydrogen:

$$U_{ss} = -\frac{1}{2}(I_s^T + A_s^T) - \frac{1}{2}F^0 \quad (6-67)$$

Lithium:

$$U_{ss} = -\frac{1}{2}(I_s^T + A_s^T) - \frac{1}{2}F^0 \quad (6-68)$$

$$U_{pp} = -\frac{1}{2}(I_p^T + A_p^T) - \frac{1}{2}F^0 + \frac{1}{12}G^1 \quad (6-69)$$

Beryllium:

$$U_{ss} = -\frac{1}{2}(I_s^T + A_s^T) - \frac{3}{2}F^0 + \frac{1}{12}G^1 \quad (6-70)$$

$$U_{pp} = -\frac{1}{2}(I_p^T + A_p^T) - \frac{3}{2}F^0 + \frac{1}{4}G^1 \quad (6-71)$$

Boron to fluorine:

$$U_{ss} = -\frac{1}{2}(I_s^T + A_s^T) - \left(Z_A - \frac{1}{2}\right)F^0 + \frac{1}{6}\left(Z_A - \frac{3}{2}\right)G^1 \quad (6-72)$$

$$U_{pp} = -\frac{1}{2}(I_p^T + A_p^T) - \left(Z_A - \frac{1}{2}\right)F^0 + \frac{1}{3}G^1 + \frac{2}{25}\left(Z_A - \frac{5}{2}\right)F^2 \quad (6-73)$$

where  $Z_A$  is the effective core charge of atom A and  $I_{s(p)}^T$  and  $A_{s(p)}^T$  are the ionization potential and electron affinity of orbital s (p) with atom type of T. The one-center two-electron integrals  $F^0$  are calculated *ab initio* based on the Slater orbitals and  $F^2$  and  $G^1$  are chosen empirically.

For second long row elements (atomic numbers from 11 to 18), HyperNMR treats their monatomic core integrals the same as used in the QCPE program,<sup>21</sup> that is,

$$U_{ss} = -\frac{1}{2}(I_s^T + A_s^T) - \frac{1}{2}F^0 \quad (6-74)$$

$$U_{pp} = -\frac{1}{2}(I_p^T + A_p^T) - \frac{1}{2}F^0 \quad (6-75)$$

## TNDO Perturbation Theory of Magnetic Shielding Constants

Equations (6-19), (6-24), and (6-25) give the basic equations for computing magnetic shielding. However, care must be taken to ensure that  $\mathbf{H}$  remains Hermitian.

In the TNDO approximation, all the integrals in equation (6-23) are zero if  $\phi_\mu^0$  and  $\phi_\nu^0$  are centered on different atoms, with the

exception of  $H_{\mu\nu}^0$ . Although this approximation is somewhat drastic, it is consistent with all the previous approximations made in TNDO. Furthermore, this approximation restores the Hermitian property of  $\mathbf{H}$ . With this approximation, the component of the shielding tensor, Equation (6-19), can be rewritten as

$$\sigma_{ij}^A = \sigma_{ij}^d(A) + \sigma_{ij}^p(A) + \sum_B^{atoms} [\sigma_{ij}^d(A, B) + \sigma_{ij}^p(A, B)] \quad (6-76)$$

with the expressions for the one-center terms,  $\sigma_{ij}^d(A)$  and  $\sigma_{ij}^p(A)$ , being

$$\sigma_{ij}^d(A) = \frac{e^2}{mc^2} \sum_{\mu}^A \sum_{\nu}^A R_{\mu\nu}(0) \left\langle \phi_{\mu}^0 \left| \frac{\mathbf{r}_{\nu} \mathbf{r}_A \delta_{ij} - r_{\nu i} r_{A j}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-77)$$

and

$$\sigma_{ij}^p(A) = -\frac{i2eh}{mc} \sum_{\mu}^A \sum_{\nu}^A \left( \frac{\partial R_{\mu\nu}}{\partial B_i^0} \right)_0 \left\langle \phi_{\mu}^0 \left| \frac{(\mathbf{r} \times \nabla)_{\alpha}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-78)$$

and the two-center terms,  $\sigma_{ij}^d(A, B)$  and  $\sigma_{ij}^p(A, B)$ ,

$$\sigma_{ij}^d(A, B) = \frac{e^2}{mc^2} \sum_{\mu}^B \sum_{\nu}^B R_{\mu\nu}(0) \left\langle \phi_{\mu}^0 \left| \frac{\mathbf{r}_{\nu} \mathbf{r}_A \delta_{ij} - r_{\nu i} r_{A j}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-79)$$

and

$$\sigma_{ij}^p(A, B) = -\frac{i2eh}{mc} \sum_{\mu}^B \sum_{\nu}^B \left( \frac{\partial R_{\mu\nu}}{\partial B_i^0} \right)_0 \left\langle \phi_{\mu}^0 \left| \frac{(\mathbf{r} \times \nabla)_{\alpha}}{|\mathbf{r}_A|^3} \right| \phi_{\nu}^0 \right\rangle \quad (6-80)$$

The derivative of the perturbed density matrix,  $\mathbf{R}$  with respect to a component of  $\mathbf{B}$  is computed numerically<sup>4</sup> by a single difference

$$\left( \frac{\partial R_{\mu\nu}}{\partial B_i^0} \right)_0 \approx i \frac{R_{\mu\nu}^i}{B_i^0} \quad (6-81)$$

with  $R_{\mu\nu} = R_{\mu\nu}^r + R_{\mu\nu}^i$  where the fact that only the imaginary component of  $\mathbf{R}$  will contribute to the derivative has been used. The perturbed Fock Matrix,  $F_{\mu\nu}^{(1)} = F_{\mu\nu}^{(1,r)} + iF_{\mu\nu}^{(1,i)}$ , with  $r$  and  $i$  standing for the real and imaginary parts, is

$$F_{\mu\mu}^{(1,r)} = F_{\mu\mu}^{(1,i)} = 0 \quad (6-82)$$

for the one-center terms where basis function  $\mu$  is taken to be centered on atom A and

$$F_{\mu\nu}^{(1,r)} = G_{\mu\nu}^{(1,r)} \quad (6-83)$$

$$F_{\mu\nu}^{(1,i)} = -\frac{eh}{2mc} \mathbf{B} \cdot \int \phi_{\mu}^0(\mathbf{r}_{\nu} \times \nabla) \phi_{\nu}^0 d\tau + G_{\mu\nu}^{(1,i)} \quad (6-84)$$

where  $\mu \neq \nu$ , but both are still centered on the same atom A. The two-center terms are

$$F_{\mu\nu}^{(1,r)} = \cos\left(\frac{e}{2hc} \mathbf{B} \cdot \mathbf{R}_{\mu} \times \mathbf{R}_{\nu}\right) \frac{1}{2} (\beta_{\mu}^T + \beta_{\nu}^T) S_{\mu\nu}^T \quad (6-85)$$

$$F_{\mu\nu}^{(1,i)} = \sin\left(\frac{e}{2hc} \mathbf{B} \cdot \mathbf{R}_{\mu} \times \mathbf{R}_{\nu}\right) \frac{1}{2} (\beta_{\mu}^T + \beta_{\nu}^T) S_{\mu\nu}^T \quad (6-86)$$

where  $\mu \neq \nu$ . These equations are the same in both TNDO/1 and TNDO/2. In Equations (6-83) and (6-84) the  $\mathbf{G}$  matrix elements are

$$G_{\mu\nu}^{(1,r)} = -R_{\mu\nu}^r F^0 \quad (6-87)$$

$$G_{\mu\nu}^{(1,i)} = R_{\mu\nu}^i F^0 \quad (6-88)$$

for TNDO/1 and

$$G_{sp_{\alpha}}^{(1,r)} = R_{sp_{\alpha}}^r (G^1 - F^0) \quad (6-89)$$

$$G_{sp_{\alpha}}^{(1,i)} = R_{sp_{\alpha}}^i \left( F^0 - \frac{1}{3} G^1 \right) \quad (6-90)$$

$$G_{p_{\alpha}p_{\beta}}^{(1,r)} = R_{p_{\alpha}p_{\beta}}^r \left( \frac{11}{25} F^2 - F^0 \right) \quad (6-91)$$

$$G_{p_{\alpha}p_{\beta}}^{(1,i)} = R_{p_{\alpha}p_{\beta}}^i \left( F^0 - \frac{1}{5} F^2 \right) \quad (6-92)$$

for TNDO/2. The index  $s$  stands for the  $s$  orbital and  $p_{\alpha}$  and  $p_{\beta}$  are two of the  $p$  orbitals ( $p_x$ ,  $p_y$  and  $p_z$ ).



## TNDO Perturbation Theory of Nuclear Spin Coupling Constants

Equations (6-40) – (6-42) give the basic equations for computing the reduced constants of nuclear spin-spin coupling. For a closed-shell molecule, the density matrices of the alpha electrons and beta electrons will be symmetric, i.e.,

$$P_{\mu\nu}^{\alpha} = -P_{\mu\nu}^{\beta} \quad (6-93)$$

Thus Equation (6-42) can be rewritten as

$$\rho_{\mu\nu}^{(1)} = 2P_{\mu\nu}^{\alpha} \quad (6-94)$$

In the finite perturbation theory, the first derivative of the perturbed spin density at the nucleus,  $\rho_{\mu\nu}^{(1)}$ , in Equation (6-41) can be calculated numerically by finite differences. That is,

$$K_{AB} = \sum_{\mu,\nu} \mathcal{H}_{\mu\nu}^{B(1)} \left[ \frac{\rho_{\mu\nu}^{(1)}(h_B)}{h_B} \right] \quad (6-95)$$

where  $h_B$  is a small value perturbing the system in order to calculate the perturbed wave function  $\Psi(h_B)$ . Substituting Equations (6-37) and (6-40) into Equation (6-95) and using the basic approximation in TNDO, the reduced spin-spin coupling constant becomes

$$K_{AB} = \frac{8\pi}{3} \beta \left[ \frac{\rho_{s_A}^{(1)}(h_B)}{h_B} \right] s_B^2(0) \quad (6-96)$$

where  $s_B^2(0)$  is the density of the valence shell s-electron at nucleus *B*.

In the TNDO approximation, Equation (6-40) contains only the one-center contribution, so that the only non-zero element of is that involving the valence s orbital on atom B. This is given by

$$\mathcal{H}_{s_B s_B}^{B(1)} = \frac{8\pi}{3} \beta s_B^2(0) \quad (6-97)$$

Hence the perturbation of the Hamiltonian Matrix,  $H_{\mu\nu}^{(1)}$ , due to the presence of the Fermi-contact nuclear spin coupling is given by

$$H_{s_B s_B}^{\alpha(1)} = +\frac{8\pi}{3}\beta s_B^2(0), \text{ for the } \alpha \text{ electrons, and}$$

$$H_{s_B s_B}^{\beta(1)} = -\frac{8\pi}{3}\beta s_B^2(0), \text{ for the } \beta \text{ electrons}$$
(6-98)

In practice it is more convenient to set

$$H_{s_B s_B}^{\alpha(1)} = +h_B, \text{ for the } \alpha \text{ electrons, and}$$

$$H_{s_B s_B}^{\beta(1)} = -h_B, \text{ for the } \beta \text{ electrons}$$
(6-99)

And comparing Equations (6-98) and (6-99), it is easy to see that

$$h_B = \frac{8\pi}{3}\beta s_B^2(0)$$
(6-100)

So the expression of the reduced coupling constant now becomes

$$K_{AB} = \left(\frac{8\pi}{3}\beta\right)^2 s_A^2(0)s_B^2(0) \left[\frac{\rho_{s_A}^{(1)}(h_B)}{h_B}\right]$$
(6-101)

In the TNDO approximation, only the valence electrons are included in the SCF MO calculation and the atom's core electrons are treated as a point charge at the corresponding nucleus. With Slater-type orbitals (STOs), the densities of the valence shell s-electrons at a nucleus are zero for all the elements except for hydrogen. Hence, empirical values of the s-orbital densities at the nucleus are needed for computing spin coupling constants.

## Self-Consistent Perturbation Theory

Commonly, a finite perturbation form of Hartree-Fock SCF MO theory is used for solving the perturbed Hartree-Fock equations, in order to produce the perturbed wave functions for the magnetic shielding and spin coupling constants. It is, however, computationally time-consuming. A complex Hartree-Fock equation needs to be solved in order to get the imaginary part of the perturbed wave function for magnetic shielding and for the spin coupling constants a set of perturbed Hartree-Fock equations needs to be solved for each nuclear spin.

In 1971, Santry<sup>5</sup> and coworkers introduced a self-consistent perturbation theory for computing nuclear spin coupling constants.

It has been shown that their self-consistent theory is more efficient than a finite perturbation form of Hartree-Fock SCF MO theory. Thus HyperNMR uses the computational approach of Santry *et al.* for nuclear spin coupling constants and extends their approach to magnetic shielding. This avoids solving the perturbed Hartree-Fock equations repeatedly and the computational efficiency is much better than for other methods.

In Santry's self-consistent perturbation theory, the perturbed eigenvectors,  $c_i^{(1)}$  ( $i = 1, 2, \dots, N/2$ ) are computed directly from the unperturbed eigenvectors,  $c_j^{(0)}$  ( $j = 1, 2, \dots, M$ ; where  $M$  is the total number of molecular orbitals) using matrix multiplication rather than matrix diagonalization.

As discussed above, the presence of the external magnetic field or the Fermi-contact Hamiltonian is a small perturbation to the unperturbed Hamiltonian. The perturbation can be treated using standard perturbation theory. In the standard perturbation theory, the wave function up to the first order perturbation is

$$\psi_j = \psi_j^{(0)} + \psi_j^{(1)} \quad (6-102)$$

with

$$\psi_j^{(1)} = \sum_i^{\text{unoccupied}} \frac{F_{ij}^{(1)}}{\epsilon_j^{(0)} - \epsilon_i^{(0)}} \psi_j^{(0)} \quad (6-103)$$

where  $j$  applies to the occupied orbitals only,  $\epsilon_j^{(0)}$  is the unperturbed eigenvalue of the  $j$ th MO, and  $F_{ij}^{(1)}$  is the first-order change in the Fock Matrix due to the perturbation being considered, that is

$$F_{ij}^{(1)} = \langle \psi_i^{(0)} | \hat{F}^{(1)} | \psi_j^{(0)} \rangle = \langle \psi_i^{(0)} | \hat{H}^{(1)} + \hat{G}^{(1)} | \psi_j^{(0)} \rangle \quad (6-104)$$

By using the LCAO approximation, Equation (6-104) becomes

$$F_{ij}^{(1)} = c_i^{(0)T} \mathbf{F}^{(1)} c_j^{(0)} \quad (6-105)$$

with  $F_{\mu\nu}^{(1)} = \langle \phi_{\mu}^{(0)} | \hat{F}^{(1)} | \phi_{\nu}^{(0)} \rangle$  being elements of  $\mathbf{F}^{(1)}$ . The matrix expression of Equation (6-103) is

$$\mathbf{c}_j^{(1)} = \sum_i^{\text{unoccupied}} \frac{\mathbf{c}_i^{(0)} \mathbf{c}_i^{(0)T}}{\varepsilon_j^{(0)} - \varepsilon_i^{(0)}} \mathbf{F}^{(1)} \mathbf{c}_j^{(0)} \quad (6-106)$$

where  $\mathbf{c}_j^{(0)}$  is a column vector,  $\mathbf{c}_i^{(0)T}$  a row vector, and  $\mathbf{F}^{(1)}$  is an  $M \times M$  matrix and a function of  $\mathbf{c}_j^{(1)}$ . These equations are solved self-consistently.

Santry noticed that  $\sum_i^{\text{unoccupied}} \frac{\mathbf{c}_i^{(0)} \mathbf{c}_i^{(0)T}}{\varepsilon_j^{(0)} - \varepsilon_i^{(0)}}$  in Equation (6-106) is

independent of the perturbation and therefore needs not be recalculated each iteration during the process of solving Equation (6-106) self-consistently. However, the  $N/2$  matrices of

$$\sum_i^{\text{unoccupied}} \frac{\mathbf{c}_i^{(0)} \mathbf{c}_i^{(0)T}}{\varepsilon_j^{(0)} - \varepsilon_i^{(0)}}, \text{ each with size of } M \times M \text{ have to be computed}$$

and stored before the self-consistent calculation. The storage of these matrices is  $O(M^3)$  ( $M$  is the total number of orbitals). The computation may be impossible for large molecular systems if main memory is used. It may be very slow if disk storage is used because of the relatively slow rate of retrieving data from disk. HyperNMR applies a new algorithm rather than using the algorithm suggested by Santry. Because of the association property, the matrix multiplications in Equation (6-106) may be carried out backwards from the right to the left. If the matrix multiplication is done in this way, then the last matrix multiplication in Equation (6-106) is

$$\mathbf{F}^{(1)} \mathbf{c}_j^{(0)} = \mathbf{A}_j^1 \quad (6-107)$$

which requires only  $M \times M$  steps of multiplication. The matrix  $\mathbf{A}_j^1$  is a one-column matrix and can be stored with  $O(M)$  computer memory before doing the summation in Equation (6-106). The second matrix multiplication

$$\mathbf{c}_i^{(0)T} \mathbf{F}^{(1)} \mathbf{c}_j^{(0)} = \mathbf{c}_i^{(0)T} (\mathbf{F}^{(1)} \mathbf{c}_j^{(0)}) = \mathbf{c}_i^{(0)T} \mathbf{A}_j^1 = A_{ij}^2 \quad (6-108)$$

takes  $M$  steps of multiplication and leads to a scalar value  $A_{ij}^2$  which can be stored for each unoccupied MO with  $O(M)$  memory. The first matrix multiplication in Equation (6-106) now becomes a one-column vector multiplying a scalar value

$$\mathbf{c}_i^{(0)} \mathbf{c}_i^{(0)T} \mathbf{F}^{(1)} \mathbf{c}_j^{(0)} = \mathbf{c}_i^{(0)} \left[ \mathbf{c}_i^{(0)T} (\mathbf{F}^{(1)} \mathbf{c}_j^{(0)}) \right] = A_{ij}^2 \mathbf{c}_i^{(0)} \quad (6-109)$$

and takes only  $M$  steps of multiplication. Thus the total number of steps of multiplication taken by Equation (6-106) in each iteration is about  $M^3$  (equivalent to two  $M \times M$  matrix multiplications) and the extra memory is only about  $O(2M)$ . Therefore this new algorithm should be faster and more efficient than the original algorithm used by Santry and coworkers.

From Equations (6-102) and (6-103), the first-order change to the unperturbed density matrix is, therefore,

$$P_{\mu\nu}^{(1)} = \sum_i^{N/2} (\mathbf{c}_{\mu i}^{(1)*} \mathbf{c}_{\nu i}^{(0)} + \mathbf{c}_{\mu i}^{(0)*} \mathbf{c}_{\nu i}^{(1)}) \quad (6-110)$$

for spin coupling constants and

$$R_{\mu\nu}^i = \sum_i^{N/2} (\mathbf{c}_{\mu i}^{(1)*} \mathbf{c}_{\nu i}^{(0)} - \mathbf{c}_{\mu i}^{(0)*} \mathbf{c}_{\nu i}^{(1)}) \quad (6-111)$$

for magnetic shielding.

## Parameterization

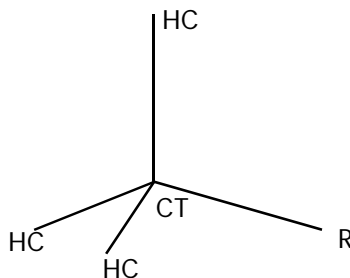
As discussed above, reparameterization is essential for the semi-empirical quantum mechanical methods to predict good chemical shifts. Furthermore, because TNDO introduces atom types, there may be several types for each element in order to reproduce chemical shifts, and the empirical parameters may not be the same for the different types within the same element.

The types and rules for assigning them are kept in two text files. The first file, NMR.TXT, lists the types and their masses and the

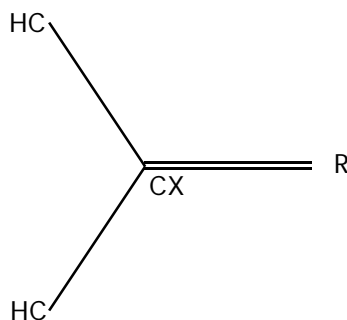
second file, NMR.RUL, defines rules for HyperNMR to assign types to atoms in molecular systems. You can edit and modify the type list and rules. The initial names and definitions of atom types are derived from the AMBER method of molecular mechanics.

The magnetic shielding property derives from all the electrons in a molecular system. This means that contributions to the magnetic shielding come not only from the valence electrons, but also from the core electrons and valence-core electron interaction. However within the NDO and NDDO approximations, the effective nucleus includes the core electron contribution and core electrons are not included explicitly in the SCF calculation. Thus it is not possible to get the absolute value of magnetic shielding for molecules within the NDO and NDDO approximation. However the shielding contributed from the core electrons in different molecules is not likely to be very system dependent. Thus the relative values of the magnetic shielding, i.e., the chemical shift, are reproducible within an all-valence electron SCF calculation.

A preliminary parameterization has been performed for TNDO/2 chemical shifts. The Slater orbital exponents,  $\zeta_s$  and  $\zeta_p$ , are taken from Slater rules and the Slater-Condon parameters,  $F^2$  and  $G^1$ , are from the conventional INDO method. The ionization potentials,  $I_s^T$  and  $I_p^T$ , electron affinities,  $A_s^T$  and  $A_p^T$ , and bond parameters,  $\beta^T$ , are reparameterized by fitting the chemical shifts to the experimental data in terms of a least squares fit. For example, the first reparameterization was done for a  $sp^3$  carbon with the type CT and three hydrogens attached to this carbon, as shown below:



In the above figure, R stands for a hydrogen or a methyl group. A massive calculation of chemical shifts for several molecular systems was carried out by varying the ionization potentials and electron affinities of s and p atomic orbitals and by varying the bond parameters for each type. The parameters for CT and HC were determined by comparing the calculated chemical shifts with the corresponding experimental values in terms of a least squares fit. Having once acquired parameters for CT and HC, the parameters for HC were then used in  $sp^2$  systems with the new atom type CX:



A massive calculation of chemical shifts for several molecular systems was done again by varying the parameters of CX. The parameters for CX were then determined by comparing the calculated chemical shifts with the corresponding experimental data in terms of a least squares fit. This approach has been used for generating the empirical parameters for chemical shift calculation.

## **Calculation of Chemical Shifts**

Tests have been performed using the preliminary parameterized parameters. The following tables list the calculated chemical shifts compared with experimental results and other calculations.

Table 5-1  $^{13}\text{C}$  Chemical shifts (in ppm) in some hydrocarbons (relative to methane)

Molecule		Exp.	Hyper-NMR	Other <sup>a</sup>
Methane $\text{CH}_4$	$\text{CH}_4$	0	0	0
Ethane $\text{C}_2\text{H}_6$	$\text{CH}_3$	-8	-10.5	14.4
Propane $\text{CH}_3\text{CH}_2\text{CH}_3$	$\text{CH}_3$	-17.7	-17.9	12.2
	$\text{CH}_2$	-18.2	-18.1	30.5
n-Butane $\text{CH}_3(\text{CH}_2)_2\text{CH}_3$	$\text{CH}_3$	-15.3	-17.4	11.6
	$\text{CH}_2$	-27.1	-25.3	28.0
n-Pentane $\text{CH}_3(\text{CH}_2)_3\text{CH}_3$	$\text{CH}_3$	-15.8	-17.6	
	$\text{CH}_2(\text{end})$	-24.7	-24.6	
	$\text{CH}_2(\text{cen})$	-36.3	-32.7	
n-Hexane $\text{CH}_3(\text{CH}_2)_4\text{CH}_3$	$\text{CH}_3$	-15.9	-17.6	
	$(\text{CH}_2)_1$	-25.0	-24.8	
	$(\text{CH}_2)_2$	-34.1	-32.1	
n-Heptane $\text{CH}_3(\text{CH}_2)_5\text{CH}_3$	$\text{CH}_3$	-16.1	-17.7	
	$(\text{CH}_2)_1$	-25.1	-24.8	
	$(\text{CH}_2)_2$	-34.5	-32.3	
	$(\text{CH}_2)_3$	-31.6	-31.5	
3-Methyl Pentane $\text{CH}_3\text{CH}_2\text{CH}(\text{CH}_3)\text{CH}_2\text{CH}_3$	$\text{CH}_3(\text{end})$	-13.3	-19.1	
	$\text{CH}_3(\text{cen})$	-20.7	-25.4	
	$\text{CH}_2$	-31.4	-30.2	
	$\text{CH}$	-38.8	-36.3	

a. Calculated by INDO with standard parameters; see P. D. Ellis, G. E. Maciel, and J. W. McIver, *J. Am. Chem. Soc.* **94**, 4069 (1972).



Table 5-2 Proton chemical shifts (in ppm) in some hydrocarbons (relative to methane)

Molecule		Exp.	Hyper-NMR	Other <sup>a</sup>
Methane CH <sub>4</sub>	CH <sub>4</sub>	0	0	0
Ethane C <sub>2</sub> H <sub>6</sub>	CH <sub>3</sub>	-0.63	-0.57	0.37
Propane CH <sub>3</sub> CH <sub>2</sub> CH <sub>3</sub>	CH <sub>3</sub>	-0.68	-0.72	0.33
	CH <sub>2</sub>	-1.10	-0.91	0.93
n-Butane CH <sub>3</sub> (CH <sub>2</sub> ) <sub>2</sub> CH <sub>3</sub>	CH <sub>3</sub>	-0.68	-0.77	0.30
	CH <sub>2</sub>	-1.10	-0.95	0.92
n-Pentane CH <sub>3</sub> (CH <sub>2</sub> ) <sub>3</sub> CH <sub>3</sub>	CH <sub>3</sub>	-0.68	-0.80	
	CH <sub>2</sub> (end)	-1.10	-1.01	
	CH <sub>2</sub> (cen)			
n-Hexane CH <sub>3</sub> (CH <sub>2</sub> ) <sub>4</sub> CH <sub>3</sub>	CH <sub>3</sub>	-0.76	-0.81	
	(CH <sub>2</sub> ) <sub>1</sub>	-1.14	-1.03	
	(CH <sub>2</sub> ) <sub>2</sub>	-1.95	-1.06	
n-Heptane CH <sub>3</sub> (CH <sub>2</sub> ) <sub>5</sub> CH <sub>3</sub>	CH <sub>3</sub>	-0.76	-0.82	
	(CH <sub>2</sub> ) <sub>1</sub>			
	(CH <sub>2</sub> ) <sub>2</sub>	-1.15	-1.08	
	(CH <sub>2</sub> ) <sub>3</sub>			
3-Methyl Pentane CH <sub>3</sub> CH <sub>2</sub> CH(CH <sub>3</sub> )- CH <sub>2</sub> CH <sub>3</sub>	CH <sub>3</sub> (end)	-0.75	-0.78	
	CH <sub>3</sub> (cen)	-0.75	-0.70	
	CH <sub>2</sub>	-1.04	-1.05	
	CH	-1.26	-1.08	

a. Calculated by INDO with standard parameters; see P. D. Ellis, G. E. Maciel, and J. W. McIver, *J. Am. Chem. Soc.* **94**, 4069 (1972).

Table 5-3  $^{13}\text{C}$  Chemical shifts (in ppm) in some hydrocarbons (relative to methane)

Molecule		Exp.	Hyper-NMR	Other
Methane $\text{CH}_4$	$\text{CH}_4$	0	0	0
Ethylene $\text{CH}_2\text{CH}_2$	$\text{CH}_2$	-125.0	-119.1	-115.0 <sup>a</sup>
Propylene $\text{CH}_2\text{CHCH}_3$	$\text{CH}_3$	-22.6	-12.0	-6.1 <sup>a</sup>
	$\text{CH}_2$	-118.6	-116.5	-112.1 <sup>a</sup>
	$\text{CH}$	-137.9	-128.2	-126.2 <sup>a</sup>
cis-2-Butane cis- $\text{CH}_3\text{CHCHCH}_3$	$\text{CH}_3$		-11.4	
	$\text{CH}$	-125.7	-130.0	-132.6 <sup>b</sup>
trans-2-butane trans- $\text{CH}_3\text{CHCHCH}_3$	$\text{CH}_3$		-9.6	
	$\text{CH}$	-127.1	-130.0	-131.3 <sup>b</sup>
2-Methyl-2- Butane $(\text{CH}_3)_2\text{CCHCH}_3$	$(\text{C}^*\text{H}_3)_2$		-16.2	
	$\text{CCHC}^*\text{H}_3$		-9.7	
	$\text{CC}^*\text{HCH}_3$	-121.0	-127.0	-132.9 <sup>b</sup>
	$\text{C}^*\text{CHCH}_3$	-133.7	-137.8	-141.5 <sup>b</sup>
2,3-dimethyl-2- Butane $(\text{CH}_3)_2\text{CC}(\text{CH}_3)_2$	$\text{CH}_3$ $\text{C}^*\text{C}(\text{CH}_3)_2$	-123.9	-17.1 -139.5	-141.7 <sup>b</sup>
Isobutylene $(\text{CH}_3)_2\text{CCH}_2$	$\text{CH}_3$		-18.0	
	$\text{C}^*=\text{CH}_2$	-143.5	-133.1	-140.9 <sup>b</sup>
	$\text{C}=\text{C}^*\text{H}_2$	-112.1	-112.6	-121.8 <sup>b</sup>

a. Calculated by INDO with their new parameters; see P. D. Ellis, G. E. Maciel, and J. W. McIver, *J. Am. Chem. Soc.* **94**, 4069 (1972).

b. E. Vauthier, S. Odier, and F. Tonnard, *Can. J. Chem.* **61**, 1417 (1983).

Table 5-4 CPU times (in sec.) for computing shielding and/or spin-spin coupling

a constants						
Molecule	Atm	Orbs	T <sub>old1</sub>	T <sub>old2</sub> <sup>b</sup>	T <sub>new1</sub> <sup>c</sup>	T <sub>new2</sub> <sup>c</sup>
Ethylene C <sub>2</sub> H <sub>4</sub>	6	12	11	61	2	3
Ethane C <sub>2</sub> H <sub>6</sub>	8	14	16	70	3	4
Fluoro-acetic acid FCH <sub>2</sub> COOH	8	23	123	548	7	15
Benzene C <sub>6</sub> H <sub>6</sub>	12	30	113	918	9	28
Anthracene C <sub>14</sub> H <sub>10</sub>	24	66	1329		58	846

a. This comparison is done on a NEC PowerMate (486/50) with Windows 3.1 and 8 Mb of RAM.

b. T<sub>old1</sub> and T<sub>old2</sub> are the times (in seconds) taken for computing the shielding constants only and both the shielding constants and nuclear spin coupling constants by the finite perturbation SCF MO semi-empirical method of Santry *et al.*, respectively.

c. T<sub>new1</sub> and T<sub>new2</sub> are the times (in seconds) taken for computing the shielding constants only and both the shielding constants and nuclear spin coupling constants by the new perturbed SCF MO semi-empirical method, respectively.

Table 5-4 shows the CPU time comparison of calculating shielding constants only, and both the shielding constants and nuclear spin coupling constants, for several molecular systems using both the original finite perturbation SCF MO algorithm and the new self-consistent perturbation theory. It can be seen that the current algorithm for computing perturbed wave functions is much faster than the original algorithm. The range of speed improvements is 10 to 30 times depending on the size of the molecular system. The finite perturbation SCF MO algorithm is only usable in the calculation of shielding constants and spin coupling constants for small molecular systems. For example, anthracene has only 24 atoms and 66 molecular orbitals, but it took 1329 seconds on a NEC 486/50 PC machine using the old algorithm. The speed of computing nuclear spin coupling constants by the finite perturbation theory is about 5 to 10 times slower than that of calculating shielding constants. Thus it may take 3 to 4 hours to compute the nuclear spin coupling constants just for this 66 orbital system using the

finite perturbation theory. With the new self-consistent perturbation algorithm of HyperNMR, it took less than 15 minutes for the calculation of both the shielding constants and spin coupling constants for anthracene. Therefore, HyperNMR may be used with much larger molecular systems than earlier programs.

## Nuclear Spin States and Transitions

### Spin Wave Functions

In quantum theory, each nucleus has a spin  $\mathbf{I}$  which is a vector related to the angular momentum  $\hbar\mathbf{I}$  and the magnetic moment  $\gamma\hbar\mathbf{I}$  ( $\gamma$  is the nuclear magnetogyric ratio). The total number of spin states for this nucleus is  $(2I + 1)$ ; each has one of the spin quantum numbers,  $-I, -I+1, -I+2, \dots, +I$ .

HyperNMR will simulate the NMR spectra only for those nuclei whose nuclear spin is  $1/2$ . For these nuclei, each isolated nucleus has two independent spin states, that is, spin up,  $\alpha$ , with spin quantum number of  $1/2$  and spin down  $\beta$ , with spin quantum number of  $-1/2$ . These two spin states are eigenstates of the nuclear spin operator,  $I_z$ , i.e.,

$$I_z\alpha = +\frac{1}{2}\alpha \quad \text{and} \quad I_z\beta = -\frac{1}{2}\beta \quad (6-112)$$

with

$$I_x = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad I_y = \frac{1}{2} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \quad I_z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (6-113)$$

With the lack of an external magnetic field, these two spin states are degenerate. This degeneracy is removed by applying an external magnetic field (Zeeman effect).

Suppose that a strong magnetic field  $\mathbf{B}^0$  is applied to a molecular system along the  $-z$  direction. The interaction energy between the external magnetic field  $\mathbf{B}^0$  and nuclear magnetic moment  $\gamma\hbar\mathbf{I}$  is then  $\gamma\hbar\mathbf{B}^0 \cdot \mathbf{I}$ . The unit of this interaction energy is in ergs if  $\mathbf{B}^0$  is in gauss. Using the conventional units of cycles per second (Hz), the interaction energy becomes  $\gamma\mathbf{B}^0 \cdot \mathbf{I} / 2\pi$ . Consequently, the Hamiltonian for a system with  $p$  nuclei is

$$H^{(0)} = \frac{1}{2\pi} \sum_i^p \gamma_i B_i I_{zi} \quad (6-114)$$

where  $B_i$  may not be the same as the external magnetic field  $B^0$  because of the screening of electrons in the molecular system. That is,

$$B_i = B^{(0)}(1 - \sigma_i) \quad (6-115)$$

The parameter  $\gamma_i$  may be different for different species of nucleus. In Equation (6-115),  $\sigma_i$  are the electron shielding constants.

The indirect spin coupling may be represented by a Hamiltonian involving the two-body interactions of all spin vectors of magnetic nuclei:

$$H^{(1)} = \sum_{i < j}^p J_{ij} \mathbf{I}_i \cdot \mathbf{I}_j \quad (6-116)$$

If the molecular system rotates rapidly enough, the dipole-dipole interaction can be neglected. Then the complete Hamiltonian is

$$H = H^{(0)} + H^{(1)} \quad (6-117)$$

For a system with  $p$  nuclei (all with spin 1/2), there is a total of  $2^p$  possible spin states. The simplest set of functions describing this many-system would be the  $2^p$  basic product functions such as

$$\begin{aligned} \varphi_n &= \alpha(1)\beta(2)\alpha(3)\cdots\beta(p) \\ &\equiv \alpha\beta\alpha\cdots\beta \end{aligned} \quad (6-118)$$

If the nuclei were actually independent, these products would themselves be stationary state wave functions in the presence of the external magnetic field. However, the nuclei actually interact via  $H^{(1)}$  which causes the basis functions of the spin wave function to mix. Because of this mixing, quantum numbers  $\alpha\beta\alpha\cdots\beta$  are no longer "good" quantum numbers.

## Stationary-state Wave Functions

The products of nuclear spins are no longer stationary functions because of the interactions of nuclear spins. The stationary wave functions may be obtained by expanding the stationary wave

functions into linear combinations of the product functions which act as basis functions. That is,

$$\phi_i = \sum_n^{2^p} c_{ni} \Phi_n \quad (6-119)$$

The coefficients,  $c_{ij}$ , are determined by solving the secular equation

$$|H_{mn} - E\delta_{mn}| = 0 \quad (6-120)$$

where  $H_{mn}$  are the Hamiltonian matrix elements of nuclear spins and  $E$  are the energies of the stationary states.

The diagonal Hamiltonian matrix elements are

$$H_{mn} = \frac{1}{4\pi} \sum_i^p \gamma_i B_i P_i + \frac{1}{4} \sum_{i < j}^p J_{ij} T_{ij} \quad (6-121)$$

where  $P_i = 1$  or  $-1$  according to whether the corresponding spin function is  $\alpha$  or  $\beta$  and  $T_{ij} = 1$  or  $-1$  according to whether spins  $i$  and  $j$  are parallel or antiparallel in  $\phi_m$ . The off-diagonal elements are

$$H_{mn} = \frac{1}{2} U J_{ij} \quad (6-122)$$

where  $U = 1$  if  $\phi_m$  differs from  $\phi_n$  by a single interchange of spin  $i$  and  $j$ , and is zero otherwise.

## Selection Rules and Intensities of Transitions

To determine the NMR spectrum, the selection rules must be found. The probability that a system originally in state  $\phi_i$  that is exposed to a time-dependent perturbation  $\mathbf{H}'(t)$  from  $t = 0$  to  $t_1$  will take a transition to  $\phi_j$  is given by

$$\frac{1}{h^2} \left| \int_0^{t_1} e^{i\omega_{mn}t} \langle \phi_m | \mathbf{H}'(t) | \phi_n \rangle \right|^2 \quad (6-123)$$

where  $\omega_{mn} = (E_m - E_n)/h$ . If a nuclear magnetic moment sitting in a static magnetic field  $\mathbf{B}^0$  is exposed to electromagnetic radiation of frequency  $\nu$ , then  $\mu_N$  will interact with the magnetic field  $\mathbf{B}^1$  of the radiation; the time dependent perturbation is

$$\begin{aligned}
\hat{H}'(t) &= -\mathbf{B}^1 \cdot \hat{\boldsymbol{\mu}}_N & (6-124) \\
&= -h \sum_i^p \gamma_i \mathbf{B}^1 \cdot \hat{\mathbf{I}}_i \\
&= -h \sum_i^p \gamma_i (B_x^1 \hat{I}_{ix} + B_y^1 \hat{I}_{iy} + B_z^1 \hat{I}_{iz})
\end{aligned}$$

Suppose the static magnetic field  $\mathbf{B}^0$  is in the  $z$  direction. Because the wave functions are then the eigenfunctions of  $\hat{I}_{iz}$ , the  $z$  component of the radiation magnetic field cannot induce an NMR transition. Thus only the  $x$  and  $y$  components of the radiation magnetic field may induce an NMR transition. If the radiation magnetic field is in the direction of  $x$ , then the transition probability is proportional to

$$\left| \left\langle \phi_m \left| \sum_i^p \gamma_i \hat{I}_{ix} \right| \phi_n \right\rangle \right|^2 \quad (6-125)$$

The selection rule can be deduced from this equation. The operator  $\hat{I}_{ix}$  will have a matrix element only between two basis functions which differ only in the spin of nucleus  $i$ . In a three-nuclei system, for example,  $\hat{I}_{ix}$  would have a non-vanishing element between  $\alpha\alpha\beta$  and  $\beta\alpha\beta$ . As a result, the total matrix element in Equation (6-125) will be non-vanishing only between states which

have values of the total spin component  $F_z = \sum_i^p I_{ix}$  differing by  $\pm 1$ . That is,

$$\Delta F_z = \pm 1 \quad (6-126)$$

## Weak Coupling Approximation

The weak coupling approximation was proposed and developed by Anderson<sup>23</sup> in 1956 and uses a perturbation method to determine an NMR spectrum (transition frequencies and intensities) instead

of solving a nuclear spin secular equation when the nuclear spin-spin coupling is weak.

In general, NMR spectra in which the signals from non-equivalent nuclei are closely spaced or overlapping must be treated with the full spin secular equation. However, in some cases the spacings of such signals are large enough for the spectrum to bear a close resemblance to a first order multiplet pattern corresponding to large chemical shifts. In these cases, a perturbation method may be more appropriate than a complete treatment.

In Anderson's perturbation approach, the first-order NMR spectrum is given by only the chemical shifts of the NMR atoms. The nuclear spin-spin coupling is considered as a perturbation to the first-order NMR spectrum. Then the transition energy, up to the second-order perturbation, between spin states  $(F_A, m_A, F_B, m_B)$  and  $(F_A, m_A, F_B, m_B)$  is given by

$$\begin{aligned} \nu = \nu_A + \sum_{R(\neq A)} J_{AR} m_R + \\ \frac{1}{2} \sum_{R(\neq A)} \frac{J_{AR}}{\nu_A - \nu_R} [F_R(F_R + 1) - m_R(m_R + 1) + 2m_A m_R] \end{aligned} \quad (6-127)$$

and the relative intensity of the corresponding transition is, to the same order,

$$(F_A - m_A + 1)(F_A + m_A) \left( 1 - 2 \sum_{R(\neq A)} \frac{J_{AR} m_R}{\nu_A - \nu_R} \right) \quad (6-128)$$

This perturbation method, if applicable, is much faster and takes much less memory than the full solution of the spin secular equation. HyperNMR uses this perturbation method to calculate the NMR spectrum up to the second-order transition energies and relative intensities if the Weak Coupling Approximation check box is checked in the Spectral Options dialog box.

## Parameter Sets

You must have the following parameter files in your HyperNMR directory to run a shielding and spin-spin coupling calculation.



- NMR\_SHFT.ABP for computing magnetic shielding constants (chemical shifts);
- NMR\_CUPL.ABP for calculating nuclear spin coupling constants.

You may want to modify these parameter files when new and better parameters become available or you might assign your own new atom types and parameters. The format of the files is described in Chapter 5.

Two of the parameters for each atom type in these files are the electronic energies of the corresponding isolated atom from TNDO/1 and TNDO/2. These energies were calculated using the CNDO and INDO methods in HyperChem with the HyperNMR parameters and restricted Hartree-Fock (RHF) wave functions for the ground state. The spin multiplicities used were 1 for Be and Mg, 2 for H, Li, B, F, Na, Al, and Cl, and 3 for C, O, Si, and S, and 4 for N and P.

## Bibliography

1. K. Wolinski, J. F. Hinton and P. Pulay, *J. Am. Chem. Soc.* **112**, 8251 (1990).
2. D. S. Wishart, B. D. Sykes and F. M. Richards, *Biochemistry* **31**, 1647 (1992).
3. A. Pastone and V. Saudek, *J. Magn. Reson.* **90**, 165 (1990).
4. P. D. Ellis, G. E. Maciel and J. W. McIver, Jr., *J. Am. Chem. Soc.* **94**, 4069 (1972).
5. A. C. Blizzard and D. P. Santry, *J. Chem. Phys.* **55**, 950 (1971).
6. J. A. Pople, J. W. McIver, Jr. and N. S. Ostlund, *J. Chem. Phys.* **49**, 2960 (1968).
7. J. A. Pople, J. W. McIver, Jr. and N. S. Ostlund, *J. Chem. Phys.* **49**, 2965 (1968).
8. G. E. Maciel, J. W. McIver, Jr., N. S. Ostlund and J. A. Pople, *J. Am. Chem. Soc.* **92**, 11 (1970).
9. L. Ducasse, J. Hoarau and M. Pesquer, *Mol. Phys.* **40**, 1249 (1980).
10. M. Kondo, I. Ando, R. Chujo and A. Nishioka, *J. Magn. Reson.* **24**, 315 (1976).

11. E. C. Vauthier, S. Odiod and F. Tonnard, *Can. J. Chem.* **60**, 957 (1982).
12. E. C. Vauthier, S. Fliszar, F. Tonnard and S. Odiod, *Can. J. Chem.* **61**, 1417 (1983).
13. J. D. Baker and M. C. Zerner, *Int. J. Quantum Chem.* **43**, 327 (1992).
14. J. A. Pople and D. V. Beveridge, *Approximate Molecular Orbital Theory*, McGraw-Hill, New York, 1970.
15. W. P. Anderson, T R. Cundari and M. C. Zerner, *Int. J. Quantum Chem.* **39**, 31(1991).
16. R. C. Bingham, M. J. S. Dewar and D. H. Lo, *J. Am. Chem. Soc.* **97**, 1285 (1975).
17. J. Li, P. C. Mello and K. Jug, *J. Comput. Chem.* **13**, 85 (1992).
18. M. J. S. Dewar and W. Thiel, *J. Am. Chem. Soc.* **99**, 4899 (1977).
19. M. J. S. Dewar, E. G. Zoebisch, E. F. Healy and J. J. P. Stewart, *J. Am. Chem. Soc.* **107**, 3902 (1985).
20. J. J. P. Stewart, *J. Comput. Chem.* **10**, 221 (1989).
21. M. Fraser and D. M. Hayes, *Quantum Chemistry Program Exchange*, QCPE Program No. 389, Indiana University.
22. H. Fukui, *Magn. Reson. Rev.* **11**, 205 (1987).
23. W. A. Anderson, *Phys. Rev.* **102**, 151 (1956).



# HyperChem<sup>®</sup> for Windows and NT

The Chemist's Developer Kit (CDK):

Customizing HyperChem

Interfacing to HyperChem

## Copyright © 2002 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

### Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperGauss, HyperChemOS, HyperNMR and ChemPlus are trademarks of Hypercube, Inc.

### Third Party Trademarks

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines, Inc.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
	The Chemist's Developer Kit . . . . .	1
	CDK for Windows or NT . . . . .	1
	Equivalent Unix CDK . . . . .	1
	Components of the CDK . . . . .	2
	Components Included with Release 5 . . . . .	2
	Other Suggested Tools . . . . .	2
	Suggested Compilers . . . . .	3
	HyperChem State Variables . . . . .	3
	Customizing HyperChem . . . . .	4
	Internal Script commands . . . . .	5
	HyperChem Command Language (Hcl) . . . . .	6
	Tool Command Language (Tcl/Tk) . . . . .	6
	Custom Menus . . . . .	6
	Interfacing to HyperChem . . . . .	6
	Dynamic Data Exchange . . . . .	7
	External Script Messages . . . . .	7
	The HyperChem Application Programming Interface (HAPI) . . . . .	8
	Overview of Chapters . . . . .	8
<b>Chapter 2</b>	<b>Architecture of HyperChem</b>	<b>11</b>
	Introduction . . . . .	11
	The Front End - Back End Architecture . . . . .	11
	The Older Master - Slave Architecture . . . . .	13
	The Open Architecture . . . . .	13
	UMSG and VMSG . . . . .	15
	IMSG and OMSG . . . . .	15
	The Newer Client - Server Architecture . . . . .	16
	The Network Architecture . . . . .	17
	Network DDE . . . . .	17

UNIX . . . . .	17
Remote Back Ends . . . . .	18
Mixing UNIX and Windows or NT . . . . .	18

### **Chapter 3 Customizing HyperChem 21**

Introduction . . . . .	21
A Flexible Development Platform . . . . .	21
What are Scripts? . . . . .	22
Type 1 (Hcl) Scripts . . . . .	22
Type 2 (Tcl/Tk) Scripts . . . . .	22
Custom Menus . . . . .	24

### **Chapter 4 HyperChem State Variables 27**

Introduction . . . . .	27
Registering of HSV's . . . . .	27
An Example of an HSV . . . . .	27
Read/Write Nature of HSV's . . . . .	28
Using HSV's . . . . .	28
Writing . . . . .	29
Reading . . . . .	29
Notifications . . . . .	29
Atom Numbering for HSV's . . . . .	30
Argument Types for HSV's . . . . .	30
Kinds of HSV's . . . . .	31
Scalar HSV's . . . . .	31
Vector HSV's . . . . .	32
Array HSV's . . . . .	33
A Finite State Machine View of HyperChem . . . . .	33
An HSV Server View of HyperChem . . . . .	35
. . . . .	35

### **Chapter 5 Custom Menus 37**

Introduction . . . . .	37
Script Menu Items . . . . .	37
Menu Files . . . . .	38
Simple Example . . . . .	43
Further Customization . . . . .	47

## Chapter 6 Type 1 (Hcl) Scripts 49

Introduction . . . . .	49
Hcl Script Commands . . . . .	49
HSV's . . . . .	49
Menu Activations . . . . .	50
Direct Commands . . . . .	50
Script Files . . . . .	52
CHEM.SCR . . . . .	53
Compiled Scripts . . . . .	54
Recursive Scripts . . . . .	54
Script Editor . . . . .	54
Examples . . . . .	54
Reactive Collision of Two Molecules . . . . .	55
Assign Target Position . . . . .	55
Assign Collision Velocities . . . . .	56
Wave Function Computation Parameters . . . . .	56
The Collision . . . . .	57
Building and Optimizing C60 . . . . .	58
Setup . . . . .	58
Drawing the First Pair of Atoms . . . . .	59
Finish First Level Pentagon . . . . .	59
Build Remaining Layers . . . . .	60
Color Bottom and Rotate . . . . .	64
Zoom Structure . . . . .	65
Create an SO2 Molecule Inside C60 . . . . .	66
Optimize SO2 inside Cavity . . . . .	66
Catalog of HSV's and Direct Script Commands . . . . .	66

## Chapter 7 Type 2 (Tcl/Tk) Scripts 95

Introduction . . . . .	95
Elements of Tcl . . . . .	96
Books . . . . .	96
Internet . . . . .	96
What is Tcl? . . . . .	97
Commands and Arguments . . . . .	97
Variables and Values . . . . .	97
Command Substitution . . . . .	97
Procedures and Control Structures . . . . .	98
Tk . . . . .	98
Hcl Embedding . . . . .	99
hcExec . . . . .	99



hcQuery . . . . .	99
Examples. . . . .	100
Calculating the Number of Atoms. . . . .	100
Calculating a Dipole Moment . . . . .	102
Labels . . . . .	103
Button . . . . .	104

## **Chapter 8 DDE Interface to HyperChem 107**

Introduction . . . . .	107
DDE versus HAPI. . . . .	107
Use of DDE in Windows Applications . . . . .	107
Basic Properties of DDE . . . . .	108
DDE Message Types . . . . .	108
DDE_INITIATE . . . . .	109
DDE_EXECUTE . . . . .	109
DDE_REQUEST . . . . .	109
DDE_ADVISE . . . . .	109
DDE Interface to Microsoft Word . . . . .	109
Red and Green Example . . . . .	110
Extended Example . . . . .	113
ActivateHC . . . . .	113
ConnectHC: . . . . .	114
ExecuteCmd . . . . .	114
GetData. . . . .	114
DisconnectHC. . . . .	115
DDE Interface to Microsoft Excel . . . . .	115
Red (and Green) . . . . .	116
Additional Macros . . . . .	117

## **Chapter 9 DDE and Visual Basic 119**

Introduction . . . . .	119
VB for GUIs or Computation . . . . .	119
VB with DDE or HAPI Calls . . . . .	119
Red and Green . . . . .	120
Basic Form and Controls . . . . .	120
Start Up (Load) . . . . .	121
A Cold Link Request. . . . .	122
A Hot Link . . . . .	123
Execute . . . . .	123
Unload . . . . .	123

A HAPI Interface to VB . . . . .	124
----------------------------------	-----

## **Chapter 10 External Tcl/Tk Interface 127**

Introduction . . . . .	127
Why External? . . . . .	127
Invoking External Tcl/Tk . . . . .	128
The THAPI package . . . . .	129
Commands . . . . .	130
hcConnect <instance>. . . . .	130
hcDisconnect . . . . .	130
hcExec hcl_script_command . . . . .	130
hcQuery hsv . . . . .	130
hcCopy source_file desination_file. . . . .	130
hcNotifyStart hsv . . . . .	131
hcNotifyStop hsv . . . . .	131
hcGetNotifyData notification_data. . . . .	131
hcSetTimeouts exec_timeout query_timeout rest_timeout . . . . .	131
hcLastError error_text. . . . .	131
hcSetErrorAction action_flag . . . . .	131
hcGetErrorAction . . . . .	132
A Notification Example . . . . .	132

## **Chapter 11 The HAPI Interface to HyperChem 137**

Introduction . . . . .	137
Towards a Chemical Operating System. . . . .	138
The Components. . . . .	138
The HAPI Calls . . . . .	139
Initialization and Termination . . . . .	140
BOOL hcInitAPI (void) . . . . .	140
BOOL hcConnect (LPSTR lszCmd) . . . . .	140
BOOL hcDisconnect (void) . . . . .	140
void hcExit(void) . . . . .	140
Discussion . . . . .	140
Text-based Basic Communication Calls . . . . .	140
BOOL hcExecTxt (LPSTR script_cmd) . . . . .	140
LPSTR hcQueryTxt (LPSTR var_name) . . . . .	140
Discussion . . . . .	140
Binary-based Basic Communication Calls . . . . .	141
BOOL hcExecBin (int cmd, LPV args, DWORD args_length) . . . . .	141
LPV hcQueryBin( int hsv, int indx1, int indx2, int* length) . . . . .	141

Discussion . . . . .	141
Binary Format . . . . .	141
Binary-based Get Integer Calls . . . . .	142
int hcGetInt (int hsv) . . . . .	142
int hcGetIntVec( int hsv, int* buff, int max_length) . . . . .	142
int hcGetIntArr (int hsv, int* buff, int max_length) . . . . .	142
int hcGetIntVecElm (int hsv, int index) . . . . .	142
int hcGetIntArrElm (int hsv, int atom_index, int mol_index) . . . . .	142
Discussion . . . . .	143
Binary-based Get Real Calls . . . . .	143
double hcGetReal (int hsv) . . . . .	143
int hcGetRealVec( int hsv, double* buff, int max_length) . . . . .	143
int hcGetRealArr (int hsv, double* buff, int max_length) . . . . .	143
double hcGetRealVecElm (int hsv, int index) . . . . .	143
double hcGetRealArrElm (int hsv, int atom_index, int mol_index) . . . . .	143
int hcGetRealVecXYZ (int hsv, index, double* x, double* y, double* z) . . . . .	143
int hcGetRealArrXYZ (int hsv, int atom_index, int mol_index, double* x, double* y, double* z) . . . . .	143
Discussion . . . . .	144
Binary-based Get String Calls . . . . .	144
int hcGetStr (int hsv, char* buff, int max_length) . . . . .	144
int hcGetStrVecElm (int hsv, int index, char* buff, int max_length) . . . . .	144
int hcGetStrArrElm (int hsv, int atom_index, int mol_index, char* buff, int max_length) . . . . .	144
Discussion . . . . .	144
Binary-based Set Integer Calls . . . . .	144
int hcSetInt (int hsv, int value) . . . . .	144
int hcSetIntVec( int hsv, int* buff, int length) . . . . .	144
int hcSetIntArr (int hsv, int* buff, int max_length) . . . . .	144
int hcSetIntVecElm (int hsv, int index, int value) . . . . .	145
int hcSetIntArrElm (int hsv, int atom_index, int mol_index, int value) . . . . .	145
Discussion . . . . .	145
Binary-based Set Real Calls . . . . .	145
int hcSetReal (int hsv, double value) . . . . .	145
int hcSetRealVec( int hsv, double* buff, int length) . . . . .	145
int hcSetRealArr (int hsv, double* buff, int length) . . . . .	145
int hcSetRealVecElm (int hsv, int index, double value) . . . . .	145
inthcSetRealArrElm(inthsv,intatom_index,intmol_index,doublevalue) . . . . .	145

int hcSetRealVecXYZ (int hsv, index, double x, double y, double z)	145
int hcSetRealArrXYZ (int hsv, int atom_index, int mol_index, double x, double y, double z)	146
Discussion	146
Binary-based Set String Calls	146
int hcSetStr (int hsv, char* string)	146
int hcSetStrVecElm (int hsv, int index, char* string)	146
int hcSetArrElm (int hsv, int atom_index, int mol_index, char* string)	146
Discussion	146
Get and Set Blocks	146
int hcGetBlock (int hsv, char* buff, int max_length)	146
int hcSetBlock (int hsv, char* buff, int length)	147
Discussion	147
Notification Calls	147
int hcNotifyStart (LPSTR hsv)	147
int hcNotifyStop (LPSTR hsv)	147
int hcNotifySetup (PFNB pCallBack, int NotifyWithText)	147
int hcNotifyDataAvail (void)	147
int hcGetNotifyData (char* hsv, char* buff, int max_length)	147
Discussion	147
Memory Allocation	147
void * hcAlloc (size_t, n_bytes)	147
hcFree (void* pointer)	148
Discussion	148
Auxiliary Calls	148
void hcShowMessage (LPSTR message)	148
void hcSetTimeouts(intExecTimeout,intQueryTimeout,intOtherTimeout)	148
int hcLastError (char* LastErr)	148
int hcGetErrorAction (void)	148
void hcSetErrorAction (int err)	148
Discussion	148
The HAPI Dynamic Link Library (HAPI.DLL)	148
How to use the HyperChem API	149
Accessing the HyperChem API from C/C++ code	150
Run-Time Dynamic Linking	150
Load-Time Dynamic Linking	151
Accessing the HyperChem API from Fortran code	152
Accessing the HyperChem API from Visual Basic Code	154
Accessing the HyperChem API from Tcl/Tk code	155
Considerations for Console-based Applications	155

The Notification Agent . . . . .	156
Examples of HAPI Calls . . . . .	157
C, C++ . . . . .	157
Text-based . . . . .	157
Binary-based . . . . .	157
Fortran . . . . .	157
Text-based . . . . .	157
Binary-based . . . . .	158
Visual Basic . . . . .	158
Text-based . . . . .	158
Binary-based . . . . .	158

## **Chapter 12 Development Using the Windows API 159**

Introduction . . . . .	159
Microsoft Development Tools . . . . .	159
Programming Assistance . . . . .	159
Language . . . . .	160
A First Example . . . . .	160
Modification of a Molecule's Coordinates . . . . .	165

## **Chapter 13 Development Using the MFC 173**

Introduction . . . . .	173
Microsoft Development Tools . . . . .	173
Programming Assistance . . . . .	173
Language . . . . .	174
A First Example . . . . .	174
Modifications . . . . .	176
Included Files . . . . .	180
Dynamic Link Library and Connecting to HyperChem . . . . .	181
Cavity. . . . .	182

## **Chapter 14 Console C and Fortran Applications 187**

Introduction . . . . .	187
Console Applications. . . . .	187
C or Fortran. . . . .	188
The Integrated Development Environment . . . . .	189
C Program . . . . .	189
Fortran Programs . . . . .	191
Reflect . . . . .	192

MiniGauss Orbitals . . . . .	195
Outline . . . . .	195
A New GUI Element . . . . .	196
The Main Program . . . . .	198
Get Molecule . . . . .	199
Wave function Calculation . . . . .	202
Displaying Orbitals and ... . . . .	202
Diffusion Limited Aggregation . . . . .	204
Further Examples . . . . .	204

## **Chapter 15 Classification of Hcl Commands 205**

The Classes . . . . .	205
General Operations . . . . .	206
Single Point . . . . .	206
Solvation . . . . .	207
Fields . . . . .	207
Customization . . . . .	207
Printing . . . . .	207
Other . . . . .	207
Cursors . . . . .	208
Mouse Mode . . . . .	208
Clipping . . . . .	208
Rotation . . . . .	208
Translation . . . . .	208
Zoom . . . . .	208
Selections . . . . .	209
Select Options . . . . .	209
Select . . . . .	209
Ask About Selection . . . . .	209
Operate on Selection . . . . .	209
Named Selections . . . . .	209
Other . . . . .	210
File Operations . . . . .	210
Molecule File . . . . .	210
Options . . . . .	210
PDB File . . . . .	210
Import/Export . . . . .	211
Other . . . . .	211
Scripts . . . . .	211
Script Files . . . . .	211
Execution . . . . .	212
Notifications . . . . .	212

OMSGs . . . . .	212
Menus . . . . .	212
Stack Operation . . . . .	213
Other . . . . .	213
Info . . . . .	213
Errors . . . . .	213
Logging . . . . .	214
Auxiliary . . . . .	214
Declarations . . . . .	214
Warnings . . . . .	214
Screen Output . . . . .	215
Version . . . . .	215
Other . . . . .	215
Viewing . . . . .	215
Alignment . . . . .	215
Redisplay . . . . .	215
Rotation. . . . .	215
Translation . . . . .	216
Window. . . . .	216
Other . . . . .	216
Rendering . . . . .	216
General Options . . . . .	216
Specific Rendering Options . . . . .	217
Show - Don't Show . . . . .	217
Coloring and Labeling . . . . .	217
Color . . . . .	217
Labels . . . . .	218
Images . . . . .	218
Model Building . . . . .	218
Options . . . . .	219
Drawing . . . . .	219
Constraints. . . . .	219
Other . . . . .	219
Stereochemistry . . . . .	219
Atom Properties . . . . .	220
Labels . . . . .	220
Coordinates and Velocities . . . . .	220
Other . . . . .	221
Molecule Properties . . . . .	221
Charge-Multiplicity . . . . .	221
Counts . . . . .	221
Labels . . . . .	221
Properties . . . . .	221

Back Ends . . . . .	222
Basic . . . . .	222
Large Communication Structures . . . . .	222
Remote Back Ends. . . . .	222
Molecular Mechanics Calculations . . . . .	222
Method . . . . .	222
Energy Components . . . . .	223
Cutoffs . . . . .	223
Scale Factors . . . . .	223
Parameters . . . . .	223
Amino Acids and Nucleic Acids . . . . .	223
Amino Acids . . . . .	223
Nucleic Acids . . . . .	224
General Residue . . . . .	224
Molecular Dynamics and Monte Carlo. . . . .	225
Basic . . . . .	225
Run Parameters. . . . .	225
Averaging . . . . .	226
Playback . . . . .	226
Monte Carlo Specific . . . . .	226
Optimization . . . . .	226
Basic . . . . .	226
Restrains . . . . .	227
General Quantum Mechanics. . . . .	227
Input Parameters . . . . .	227
Output Results . . . . .	228
Semi-empirical Calculations . . . . .	228
General . . . . .	228
Huckel . . . . .	228
ZINDO . . . . .	228
Ab Initio Calculations . . . . .	229
Input Options . . . . .	229
Basis Set . . . . .	229
2-electron Integrals . . . . .	229
Results . . . . .	229
TNDO . . . . .	230
DFT . . . . .	230
Configuration Interaction . . . . .	230
Infrared Spectra . . . . .	230
Animations . . . . .	230
Spectra . . . . .	231
UV Spectra . . . . .	231
Plotting . . . . .	231



General Options . . . . .	231
2D . . . . .	232
3D . . . . .	232
Grid . . . . .	233
<b>Chapter 16 Listing of Tcl Commands</b>	<b>235</b>
The Tcl Commands . . . . .	235
<b>Chapter 17 Classification of HAPI Calls</b>	<b>241</b>
The API functions . . . . .	241
Functions for Initialization and Termination . . . . .	241
Functions for Text-based Communication . . . . .	245
Functions for Binary Communication . . . . .	248
Binary Execute and Query . . . . .	248
Functions for Binary ‘Get’ . . . . .	251
Functions for Binary ‘Set’ . . . . .	268
Functions for Processing Notifications . . . . .	284
Functions For Memory Allocation . . . . .	290
Auxiliary Functions . . . . .	292
<b>Chapter 18 Index</b>	<b>299</b>

# Chapter 1

## Introduction

### The Chemist's Developer Kit

The Chemist's Developer Kit (CDK) allows you to:

- Customize HyperChem For Your Own Purposes
- Interface Your Own Programs to HyperChem

### CDK for Windows or NT

This manual describes the Chemist's Developer Kit (CDK) to be used in association with Release 5.0 or greater of HyperChem® for Windows and NT. The CDK allows you to *customize* HyperChem for your own special purposes, by modifying its menus and introducing scripts that each custom menu item can invoke. Alternatively, you can use the CDK to *interface* HyperChem to external programs written in Visual Basic, C, Fortran, etc. The CDK for Windows and NT, as described here, allows you to extend HyperChem in a multitude of ways but always within the confines of Microsoft Windows 95 (or greater) or Windows NT 3.51 (or greater).

### Equivalent Unix CDK

A similar but somewhat modified CDK and manual will be made available for Unix. That manual will describe how to, correspondingly, customize Unix versions of HyperChem and how to interface a UNIX version of HyperChem to programs written for Unix. It will also describes how to interface a Windows or NT version of HyperChem to your Unix program.

## Components of the CDK

The CDK is included as a part of the general Release 5.0 of HyperChem for Windows 95 and NT. Example scripts and programs are found in sub-directories on the HyperChem CD-ROM associated with the the relevant programming environment - C, Fortran, etc. The current CDK has only limited capability with older versions of HyperChem.

### Components Included with Release 5

The CDK essentially consists of:

- A HyperChem Executable (Release 5.0 or greater)
- This Manual
- Example Custom Menu Files (\*.MNU)
- Example Hcl Scripts (\*.SCR)
- Example Tcl/Tk Scripts (\*.TCL)
- The HyperChem Application Programming Interface Library (HC.H, HCLoad.C, HSV.H, HAPI.DLL and HAPI.LIB for C/C++ environments)
- Related HAPI Files for Other Languages
- Example Programs Interfaced to HyperChem

### Other Suggested Tools

To customize HyperChem no other tools are required. However, to explore interfacing HyperChem to other Windows programs, one or more of the following may be useful:

- Microsoft Word
- Microsoft Excel
- Microsoft Visual Basic

These three have been used in this manual to illustrate the low-level DDE interfaces to HyperChem described in Chapters 8 and 9. Other Windows word processors (for example, WordPerfect) or spreadsheets (for example, Quattro Pro) could be used in place of Word and Excel but the authors have little experience with them. Visual Basic is preferred by the authors as a rapid

prototyping language for preparing a visual interface to HyperChem. Again, other such visual tools are available from Borland and other manufacturers.

Visual Basic may also be used with the higher-level HyperChem Application Programming Interface (HAPI) replacing the lower-level DDE interface.

## Suggested Compilers

To explore compiled C, C++, or Fortran programs interfaced to HyperChem, appropriate compilers and development environments are required. We have used

- Microsoft Visual C++ 4.0
- Microsoft Fortran PowerStation 4.0

Equivalent compiler tools from other manufacturers, such as Watcom, may be used but the examples of the CDK have only been tested with the Microsoft tools.

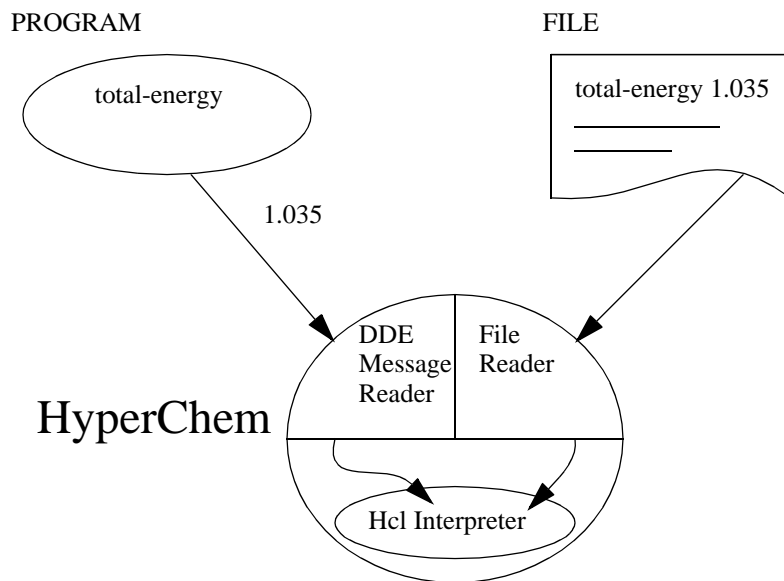
## HyperChem State Variables

The CDK and much of what follows are possible because HyperChem implements the concept of a *state variable*. A HyperChem State Variable (HSV) is one of hundreds of variables and data structures that is registered at the instantiation of HyperChem and is henceforth available for flexible and robust reading and writing at any later time. An example of an HSV is the total energy of the system which is represented by the hyphenated string,

total-energy

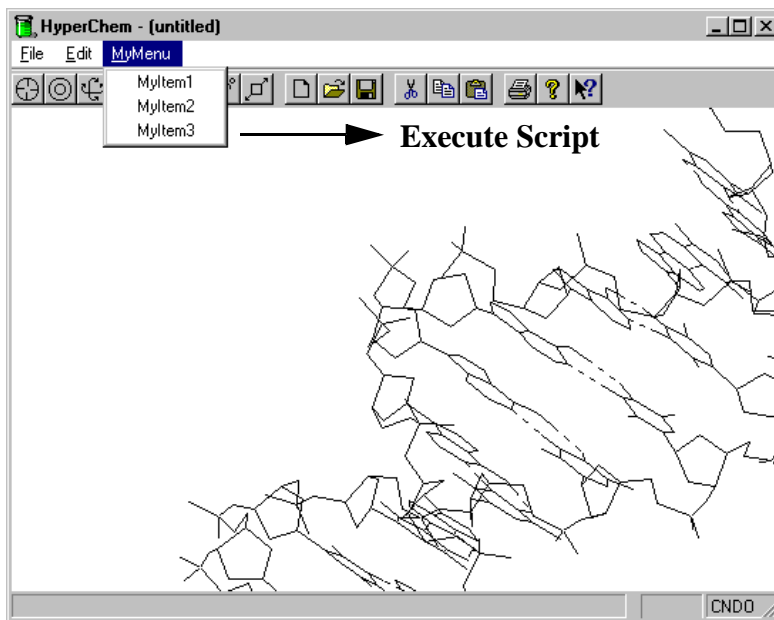
At any time, even when the system still has no molecule or no calculation has yet been performed on the molecular system (and the total energy as the ini-

tialized value zero), the HSV, *total-energy*, is available for reading and writing, either within a HyperChem script or from an external program.



## Customizing HyperChem

HyperChem has two quite unique features that allow you to customize it. The first of these is that it has its own set of *script commands* that can activate essentially any of the program's functionality. The second is that HyperChem allows its whole menu structure to be replaced by *custom menus*.



## Internal Script commands

Script commands each consist of a single line of text, such as *do-molecular-dynamics* and *menu-file-open*, that invoke HyperChem actions or read and write HyperChem State Variables (HSV's) via a Scripting Interface (SI) that is a superset of the interactive Graphical User Interface (GUI) available via mouse and keyboard. Thus a molecular dynamics trajectory can be initiated by mouse clicks on the appropriate menu items and dialog boxes or by the above script commands. Alternatively, one can bring up the `<File/Open...>`<sup>1</sup> dialog box by clicking on the appropriate menu item or by executing the appropriate script command, *menu-file-open*. Script commands can be of two types, Hcl or Tcl/Tk.

1. Here and throughout the text we will periodically use angular brackets to delimit menu items (and other text) so as not to confuse it with surrounding text.

## HyperChem Command Language (Hcl)

Scripts can be written in the *HyperChem Command Language (Hcl)*, pronounced “hickle”, that has been part of HyperChem since its first public release. These scripts, now referred to in Release 5.0 as Hcl scripts consist of a simple sequence of Hcl commands (strings), such as the *do-molecular-dynamics* script command referred to above. Hcl scripts are stored in \*.SCR files that contain only straight line code consisting of a sequence of single line Hcl commands. Hcl scripts contain no Variables or Control Statements such as IF, ELSE, DO, etc.

## Tool Command Language (Tcl/Tk)

With Release 5.0, HyperChem can now execute a *Tcl/Tk Script* that consist of Hcl script commands embedded inside a conventional Tcl/Tk script. A Tcl/Tk script is one that is interpreted by the well known public domain Tcl command interpreter that is now part of HyperChem Release 5.0. The Tcl interpreter allows a rich control structure of variables, loops, conditionals, etc. It is augmented by a tool kit (Tk) that allows a Tcl script to define new graphical elements, such as dialog boxes, allowing you to extend HyperChem’s GUI.

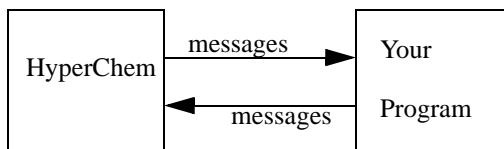
## Custom Menus

HyperChem Release 5.0 allows a user to replace the standard menu structure of the shipped product with a totally new and custom menu structure. Each custom menu item can have its own button text and can be tied to the execution of an arbitrary script file, either \*.SCR or \*.TCL. Since all the conventional menu items of HyperChem Release 5.0 have equivalent script commands such as *menu-file-open*, the custom menus can replicate the standard HyperChem product as well as define essentially any new product that one likes. The custom menus allow a Tabula Rasa, or blank page, on which a new chemistry product can be written.

# Interfacing to HyperChem

A principal component of the CDK is the documentation, libraries, and examples that allow you to interface your own codes to HyperChem. Customizing HyperChem, as implied above, means the *internal* execution of script commands brought about by reading them from a simple text file (\*.SCR or \*.TCL). Interfacing to HyperChem on the other hand, implies that an *exter-*

*nal* program (your own) executes the equivalent of script commands by sending them as *messages* to a running copy of HyperChem.



That is, an external program can drive HyperChem from outside by sending it script messages. Interfacing to HyperChem does not imply that you statically link your code together with HyperChem code but rather that you completely control HyperChem from outside and read and write to HyperChem's data structures via external messages.

## Dynamic Data Exchange

With most computer operating systems, running programs can send messages to one another. In the Unix environment these messages are often implemented with Pipes or Sockets and the Unix version of HyperChem can accept messages sent by external programs in this fashion. Microsoft Windows and NT include a capability, referred to as Dynamic Data Exchange (DDE), that is an equivalent capability allowing one Windows or NT program to communicate with another by sending it DDE messages.

## External Script Messages

In the Microsoft Windows or NT environment a program external to and independent of HyperChem can drive or control HyperChem by sending it DDE messages. HyperChem responds to a complete set of *script messages*, implemented via DDE, that are analogous and essentially identical to the *script commands* discussed above. A script command is text placed in a file while a script message is the corresponding text placed in a message. Thus a HyperChem molecular dynamics trajectory can be initiated by running a script with the text *do-molecular-dynamics* in a \*.scr or \*.tcl file or by sending a DDE message containing the same text.

These external DDE messages can be sent to HyperChem from essentially any well-designed Microsoft Windows or NT program. Thus, HyperChem can be driven from a word processor like Microsoft Word, a spreadsheet like



Microsoft Excel, a simple Visual Basic program or a C, C++, or Fortran program.

## The HyperChem Application Programming Interface (HAPI)

The CDK includes a dynamic link library (HAPI.DLL) and a static library (HAPI.LIB) that makes it particularly easy to interface your compiled Visual Basic, C, C++, or Fortran program to HyperChem. Instead of using DDE, an external program can just make HAPI calls to HyperChem. These HAPI calls are part of a higher-level interface that is included with the CDK and which abstracts away from operating system and machine dependencies to give you a portable interface to HyperChem. Everything you can do with the lower-level DDE interface you can accomplish with the HyperChem API.

## Overview of Chapters

The *CDK Manual* contains the following chapters:

- Chapter 1, this “*Introduction*,” discusses the general nature of the components of the Chemist’s Developer Kit and gives an overview of the rest of the chapters.
- Chapter 2, “*Architecture of HyperChem*,” discusses the general architecture of HyperChem including its *Front End - Back End* Architecture, its *Network* Architecture, its *Client-Server* Architecture, and its *Open* Architecture. Each of these architectural aspects is important for the richer understanding of HyperChem that is desirable in order for you to be able to easily customize it or interface your own code to it. Of particular importance for the CDK is an understanding of the Open Architecture of HyperChem.
- Chapter 3, “*Customizing HyperChem*,” describes script commands and the execution of script files, either HyperChem Hcl script files or Tcl/Tk Script files. Hcl script files contain simple sequences of script commands. Tcl/Tk Script files contain normal Tcl/Tk code with embedded Hcl script commands. This chapter, in addition, gives a full description of the custom menu capability.
- Chapter 4, “*HyperChem State Variables*,” describes the concept of HyperChem State Variables (HSV’s) in detail. The basic data structures of HyperChem are HSV’s. These are registered by HyperChem at instantiation and are made available for reliable Reading and Writing by scripts and by external programs. This chapter provides background for

later chapters which use HSV's in scripts and in the HyperChem Application Programming Interface (HAPI).

- Chapter 5, "*Custom Menus*," describes the concept of custom menus and the syntax of a menu file. A menu file describes a set of menus and the scripts that are executed upon selecting a menu item. This chapter fully defines the use of custom menus in HyperChem.
- Chapter 6, "*Type 1 (Hcl Scripts)*," describes the HyperChem Command Language (Hcl) and scripts based upon it. These scripts are straight-line scripts that access HyperChem data and functionality but include no control structures. Even without the superstructure provided by the control structures of other languages, however, Hcl scripts can accomplish significant tasks on their own.
- Chapter 7, "*Type 2 (Tcl/Tk) Scripts*," describes a new feature of HyperChem, the inclusion of a very flexible and extensible scripting language referred to as the Tool Command Language (Tcl). It includes an extension called the Toolkit (Tk) that can be used to build additional graphical user interfaces for HyperChem. All Type 1 (Hcl) script commands are fully imbedded in the new Tcl/Tk interpreter.
- Chapter 8, "*DDE Interface to HyperChem*," describes the low level DDE interface to HyperChem that allows the direct control of HyperChem by programs like Word or Excel. A first example of HyperChem interfacing is given by having these external programs execute simple script messages to affect the visual appearance of the HyperChem screen. Subsequent, more complicated examples, are also described.
- Chapter 9, "*DDE and Visual basic*," describes how programs written in Visual Basic can be interfaced to HyperChem via DDE. This extends the discussion of the last chapter to a more fully programmable system like Visual Basic. It contrasts with later Visual Basic applications that use the HAPI library.
- Chapter 10, "*The External Tcl/Tk Interface*," describes an interface to HyperChem where external forms of Tcl and Tk are used to interface to HyperChem. Use of theTcl/Tk interpreter from outside HyperChem allows for certain operations, such as notification, that are not possible with the interpreter imbedded right into HyperChem.
- Chapter 11, "*HAPI Interface to HyperChem*," describes a higher level library for interfacing to HyperChem for use by Visual Basic, C, C++, and Fortran programs. This HyperChem Application Programming Interface (HAPI) is described along with the calls that it contains. The

HAPI library is a fundamental component of the CDK. It is illustrated with a Visual Basic example.

- Chapter 12, “*Development Using the Windows API*,” describes the development of Windows and NT programs that interface to HyperChem. The type of development presented in this chapter uses the Window’s System Developer’s Kit (SDK) approach, along with HAPI calls. The SDK approach is a detailed and more fundamental way to develop Windows programs than the approach of the next chapter.
- Chapter 13, “*Development Using the MFC*,” describes the development of Windows and NT programs that interface to HyperChem and use the Microsoft Foundation Classes (MFC). The MFC, in conjunction with C++, enables one to build a Windows program much more quickly but with somewhat less flexibility than with the SDK of the last chapter. Development of interfaces to HyperChem using the MFC are described.
- Chapter 14, “*Console C and Fortran Applications*,” describes the interface between Microsoft “console” applications (having no GUI) and HyperChem. The emphasis here is on taking “legacy” Fortran programs and having HyperChem provide a GUI for them.
- The three Appendices describe the complete set of Hcl script commands, the set of Tcl/Tk script commands, and the details of each HAPI call.

Finally, the HyperChem CD-ROM and the associated installation of HyperChem 5.0 will provide you with a number of examples of the scripts, menu files, and applications either discussed or perhaps not discussed in this CDK manual. However, further material associated with the CDK will be found on Hypercube’s WWW site (<http://www.hyper.com>) and you should regularly check that site for additional help.

# Chapter 2

## Architecture of HyperChem

### Introduction

This chapter contains information on the following architectural features of HyperChem:

- The Front End - Back End Architecture
- The Master - Slave Architecture
- The Open Architecture
- The Client - Server Architecture
- The Network Architecture

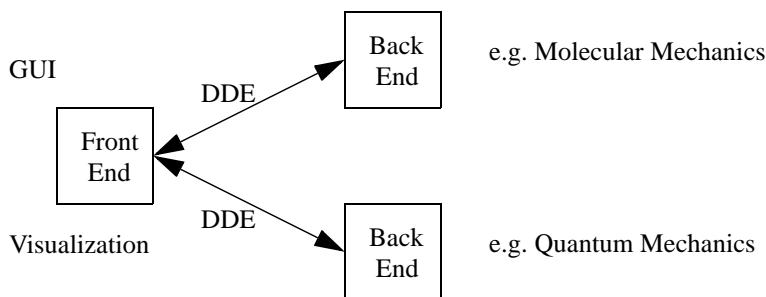
The discussion here is given as general background on how HyperChem is constructed so that you are in a better position to appreciate how and why customization and interfacing become possible. The material of this chapter is not strictly required for what follows in later chapters but should assist you in understanding the CDK.

Since HyperChem, like most large commercial products, has an evolving design and is certainly not the product of a single, totally rational, design process, what is presented here is somewhat of a combination snapshot of both the way it is and the way it is becoming. Nevertheless, what is described here represents Release 5.0 in most regards.

### The Front End - Back End Architecture

HyperChem basically consists of one monolithic *Front End* and numerous *Back Ends*. In the Windows and NT environment each of these components has its own icon and is a completely separate program. For example, the green beaker icon represents the HyperChem front end program. The falling red

apple icon represents one of the back end programs - in this case, HyperNewton.



The front end is the program that you interact with. It accepts input from you, via the mouse and the keyboard, which constitute the GUI, and it provides you with visualization services. For example, it may render a drawing of a molecule for you.

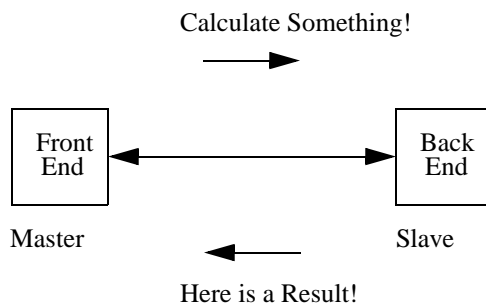
The back ends generally perform the compute intensive computations. HyperChem Release 5.0 comes with 5 back ends - HyperMM+, HyperNewton, HyperEHT, HyperNDO, and HyperGauss. In principle, the back ends compute only the energy of interaction of atoms and the first and second derivatives of these energies. These energetic quantities feed the front end which then computes chemically relevant properties. In practice, the subdivision of labor between a back end program and the HyperChem front end is more complicated than this and depends on the situation.

One of the first things you might wish to consider as an application of the CDK is to implement your own back end to replace one of the HyperChem back ends. For example, you might like to have your own unique force field in replacement of the MM+, Amber, etc. force fields of HyperChem.

In the Windows or NT environment, the HyperChem front end and the various back ends communicate via DDE although one might have thought that they would communicate through files. The HyperChem front end - back end DDE communication, however, is more of a *live link* than would have been possible via normal file reading and writing.

## The Older Master - Slave Architecture

The normal front end - back end relationship in HyperChem Release 5.0 is a master - slave relationship, i.e. the front end is the master over a back end slave.



The back ends do not initiate anything and do only as they are told by the front end. If the front end requires the energy of a molecule, for example, it will know whether a back end slave is idle, it will send the slave a molecule with the instructions, “compute its energy,” and it will then wait for the slave to return the result. The slave, when it is sent a molecule and the instructions for computing some energetic quantity, will do so, returning the result to the front end when it is finished. The slave will then return to an idle state waiting for further instructions.

The Master - Slave Architecture is considered an older architecture by Hypercube. It is slowly being replaced by a newer Client - Server Architecture as described in the next section. However, the existing back ends being shipped by Hypercube as part of Release 5.0 all still use the master - slave relationship. The protocol between a master and a slave in HyperChem is, and has always been, an unpublished proprietary protocol. As such, knowing the explicit details of the protocol is essentially irrelevant to the CDK and to you, the user. It is presented here so that you can understand the basic architecture of HyperChem. The newer client-server protocol described below is the protocol used by the CDK and is the one that you should expect to use in interfacing to HyperChem.

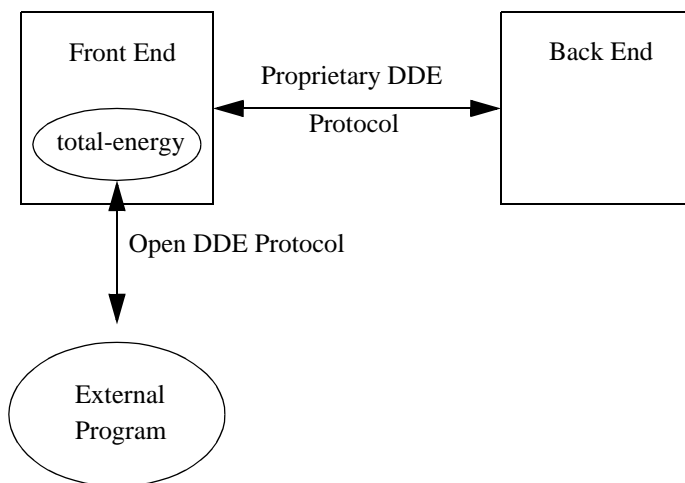
## The Open Architecture

HyperChem has an *Open Architecture* in that:

- It can be driven by messages coming from other programs
- Other programs can read and write its internal data structures

The normal operation of HyperChem is via its GUI where you use the mouse and keyboard to operate HyperChem's menus and dialog boxes. HyperChem attempts to allow an external program to operate it in the same way that you do sitting in front of the screen, except that the external program sends script messages (rather than clicking on a key or a mouse button, which a program can't do).

For example, an external program can access an internal variable, such as the total energy, either for reading or writing, by sending a DDE message to the HyperChem front end.



Only front end variables are available to external programs and all information and state held solely by the back end is private to HyperChem. For example, a back end program probably calculated the total energy but conveyed it to the front end where it resides in a front end data structure.

These front end variables that are made available to external programs for reading and writing are referred to as HyperChem State Variables (HSV) and are part of the front end state. The front end HSV's are available for reading and writing at any time whether or not a back end has computed a value. The default value, in this case, is zero.

## UMSG and VMSG

Another way of looking at the open architecture of HyperChem is by way of its message interface. In the normal use of HyperChem, via the GUI, a user provides input with mouse clicks and keyboard clicks. We refer to these inputs as user messages. Each constitutes a UMSG. The result of a UMSG is that the internal state of HyperChem changes and a visual change may appear on the screen. We refer to these visual changes as HyperChem having emitted a VMSG. While the user is not explicitly aware of having sent and received these messages, HyperChem does indeed operate this way.



## IMSG and OMSG

The open architecture implies that messages equivalent to a UMSG or VMSG can be sent and received by external programs as well as interacting human users. For every UMSG there is expected to be an equivalent input message (IMSG). However, IMSGs are a superset of UMSGs because there are many times a program will want to interface to HyperChem in a way that a human user would not. For example, a program may want to initiate a molecular dynamics trajectory without bringing up a dialog box that a human must respond OK to. The IMSG may or may not generate an output message (OMSG) that augments the VMSG. These IMSGs and OMSGs constitute real data that are conveyed to HyperChem, usually as text strings, by a script or a DDE message.



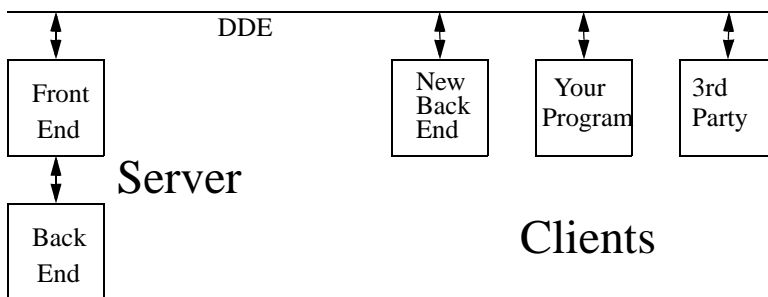
For example, a user clicking with the mouse on the menu item <File/Open> will bring up the File Open dialog box. A program external to HyperChem can do the identical thing by sending an IMMSG,

menu-file-open

The resulting OMSG, in this case, is nil but the VMSG is identical to that generated by the UMSG.

## The Newer Client - Server Architecture

For a number of reasons it makes sense to move away from the master - slave architecture. In particular, if the services of the HyperChem front end are to be made available to you, other users, and third party programmers, you must be in command of the situation, not HyperChem. Thus HyperChem should act as a universal *server* to you, the *client*.



The HyperChem front end has essentially always acted as a server to external programs as, for example, with ChemPlus, HyperNMR, or programs like Microsoft Excel. The CDK, however, extends and documents this capability, making it possible to even replace Hypercube's proprietary back ends with third party back ends.

These client programs are generally started by the HyperChem front end via a custom menu in HyperChem. When the program begins executing it requests services from HyperChem such as asking it to send a copy of the coordinates of the molecule currently on the screen. It might then compute properties of the molecule and then ask HyperChem to display these properties.

A client program need not reflect just a back end operation but could instead augment the front end GUI or visualization capability of HyperChem. Your program can be of arbitrary design using HyperChem only for functionality

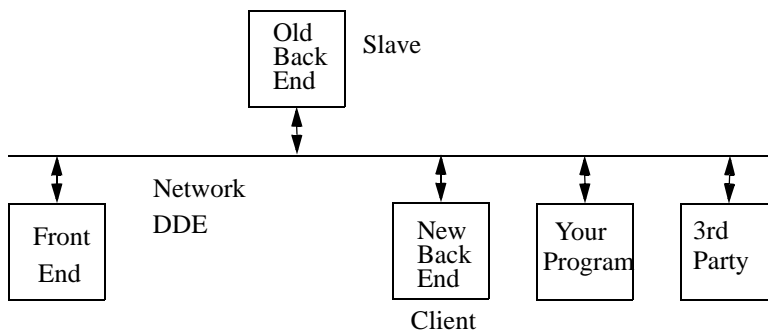
you do not want to reproduce. HyperChem can act as a GUI server, a computational server, or a visualization server all at the same time. The HyperChem back ends are not directly accessible to external programs but only through the HyperChem front end acting as a proxy.

## The Network Architecture

As described above, the HyperChem front end, the HyperChem back ends, and third party programs that are DDE compliant, such as Excel or programs built with the Hypercube CDK, can communicate with each other via messages. This communication is initially assumed to occur on a single PC running Windows or NT.

### Network DDE

Microsoft, however, has implemented Network DDE, in Windows 95 and NT. Thus it is possible to place any of the various components of a solution onto different PCs as long as they are connected by an appropriate network.



An old back end communicates with the front end by means of Hypercube's proprietary protocol while a new back end uses the open client-server protocol of the CDK.

## UNIX

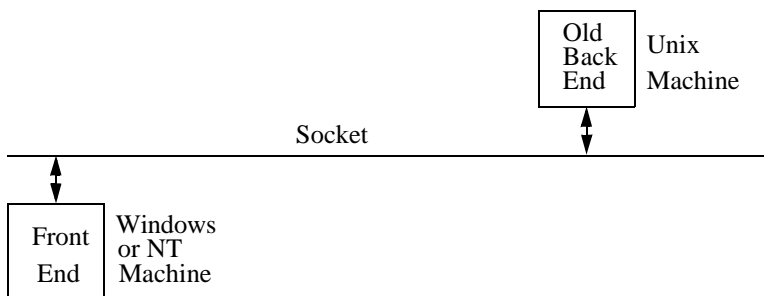
While the following discussion is not strictly relevant to this Windows CDK, a completely analogous CDK and capability is available for Unix versions of HyperChem. In this case DDE is replaced by a Pipe and Socket capability. The Unix versions of HyperChem have a front end - back end architecture as well so that essentially everything we say here applies equally to the Unix

world once DDE messages and communication are replaced by their Unix equivalents. The HyperChem API, on the other hand, is portable across platforms.

## Remote Back Ends

Hypercube, Inc. supports the mixing of Windows and NT with Unix in the sense that Remote Unix Back Ends are available for Unix machines from Digital Equipment Corporation, IBM, Silicon Graphics, and SUN. This allows a desktop PC to use a Unix machine to perform compute intensive back end calculations.

These Unix back ends communicate in the older proprietary way with the Windows or NT front end via a proprietary socket protocol. On the Windows or NT side, DDE is replaced by equivalent socket (the WINSOCK standard) messages.



## Mixing UNIX and Windows or NT

For completeness it ought to be possible to combine the Windows and NT CDK and the Unix CDK to allow a Unix 3rd party client application to communicate properly with a Windows or NT desktop front end server. Alternatively an NT client might like to use a UNIX machine for HyperChem visualization services. It is not yet possible, however, to mix windows and Unix for these CDK functions. One possible way to obtain this capability (Windows front end HyperChem server and Unix back end client) would be to have a 3rd-party Windows program receive Unix socket messages and translate them into DDE messages for Windows HyperChem.

This and the other network aspects of HyperChem are pointed out here for completeness. This manual and the Windows and NT version of the CDK do

not attempt to describe the equivalent Unix product but relate only to the Windows and NT programs described here.



# Chapter 3

## Customizing HyperChem

### Introduction

This chapter describes how HyperChem can be customized via:

- Scripting
- Custom Menus

These two capabilities allow you to automate many of the computations you perform with HyperChem or to customize HyperChem for your own purposes. This chapter does not describe the interfacing of external programs to HyperChem which, in itself, is a form of customization; that is left for later chapters. Here we focus on how HyperChem can be customized through the process of writing a set of scripts and through your ability, in Release 5.0, to redefine the whole menu structure of the program if you chose to do so.

### A Flexible Development Platform

Scripting and custom menus, by themselves, allow you to customize HyperChem in a very large variety of ways. It is possible to, in essence, start from scratch with just a bare Window (no menus) and add only the capabilities you wish. Any menu item can be tied to the execution of an arbitrary script. With Release 5.0 and the CDK, we at Hypercube, Inc. are working towards a Chemical Operating System that allows end users and third-party developers to develop their own products using our tools. With the ability to interface external programs to HyperChem and have them access any of the capabilities of a custom HyperChem, a very flexible chemical development tool becomes available.

## What are Scripts?

A script consists of a sequence of individual *script commands* that are held in a script file. Each script command consists of simple text that can be executed to invoke a HyperChem action, to read or write HyperChem data, etc. A script can be created with a text editor such as the Windows Notepad. A script can be executed in HyperChem by simply opening the script with the HyperChem <Script/Open...> menu item. Alternatively, scripts can be created and executed via the Script Editor of ChemPlus.

There are two kinds of scripts. The first consists of a simple straight-line sequence of commands without any control structures (such as for-loops, if-statements, etc.). These are referred to as Type 1 scripts. The second, much richer, type of script has elaborate control structures and is referred to as a Type 2 script. Type 2 scripts contain Type 1 script commands within them.

### Type 1 (Hcl) Scripts

The first kind of script, the only kind available prior to Release 5.0, is referred to as a Type 1 Script. These consist of a sequence of text lines of the form,

window-color green

The above script command, executed within a script, changes the background workspace (window) color to green. Executing this script command is equivalent to using the mouse and the <File/Preferences...> dialog box in the GUI to change the window color. We will often use this simple script command, one of potentially hundreds, to illustrate many of the basic ideas in the CDK.

With Release 5.0 and the CDK, one now refers to these script commands as being part of the HyperChem Command Language (Hcl). Hcl commands in the HyperChem Command Language, minus any possible arguments, are identifiable as a contiguous sequence of words separated by hyphens, e.g. *menu-file-start-log*.

Type 1 scripts are held in \*.SCR files, i.e they have a default SCR file extension.

### Type 2 (Tcl/Tk) Scripts

The newer scripts in Release 5.0 are referred to as Type 2 Scripts or as Tcl/Tk Scripts. These scripts use a public domain Tcl/Tk interpreter that is part of HyperChem Release 5.0. The interpreter can interpret Hcl script commands

imbedded in normal Tcl/Tk code. The Tcl/Tk interpreter was developed by John Ousterhout and collaborators at the University of California at Berkeley and at Sun Microsystems. It is described in the following books that may be important to you in becoming an efficient developer of Type 2 Scripts:

- Eric F. Johnson, *Graphical Applications with Tcl & Tk*, 1996, M&T Books, New York, N.Y., ISBN 1-55851-471-6.
- John K. Ousterhout, *Tcl and the Tk Toolkit*, 1994, Addison-Wesley, Reading, Mass., ISBN 0-201-6337-X.
- Brent Welch, *Practical Programming in Tcl and Tk*, Prentice Hall, 1994.

It is also described at the following World Wide Web sites:

- <http://www.sunlabs.com:80/research/tcl/>
- <http://www.sco.com/Technology/tcl/Tcl.html>

The Tool Command Language orTcl part of Tcl/Tk describes a very general scripting language that can embed custom scripting operations such as the Hcl script commands. The Tk part of Tcl/Tk describes an extension to Tcl that makes it possible to easily create a graphical user interface (GUI) having menus, dialog boxes, etc. Thus a Type 2 script consists of a sequence of script lines that are a mixture of Tcl/Tk (“tickle”) lines and Hcl (“hickle”) lines.

Type 2 scripts add variables, do- or while-loops, if-statements, and so on to a script, so that a script can now be a very general purpose program having its own GUI, visualization, etc. An simple example of a Type 2 script is:

```
TclOnly
set i 10
while { $i > 0 } {
incr i -1
if { ($i - 2*($i/2)) == 1 } then {
hcExec window-color black } else {
hcExec window-color white }
}
Exit
```



This script causes the background window color to alternate between black and white. The while-loop has the variable *i* going from 10,9,8... to 1. For even values of *i* the window color is set to black while for odd values of *i* it is set to white. The *TclOnly* command indicates that no Tk window is needed. It should be the first command of any simple script that requires no addition to the HyperChem graphical user interface. The *hcExec* (Hyperchem execute) command is always followed by the appropriate Hcl command, which in this case sets the color of the HyperChem workspace window. Note that Tcl/Tk is case sensitive; a command with the wrong case, such as *Tclonly*, would be an invalid command. A Tcl script is normally exited with the *Exit* command.

Type 2 scripts are held in \*.TCL files, i.e they have a default TCL file extension.

The whole while statement above is really a single Tcl/Tk command. The while command contains other embedded Tcl/Tk commands plus embedded Type 1 script commands. The *hcExec* command embeds Hcl commands that can be menu invocations, such as *menu-file-open*, direct commands, such as *do-molecular-dynamics*, or HSV writes, such as *window-color green*. The corresponding *hcQuery* command embeds HSV reads. Thus, the following Tcl script creates a Tk window with a message, called .msg, contained in the window and displaying the coordinates of each of the atoms:

```
message .msg -text [hcQuery "coordinates"]
pack .msg
```

The *pack* command places widgets, such as .msg, in the main Tk window and controls their layout. Tk widgets are normally named to begin with a “dot”.

## Custom Menus

With Release 5.0 of HyperChem, you can now redefine each and every menu item to fully customize HyperChem’s menus. When HyperChem is first invoked, it has a standard set of menus as with any other Windows product. However, if one subsequently executes the Hcl script command:

```
load-user-menu custom.mnu
```

then the standard menus are discarded and replaced by a new set of menus defined by a *menu file* which, in the above case, is the file *custom.mnu*. These menu files define the custom text of each menu button (for each of the new custom menus), and the action that is taken when each menu button is

pushed. The custom action for each menu button, as defined in the file, can be any Hcl script command.

For example, to re-implement the <File/New> menu item exactly as it is shipped in the default product, the custom menu file should request the action,

```
menu-file-new
```

when the appropriate button with text “New” on the “File” menu is pushed. A menu file is basically just a description of the text and the keyboard assist for each new menu button plus an associated Hcl script command for the button.

Because any of the button actions (Hcl script commands) could be of the form,

```
read-tcl-script xxx.tcl
```

it is also possible to have any menu button execute a Tcl script. This results in HyperChem being capable of having very generic menus and functional capability. That is, since a Tcl/Tk script can be executed upon pushing any menu button, and since a Tcl/Tk script can, in principle, accomplish any programming task, there is no fundamental limitation to the generality of HyperChem.

The following menu file, for example, creates a version of HyperChem that has a single menu (“Custom”) with a single menu item (“Version”) which does nothing other than inform the user of the current release number when the menu button is pushed,

```
MENU "&Custom"
  ITEM "&Version",query-value version
END
```

New keyboard accelerators are not available for custom menu items. The default HyperChem keyboard accelerators are always active. However, keyboard assists or shortcuts, usable with the Alt key and indicated by the ampersand, are available.

Before continuing with the description of user-defined menus and scripts for customizing HyperChem, we will first elaborate on the concept of HyperChem State Variables (HSV's). These are fundamental to the whole concept of customization or interfacing.



# Chapter 4

## HyperChem State Variables

### Introduction

This chapter describes a very important feature of HyperChem, the concept of its current state and the variables that represent that state. These HyperChem State Variables (HSV's) define the current state of visualization and computation in HyperChem, e.g. the color of a window or the total energy of a molecule, and can be read and written by scripts or by external programs. these HSV's are registered when HyperChem is invoked and can be reliably queried or modified from then on.

### Registering of HSV's

When HyperChem is first invoked, one of the things it does is to register, for reliable import and export, a large number (hundreds) of data structures. In cases these data structures are just simple variables rather than more complicated arrays, lists, etc. By registering these data structures we mean that of all the internal data structures that come and go dynamically in HyperChem, these ones can be requested to be read or written at any time. Any registered variable can be reliably and robustly accessed. If the HSV value is not yet available, HyperChem will issue a simple warning at the attempted access. You needn't worry about writing an HSV value to HyperChem and causing its internal operation to be badly perturbed, apart from the intended effect of having a new value for an HSV.

### An Example of an HSV

An example of a HyperChem State Variable is *window-color*. The names for these variables are fixed inside the code of HyperChem and may or may not always be optimally descriptive. The variable *window-color* defines the background of the HyperChem workspace (the area displaying a molecular system). It is commonly black but can be changed via the <File/Preferences...>

menu item to be any one of the 8 basic HyperChem colors. The variable is an enumerated variable of type *Enum*. That is, it is one of a small number of pre-defined values represented internally in the computer by an integer but in your program code as some member of the set {Black, Blue, Green, Cyan, Red, Violet, Yellow, or White}.

Another example of an HSV with more chemical content is *dipole-moment*. This is a R/W variable also and behaves just as *window-color*. However, we prefer to use *window-color* in many of our examples because it is immediately visually obvious when this variable changes. The consequences of writing a new value for the dipole moment are also a little obscure. In actuality no harm comes from writing an arbitrary value for the dipole moment to HyperChem; it just may not correspond to the correct dipole moment for the molecule on the screen. It will be overwritten when a calculation, such as a wave function calculation, is performed that computes the dipole moment.

## Read/Write Nature of HSV's

All HSV's can be classified as Read-Only (R) or as Read-Write (R/W). An example of a variable that is read-only is *selected-atom-count*. HyperChem makes extensive use of "atom selections" and this variable reports the total number of currently selected atoms. While it is possible to select atoms via a script, it makes no sense to write new values of this variable that would contradict the number of atoms that HyperChem determines to be currently selected. This number will be automatically updated by HyperChem if an external program were to select some atoms in HyperChem.

## Using HSV's

Here we describe the reading and writing of the simplest type of HSV, a scalar. Later, we will describe the reading and writing of HSV's that have a more complicated structure. We use the simple HSV, *window-color*, as our example.

## Writing

An HSV is written by simply giving its new value after the variable, with or without an equal sign. Thus the following are all appropriate Hcl script commands for turning the background screen color to green:

```

window-color green
window-color = green
WInDOW-CoLOR=GReeN

```

Hcl script commands are case insensitive and spacing is ignored except within the HSV name.

## Reading

Two equivalent syntactical methods are available for reading HSV's. The first is through a Hcl command, *query-value*, which takes an HSV as an argument:

```

query-value window-color

```

The second and completely equivalent way to ask for the value of an HSV is to simply name the HSV placing a question mark after it, separated by at least one space:

```

window-color ?

```

Either of these procedures returns to the questioner (provided the HyperChem screen is green) the output message (OMSG) string:

```

window-color = Green

```

The output message is returned to an external program, if that is who sent the original query or, by default, to a message box on the screen if the original query came from an internal script. Through the use of an Hcl script command, *omsgs-to-file*, the message string generated by the Hcl script can be placed in a file rather than appear in a message box on the screen. Finally, the returning OMSG string could have been shortened to just "Green" rather than "window-color = Green" if the HSV, *query-response-has-tag*, had been set to *false* prior to issuing the original query.

## Notifications

It is possible to make a request to HyperChem that you be sent an OMSG should the value of an HSV change. The message you might eventually receive is identical to that which you could receive by performing a *query-value*. The OMSG is repeatedly sent to you whenever the variable changes it

value. For data structures more complex than simple variables, the notification is sent if any member of the data structure changes. For example, monitoring the window-color is accomplished with the script command,

```
notify-on-update window-color
```

The notification can be cancelled at any later time via the script command

```
cancel-notify window-color
```

Notifications are quite powerful and can be extremely useful. For example, you could ask for a notification of the total-energy during an optimization and easily plot a graph of the optimization with the values sent to you by HyperChem. These notifications are really only meaningful in the context of an external program interfaced to HyperChem rather than in the context of an internal script.

## Atom Numbering for HSV's

It is important in using HSV's to understand how HyperChem numbers atoms. Each molecule (connected graph) in the HyperChem workspace has its own number (1, 2,...). The atoms in any molecule are numbered starting at 1 also. Thus a unique id for an atom includes two numbers - the molecule number and the atom number within the molecule (atom-in-molecule number).

Thus if a number is to be used as a unique index for an atom, it has the form, (atom number, molecule number). For example, (2,1) is the second atom in the first molecule. The atom index always comes before the molecule index in conventional HSV usage.

## Argument Types for HSV's

The types of arguments that HSV variables have are the following:

<b>Boolean</b>	Yes or no, true or false, 0 or 1.
<b>string</b>	Text (letters, characters, or symbols, in upper- or lower-case, unlimited number of characters). Enclose a string in quotes (“ ”) if it contains spaces, tabs, or newline characters.
<b>filename</b>	A type of string requiring a DOS filename.
<b>enum</b>	A type of string requiring one of a limited set of possibilities.
<b>int</b>	An integer.

**float** A floating point (decimal) number. For an angle, the number is in degrees.

An int or float may have limits which are checked. For example, create-atom takes an int that is restricted to the range (1..103) and creates an atom with this atomic number at the origin of the molecular system. Floats may have similar limits. A string may not always need to be enclosed in quotes (“string”) but it is safer to do so. Appropriate values for an enum depend on the context, of course.

## Kinds of HSV's

A number of different types of HSV's are available. They are classified by how they are assigned values. The simplest are just scalars.

### Scalar HSV's

A scalar HSV is one which does not use an index. A simple example is *max-iterations*, the maximum number of allowed iterations in a self-consistent-field (SCF) calculation. The argument is an int in the range (1..32767) and an assignment (write) looks as follows:

```
max-iterations = 100
```

The equal sign is not strictly necessary and white space will do:

```
max-iterations 100
```

The number of arguments for a scalar HSV, while normally one, is not restricted to one. For example the HSV, *dipole-moment-components*, takes three float arguments - the x, y, and z components. It is defined as a scalar since you do not use an array index with it but assign or query all its components simultaneously. It is written or assigned, with a flexible syntax, as follows:

```
dipole-moment-components 1.0 2.0 3.0
```

```
dipole-moment-components = 1.0 2.0 3.0
```

```
dipole-moment-components = 1.0, 2.0, 3.0
```

That is, the arguments may begin with an equal sign or not, be separated by commas or not, etc. The syntax is flexible but equal signs and commas are suggested as an appropriate convention.



## Vector HSV's

A vector HSV is one which takes a single index. An example is the HSV, *alpha-orbital-occupancy*. This variable describes the number of electrons in an alpha (spin up) molecular orbital from an unrestricted Hartree-Fock (UHF) calculation. When the calculation is a restricted Hartree-Fock (RHF) calculation having no beta occupied orbitals different from alpha occupied orbitals then the *alpha-orbital-occupancy* variable describes the total occupancy of alpha and beta electrons in an orbital. Thus the arguments are 0 and 1 for UHF calculations and 0 and 2 for RHF calculations. Thus, for H<sub>2</sub> you can externally set the first (HOMO) orbital to be occupied and the second (LUMO) orbital to also be occupied (corresponding to H<sub>2</sub><sup>2+</sup>) by the following script commands:

```
alpha-orbital-occupancy(1) 2
```

```
alpha-orbital-occupancy(2) 2
```

or, equivalently,

```
alpha-orbital-occupancy(1) = 2
```

```
alpha-orbital-occupancy(2) = 2
```

For a standard minimal basis calculation on H<sub>2</sub>, the query,

```
alpha-orbital-occupancy ?
```

would return the normal result

```
alpha-orbital-occupancy(1) = 2
```

```
alpha-orbital-occupancy(2) = 0
```

That is, even though a query such as

```
alpha-orbital-occupancy(1) ?
```

is perfectly valid, it is possible to query for all relevant indices (1..2) at once, as we have done above. It is, of course, also possible to use the alternative syntax,

```
query-value alpha-orbital-occupancy(1)
```

## Array HSV's

An array HSV is defined to be one which takes two indices. These two indices are *always* the atom index and the molecule index (iat, imol). If one thinks of this combination as a single unique index, then an array is just a vector where the index is a unique atom number. A simple example is the mass of an atom which is represented by the HSV, *atom-mass*. The atomic masses of H<sub>2</sub> would be assigned as follows,

$$\text{atom-mass}(1,1) = 1.008$$

$$\text{atom-mass}(2,1) = 1.008$$

That is, the first and second atoms of molecule one are assigned. Another example is the Cartesian coordinates of the atoms. This is an array with 3 arguments analogous to the dipole moment components above. The coordinates of H<sub>2</sub>, again, could be assigned as follows,

$$\text{coordinates}(1,1) = 0.0, 0.0, 0.0$$

$$\text{coordinates}(2,1) = 0.0, 0.0, 0.74$$

An alternative to this is to assign all coordinates at once,

$$\text{coordinates} = 0\ 0\ 0\ 0\ 0\ 0.74$$

Note again that there is flexibility in using commas or equal signs or not using them.

## A Finite State Machine View of HyperChem

HyperChem has the characteristics of a Finite State Machine (FSM). By this it is meant that HyperChem has a finite set of internal states; an input received while in a particular state causes a transition to a new state with the consequent emission of an output. The set of states and the set of inputs/outputs associated with the set of possible state transitions fully characterizes an FSM.

In the HyperChem case, the inputs are messages and the outputs are messages. these messages are of four kinds (two input and two output):

- User message (UMSG) - an input message coming from the user using the mouse or keyboard.

- Visual message (VMSG) - an output message representing a visual change in the screen.
- Input message (IMSG) - an input message coming from a script or from a Dynamic Data Exchange (DDE) message.
- Output message (OMSG) - an output message going to a script directed target or to an external receiver of DDE messages.

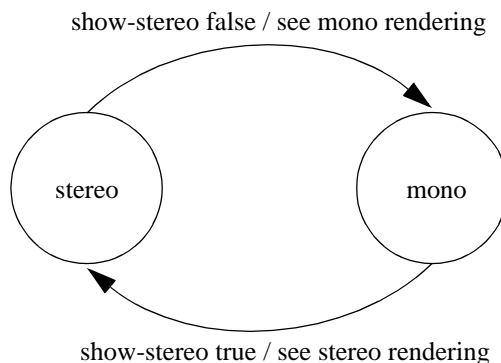
The normal interactive use of HyperChem has UMSGs as input and VMSGs as output. That is, when you click the mouse, something changes on the screen. It is possible to drive HyperChem with IMSGs rather than a mouse or keyboard. With an IMSG for input, one obtains possibly both a VMSG and an OMSG for output. That is, an IMSG, in addition to causing a visual change on the screen may result in an OMSG being sent to a receiver associated with the sender of the IMSG.

IMSGs are meant to be a superset of UMSGs. It is intended that anything you can do with the mouse you can do with an IMSG. In addition, IMSGs will trigger actions and state changes that are impossible by direct interaction. The target for OMSGs depends on the source of the IMSG and on the redirection of OMSGs by previous IMSGs. These generic concepts have two current implementations. One is implemented by the Hcl scripting language and the other is implemented by dynamic data exchange (DDE).

As an example of the state machine aspects of HyperChem, consider two states of HyperChem shown below as circles. The state is whether a stick rendering is shown on the screen as a normal image or as a stereographic image. Equivalently, the two states are described by an internal HyperChem State Variable (HSV) called *show-stereo*. when this variable is false, a normal (mono) image is displayed but when this variable is true, a double (stereo) image is displayed. The transitions between the two states are represented as arrows. the transition is triggered by an <input> and results in an <output> and each transition (arrow) is labelled:

transition: <input>/<output>

The input in this case is an IMSG of type "show-stereo", the name of the HSV, with an argument (part of the data of the message) of either "true" or "false". The output message is simply a VMSG that changes the screen display. No OMSG is emitted in this case.



## An HSV Server View of HyperChem

The HyperChem State Machine acts as a universal HSV Server in a Client-Server architecture. This is consistent with HyperChem attempting to be a core component of any chemical computation. Other programs, scripts, and third-party processes are clients that HyperChem serves. Consider a third-party application such as one that you would potentially write. If you want the molecular coordinates to perform a calculation with, you must make a request to HyperChem for the coordinates rather than expect HyperChem to send them on its own. HyperChem is sitting in a loop servicing requests like this as they come in. When it receives the request from you for coordinates, it sends them and then looks for another request. Some of these requests come from a user interacting with a mouse and some come from you and your program requesting further data, requesting the ability to access certain HyperChem functionality, or requesting a change in HyperChem's internal state via the setting of a HyperChem State Variable (HSV). These server requests are what we have called UMSGs or IMSGs above. A UMSG comes from an interacting user clicking on a mouse or the keyboard while an IMSG is an internal request in a script or an external request coming in a DDE message.

Further information on HSV's is contained in Chapter 6, Appendix A, or the HyperChem Reference Manual.



# Chapter 5

## Custom Menus

### Introduction

This chapter describes in detail the customizing of HyperChem via your ability to add menu items to the <Script> menu plus your ability to redefine completely the menu structure with a menu (\* .mnu) file.

### Script Menu Items

There has been the ability in earlier releases of HyperChem to add menu items to the <Script> menu. This capability is still there, with two slightly different ways of invoking it (using the “third” menu item as an example):

(1) `change-user-menuitem 3, "Sample", "sample.scr"`

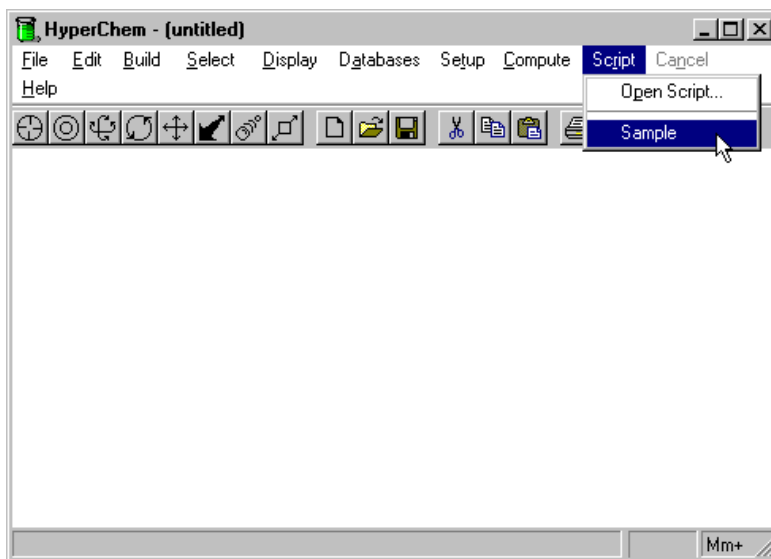
`script-menu-enabled(3) = true`

(2) `script-menu-caption(3) = "Sample"`

`script-menu-command(3) = "read-script sample.scr"`

`script-menu-enabled(3) = true`

Executing the above script, `Example1.scr` from the HyperChem CD-ROM, illustrates this behavior. For simplicity, the `sample.scr` file contains only the single Hcl script command, `window-color=green`, which gives a quick visual feedback of the effect of the new menu item. The new menu item is shown below:



Up to ten new menu items, with vector indices 1..10, are available under the <Script> menu. Each new menu item shows up in the order associated with the vector index which was assigned to it, such as the “3” above. Menu items that have not been assigned a caption do not show up.

## Menu Files

What is new in Release 5.0 of HyperChem is that, in addition to adding new menu items in the <Script> menu, all the menu items can now be designed from scratch and instead of just executing a Hcl script any menu item can now execute a much more generic Tcl/Tk script.

This new functionality comes about via the three new Hcl script commands,

```
load-user-menu <menu file (*.mnu)>
```

```
load-default-menu
```

```
switch-to-user-menu
```

The first of these loads and switches HyperChem to a set of custom menus defined in a menu file. The remaining two switch back and forth between the default *hard-wired* menus and the user *custom* menus, provided they have been defined earlier by the first script command of the above three.

A menu file need not have the \*.mnu extension since it simply a text file but it is convenient for menu files to be recognized with their own extension. These menu files define the left-to-right and top-to-bottom structure of a complete set of menus via sequential text lines of the form,

```
MENU <text-string>
    ITEM <text-string>, <Hcl-script-command>
    ITEM ...
END
```

The text string is the text that you see on the menu or menu item, including the naming of a possible keyboard assist via the ampersand, "&", being placed prior to the character that is to be used with the Alt key, as per the Microsoft Windows standard. If a menu has no subset of items but is its own menuitem and can be activated by clicking on it alone, it is defined in the form,

```
MENUITEM <text-string>, <Hcl-script-command>
```

Cascading menus are not supported. The menu file that corresponds to the default set of hard-wired menus is on the HyperChem CD-ROM as default.mnu. It looks as follows:

```
;
; 1996 (c) Hypercube, Inc.
; User Customizable Menu
;
MENU "&File"
    ITEM "&New\tCtrl+N",menu-file-new
    ITEM "&Open...\tCtrl+O",menu-file-open
    ITEM "&Merge...",menu-file-merge
    ITEM "&Save\tCtrl+S",menu-file-save
    ITEM "Save &As...\tCtrl+A",menu-file-save-as
    SEPARATOR
    ITEM "S&tart Log...",menu-file-start-log
    ITEM "Stop Lo&g",menu-file-stop-log
    ITEM "&Log Comments...",menu-file-log-comments
    SEPARATOR
    ITEM "&Import...",menu-file-import
    ITEM "&Export...", menu-file-export
    SEPARATOR
    ITEM "&Print...\tCtrl+P", menu-file-print
    SEPARATOR
    ITEM "Pr&eferences...",menu-file-preferences
```



## Menu Files

```
SEPARATOR
ITEM "E&xit",menu-file-exit
END
MENU"&Edit"
ITEM "&Clear\tDelete",menu-edit-clear
ITEM "C&ut\tCtrl+X",menu-edit-cut
ITEM "C&opy\tCtrl+C",menu-edit-copy
ITEM "Copy ISIS S&ketch",menu-edit-copy-isis-sketch
ITEM "&Paste\tCtrl+V",menu-edit-paste
SEPARATOR
ITEM "Cop&y Image\tF9",menu-edit-copy-image
SEPARATOR
ITEM "&Invert",menu-edit-invert
ITEM "&Reflect",menu-edit-reflect
SEPARATOR
ITEM "Rotat&e...",menu-edit-rotate
ITEM "&Translate...",menu-edit-translate
ITEM "&Zoom...",menu-edit-zoom
ITEM "Z C&lip...",menu-edit-z-clip
SEPARATOR
ITEM "&Align Viewer...",menu-edit-align-viewer
ITEM "Align &Molecules...",menu-edit-align-molecules
SEPARATOR
ITEM "Set &Bond Length...",menu-edit-set-bond-length
ITEM "Set Bon&d Angle...",menu-edit-set-bond-angle
ITEM "Set Bo&nd Torsion...",menu-edit-set-bond-torsion
END
MENU"&Build"
ITEM "&Explicit Hydrogens",menu-build-explicit-hydrogens
ITEM "&Default Element...",menu-build-default-element
ITEM "&Add Hydrogens",menu-build-add-hydrogens
ITEM "&Model Build",menu-build-model-build
SEPARATOR
ITEM "Allow &Ions",menu-build-allow-ions
ITEM "&United Atoms",menu-build-united-atoms
ITEM "A&ll Atoms",menu-build-all-atoms
SEPARATOR
ITEM "Calculate T&ypes",menu-build-calculate-types
ITEM "Compile Type &Rules",menu-build-compile-type-rules
SEPARATOR
ITEM "&Set Atom Type...",menu-build-set-atom-type
ITEM "Set &Mass...",menu-build-set-mass
ITEM "Set C&harge...",menu-build-set-charge
```

```

ITEM "&Constrain Geometry...",menu-build-constrain-geometry
ITEM "Constrain &Bond Length...",menu-build-constrain-bond-length
ITEM "Constrain Bond An&gle...",menu-build-constrain-bond-angle
ITEM "Constrain Bond &Torsion...",menu-build-constrain-bond-
torsion
END
MENU"&Select"
ITEM "&Atoms",menu-select-atoms
ITEM "&Residues",menu-select-residues
ITEM "&Molecules",menu-select-molecules
SEPARATOR
ITEM "M&ultiple Selections",menu-select-multiple-selections
ITEM "&Select Sphere", menu-select-select-sphere
SEPARATOR
ITEM "S&elect All",menu-select-select-all
ITEM "&Complement Selection",menu-select-complement-selection
ITEM "Se&lect...",menu-select-select
ITEM "&Name Selection...",menu-select-name-selection
SEPARATOR
ITEM "E&xtend Ring",menu-select-extend-ring
ITEM "Ex&tend Side Chain",menu-select-extend-side-chain
ITEM "Exten&d to sp3",menu-select-extend-to-sp3
ITEM "Select Bac&kbone",menu-select-select-backbone
END
MENU"&Display"
ITEM "&Scale to Fit\tSpace",menu-display-scale-to-fit
ITEM "&Overlay",menu-display-overlay
SEPARATOR
ITEM "Show &All",menu-display-show-all
ITEM "Sho&w Selection Only",menu-display-show-selection-only
ITEM "&Hide Selection",menu-display-hide-selection
SEPARATOR
ITEM "&Rendering..."menu-display-rendering
ITEM "Last Renderin&g\tF2"menu-display-last-rendering
SEPARATOR
ITEM "Show &Isosurface\tF3"menu-display-isosurface
ITEM "Isosur&face...\tF4"menu-isosurface-control
SEPARATOR
ITEM "Show H&ydrogens",menu-display-show-hydrogens
ITEM "Show Periodic &Box",menu-display-show-periodic-box
ITEM "Show &Multiple Bonds",menu-display-show-multiple-bonds
ITEM "Show Hy&drogen Bonds",menu-display-show-hydrogen-bonds
ITEM "Recomp&ute H Bonds",menu-display-recompute-h-bonds

```

```

ITEM "Show Inertial A&xes",menu-display-show-inertial-axes
ITEM "Show Dipole Momen&t",menu-display-show-dipole-moment
SEPARATOR
ITEM "&Labels...",menu-display-labels
ITEM "&Color...",menu-display-color
ITEM "&Element Color...",menu-display-element-color
END
MENU "D&atabases"
ITEM "&Amino Acids...",menu-databases-amino-acids
ITEM "&Make Zwitterion",menu-databases-make-zwitterion
ITEM "&Remove Ionic Ends",menu-databases-remove-ionic-ends
SEPARATOR
ITEM "&Nucleic Acids...",menu-databases-nucleic-acids
ITEM "Add &Counter Ions",menu-databases-add-counter-ions
SEPARATOR
ITEM "M&utate...",menu-databases-mutate
END
MENU "Se&tup"
ITEM "&Molecular Mechanics...",menu-setup-molecular-mechanics
ITEM "&Semi-empirical...",menu-setup-semi-empirical
ITEM "&Ab Initio...",menu-setup-ab-initio
SEPARATOR
ITEM "&Periodic Box...",menu-setup-periodic-box
ITEM "&Restrains...",menu-setup-restraints
ITEM "Set &Velocity...",menu-setup-set-velocity
SEPARATOR
ITEM "S&elect Parameter Set...",menu-setup-select-parameter-set
ITEM "C&ompile Parameter File",menu-setup-compile-parameter-file
SEPARATOR
ITEM "&Reaction Map...", menu-setup-reaction-map
END
MENU "&Compute"
ITEM "&Single Point",menu-compute-single-point
ITEM "&Geometry Optimization...",menu-compute-geometry-
optimization
SEPARATOR
ITEM "&Molecular Dynamics...",menu-compute-molecular-dynamics
ITEM "&Langevin Dynamics...",menu-compute-langevin-dynamics
ITEM "Monte &Carlo...",menu-compute-monte-carlo
SEPARATOR
ITEM "&Vibrations",menu-compute-vibrations
ITEM "&Transition State...",menu-compute-transition-state
SEPARATOR

```

```

ITEM "&Plot Molecular Properties...", menu-compute-plot-molecular-
properties
ITEM "&Orbitals...", menu-compute-orbitals
SEPARATOR
ITEM "Vibrational Spectrum...", menu-compute-vibrational-spectrum
ITEM "&Electronic Spectrum...", menu-compute-electronic-spectrum
END
MENU "Script"
ITEM "Open Script...", menu-script-open-script
ITEM "&Compile Script...", menu-script-compile-script
END
MENUITEM "Cancel", menu-cancel
MENU "&Help"
ITEM "&Index", menu-help-index
ITEM "&Keyboard", menu-help-keyboard
ITEM "&Commands", menu-help-commands
ITEM "&Tools", menu-help-tools
ITEM "&Scripts \& DDE", menu-help-scripts-&-DDE
ITEM "&Glossary", menu-help-glossary
ITEM "&Using Help", menu-help-using-help
SEPARATOR
ITEM "&About HyperChem", menu-help-about-hyperchem
END

```

An extension of this menu file, called `chemplus.mnu`, is on the HyperChem CD-ROM to enable users of Release 5 to continue using ChemPlus from menu items. [Although ChemPlus 1.5 continues to work with HyperChem 5.0, the ChemPlus modules cannot be activated by menu items except through a menu file such as described here].

## Simple Example

The custom menus can be illustrated by executing the script, `Example2.scr` from the HyperChem CD-ROM:

```

script-menu-caption(1) = "Rotate Menus"
script-menu-command(1) = "load-user-menu rotate.mnu"
script-menu-enabled(1) = true

```

This script puts a new menu item in the <Script> menu. The new menu button has the text, "Rotate Menus" on it. Pushing this menu button will execute a script command, as shown above, that loads a custom set of user menus (`rotate.mnu`) rather than the hard-wired default menus that come with the

product. A set of custom menus is characterized by a menu file (\*.mnu), which in this case is the set of menus defined in `rotate.mnu` (from the HyperChem CD-ROM):

```
MENU "&File"
    ITEM "&New\tCtrl+N",menu-file-new
    ITEM "&Open...\tCtrl+O",menu-file-open
    ITEM "E&xit",load-default-menu
END
MENU"&Rotate"
    ITEM "&X Axis",read-tcl-script rotatex.tcl
    ITEM "&Y Axis",read-tcl-script rotatey.tcl
    ITEM "&Z Axis",read-tcl-script rotatez.tcl
END
MENUITEM "Ca&ncel",menu-cancel
```

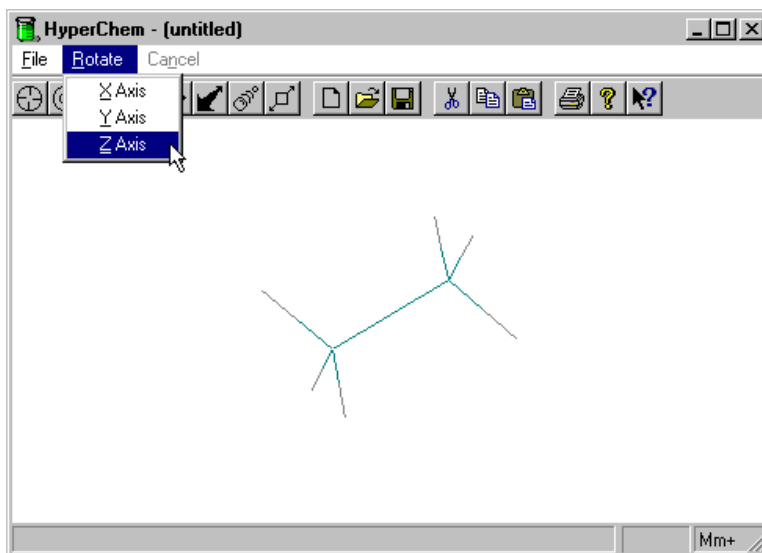
This menu file is characteristic of all menu files in that it simply defines the individual menus and menu items from left to right and top to bottom. The key words MENU, MENUITEM and ITEM introduce a normal menu, a menu without any underlying items, and an individual menu item within a menu. Each menu is defined by the text string label for that menu while each menu item is defined by the text string plus a Hcl script command that is executed when that menu item is chosen. Each text string can be assigned a keyboard assist with the ampersand, "&".

The way to have this file define menus that are identical to the default hard-wired menus is to define a menu item with a script command that is a *menu invocation* such as,

```
menu-file-new
```

A menu item can be set to execute any Hcl script command. By using the script commands, *read-script* and *read-tcl-script*, any generic Hcl script or Tcl/Tk script stored in a file can be triggered by each custom menu item. In the above case, a Tcl script file is attached to each member of a set of three menu items for continuously rotating the molecular system about the x, y, or z axis. The menu item <File/Exit> has been set to execute a script command that returns to the hard-wired default menus. In this way it is simple to go back and forth between the two sets of menus by just clicking on menu items, i.e. by clicking on <Script/Rotate Menus> to obtain the special menus for rotation only and by clicking on <File/Exit> to return to the default menus. Executing

the original Example2.scr script and then clicking on <Script/Rotate Menus> gives a version of HyperChem that is shown below.



The rotatez.tcl script, which is included on the HyperChem CD-ROM, rotates the molecule about the z axis and is as follows:

```
TclOnly
hcExec "query-response-has-tag false"
hcExec "cancel-menu = true"
for {set i 0} {$i < 36500} {incr i} {
  hcExec "rotate-molecules x 5"
  set we_quit [hcQuery cancel-menu]
  if { $we_quit == "false" } { break }
  update
}
```

This is not the place to describe Tcl/Tk scripts in detail but this script is fairly simple and illustrates many of the general ideas of how universal is the ability to customize HyperChem. The first Tcl command, TclOnly [note that Tcl/Tk is case sensitive] indicates that this script needs no graphical user interface

GUI) or extra windows. By default, without this command, a simple window is put up where any Tk widgets are placed. Since we do not want this window, we indicate that this Tcl/Tk script is a Tcl only script without any extra visual elements.

The HyperChem command language, Hcl, and the HyperChem GUI do not have any capability for continuously rotating a molecule. The GUI allows for manually rotating a molecule and the script command,

```
rotate-molecules x <angle>
```

rotates the molecular system about, in this case, the x axis by <angle> degrees as a one-shot, non-continuous motion. To see continuous motion, the <angle> must be small and the script command repeated a large number of sequential times even for a “single” rotation. The way to obtain continuous rotation, of course, is to have a do, while or for-loop of these individual rotation commands. The Tcl code above has a simple “for-loop over i” of 36500 iterations. It remains only to describe the Hcl script commands that are embedded inside Tcl and the handling of the Cancel menu.

The Tcl language embeds all possible Hcl commands into the language through the syntax,

```
hcExec <Hcl script command> and
hcQuery <name of HSV>
```

The basic idea here is to place the Hcl script command, *rotate-molecules* (about the z axis by 5 degrees), inside the for-loop as is done in the above script. The remaining complexity of the Tcl script is all associated with the handling of the Cancel button. Without the Cancel button, the rotation would go on forever, or at least for 7200 complete rotations. The trick is to sense whether the Cancel button has been pushed while executing the loop and break out of the loop if this is so. As with all queries of HSV’s, there is the possibility of the answer coming back with a tag, such as “cancel-menu=true” or as just the raw value, “true”. We first ensure that only raw values are returned so as to simplify the parsing required. We thus execute the Hcl script command,

```
query-response-has-tag = false
```

Next, we disable (gray-out) the usual menus and enable (blacken) the Cancel menu via the following script command, just prior to beginning the rotation,

```
cancel-menu = true
```

This (Cancel-enabled, Normal-disabled) state of HyperChem can be monitored by enquiring about the HSV, *cancel-menu*. If it is true, then the Cancel button is enabled and it can be clicked upon at any time. If you do click on

this Cancel button, it becomes disabled and the value of the HSV becomes false. The Tcl script looks for this cancel operation each iteration and breaks out of the loop if it occurs. The only other Tcl command that needs comment is the update command. Without this, Tcl would not temporarily release control to Windows and sense the change made to the cancel-menu HSV in HyperChem. It would be busy just executing “its own for-loop code”. When you want an immediate update of a Tk dialog box or a HyperChem HSV, it is best to place an appropriate update command inside a Tcl script.

## Further Customization

The name of the HyperChem window can be modified to identify specific custom versions of HyperChem. This is very straightforward using the following Hcl script command.

```
custom-title <addendum to HyperChem name>
```

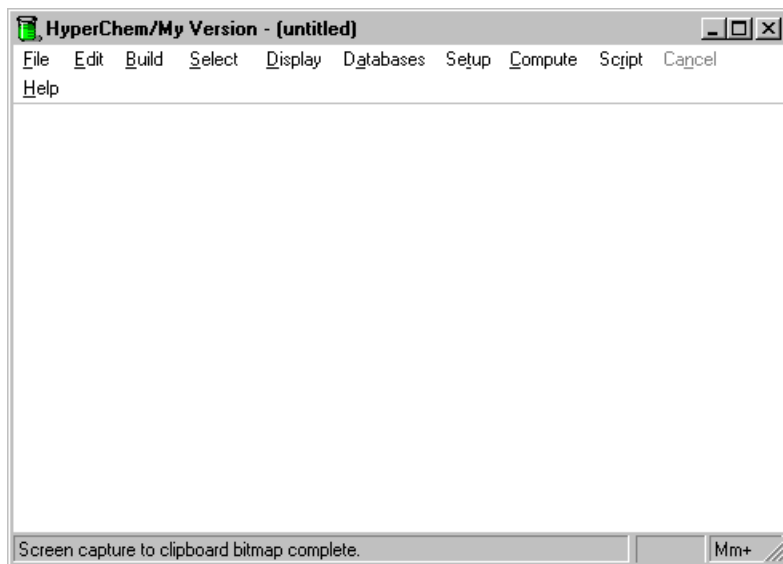
In addition, if so desired, the tool bar can be eliminated using the *hide-toolbar* HSV. Thus, the Hcl script, stored on the HyperChem CD-ROM as `Example3.scr`,

```
custom-title = "My Version"
```

```
hide-toolbar = true
```

results in the following window, having a custom name and no toolbar. It is still possible without a toolbar to change the HyperChem cursor tool using the *mouse-mode* HSV.





# Chapter 6

## Type 1 (Hcl) Scripts

### Introduction

Type 1 scripts, the simplest scripts, are simple sequences of script commands each of which conform to the *HyperChem Command Language* (Hcl - pronounced “hickle”). These scripts have been part of HyperChem since its inception but with the addition, in Release 5.0, of Tcl/Tk scripting capability, it is necessary to clarify our terminology somewhat. Thus, what was previously referred to simply as a script command or script message now becomes a Hcl command or Hcl script command. These Hcl script commands can constitute the totality of a script, be imbedded in a Tcl script, or be the content of certain external messages sent to HyperChem by other programs. A script that consists solely of Hcl commands is a Type 1 or Hcl script.

### Hcl Script Commands

A Hcl script consists of a sequence of Hcl script commands, each of which is a single line of Ascii text. These Hcl script commands are of four types:

- An HSV read
- An HSV write
- A menu activation
- A direct command

### HSV's

The HyperChem State Variables (HSV's) have been described in Chapter 4. HyperChem makes these variables available to external programs and to internal scripts, for reading and possibly for writing. They need not be described again here other than for completeness in describing Hcl script commands. Using *dipole-moment* as an example of an HSV, a Hcl script com-

mand can read the value (enquire about the current value that HyperChem maintains) by the Hcl script command,

```
query-value dipole-moment
```

or, alternatively, via the equivalent but slightly different syntax,

```
dipole-moment ?
```

You can write the value (assuming you wish to assign a value of 2.5 Debyes to the dipole moment) via the Hcl script command,

```
dipole-moment 2.5
```

## Menu Activations

A menu activation is a replacement for the user clicking with the mouse on a menu item (menu button). Every menu item of HyperChem has an equivalent Hcl script command that accomplishes exactly the same effect as clicking on the menu item. These Hcl script commands all start with “menu-” and then name the particular menu and finally a string representation of the particular menu item, all separated by hyphens. Thus, the equivalent of clicking on the menu item <File/Save As...> is to open a \*.scr file and execute the Hcl script command,

```
menu-file-save-as
```

All these menu activations are listed in the HyperChem Reference Manual but each can be inferred just by looking at the HyperChem menus and putting a hyphen between every word of the text of the menu item. Thus, if the *Setup* Menu contains the menu button *Select Parameter Set...*, the proper script command is,

```
menu-setup-select-parameter-set
```

One exception, of sorts, to this rule is the “Model Build...” menu item which sometimes reads, “Add H & Model Build...” The script command is always,

```
menu-build-model-build
```

## Direct Commands

To open (read in) a file you normally click with the mouse on the menu item <File/Open...> and bring up a dialog box to be filled out with appropriate dialog box values prior to hitting the OK button which initiates the reading in of the file. Suppose you wanted to read in a Protein Data Bank (PDB) file. Then, prior to hitting OK, you need to set the default file type to be \*.ENT in the dialog box in addition to choosing the name of the appropriate file that you

want to read in. You might like a script to automate these actions but if you used the script command

```
menu-file-open
```

the script would stop with the File Open dialog box sitting on the screen waiting for you to hit the OK button to initiate the reading of the file. This is anything but automation if you have to be there to click on OK! Scripts must go beyond imitating the actions of a user at a keyboard or with mouse in hand.

To solve this problem, a script command is needed that initiates the reading of a file using the current values of the dialog box settings without bringing up the dialog box at all. If different settings are needed than are in the prospective dialog box, these values can be set prior to calling for the opening of the file. The script command *open-file filename* does the job. It is a *direct command* that bypasses a dialog box to get the job done using the current dialog box settings. If you wanted to read, for example, a glucagon PDB file you could simply execute the script,

```
file-format pdb
```

```
open-file glucagon.ent
```

This is completely equivalent to invoking the <File/Open...> dialog box, selecting PDB as the file format, filling in the File Name as glucagon.pdb, and hitting OK to dismiss the dialog box and initiate the reading of the file.

Other direct script commands cause actions that have no immediate mapping to a GUI action. The direct script commands consist of the name of the command followed by one or more arguments, each separated by at least one space or a comma:

```
hcl-command-name <argument1>, <argument2>, ...
```

An argument is one of:

## Arguments

The types of arguments for variables or commands are the following:

<b>Boolean</b>	Yes or no, true or false, 0 or 1.
<b>string</b>	Text (letters, characters, or symbols, in upper- or lower-case, unlimited number of characters). Enclose a string in quotes (“ ”) if it contains spaces, tabs, or newline characters.
<b>filename</b>	A type of string requiring a DOS filename.

<b>enum</b>	A type of string requiring one of a limited set of possibilities.
<b>int</b>	An integer.
<b>float</b>	A floating point (decimal) number. For an angle, the number is in degrees.

Another example of a direct command is:

```
do-molecular-dynamics
```

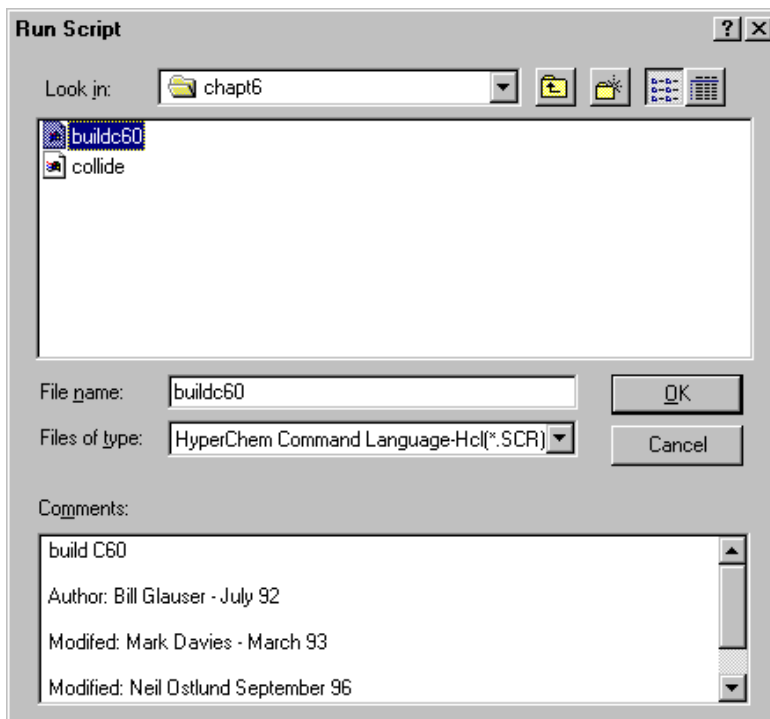
This allows a script to perform molecular dynamics calculations without the manual intervention of a dialog box. Prior to executing this direct command, the script could set all the relevant HSV's for molecular dynamics. This is equivalent to what one is really doing in filling out the molecular dynamics dialog box. For example, the length of the MD trajectory, the dynamics run time, could be set to 100ps via the script command,

```
dynamics-run-time 100
```

Alternatively, one can just accept default values for the relevant HSV's which are just the current values in the dialog box left over from the last time it was invoked. As with almost any situation in HyperChem, reasonable default values are available and are used when you choose not to specify further details.

## Script Files

Type 1 scripts are normally stored in files with a default file extension of \*.SCR. These files consist of nothing more than a sequence of individual Hcl script commands, stored in an Ascii text file with a file name that has the SCR file extension. A script is normally executed by opening a \*.SCR script file with the <Script/Open...> menu item.



A Hcl script file is simply a text file and while it is not essential to use the \*.scr extension, that is the useful default extension used by HyperChem.

A trivial example of a \*.scr file is `version.scr`, one that contains the single line,

```
version ?
```

This script file, when opened, queries for the Release Number or Version of HyperChem, in case you are not sure what it is. That is, it requests the value of the HSV, "version". The result will be reported to you in a message box on the screen.

## CHEM.SCR

When HyperChem is started, one of things it does before handing over control to you as a user is to execute a default initialization Hcl script. If HyperChem

finds a script, CHEM.SCR, in its path then it executes that script as a final part of its initialization. If this file does not exist or HyperChem cannot find it in its path, then no such specialized initialization is performed. This allows you to set up a script and place it in CHEM.SCR to customize your copy of HyperChem right from its instantiation, without ever having to explicitly execute a script. If you have your own customization of HyperChem that you like to use on a semi-permanent basis, you should place the relevant script into CHEM.SCR.

## Compiled Scripts

A script can be compiled, if desired. The compilation results in a `*.ocr` file. Thus, the following script command compiles `xxx.scr` into `xxx.ocr`,

```
compile-script-file xxx.scr xxx.ocr
```

Compiled scripts can be opened by HyperChem just as text scripts (see the file filter in the <Script/Open...> dialog box).

## Recursive Scripts

Scripts files can contain script commands to open other script files. Thus, the script file `a.scr` could contain a script command,

```
read-script b.scr
```

When this second script, `b.scr`, completes, control is returned to the script command in `a.scr` following the above call to `b.scr`.

## Script Editor

The companion product to HyperChem, ChemPlus, includes a script editor that makes it easy to create and execute individual Hcl script commands or whole or partial Hcl script files.

## Examples

This section begins a description of a number of example Hcl scripts. There are a large number of HSV's and direct script commands contained in HyperChem and learning them all is not a simple matter. One of the best ways is to study example scripts and ultimately to write a number your own scripts.

The examples here will give you some idea of the power of the HyperChem Command Language but its real power ultimately comes in conjunction with the control structures that Tcl can add or the GUI that Tk can add. Hcl scripts are rich in chemistry but it is sometimes difficult to get what you want done easily without the true programming power of variables, do-loops, if-statements, etc.

In addition to this section a good place to start learning Hcl scripts is the `test.scr` script that is explained and described in Chapter 10 of the Reference manual.

## Reactive Collision of Two Molecules

This example illustrates a potential process for studying chemical reactions. HyperChem's molecular dynamics (MD) calculations use forces computed by any of the basic computation methods - molecular mechanics, semi-empirical or ab initio quantum mechanics. When the forces are computed by a quantum mechanical method, bond breaking is quite straight-forward and an MD trajectory can describe a reactive or non-reactive collision of two molecular systems. The actual calculation of a rate constant for the chemical reaction is much more complicated and involves performing the collision over and over again with a Boltzman weighted set of initial conditions and averaging over the results. A Tcl script could do this but here we just illustrate how to set up a simple script so that any two molecules on the screen could be set to collide with each other and you can see what happens. To reiterate, this will show you only one of the many possible outcomes of collisions between the two molecules. It nevertheless can be very informative.

This script assumes there are two molecules and two molecules only on the screen. A collision is going to be initiated between a moving molecule 1 and a stationary molecule 2. Molecule 1's center of mass is going to be set to move in the direction of the center of mass of molecule 2.

### Assign Target Position

The first thing we are going to do is to make the named selection POINT correspond to molecule 2. POINT is a pre-named selection such that POINT can be referred to by other parts of HyperChem as the center-of-mass of the selection. That is POINT can be both a name for a particular selection as well as the center-of-mass of that selection. We select the atoms of molecule 2 and assign them to POINT. We can select all the atoms of one molecule by having the selection unit be molecules (menu item <Select/Molecules>) and selecting any atom of that molecule.



```
select-none                ; start clean
selection-target molecules ; want to select all of molecule 2
select-atom 1,2           ; select atom 1 of molecule 2 => all atoms
name-selection POINT      ; name the selection POINT
set-velocity POINT 0      ; make sure molecule 2 is stationary
select-none                ; de-select
```

## Assign Collision Velocities

The next step is to assign the velocities of the atoms of molecule 1 to be such that the center of mass of molecule 1 is aimed at the center of mass of molecule 2. We do this by selecting all of the atoms of molecule 1 and give them a velocity in the direction of POINT. The scalar value of the assigned velocity here is 200 Angstroms/picosecond. This is rather a high velocity of collision. The center of mass of this second selection, directed at POINT, is the direction of each assigned velocity.

```
select-atom 1,1           ; select all atoms of molecule 1
set-velocity POINT 200    ; veloc=200 in direction of POINT
select-none                ; done
```

## Wave Function Computation Parameters

Prior to initiating the collision we must set up the parameters for the resulting quantum mechanical computations. In this case we use the CNDO semi-empirical method but it might be any semi-empirical or ab initio method. It is appropriate to try to accelerate the convergence and we do not want to perform a configuration interaction calculation since in that case no forces would be calculated. We will be looking at a classical MD trajectory for the ground state potential energy surface for these two molecules. Configuration Interaction in HyperChem is for exploring excited states. A convergence of 0.01 is usually sufficient and it should definitely converge in less than 100 iterations if it converges at all.

The final setting below is to request a UHF rather than an RHF calculation. This is sometimes a controversial subject but RHF calculations do not have the correct asymptotic behavior at long distances when bonds are being broken. UHF calculations sometimes have their own problems but allow a much more generic approach for arbitrary chemical reactions. The UHF calculations usually allow bond-breaking to lead to the correct open-shell behavior of intermediates and products, as compared to RHF calculations.

```

calculation-method=semiempirical      ; faster than ab initio
semi-empirical-method=cndo            ; why not
accelerate-scf-convergence=true        ; fast is good
configuration-interaction=noci         ; no gradients with ci
excited-state=false                   ; could also study lowest state
scf-convergence=0.01                  ; 0.1 to 0.0001 ?
max-iterations=100                     ; better converge better than this
uhf=true                                ; essential for bond breaking

```

## The Collision

The last step is to initiate the molecular dynamics trajectory correctly. Most of these parameters are not set below but have been set in the dialog box. The important values are the step size and the length of the trajectory. If gradients get very high a smaller step size may be necessary. A larger collision velocity may also require smaller steps. It is essential to set dynamics-restart to be true as this uses the velocities assigned above rather than attempts to equilibrate the velocities, with random numbers, according to the temperature.

```

dynamics-restart=true                  ; use velocities we have
do-molecular-dynamics                  ; let 'er rip

```

This example can be run any time there are two molecules on the screen. In some instances, it may be necessary to set the charge and multiplicity for charged or open-shell systems. Try it by placing two methane molecules on the screen and see if you can form  $\text{H}_2 + \text{C}_2\text{H}_6$  in a collision? The example can be expanded on in a great many ways. For example, one might like to see molecular orbitals change during the collision.

One of the potential problems in watching molecular dynamics of chemical reaction is that of the “standard model” of HyperChem which treats a molecule as a connected graph. Overlapping spheres is the best way of viewing chemical reactions because they display only the position of atoms and not bonds that may no longer exist. Unfortunately, HyperChem uses the molecular graph to speed up the overlapping spheres rendering; it avoids working out the intersection of spheres when there is no “bond”. Thus one may see rendering artifacts sometimes when two atoms are near each other in a “product” when the reactant molecular graph “says” that they are far apart. An interesting script that you might wish to explore would be one that dynamically recomputes the graph to reflect the changing bonds of the reaction. Because of this issue, sometimes one “bonds everything” so that there are no artifacts in a spheres rendering; quantum mechanical calculations, of course, pay no attention to “bonds” but only care about the position of

atoms. Bonds are something to be derived from the results of the such quantum mechanical calculations.

## Building and Optimizing C<sub>60</sub>

The molecule C<sub>60</sub> continues to be of great interest to chemists. The power of HyperChem's model builder is illustrated by its capability for building the correct 3D structure of the molecule from a simple drawing of its 2D molecular graph. This can be accomplished without any need for a script if you simply draw the molecular graph using a mouse. This example, however, creates C<sub>60</sub> using a script as an illustration of the general scripting capability and of the process that one uses in a script to create arbitrary molecular structures. In combination, for example, with Tcl and a Tk GUI it would certainly be possible to add to HyperChem, yourself, the template building capabilities that are the only building capabilities that many other molecular modeling programs contain. That is, the HyperChem model builder has a richness that makes possible many other molecule creation procedures.

## Setup

The first part of this and any other script should set up the appropriate HSV environment for the script. Here, this includes options for the model builder, for the rendering and for the optimization that comes later. It is particularly important to note that one should draw with explicit hydrogens (hydrogens are not added unless they are explicitly drawn). This means that for intermediate structures used to build up to C<sub>60</sub>, the model builder will not add inappropriate intermediate hydrogens to the dangling valencies.

```
file-needs-saved no           ; no user intervention - new
menu-file-new                ; clean slate for new
render-method sticks         ; use sticks for drawing
calculation-method molecular-mechanics      ; for later optimization
molecular-mechanics-method mm+ ; universal method
selection-target atoms       ; select individual atoms
show-multiple-bonds yes     ; lets see aromatic bonds
allow-ions yes               ; valence of 4 for S
explicit-hydrogens yes      ; build won't add wrong hydrogens
multiple-selections yes     ; going to select group
```

## Drawing the First Pair of Atoms

To draw a molecule, one uses *create-atom* to place an atom, of a specified atomic number, down onto the workspace. This script command places all atoms at the origin and the fact that two atoms may have the same coordinates is not particularly important here as their final positions will be chosen by the model builder, which cares here only about the graph (what is bonded to what!) not the arbitrary initial coordinates. If the coordinates were to be important because, say, the model builder was not to be used, then any atom could be translated (*translate-selection*) to some arbitrary Cartesian coordinates.

Once, for example, the first two atoms are placed onto the workspace, a bond is placed between them with the script command, *set-bond*. The order of the arguments is atom and molecule of the first atom and then atom and molecule of the second atom. In this case, the two atoms are both numbered 1 but in molecules 1 and 2. The two atoms are in separate molecules, *until the bond is drawn*, because they are not part of the same connected molecular graph which the bonds define.

```
;build first bonded pair of atoms
create-atom 6                ; place C at origin
create-atom 6                ; place 2nd C at origin
set-bond 1 1 1 2 a          ; create aromatic bond between
menu-build-model-build      ; build 3D structure
```

## Finish First Level Pentagon

The following script code then creates the remaining three atoms of the first pentagon, building the structure (applying the model builder) as we go along. It is not really necessary to apply the model builder at every step. Finally the ring closing bond is applied and the atoms are all colored red. The top and bottom pentagons will be colored red just to distinguish them as we rotate the final structure. HyperChem generally uses a “select then operate” algorithm where you first select a subset of atoms of the molecular system, apply some operation to this subset, and then de-select or not for the next operation. The operation here is one of many possible ones that could be applied to a selection. Specifically it is the *color-selection* operation.

```
create-atom 6                ; create the remaining atoms
set-bond 2 1 1 2 a
```

## Examples

```
menu-build-model-build
create-atom 6
set-bond 3 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 4 1 1 2 a
menu-build-model-build
set-bond 5 1 1 1 a ; set ring-closure bond
menu-build-model-build
select-atom 1 1 ; select each of the five atoms
select-atom 2 1
select-atom 3 1
select-atom 4 1
select-atom 5 1
color-selection red ; color them red
select-none ; de-select everything
```

## Build Remaining Layers

The following code is repetitive and just adds the remaining 55 atoms according to the correct graph, building as it goes along.

```
create-atom 6 ; build second tier
set-bond 1 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 2 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 3 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 4 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 5 1 1 2 a
menu-build-model-build ; build third tier
create-atom 6
set-bond 6 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 6 1 1 2 a
menu-build-model-build
```

```
create-atom 6
set-bond 7 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 7 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 8 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 8 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 9 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 9 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 10 1 1 2 a
menu-build-model-build
create-atom 6
set-bond 10 1 1 2 a
menu-build-model-build
set-bond 11 1 19 1 a
set-bond 12 1 14 1 a
set-bond 13 1 16 1 a
set-bond 15 1 18 1 a
set-bond 17 1 20 1 a
create-atom 6 ; build fourth tier
set-bond 11 1 1 2 a
create-atom 6
set-bond 12 1 1 2 a
create-atom 6
set-bond 13 1 1 2 a
create-atom 6
set-bond 14 1 1 2 a
create-atom 6
set-bond 15 1 1 2 a
create-atom 6
set-bond 16 1 1 2 a
create-atom 6
set-bond 17 1 1 2 a
```

## Examples

```
create-atom 6
set-bond 18 1 1 2 a
create-atom 6
set-bond 19 1 1 2 a
create-atom 6
set-bond 20 1 1 2 a
set-bond 21 1 22 1 a           ; tier of five pentagons
set-bond 23 1 24 1 a
set-bond 25 1 26 1 a
set-bond 27 1 28 1 a
set-bond 29 1 30 1 a
menu-build-model-build
create-atom 6                 ; fifth tier
set-bond 21 1 1 2 a
create-atom 6
set-bond 22 1 1 2 a
create-atom 6
set-bond 23 1 1 2 a
create-atom 6
set-bond 24 1 1 2 a
create-atom 6
set-bond 25 1 1 2 a
create-atom 6
set-bond 26 1 1 2 a
create-atom 6
set-bond 27 1 1 2 a
create-atom 6
set-bond 28 1 1 2 a
create-atom 6
set-bond 29 1 1 2 a
create-atom 6
set-bond 30 1 1 2 a
menu-build-model-build
set-bond 32 1 34 1 a
set-bond 33 1 36 1 a
set-bond 35 1 38 1 a
set-bond 37 1 40 1 a
set-bond 31 1 39 1 a
create-atom 6
set-bond 31 1 1 2 a
create-atom 6
set-bond 32 1 1 2 a
create-atom 6
```

```
set-bond 33 1 1 2 a
create-atom 6
set-bond 34 1 1 2 a
create-atom 6
set-bond 35 1 1 2 a
create-atom 6
set-bond 36 1 1 2 a
create-atom 6
set-bond 37 1 1 2 a
create-atom 6
set-bond 38 1 1 2 a
create-atom 6
set-bond 39 1 1 2 a
create-atom 6
set-bond 40 1 1 2 a
menu-build-model-build
set-bond 41 1 42 1 a
set-bond 43 1 44 1 a
set-bond 45 1 46 1 a
set-bond 47 1 48 1 a
set-bond 49 1 50 1 a
create-atom 6
set-bond 41 1 1 2 a
set-bond 49 1 51 1 a
create-atom 6
set-bond 42 1 1 2 a
set-bond 44 1 52 1 a
create-atom 6
set-bond 43 1 1 2 a
set-bond 46 1 53 1 a
create-atom 6
set-bond 45 1 1 2 a
set-bond 48 1 54 1 a
create-atom 6
set-bond 47 1 1 2 a
set-bond 50 1 55 1 a
menu-build-model-build
create-atom 6
set-bond 51 1 1 2 a
create-atom 6
set-bond 52 1 1 2 a
create-atom 6
set-bond 53 1 1 2 a
```



## Examples

```
create-atom 6
set-bond 54 1 1 2 a
create-atom 6
set-bond 55 1 1 2 a
menu-build-model-build
set-bond 56 1 57 1 a
set-bond 57 1 58 1 a
set-bond 58 1 59 1 a
set-bond 59 1 60 1 a
set-bond 60 1 56 1 a
menu-build-model-build
```

### Color Bottom and Rotate

We now color the bottom pentagon, switch to a solid rendering and rotate the molecule to visualize the structure. Because Hcl scripts have no control structures we have to repeat the *rotate-molecules* script command to get the effect we want. A Tcl script would use a do-loop.

```
select-atom 56 1
select-atom 57 1
select-atom 58 1
select-atom 59 1
select-atom 60 1
color-selection red
select-none
render-method spheres ; easier to see 3D shape
rotate-molecules x 5 ; rotate around horizontal
rotate-molecules x 5 ; rotate by 5 degrees at time
rotate-molecules x 5 ; rotate until see whole
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
```

```
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
rotate-molecules x 5
```

## Zoom Structure

We now switch back to a sticks structure and zoom in on the molecule for effect. We see a perspective picture as we zoom.

```
render-method sticks
align-viewer z                ; view molecule along z axis
front-clip 0                  ; don't let it get clipped
show-perspective true         ; pretty perspective view
translate-view 0 0 1          ; translate 1 Angstrom in Z
translate-view 0 0 1          ; repeat according to taste
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
translate-view 0 0 1
```

```
translate-view 0 0 1
translate-view 0 0 -30           ; go back where we were
show-perspective false         ; turn off for other renderings
```

### Create an SO<sub>2</sub> Molecule Inside C<sub>60</sub>

Next we create an SO<sub>2</sub> molecule and place it at the center inside the C<sub>60</sub> structure. We switch back to a solid rendering and cut away the front of C<sub>60</sub> so that we can see the SO<sub>2</sub> molecule inside. If the selection unit (target) is molecules then selecting any atom of a molecule selects the whole molecule. The SO<sub>2</sub> is built without affecting the C<sub>60</sub> by selecting a subset (SO<sub>2</sub>) before building. This performs an incremental build (an important capability of the model builder) on only the selection (“select and operate”).

```
selection-target molecules           ; select whole molecules
select-atom 1 1                     ; select whole C60
color-selection violet               ; color violet for variety
select-none                          ; OK I'm done
create-atom 16                       ; create SO2 - Sulfur first
create-atom 8                        ; then the Oxygen
set-bond 1 2 1 3 d                  ; double bonds
create-atom 8
set-bond 1 2 1 3 d
select-atom 1 2                     ; select the SO2 for a build
menu-build-model-build              ; incremental build
select-none                          ; have good SO2 now
render-method disks
selection-target atoms
front-clip 54
```

### Optimize SO<sub>2</sub> inside Cavity

```
do-optimization
exit-script
```

Other scripts can be found on the HyperChem CD-ROM. You may wish to explore them as a CDK learning tool. To complete this chapter, we catalog all Hcl script commands.

## Catalog of HSV's and Direct Script Commands

What follows is an alphabetic listing of all HSV *variables* and all direct *commands*. Each entry contains an indication whether it is a variable or command and for a variable whether it is a Readonly variable or a Read/Write variable.

For variables the next line describes the type of variable it is while for commands the argument list is described. The third line of each entry gives a brief description of the entry. These entries are the result of running the script command, *print-variable-list*, which is the final arbiter of the complete list of script commands. The list of menu activations is left off this list since they can be inferred from HyperChem by just looking at its menus. Alternatively, the menu activations are described in the HyperChem Reference Manual.

abinitio-buffer-size: Variable, Read/Write.  
 Type: integer in range (1 .. 32767).  
 Two electron integral buffer size.

abinitio-calculate-gradient: Variable, Read/Write.  
 Type: boolean.  
 Enable Ab Initio gradient calculation (Single Point only).

abinitio-cutoff: Variable, Read/Write.  
 Type: float in range (0 .. 1e+010).  
 Two electron integral cutoff.

abinitio-d-orbitals: Variable, Read/Write.  
 Type: boolean.  
 Either five (False) or six (True).

abinitio-direct-scf: Variable, Read/Write.  
 Type: boolean.  
 Enable Ab Initio Direct SCF calculation.

abinitio-f-orbitals: Variable, Read/Write.  
 Type: boolean.  
 Either seven (False) or ten (True).

abinitio-integral-format: Variable, Read/Write.  
 Type: enum(raffenetti, regular).  
 Either regular or raffennetti.

abinitio-integral-path: Variable, Read/Write.  
 Type: string.  
 Path for storing integrals.

abinitio-mo-initial-guess: Variable, Read/Write.  
 Type: enum(core-hamiltonian, projected-huckel, projected-cndo, projected-indo).  
 Either core-hamiltonian, projected-huckel, projected-cndo, projected-indo.

abinitio-mp2-correlation-energy: Variable, Read/Write.  
 Type: boolean.  
 Enable Ab Initio MP2 correlation energy.

abinitio-mp2-frozen-core: Variable, Read/Write.  
 Type: boolean.  
 Enable Ab Initio MP2 frozen core.

abinitio-scf-convergence: Variable, Read/Write.  
 Type: float in range (0 .. 100).  
 SCF Convergence for Ab Initio.

abinitio-use-ghost-atoms: Variable, Read/Write.  
 Type: boolean.  
 Include or ignore ghost atoms.

accelerate-scf-convergence: Variable, Read/Write.  
 Type: boolean.  
 Whether to use DIIS procedure.

add-amino-acid: Command.  
 Arg list: string.  
 String-1 gives the name of an amino acid residue to add to the system.

add-nucleic-acid: Command.

Arg list: string.  
String-1 names the nucleotide to add to the current system.

align-molecule: Command.  
Arg list: .  
Align the inertial axes of the molecular system.

align-viewer: Command.  
Arg list: enum(x, y, z, line).  
Align the viewer's line-of-sight with the indicated axis or LINE.

allow-ions: Variable, Read/Write.  
Type: boolean.  
Whether to allow excess valence on atoms.

alpha-orbital-occupancy: Variable, Read/Write.  
Type: vector of float.  
(i) Number of electrons in the i-th MO.

alpha-scf-eigenvector: Variable, Read/Write.  
Type: vector of float-list.  
(i) Coefficients for the i-th MO.

amino-alpha-helix: Command.  
Arg list: (void).  
Subsequent additions of amino acid residues are to use alpha-helix torsions.

amino-beta-sheet: Command.  
Arg list: (void).  
Subsequent additions of amino acid residues are to use beta-sheet torsions.

amino-isomer: Variable, Read/Write.  
Type: enum(l, d).  
Whether amino acids are l or d.

amino-omega: Variable, Read/Write.  
Type: float angle in range (-360 .. 360).  
The Omega amino acid backbone angle.

amino-phi: Variable, Read/Write.  
Type: float angle in range (-360 .. 360).  
The Phi amino acid backbone angle.

amino-psi: Variable, Read/Write.  
Type: float angle in range (-360 .. 360).  
The Psi amino acid backbone angle.

animate-vibrations: Variable, Read/Write.  
Type: boolean.  
Whether or not to animate vibrations.

append-dynamics-average: Command.  
Arg list: string.  
Add a named selection to dynamics average gathering.

append-dynamics-graph: Command.  
Arg list: string.  
Add a named selection to dynamics graph display.

append-omsgs-to-file: Command.  
Arg list: string.  
String-1 gives the name of a file to which o-msgs are to be appended.

assign-basisset: Command.  
Arg list: string.  
Assign a basis set to a selection or system.

atom-basisset: Variable, Read/Write.  
Type: array of string.  
(iat, imol) The basis set of atom iat in molecule imol.

atom-charge: Variable, Read/Write.  
Type: array of float.  
(iat, imol) The charge of atom iat in molecule imol.

atom-color: Variable, Read/Write.  
 Type: array of enum(ByElement, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
 (iat, imol) The current color of the atom.

atom-count: Variable, Readonly.  
 Type: vector of integer.  
 (imol) The number of atoms in molecule imol.

atom-extra-basisset: Variable, Read/Write.  
 Type: array of string, float.  
 (iat, imol) The basis set of atom iat in molecule imol.

atom-info: Variable, Readonly.  
 Type: (unknown).  
 Funny composite to support backends.

atom-label-text: Variable, Readonly.  
 Type: array of string.  
 (iat, imol) RO. The text of the current atom label.

atom-labels: Variable, Read/Write.  
 Type: enum(None, Symbol, Name, Number, Type, Charge, Mass, Basis-Set, Chirality).  
 Label for atoms.

atom-mass: Variable, Read/Write.  
 Type: array of float.  
 (iat, imol) The mass of atom iat in molecule imol.

atom-name: Variable, Read/Write.  
 Type: array of string.  
 (iat, imol) The name of atom iat in molecule imol.

atom-spin-density-at-nucleus: Variable, Read/Write.  
 Type: array of float.  
 (iat, imol) The electron density of nucleus of atom iat in molecule imol.

atom-spin-population: Variable, Read/Write.  
 Type: array of float.  
 (iat, imol) The spin density of atom iat in molecule imol.

atom-type: Variable, Read/Write.  
 Type: array of string.  
 (iat, imol) The type of atom iat in molecule imol.

atomic-number: Variable, Read/Write.  
 Type: array of integer.  
 (iat, imol) The atomic number of atom iat in molecule imol.

atomic-symbol: Variable, Readonly.  
 Type: array of string.  
 (iat, imol) The element symbol of the atom.

auxilliary-basis: Variable, Read/Write.  
 Type: integer in range (1 .. 3).  
 1=A1, 2=A2, 3=P1

back-clip: Variable, Read/Write.  
 Type: float.  
 Set back clipping plane.

backend-active: Variable, Read/Write.  
 Type: boolean.  
 Whether current channel is an active backend.

backend-communications: Variable, Read/Write.  
 Type: enum(Local, Remote).  
 Whether to compute on local or remote host.

backend-host-name: Variable, Read/Write.  
 Type: string.  
 The name of remote host for backend communications.

backend-process-count: Variable, Read/Write.  
 Type: integer in range (1 .. 32).

- The number of processes to run.  
backend-user-id: Variable, Read/Write.  
Type: string.  
The user id to use on the remote host for backend communications.
- backend-user-password: Variable, Read/Write.  
Type: string.  
The password for user id to use on the remote host for backend communications.
- balls-highlighted: Variable, Read/Write.  
Type: boolean.  
Balls and Balls-and-Cylinders should be highlighted when shaded.
- balls-radius-ratio: Variable, Read/Write.  
Type: float in range (0.001 .. 1).  
Size of the Balls relative to the maximum value.
- balls-shaded: Variable, Read/Write.  
Type: boolean.  
Balls and Balls-and-Cylinders should be shaded.
- basisset-count: Variable, Readonly.  
Type: integer.  
Number of coefficients required to describe a molecular orbital.
- bend-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.
- beta-orbital-occupancy: Variable, Read/Write.  
Type: vector of float.  
(i) Number of electrons in the i-th MO.
- beta-scf-eigenvector: Variable, Read/Write.  
Type: vector of float-list.  
(i) Coefficients for the i-th MO.
- bond-color: Variable, Read/Write.  
Type: enum(ByElement, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
The color used for drawing atoms and bonds.
- bond-spacing-display-ratio: Variable, Read/Write.  
Type: float in range (0 .. 1).  
Bond spacing display ratio.
- builder-enforces-stereo: Variable, Read/Write.  
Type: boolean.  
Whether the model builder implicitly enforces any existing stereochemistry.
- calculation-method: Variable, Read/Write.  
Type: enum(MolecularMechanics, SemiEmpirical, AbInitio, DFT).  
Whether molecular mechanics, semi-empirical, or ab initio.
- cancel-menu: Variable, Read/Write.  
Type: boolean.  
Whether the cancel menu is up, or the normal one.
- cancel-notify: Command.  
Arg list: string.  
String-1 names a variable to stop watching.
- change-stereochem: Command.  
Arg list: integer, integer.  
Immediately change the stereochemistry about (iat, imol).
- change-user-menuitem: Command.  
Arg list: integer, string, string.  
Change the text and procedure associated with the specified user MenuItem.
- chirality: Variable, Read/Write.  
Type: array of string.

- (iat, imol) A, R, S, or ?, for achiral, R, S, or unknown chirality.
- ci-criterion: Variable, Read/Write.  
Type: enum(Energy, Orbital).  
One of: energy, orbital.
- ci-excitation-energy: Variable, Read/Write.  
Type: float in range (0 .. 10000).  
When ci-criterion=energy, maximum excitation energy.
- ci-occupied-orbitals: Variable, Read/Write.  
Type: integer in range (0 .. 32767).  
When ci-criterion=orbital, count of occupied orbitals included.
- ci-state-to-optimize: Command.  
Arg list: unregistered.  
Which CI state to optimize with conjugate directions
- ci-unoccupied-orbitals: Variable, Read/Write.  
Type: integer in range (0 .. 32767).  
When ci-criterion=orbital, count of unoccupied orbitals included.
- clip-cursor: Variable, Read/Write.  
Type: float in range (0 .. 1000).  
Select Z axis clip cursor tool.
- clip-icon-step: Variable, Read/Write.  
Type: float in range (0 .. 1000).  
Select clip step.
- color-element: Command.  
Arg list: integer, enum().  
Element Int-1 gets color String-2 as its default color.
- color-selection: Command.  
Arg list: string.  
String-1 names a color for the current selection.
- compile-script-file: Command.  
Arg list: string, string.  
Compile file string-1, writing result to string-2
- configuration: Variable, Read/Write.  
Type: integer.  
The current UV configuration of the system.
- configuration-interaction: Variable, Read/Write.  
Type: enum(NoCI, SinglyExcited, Microstate).  
One of: no-ci, singly-excited, microstate.
- connectivity-in-pdb-file: Variable, Read/Write.  
Type: boolean.  
Whether connectivity information is to be included in a PDB file.
- constrain-bond-angle: Command.  
Arg list: float angle in range (-360 .. 360).  
Float-1 gives the angle constraint for the three currently selected atoms.
- constrain-bond-down: Command.  
Arg list: integer, integer, integer, integer.  
Constrain the bond from (iat1, imol1) to (iat2, imo2) to be down.
- constrain-bond-length: Command.  
Arg list: float in range (0 .. 100).  
Float-1 gives the length constraint for the two currently selected atoms.
- constrain-bond-torsion: Command.  
Arg list: float angle in range (-360 .. 360).  
Float-1 gives the torsion constraint for the four currently selected atoms.
- constrain-bond-up: Command.  
Arg list: integer, integer, integer, integer.  
Constrain the bond from (iat1, imol1) to (iat2, imo2) to be up.
- constrain-change-stereo: Command.  
Arg list: integer, integer.



Constrain atom (iat, imol) to change the current stereochemistry.  
constrain-drawing: Variable, Read/Write.  
Type: boolean.  
Whether to constrain bond lengths and angles to canonicalize drawing of molecule

constrain-fix-stereo: Command.  
Arg list: integer, integer.  
Constrain atom (iat, imol) to enforce the current stereochemistry.

constrain-geometry: Command.  
Arg list: string.  
String-1 describes the geometry constraint around the currently selected atom.

coordinates: Variable, Read/Write.  
Type: array of float, float, float.  
(iat, imol) The x, y, and z coordinates of atom iat in molecule imol.

coordination: Variable, Readonly.  
Type: array of integer.  
(iat, imol) The coordination number for the specified atom.

correlation-functional: Variable, Read/Write.  
Type: enum(None, Perdew86, VWN, LYP, PZ81, PW91, PBE96, HCTH98).  
Perdew, LYP, etc.

cpk-max-double-buffer-atoms: Variable, Read/Write.  
Type: integer in range (0 .. 2147483647).  
Maximum number of double buffered atoms in cpk rendering mode.

create-atom: Command.  
Arg list: integer in range (0 .. 103).  
Create a new atom at the origin with atomic number nAtno.

current-file-name: Variable, Readonly.  
Type: string.  
The name of the current file.

custom-title: Variable, Read/Write.  
Type: string.  
Custom Title string, append string to title.

cutoff-inner-radius: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
The distance (in Angstroms) to begin a switched cutoff.

cutoff-outer-radius: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
The distance (in Angstroms) at which nonbonded interactions become zero.

cutoff-type: Variable, Read/Write.  
Type: enum(None, Switched, Shifted).  
Electrostatic cutoff to apply to molecular mechanics calculations.

cycle-atom-stereo: Command.  
Arg list: integer, integer.  
Advance the stereo constraint about atom (iat, imol).

cycle-bond-stereo: Command.  
Arg list: integer, integer, integer, integer.  
Advance the stereo constraint along the bond (iat1, imol1)--(iat2, imol2).

cylinders-color-by-element: Variable, Read/Write.  
Type: boolean.  
Color Cylinders using element colors.

cylinders-width-ratio: Variable, Read/Write.  
Type: float in range (0 .. 1).  
Width of the Cylinders relative to the maximum value.

d-orbitals-on-second-row: Variable, Read/Write.  
Type: boolean.

Include D orbitals on second row.  
 declare-float: Command.  
   Arg list: string.  
   Declare a new floating-point variable.  
 declare-integer: Command.  
   Arg list: string.  
   Declare a new integer variable.  
 declare-string: Command.  
   Arg list: string.  
   Declare a new string variable.  
 default-element: Variable, Read/Write.  
   Type: integer in range (0 .. 103).  
   The atomic number of the default element for drawing operations.  
 delete-atom: Command.  
   Arg list: integer, integer.  
   Delete the specified atom.  
 delete-file: Command.  
   Arg list: string.  
   filename to be deleted.  
 delete-named-selection: Command.  
   Arg list: string.  
   Remove the named selection String-1 from the list of named selections.  
 delete-selected-atoms: Command.  
   Arg list: (void).  
   Delete the currently selected atoms.  
 dipole-moment: Variable, Read/Write.  
   Type: float in range (-1e+010 .. 1e+010).  
   Dipole moment.  
 dipole-moment-components: Variable, Read/Write.  
   Type: float, float, float.  
   Dipole moment components.  
 do-langevin-dynamics: Command.  
   Arg list: (void).  
   Perform a Langevin dynamics computation on the system.  
 do-molecular-dynamics: Command.  
   Arg list: (void).  
   Perform a molecular dynamics computation on the system.  
 do-monte-carlo: Command.  
   Arg list: (void).  
   Perform a Monte Carlo computation on the system.  
 do-optimization: Command.  
   Arg list: (void).  
   Perform a structure optimization on the system.  
 do-qm-calculation: Variable, Read/Write.  
   Type: boolean.  
   For single-point QM calculations, whether to re-compute wave function.  
 do-qm-graph: Variable, Read/Write.  
   Type: boolean.  
   For single-point QM calculations, to graph some data.  
 do-qm-isosurface: Variable, Read/Write.  
   Type: boolean.  
   For single-point QM calculations, to generate iso-surface of results.  
 do-single-point: Command.  
   Arg list: (void).  
   Perform a single-point computation on the system.  
 do-vibrational-analysis: Command.  
   Arg list: (void).

Perform a vibrational analysis computation on the system.

dot-surface-angle: Variable, Read/Write.  
Type: float angle in range (-90 .. 90).  
Dot surface angle.

double-buffered-display: Variable, Read/Write.  
Type: boolean.  
Whether display operations are double-buffered.

dynamics-average-period: Variable, Read/Write.  
Type: integer in range (1 .. 32767).  
Computation results from dynamics run.

dynamics-bath-relaxation-time: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Bath relaxation time for dynamics.

dynamics-collection-period: Variable, Read/Write.  
Type: integer in range (1 .. 32767).  
Dynamics data collection interval.

dynamics-constant-temp: Variable, Read/Write.  
Type: boolean.  
Whether to keep temperature fixed at dynamics-simulation-temp.

dynamics-cool-time: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Time taken to change from dynamics-simulation-temp to dynamics-final-temp.

dynamics-final-temp: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Temperature to cool back to when annealing.

dynamics-friction-coefficient: Variable, Read/Write.  
Type: float in range (0 .. 1000000).  
Friction coefficient for Langevin dynamics.

dynamics-heat-time: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Time taken to change from dynamics-starting-temp -> dynamics-simulation-temp.

dynamics-info-elapsed-time: Variable, Readonly.  
Type: float in range (0 .. 1e+010).  
Elapsed time in dynamics run.

dynamics-info-kinetic-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Computation results from dynamics run.

dynamics-info-last-update: Variable, Readonly.  
Type: boolean.  
Last update from dynamics run.

dynamics-info-potential-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Computation results from dynamics run.

dynamics-info-temperature: Variable, Readonly.  
Type: float in range (0 .. 1e+010).  
Computation results from dynamics run.

dynamics-info-total-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Computation results from dynamics run.

dynamics-playback: Variable, Read/Write.  
Type: enum(none, playback, record).  
Playback a recorded dynamics run.

dynamics-playback-end: Variable, Read/Write.  
Type: integer in range (0 .. 32767).  
End playback of recorded dynamics run.

dynamics-playback-period: Variable, Read/Write.

Type: integer in range (1 .. 32767).  
Dynamics playback interval.

dynamics-playback-start: Variable, Read/Write.  
Type: integer in range (0 .. 32767).  
Start playback of recorded dynamics run.

dynamics-restart: Variable, Read/Write.  
Type: boolean.  
Use saved velocities.

dynamics-run-time: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Total integration time at dynamics-simulation-temp.

dynamics-seed: Variable, Read/Write.  
Type: integer in range (-32768 .. 32767).  
Seed for dynamics initialization random number generator.

dynamics-simulation-temp: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
High temperature for the dynamics run.

dynamics-snapshot-filename: Variable, Read/Write.  
Type: string.  
Name file of to store dynamics run.

dynamics-snapshot-period: Variable, Read/Write.  
Type: integer in range (1 .. 32767).  
Set recording interval of dynamics run.

dynamics-starting-temp: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Starting temperature for the dynamics run.

dynamics-temp-step: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Step size (K) by which temperature is changed.

dynamics-time-step: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Size of the step for integration.

error: Variable, Read/Write.  
Type: string.  
The current error.

errors-are-not-omsgs: Command.  
Arg list: (void).  
Specifies that error messages are to appear in message boxes.

errors-are-omsgs: Command.  
Arg list: (void).  
Specifies that error messages should be treated like o-msgs.

estatic-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.

exchange-functional: Variable, Read/Write.  
Type: enum(None, Hartree-Fock, Slater, Becke88, PW91, Gill96, PBE96, HCTH98, B3-LYP, B3-PW91, EDF1, Becke97).  
Slater, Becke88, etc.

excited-state: Variable, Read/Write.  
Type: boolean.  
False for lowest state, true for next-lowest state.

execute-client: Command.  
Arg list: string.  
Run a client application.

execute-hyperchem-client: Command.  
Arg list: string.  
Run a client application. App can reliably connect to instance of HyperChem.

execute-string: Command.  
Arg list: string.  
Execute the string variable as a script.

exit-script: Command.  
Arg list: (void).  
Exit the current script.

explicit-hydrogens: Variable, Read/Write.  
Type: boolean.  
Whether hydrogens are to be drawn explicitly.

export-dipole: Variable, Read/Write.  
Type: boolean.  
Whether or not to export dipole moment data to .EXT file.

export-ir: Variable, Read/Write.  
Type: boolean.  
Whether or not to export IR data to .EXT file.

export-orbitals: Variable, Read/Write.  
Type: boolean.  
Whether or not to export orbital data to .EXT file.

export-property-file: Command.  
Arg list: string.  
Writes properties to the named file.

export-uv: Variable, Read/Write.  
Type: boolean.  
Whether or not to export UV data to .EXT file.

factory-settings: Command.  
Arg list: (void).  
Reset chem to its out-of-the-box state.

field-direction: Variable, Read/Write.  
Type: integer in range (1 .. 3).  
direction (X, Y or Z) of the static electric field applied to the system

field-strength: Variable, Read/Write.  
Type: float in range (-1000 .. 1000).  
strength (a.u.) of the static electric field applied to the system

file-diff-message: Command.  
Arg list: string, string, string, string.  
Compare file1 to file2; if they are the same say string3, else say string4.

file-format: Variable, Read/Write.  
Type: string.  
The molecule file format.

file-needs-saved: Variable, Read/Write.  
Type: boolean.  
Whether the current system needs to be saved.

front-clip: Variable, Read/Write.  
Type: float.  
Set front clipping plane.

global-inhibit-redisplay: Variable, Readonly.  
Type: boolean.  
Whether redisplay of the system is inhibited (readonly)

gradient-x: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Molecular gradient in the X direction

gradient-y: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Molecular gradient in the Y direction

gradient-z: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Molecular gradient in the Z direction

gradients: Variable, Read/Write.  
 Type: array of float, float, float.  
 (iat, imol) The x, y, and z gradients of atom iat in molecule imol.

graph-beta: Variable, Read/Write.  
 Type: boolean.  
 If true and UHF, graph beta-spin orbitals instead of alpha.

graph-contour-increment: Variable, Read/Write.  
 Type: float in range (-1e+010 .. 1e+010).  
 Increment between contour lines.

graph-contour-increment-other: Variable, Read/Write.  
 Type: boolean.  
 Whether to use graph-increment-other (true) or use defaults (false).

graph-contour-levels: Variable, Read/Write.  
 Type: integer in range (1 .. 32767).  
 The number of contour levels to plot.

graph-contour-start: Variable, Read/Write.  
 Type: float in range (-1e+010 .. 1e+010).  
 Value for first contour line.

graph-contour-start-other: Variable, Read/Write.  
 Type: boolean.  
 Whether to use graph-contour-start (true) or use defaults (false).

graph-data-row: Variable, Readonly.  
 Type: vector of float-list.  
 (i) The values on the i-th row of graph data.

graph-data-type: Variable, Read/Write.  
 Type: enum(electrostatic, charge-density, orbital, orbital-squared, spin-density).  
 The type of wavefunction data to plot.

graph-horizontal-grid-size: Variable, Read/Write.  
 Type: integer in range (2 .. 8192).  
 Number of data grid points for plotting in the horizontal direction.

graph-orbital-offset: Variable, Read/Write.  
 Type: integer in range (0 .. +Inf).  
 Display orbital offset.

graph-orbital-selection-type: Variable, Read/Write.  
 Type: enum(lumo-plus, homo-minus, orbital-number).  
 Display orbital type.

graph-plane-offset: Variable, Read/Write.  
 Type: float in range (-1e+010 .. 1e+010).  
 Offset along viewer's Z axis of the plane of the data to plot.

graph-vertical-grid-size: Variable, Read/Write.  
 Type: integer in range (2 .. 8192).  
 Number of data grid points for plotting in the vertical direction.

grid-max-value: Variable, Readonly.  
 Type: float in range (-1e+010 .. 1e+010).  
 The isosurface maximum grid value.

grid-min-value: Variable, Readonly.  
 Type: float in range (-1e+010 .. 1e+010).  
 The isosurface minimum grid value.

hbond-energy: Variable, Readonly.  
 Type: float in range (-1e+010 .. 1e+010).  
 Results from backend computation.

heat-of-formation: Variable, Read/Write.  
 Type: float in range (-1e+010 .. 1e+010).  
 Heat of formation.

help: Command.  
 Arg list: string.  
 Give help on topic String-1.

hide-errors: Variable, Read/Write.  
Type: boolean.  
Whether to display error messages on the screen (channel specific).

hide-messages: Variable, Read/Write.  
Type: boolean.  
Whether to display MESSAGE value on the screen.

hide-toolbar: Variable, Read/Write.  
Type: boolean.  
Command to toggle the toolbar.

hide-warnings: Variable, Read/Write.  
Type: boolean.  
Whether to display warning messages on the screen (channel specific).

huckel-constant: Variable, Read/Write.  
Type: float in range (0 .. 10).  
Extended Huckel constant.

huckel-scaling-factor: Variable, Read/Write.  
Type: float in range (0 .. 100000).  
Extended Huckel scaling factor.

huckel-weighted: Variable, Read/Write.  
Type: boolean.  
Extended Huckel weighting factor.

hydrogens-in-pdb-file: Variable, Read/Write.  
Type: boolean.  
Should Hydrogens be written into a PDB file?

ignore-script-errors: Variable, Read/Write.  
Type: boolean.  
Whether script errors should be ignored, otherwise offer to abort on errors.

image-color: Variable, Read/Write.  
Type: boolean.  
Save image in color?

image-destination-clipboard: Variable, Read/Write.  
Type: boolean.  
Save image to the Windows clipboard?

image-destination-file: Variable, Read/Write.  
Type: boolean.  
Save image to a file?

image-destination-filename: Variable, Read/Write.  
Type: string.  
Save image to a given file

image-file-bitmap: Variable, Read/Write.  
Type: boolean.  
Save in bitmap format?

image-file-bitmapRGB: Variable, Read/Write.  
Type: boolean.  
Save in bitmap-RGB format?

image-file-metafile: Variable, Read/Write.  
Type: boolean.  
Save in Windows metafile format?

image-include-cursor: Variable, Read/Write.  
Type: boolean.  
Include cursor in image?

image-source-window: Variable, Read/Write.  
Type: enum(TopLevel, Workspace, HyperChem, FullScreen).  
Extent of image to capture.

import-dipole: Variable, Read/Write.  
Type: boolean.  
Whether or not to import dipole moment data from .EXT file.

import-ir: Variable, Read/Write.  
 Type: boolean.  
 Whether or not to import IR data from .EXT file.

import-orbitals: Variable, Read/Write.  
 Type: boolean.  
 Whether or not to import orbital data from .EXT file.

import-property-file: Command.  
 Arg list: string.  
 Reads properties from the named file.

import-uv: Variable, Read/Write.  
 Type: boolean.  
 Whether or not to import UV data from .EXT file.

info-access: Variable, Readonly.  
 Type: string.  
 (RO) Access (R, W, RW) for info-variable-target.

info-enum-id-of: Variable, Readonly.  
 Type: string.  
 (RO) Binary id of info-enum-target value for info-variable-target.

info-enum-list: Variable, Readonly.  
 Type: string.  
 (RO) If enumerated type, list of enumerated values for info-variable-target.

info-factory-setting: Variable, Readonly.  
 Type: string.  
 (RO) factory setting value for info-variable-target.

info-id-of: Variable, Readonly.  
 Type: integer.  
 (RO) Binary id of info-variable-target.

info-type-of: Variable, Readonly.  
 Type: string.  
 (RO) Type of info-variable-target.

info-type-of-element: Variable, Readonly.  
 Type: string.  
 (RO) If info-type-of is array or vector, type of elements.

info-variable-target: Variable, Read/Write.  
 Type: string.  
 Variable for which info is required.

inhibit-redisplay: Variable, Read/Write.  
 Type: boolean.  
 Whether redisplay of the system is inhibited by current channel.

ir-animate-amplitude: Variable, Read/Write.  
 Type: float in range (0 .. 10).  
 The distance in angstroms to move the fastest atom during vibrations(0..10)

ir-animate-cycles: Variable, Read/Write.  
 Type: integer in range (0 .. +Inf).  
 The number of cycles (length of time) to animate. 0 means forever.

ir-animate-steps: Variable, Read/Write.  
 Type: integer in range (3 .. +Inf).  
 The number of steps to use in animating vibrations (3 -- BIG)

ir-band-count: Variable, Read/Write.  
 Type: integer.  
 Number of ir bands.

ir-frequency: Variable, Read/Write.  
 Type: vector of float.  
 (i) Frequency of the i-th IR band.

ir-intensity: Variable, Read/Write.  
 Type: vector of float.



- (i) Intensity of the i-th IR band.
- ir-intensity-components: Variable, Read/Write.  
Type: vector of float, float, float.  
(i) Intensity components (x,y, and z) of the i-th IR band.
- ir-normal-mode: Variable, Read/Write.  
Type: vector of float-list.  
(i) Normal node. This is a vector holding x, y, and z for each atom.
- is-extended-hydrogen: Variable, Readonly.  
Type: array of boolean.  
(iat, imol) RO. Is the atom an extended hydrogen?
- is-ring-atom: Variable, Readonly.  
Type: array of boolean.  
(iat, imol) RO. Is the atom in a ring?
- isosurface-grid-step-size: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
The isosurface grid stepsize.
- isosurface-hide-molecule: Variable, Read/Write.  
Type: boolean.  
Show only the isosurfaces.
- isosurface-map-function: Variable, Read/Write.  
Type: boolean.  
Display a mapped function isosurface.
- isosurface-map-function-display-legend: Variable, Read/Write.  
Type: boolean.  
Display the isosurface mapped function range legend.
- isosurface-map-function-range: Variable, Read/Write.  
Type: float, float.  
Set the isosurface mapped function range.
- isosurface-mesh-quality: Variable, Read/Write.  
Type: enum(coarse, medium, fine).  
Use coarse, medium or fine grid settings.
- isosurface-render-method: Variable, Read/Write.  
Type: enum(wire-mesh, Jorgensen-Salem, lines, flat-surface, shaded-surface, Gouraud-shaded-surface, translucent-surface).  
The method used to render the isosurfaces.
- isosurface-threshold: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
The isosurface threshold value?
- isosurface-transparency-level: Variable, Read/Write.  
Type: float in range (0 .. 1).  
The isosurface level of transparency (0 = opaque, 1 = transparent)
- isosurface-x-min: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
The smallest x coordinate of the grid data.
- isosurface-x-nodes: Variable, Read/Write.  
Type: integer in range (2 .. 128).  
The number of isosurface x nodes.
- isosurface-y-min: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
The smallest y coordinate of the grid data.
- isosurface-y-nodes: Variable, Read/Write.  
Type: integer in range (2 .. 128).  
The number of isosurface y nodes.
- isosurface-z-min: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
The smallest z coordinate of the grid data.
- isosurface-z-nodes: Variable, Read/Write.  
Type: integer in range (2 .. 128).

The number of isosurface z nodes.  
 keep-atom-charges: Variable, Read/Write.  
 Type: boolean.  
 Keep atom charges when switch calculation methods.

load-default-menu: Command.  
 Arg list: (void).  
 Load the Hyperchem default menu.

load-user-menu: Command.  
 Arg list: string.  
 Load user customized menu.

log-comment: Command.  
 Arg list: string.  
 Write String-1 into the current logfile.

magnetic-field-direction: Command.  
 Arg list: unregistered.  
 direction (X, Y, or Z) of static magnetic field

magnetic-field-strength: Command.  
 Arg list: unregistered.  
 strength (a.u.) of the static magnetic field applied to system

max-iterations: Variable, Read/Write.  
 Type: integer in range (1 .. 32767).  
 Maximum number of SCF iterations.

mechanics-data: Variable, Readonly.  
 Type: (unknown).  
 Funny composite to support backends.

mechanics-dielectric: Variable, Read/Write.  
 Type: enum(Constant, DistanceDependent), enum(Constant, DistanceDependent), enum(Constant, DistanceDependent).  
 The method for calculating dielectric permittivity.

mechanics-dielectric-scale-factor: Variable, Read/Write.  
 Type: float, float, float, float.  
 Constant to multiply distance-dependent dielectric by.

mechanics-electrostatic-scale-factor: Variable, Read/Write.  
 Type: float, float, float, float.  
 Scale factor for 1-4 dielectric interactions.

mechanics-info: Variable, Readonly.  
 Type: (unknown).  
 Funny composite to support backends.

mechanics-mmp-electrostatics: Variable, Read/Write.  
 Type: enum(BondDipoles, ConnectedBondDipoles, AtomicCharges, None).  
 The type of electrostatic interaction to use in MM+ calculations.

mechanics-print-level: Variable, Read/Write.  
 Type: integer in range (0 .. 9).  
 Print level for molecular mechanics.

mechanics-van-der-waals-scale-factor: Variable, Read/Write.  
 Type: float, float, float, float.  
 Scale factor for 1-4 van der Waals interactions.

merge-file: Command.  
 Arg list: string.  
 String-1 names the molecule file to be merged with the current system.

message: Variable, Read/Write.  
 Type: string.  
 string1 is an output message.

module-name: Variable, Read/Write.  
 Type: string.  
 Current name of module.

molecular-mechanics-method: Variable, Read/Write.  
Type: enum(mm+, amber, bio+, opl, amber94, charmm22, tndo).  
The type of molecular mechanics method to perform.

molecular-symmetry: Variable, Read/Write.  
Type: string.  
Point group of the system, if known.

molecule-count: Variable, Readonly.  
Type: integer.  
The number of molecules in the system.

molecule-name: Variable, Readonly.  
Type: vector of string.  
(i) The name of the i-th molecule.

moments-of-inertia: Variable, Readonly.  
Type: boolean.  
Return the moments of inertia of selected system.

monte-carlo-cool-steps: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Steps taken to change from dynamics-simulation-temp -> dynamics-final-temp.

monte-carlo-heat-steps: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Steps taken to change from dynamics-starting-temp -> dynamics-simulation-temp.

monte-carlo-info-acceptance-ratio: Variable, Readonly.  
Type: float in range (0 .. 1).  
Computation result from Monte Carlo run.

monte-carlo-max-delta: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Maximum allowed size of the displacement step in Angstroms.

monte-carlo-run-steps: Variable, Read/Write.  
Type: float in range (0 .. 1e+010).  
Total number of steps at dynamics-simulation-temp.

mouse-mode: Variable, Read/Write.  
Type: enum(Drawing, Selecting, Rotating, ZRotating, Translating, ZTranslating, Zooming, Clipping).  
The function of the cursor in the drawing area.

mp2-energy: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
MP2 energy.

multiple-selections: Variable, Read/Write.  
Type: boolean.  
Allow multiple selections.

multiplicity: Variable, Read/Write.  
Type: integer in range (1 .. 6).  
Spin multiplicity.

mutate-residue: Command.  
Arg list: string.  
Change selected residue into String-1.

name-molecule: Command.  
Arg list: string.  
Give name String-1 to molecule of current selection

name-selection: Command.  
Arg list: string.  
Name the current selection String-1.

named-selection-count: Variable, Readonly.  
Type: integer.  
The number of named selections.

named-selection-name: Variable, Readonly.

Type: vector of string.  
 (i) The name of the i-th named selection.  
 named-selection-value: Variable, Readonly.  
 Type: vector of float.  
 (i) The value of the i-th named selection (bond length, angle, etc).  
 negatives-color: Variable, Read/Write.  
 Type: enum(ByElement, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
 The color of the negatives.  
 neighbors: Variable, Readonly.  
 Type: array of (unknown).  
 (iat, imol) The neighbor list for the specified atom.  
 no-source-refs-in-errors: Command.  
 Arg list: (void).  
 Controls presentation of filename, line number in error messages.  
 non-standard-pdb-names: Variable, Read/Write.  
 Type: boolean.  
 If true, then look for left-justified pdb element names .  
 nonbond-energy: Variable, Readonly.  
 Type: float in range (-1e+010 .. 1e+010).  
 Results from backend computation.  
 notify-on-update: Command.  
 Arg list: string.  
 String-1 gives name of variable whose value-changes are desired.  
 notify-with-text: Variable, Read/Write.  
 Type: boolean.  
 For DDE channels only. Are notifications to be text (otherwise binary).  
 nucleic-a-form: Command.  
 Arg list: (void).  
 Subsequent additions of nucleotides will use a-form torsion angles.  
 nucleic-alpha: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA alpha backbone torsion.  
 nucleic-b-form: Command.  
 Arg list: (void).  
 Subsequent additions of nucleotides will use b-form torsion angles.  
 nucleic-backwards: Variable, Read/Write.  
 Type: boolean.  
 Build backward DNA.  
 nucleic-beta: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA beta backbone torsion.  
 nucleic-chi: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA chi backbone torsion.  
 nucleic-delta: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA delta backbone torsion.  
 nucleic-double-strand: Variable, Read/Write.  
 Type: boolean.  
 Build double strand DNA.  
 nucleic-epsilon: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA epsilon backbone torsion.  
 nucleic-gamma: Variable, Read/Write.  
 Type: float angle in range (-360 .. 360).  
 DNA gamma backbone torsion.  
 nucleic-sugar-pucker: Variable, Read/Write.

Type: enum(2-endo, 3-endo).  
Select sugar pucker of DNA.

nucleic-z-form: Command.  
Arg list: (void).  
Subsequent additions of nucleotides will use z-form torsion angles.

nucleic-zeta: Variable, Read/Write.  
Type: float angle in range (-360 .. 360).  
DNA zeta backbone torsion.

omsg-file: Variable, Read/Write.  
Type: string.  
String is file to append omgs.

omsgs-not-to-file: Command.  
Arg list: (void).  
Directs that o-msgs are not to be written to a file, but to a messagebox.

omsgs-to-file: Command.  
Arg list: string.  
String-1 gives the name of the new file to which o-msgs are to be written.

one-line-arrays: Variable, Read/Write.  
Type: boolean.  
Whether to emit arrays all on one line.

open-file: Command.  
Arg list: string.  
String-1 names the molecule file to be opened.

optim-algorithm: Variable, Read/Write.  
Type: enum(SteepestDescents, FletcherReeves, PolakRibiere, Newton-Raphson, EigenvectorFollow, ConjugateDirections).  
The algorithm to use for structure optimizations.

optim-converged: Variable, Readonly.  
Type: boolean.  
Whether optimization has converged.

optim-convergence: Variable, Read/Write.  
Type: float in range (0 .. 100).  
Optimization gradient convergence.

optim-max-cycles: Variable, Read/Write.  
Type: integer in range (1 .. +Inf).  
Maximum number of optimization steps.

orbital-count: Variable, Read/Write.  
Type: integer.  
Number of molecular orbitals available.

orbital-symmetry: Variable, Read/Write.  
Type: vector of string.  
(i) Irreducible representation of the i-th orbital.

parameter-set-changed: Variable, Read/Write.  
Type: boolean.  
Toggles the state of the backend parameters.

path: Variable, Read/Write.  
Type: string.  
Current directory.

pause-for: Command.  
Arg list: integer in range (0 .. 32767).  
HyperChem pauses for Int-1 seconds.

periodic-boundaries: Variable, Read/Write.  
Type: boolean.  
Whether to use periodic boundary conditions.

periodic-box-size: Variable, Readonly.  
Type: (unknown).  
Return the size of the periodic box.

polar-field-strength: Variable, Read/Write.

Type: float in range (-1000 .. 1000).  
 Electric field strength for polarizability calculations  
 polarizability-calculated: Variable, Read/Write.  
 Type: boolean.  
 calculate ab initio or semiempirical polarizability  
 pop-no-value: Command.  
 Arg list: string.  
 variable: pop stack, don't restore value.  
 pop-value: Command.  
 Arg list: string.  
 variable: Restore pushed value.  
 positives-color: Variable, Read/Write.  
 Type: enum(ByElement, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
 The color of the positives.  
 print: Command.  
 Arg list: (void).  
 Print to default printer.  
 print-variable-list: Command.  
 Arg list: string.  
 Write a summary of the state variables to file String-1  
 printer-background-white: Variable, Read/Write.  
 Type: boolean.  
 Force printer background color to white.  
 push: Command.  
 Arg list: string.  
 variable: Push copy of current value onto stack.  
 quantum-print-level: Variable, Read/Write.  
 Type: integer in range (0 .. 9).  
 Print level for quantum mechanics.  
 quantum-total-charge: Variable, Read/Write.  
 Type: integer in range (-32768 .. 32767).  
 The total charge of the quantum mechanical system.  
 query-response-has-tag: Variable, Read/Write.  
 Type: boolean.  
 Do HSV responses have identifying tags?  
 query-value: Command.  
 Arg list: .  
 String-1 names the state variable whose value should be emitted as an o-  
 msg.  
 read-binary-script: Command.  
 Arg list: string.  
 String-1 names the compiled script file that should be read.  
 read-script: Command.  
 Arg list: string.  
 String-1 names the file that should be read as a command script.  
 read-tcl-script: Command.  
 Arg list: string.  
 String-1 names the file that should be read and executed as a Tcl/Tk  
 script.  
 remove-all-stereo-constraints: Command.  
 Arg list: (void).  
 Remove all stereo constraints.  
 remove-stereo-constraint: Command.  
 Arg list: integer, integer.  
 Remove any stereo constraints from atom (iat, imol)  
 render-method: Variable, Read/Write.

Type: enum(sticks, balls, balls-and-cylinders, spheres, dots, sticks-and-dots).  
How the system is to be displayed.  
reorder-selections: Variable, Read/Write.  
Type: boolean.  
Should atoms in selections be reordered to make proper angles, etc.?  
request: Command.  
Arg list: string.  
Displays String-1 in a modeless dialog until the user click OK.  
residue-charge: Variable, Readonly.  
Type: array of float.  
(ires, imol) The net charge on the residue.  
residue-coordinates: Variable, Readonly.  
Type: array of float, float, float.  
(ires, imol) The center-of-mass for the residue.  
residue-count: Variable, Readonly.  
Type: vector of integer.  
(imol) The number of residues in molecule imol.  
residue-label-text: Variable, Readonly.  
Type: array of string.  
(ires, imol) Text of the label for the residue.  
residue-labels: Variable, Read/Write.  
Type: enum(None, Name, Sequence, NameSequence).  
Label for residues.  
residue-name: Variable, Readonly.  
Type: array of string.  
(ires, imol) The name of the residue.  
restraint: Command.  
Arg list: string, float, float.  
(selection-name, value, force-constant)  
restraint-tether: Command.  
Arg list: .  
(selection-name, [POINT|x,y,z], force-constant)  
revert-element-colors: Command.  
Arg list: (void).  
Use default color scheme for displaying atoms.  
rms-gradient: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.  
rotate-molecules: Command.  
Arg list: enum(x, y, z, line), float.  
rotate-molecules (axis, angle)  
rotate-viewer: Command.  
Arg list: enum(x, y, z, line), float.  
rotate-viewer (axis, angle)  
scf-atom-energy: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
SCF atom energy.  
scf-binding-energy: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
SCF binding energy.  
scf-convergence: Variable, Read/Write.  
Type: float in range (0 .. 100).  
Convergence required for QM SCF computations.  
scf-core-energy: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
SCF core energy.  
scf-electronic-energy: Variable, Read/Write.

Type: float in range (-1e+010 .. 1e+010).  
 SCF electronic energy.

scf-orbital-energy: Variable, Read/Write.  
 Type: vector of float.  
 (i) Eigenvalues of the Fock matrix.

screen-refresh-period: Variable, Read/Write.  
 Type: integer in range (1 .. 32767).  
 How frequently to update system on the screen.

script-menu-caption: Variable, Read/Write.  
 Type: vector of string.  
 Caption for menu button.

script-menu-checked: Variable, Read/Write.  
 Type: vector of boolean.  
 If checked.

script-menu-command: Variable, Read/Write.  
 Type: vector of string.  
 Command for menu button.

script-menu-enabled: Variable, Read/Write.  
 Type: vector of boolean.  
 If greyed.

script-menu-help-file: Variable, Read/Write.  
 Type: vector of string.  
 Help file for menu item.

script-menu-help-id: Variable, Read/Write.  
 Type: vector of integer.  
 Context id for help on button.

script-menu-in-use: Variable, Read/Write.  
 Type: vector of boolean.  
 Is somebody claiming this menu item?

script-menu-message: Variable, Read/Write.  
 Type: vector of string.  
 Status message.

script-refs-in-errors: Variable, Read/Write.  
 Type: boolean.  
 Whether to include script file line numbers in errors.

select-atom: Command.  
 Arg list: integer, integer.  
 Selects atom int-1 in molecule int2. Honors selection-target.

select-name: Command.  
 Arg list: string.  
 String-1 specifies the name of a selection to become the current selection.

select-none: Command.  
 Arg list: (void).  
 Unselect all atoms.

select-path: Command.  
 Arg list: integer, integer, integer, integer.  
 Selects atoms in path from int-1 in molecule int-2 to int-3 in molecule int-4.

select-residue: Command.  
 Arg list: integer, integer.  
 Selects residue int-1 in molecule int-2, disregarding selection-target.

select-sphere: Variable, Read/Write.  
 Type: boolean.  
 Whether double-button dragging selects in a sphere or in a rectangle.

selected-atom: Variable, Readonly.  
 Type: vector of integer, integer.  
 (i) The atom and molecule indices of the i-th selected atom.

selected-atom-count: Variable, Readonly.



Type: integer.  
The number of selected atoms.

selection-color: Variable, Read/Write.  
Type: enum(ThickLine, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
How to display the selection.

selection-target: Variable, Read/Write.  
Type: enum(Molecules, Residues, Atoms).  
The type of target for selection operations.

selection-value: Variable, Readonly.  
Type: float.  
The value of the current selection (bond length, angle, etc).

semi-empirical-method: Variable, Read/Write.  
Type: enum(ExtendedHuckel, CNDO, INDO, MINDO3, MNDO, AM1, PM3, ZINDO1, ZINDO5, MND0d, TNDO).  
The type of semi-empirical computation to perform.

serial-number: Variable, Readonly.  
Type: string.  
The serial number of this copy of HyperChem, read-only.

set-atom-charge: Command.  
Arg list: float in range (-100 .. 100).  
Float-1 provides the charge for the currently selected atom(s).

set-atom-charges-to-formal: Command.  
Arg list: (void).  
Sets all atom charges to the value of formal charge.

set-atom-type: Command.  
Arg list: string.  
String-1 provides the type for the currently selected atom(s).

set-bond: Command.  
Arg list: integer, integer, integer, integer, enum().  
Set bond between (iat1, imol1) and (iat2, imol2) to be bond type.

set-bond-angle: Command.  
Arg list: float angle in range (0 .. 180).  
Set the bond angle for the current selection.

set-bond-length: Command.  
Arg list: float in range (0 .. 3200).  
Set the bond length for the current selection.

set-bond-torsion: Command.  
Arg list: float angle in range (-360 .. 360).  
Set the torsion angle for the current selection.

set-velocity: Command.  
Arg list: .  
Set the velocity for the selected atoms.

show-aromatic-rings-as-circles: Variable, Read/Write.  
Type: boolean.  
Whether aromatic rings are shown with circles or dotted bonds

show-axes: Variable, Read/Write.  
Type: boolean.  
Whether to display inertial axes.

show-dipoles: Variable, Read/Write.  
Type: boolean.  
Whether to display dipole.

show-gradient-vectors: Variable, Read/Write.  
Type: boolean.  
Show atomic gradient vectors

show-hydrogen-bonds: Variable, Read/Write.  
Type: boolean.  
Whether hydrogen bonds are displayed.

show-hydrogens: Variable, Read/Write.  
 Type: boolean.  
 Whether Hydrogens are displayed.

show-isosurface: Variable, Read/Write.  
 Type: boolean.  
 Whether an isosurface should be displayed, if one is available.

show-multiple-bonds: Variable, Read/Write.  
 Type: boolean.  
 Whether multiple bonds are drawn with multiple lines.

show-periodic-box: Variable, Read/Write.  
 Type: boolean.  
 Whether the periodic box is displayed when it exists.

show-perspective: Variable, Read/Write.  
 Type: boolean.  
 Whether the system should be displayed in perspective.

show-ribbons: Variable, Read/Write.  
 Type: boolean.  
 Whether the system should be displayed with ribbons.

show-stereo: Variable, Read/Write.  
 Type: boolean.  
 Whether the system should be displayed as a stereo pair.

show-stereochem-wedges: Variable, Read/Write.  
 Type: boolean.  
 Whether stereochemistry constraints get displayed on the screen.

show-vibrational-vectors: Variable, Read/Write.  
 Type: boolean.  
 Whether or not to display per-atom vibrational vectors.

small-density-cutoff: Variable, Read/Write.  
 Type: float in range (0 .. 0.001).  
 value of density used as cutoff value

solvate-system: Command.  
 Arg list: (void).  
 Solvate current system using default box size.

solvate-system-in-this-box: Command.  
 Arg list: float, float, float.  
 The three Float args give the size of the box for solvation.

source-refs-in-errors: Command.  
 Arg list: (void).  
 Controls presentation of filename, line number in error messages.

spheres-highlighted: Variable, Read/Write.  
 Type: boolean.  
 CPK overlapping spheres should be highlighted when shaded.

spheres-shaded: Variable, Read/Write.  
 Type: boolean.  
 CPK overlapping spheres should be shaded.

start-logging: Command.  
 Arg list: string, boolean.  
 Begin append of logging computation results to file String-1.

state-to-optimize: Variable, Read/Write.  
 Type: enum(SCF, MP2, CI).  
 Which energy to use for conjugate directions.

status-message: Variable, Read/Write.  
 Type: string.  
 The text of the last status message.

sticks-width: Variable, Read/Write.  
 Type: integer in range (0 .. 25).  
 Sticks rendering width in pixels.

stop-logging: Command.  
Arg list: (void).  
Don't log computation results any more.

stretch-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.

switch-to-user-menu: Command.  
Arg list: (void).  
Change menu to the user customized menu.

tndo-version: Variable, Read/Write.  
Type: integer in range (1 .. 2).  
1, 2, etc. as versions of TNDO

toggle: Command.  
Arg list: string.  
Invert value of boolean variable.

torsion-energy: Variable, Readonly.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.

total-energy: Variable, Read/Write.  
Type: float in range (-1e+010 .. 1e+010).  
Results from backend computation.

translate-merged-systems: Variable, Read/Write.  
Type: boolean.  
Should newly merged/pasted molecules be translated off to one side?

translate-selection: Command.  
Arg list: float, float, float.  
Translate the selection by (dx, dy, dz)

translate-view: Command.  
Arg list: float, float, float.  
Translate the view by (dx, dy, dz).

translate-whole-molecules: Variable, Read/Write.  
Type: boolean.  
Select translation of entire molecule.

uhf: Variable, Read/Write.  
Type: boolean.  
Perform UHF(true) or RHF(false) calculation.

un-select-atom: Command.  
Arg list: integer, integer.  
Un-selects atom int-1 in molecule int2. Honors selection-target.

un-select-residue: Command.  
Arg list: integer, integer.  
Un-selects residue int-1 in molecule int-2, disregarding selection-target.

unconstrain-bond-angle: Command.  
Arg list: (void).  
Remove any angle constraint for the three currently selected atoms.

unconstrain-bond-length: Command.  
Arg list: (void).  
Remove any constraints on two currently selected atoms.

unconstrain-bond-torsion: Command.  
Arg list: (void).  
Remove any torsion constraint for the four currently selected atoms.

use-auxilliary-basis: Variable, Read/Write.  
Type: boolean.  
As opposed to numerical integration

use-electric-field: Command.  
Arg list: unregistered.  
use an applied electric field

use-fast-translation: Variable, Read/Write.  
     Type: boolean.  
     Use bitmap for XY translations.

use-magnetic-field: Command.  
     Arg list: unregistered.  
     use an applied magnetic field

use-mullikan-approximation-to-density: Variable, Read/Write.  
     Type: boolean.  
     speed things up in dft only

use-no-restraints: Command.  
     Arg list: (void).  
     Ignore all restraints.

use-parameter-set: Command.  
     Arg list: string.  
     Use parameter set string1 for current molecular mechanics methods.

use-restraint: Command.  
     Arg list: string, boolean.  
     (selection-name, if-use)

use-small-density-cutoff: Variable, Read/Write.  
     Type: boolean.  
     minimize points in numerical integration

uv-band-count: Variable, Read/Write.  
     Type: integer.  
     The total number of uv bands.

uv-dipole-components: Variable, Read/Write.  
     Type: vector of float-list.  
     (i) The components of the dipole moment for the i-th state.

uv-energy: Variable, Read/Write.  
     Type: vector of float.  
     (i) The energy of the i-th uv band.

uv-oscillator-strength: Variable, Read/Write.  
     Type: vector of float.  
     (i) For the current state.

uv-spin: Variable, Read/Write.  
     Type: vector of float.  
     (i) The total spin of the i-th state.

uv-total-dipole: Variable, Read/Write.  
     Type: vector of float.  
     (i) The total dipole of the i-th excited state.

uv-transition-dipole: Variable, Read/Write.  
     Type: vector of float, float, float.  
     (i) The components of the transition dipole for the i-th state.

variable-changed: Command.  
     Arg list: string.  
     Declare that the named variable has changed.

velocities: Variable, Read/Write.  
     Type: array of float, float, float.  
     (iat, imol) The x, y, and z velocity components of atom iat in molecule imol.

velocities-in-hin-file: Variable, Read/Write.  
     Type: boolean.  
     Should velocities be written into a hin file?

version: Variable, Readonly.  
     Type: string.  
     The version number of HyperChem, read-only.

vibrational-mode: Variable, Read/Write.  
     Type: integer.  
     The index of the current normal mode.

view-in-hin-file: Variable, Read/Write.  
Type: boolean.  
Should view be written into a hin file? (useful for comparing hin files)

wall-eyed-stereo: Variable, Read/Write.  
Type: boolean.  
Wall eyed stereo.

warning: Variable, Read/Write.  
Type: string.  
The current warning.

warning-type: Variable, Read/Write.  
Type: enum(none, log, message).  
Destination for warning messages.

warnings-are-not-omsgs: Command.  
Arg list: (void).  
Specifies that warning messages are to appear in message boxes.

warnings-are-omsgs: Command.  
Arg list: (void).  
Specifies that warning messages should be treated like o-msgs.

window-color: Variable, Read/Write.  
Type: enum(Monochrome, Black, Blue, Green, Cyan, Red, Violet, Yellow, White).  
The background color for the window.

window-height: Variable, Read/Write.  
Type: integer.  
Height of the HyperChem window in pixels.

window-width: Variable, Read/Write.  
Type: integer.  
Width of the HyperChem window in pixels.

workspace-height: Variable, Read/Write.  
Type: integer.  
Height of the workspace in pixels.

workspace-width: Variable, Read/Write.  
Type: integer.  
Width of the workspace in pixels.

write-atom-map: Command.  
Arg list: string.  
Writes a mapping of backend atom numbers to HyperChem (atom, molecule) pairs.

write-file: Command.  
Arg list: string.  
String-l names the file into which the current system should be written.

x-y-rotation-cursor: Variable, Read/Write.  
Type: float angle in range (0 .. 3600).  
Select X-Y axis rotation cursor tool.

x-y-rotation-icon-step: Variable, Read/Write.  
Type: float angle in range (0 .. 3600).  
Select X-Y axis rotation step.

x-y-translation-icon-step: Variable, Read/Write.  
Type: float in range (0 .. 1000).  
Select X-Y translation step.

z-rotation-cursor: Variable, Read/Write.  
Type: float angle in range (0 .. 3600).  
Select Z axis rotation cursor tool.

z-rotation-icon-step: Variable, Read/Write.  
Type: float angle in range (0 .. 3600).  
Select Z axis rotation step.

z-translation-cursor: Variable, Read/Write.  
Type: float in range (0 .. 1000).

Select Z axis translation cursor tool.  
z-translation-icon-step: Variable, Read/Write.  
Type: float in range (0 .. 1000).  
Select Z translation step.  
zindo-1-pi-pi: Variable, Read/Write.  
Type: float in range (0 .. 2).  
Overlap weighting factors.  
zindo-1-sigma-sigma: Variable, Read/Write.  
Type: float in range (0 .. 2).  
Overlap weighting factors.  
zindo-s-pi-pi: Variable, Read/Write.  
Type: float in range (0 .. 2).  
Overlap weighting factors.  
zindo-s-sigma-sigma: Variable, Read/Write.  
Type: float in range (0 .. 2).  
Overlap weighting factors.  
zoom: Command.  
Arg list: float in range (0.01 .. 50).  
Set the magnification.  
zoom-cursor: Variable, Read/Write.  
Type: float in range (1 .. 1000).  
Select zoom cursor tool.  
zoom-icon-step: Variable, Read/Write.  
Type: float in range (1 .. 1000).  
Select zoom step.



# Chapter 7

## Type 2 (Tcl/Tk) Scripts

### Introduction

This chapter describes a new scripting language for HyperChem that appears for the first time in Release 5.0. This language is called the *Tool Command Language* or just Tcl (“tickle”) for short. Tcl was developed by Professor John Ousterhout and his students at the University of California, Berkeley and placed in the public domain. Hypercube has build the language directly into the HyperChem product. Tcl is a general purpose, but relatively simple, interpreted scripting language. It is *embeddable*. That is, one can extend the core Tcl language with additional commands; Hypercube has imbedded the full *HyperChem Command Language* (Hcl) into Tcl. This makes it an extremely powerful language for molecular modeling. A HyperChem script can now be made to do almost anything you wish with a relative modest amount of programming.

Since Tcl is imbeddable, it commonly comes with an extra set of commands that allow it to be used for building graphical user interfaces (GUI’s). The extra module is called the *Toolkit* (Tk) and the combination is called Tcl/Tk. Within HyperChem it might be called Tcl/Tk/Hcl but we simply refer to it as Tcl/Tk (or just Tcl when we are not concerned with graphical elements).

Because the Tcl/Tk/Hcl combination is so powerful it is now possible to write whole graphical applications as simple scripts. They can be written quickly and they can be debugged quickly since Tcl is an interpreted language. One might first think that a scripting language is just a way of customizing HyperChem but there are many situations where Tcl can be used to write completely new applications, reusing whatever functionality HyperChem conveniently can provide. In other situations Tcl is a convenient glue language for interfacing HyperChem to your own applications, possibly written in C, C++, or Fortran. HyperChem contains many capabilities that you do not need or want to replicate, but can just use - via this new scripting capability.



## Elements of Tcl

This manual cannot be a tutorial or reference for Tcl! The language is extensive and competent Tcl programmers must obtain the appropriate reference books and materials as with any other programming language. Nevertheless, with a few ideas described here and the examples that are provided, you should be able to write your first Tcl/Tk program and can begin to extend HyperChem, in some useful way, for your own purposes. Appendix B contains a very brief outline of the Tcl commands.

The obvious way to start learning the language is through books on the subject or through user groups and other material on the Internet and World Wide Web.

### Books

The obvious book to use is the one by Ousterhout himself. It is the oldest, however, and may not be up to date compared with the latest releases of the software.

- John K. Ousterhout, *Tcl and the Tk Toolkit*, 1994, Addison-Wesley, Reading, Mass., ISBN 0-201-6337-X.
- Brent Welch, *Practical Programming in Tcl and Tk*, Prentice Hall, 1994.
- Eric F. Johnson, *Graphical Applications with Tcl & Tk*, 1996, M&T Books, New York, N.Y., ISBN 1-55851-471-6.

### Internet

Tcl resources are also readily available on the internet. The Usenet newsgroup comp.lang.tcl may be useful to you. Some of the WWW sites where Tcl resources are available are:

- <http://www.sunlabs.com:80/research/tcl/>
- <http://www.sco.com/Technology/tcl/Tcl.html>
- <http://www.elf.org/tcltk-man-html/contents.html>

## What is Tcl?

The Tcl language is an interpreted scripting language that, after you are familiar with it, is really quite simple and easy to use. However, to chemists and those with limited programming experience outside C and Fortran, it may seem somewhat strange to begin with. The language may seem to you to be very oriented toward strings rather than numbers, but this is perhaps one of its strengths.

### Commands and Arguments

Tcl scripts (\*.TCL) consist of a sequence of Tcl commands. Each Tcl command consists of the name of the command followed by its arguments:

```
command <argument1> <argument2> ...
```

Every command returns a *string*. For example, the command,

```
incr $x 3
```

increments the *value* of x by three and returns the string corresponding to the new value..

### Variables and Values

Tcl allows you to store values in *variables* and use values in commands. For example, the command,

```
set x 3.5
```

stores the value 3.5 in the variable x. The value of x is the string, “3.5”, obtained by using the syntax, \$x. To perform arithmetic like operations, one must use a command, *expr*, that concatenates its arguments into a single string, evaluates it as an arithmetic expression, and converts the numerical result back into a string to return it. Thus,

```
expr $x * 3
```

returns the string, “10.5”.

### Command Substitution

This allows you to use the result of one command (a string) as an argument in another command. Thus,

```
set y [expr $x + 0.5]
```

is the way to set y to the value of 4.0, or in general the way to perform arithmetic. The square brackets invoke command substitution, i.e. everything

inside the square brackets is evaluated as a separate Tcl script and the result of the script is substituted in place of the bracketed command.

## Procedures and Control Structures

It is possible to write Procedures in Tcl. For example, a procedure called `power` which when passed `x` and `n` computes  $x^n$  is as follows:

```
proc power {x n} {
    set result 1
    while {$n > 0} {
        set result [expr $result * x]
        set n [expr $n - 1]
    }
    return $result
}
```

The braces, `{ }`, are like the double quotes that you place around words that have spaces in them to make them into single words - “two words”, for example. Braces nest, i.e. the last argument of the `proc` command starts after the open brace on the first line and contains everything up to the close brace on the last line. No substitutions occur within braces - all the characters between the two braces are passed verbatim to `proc` as its third argument.

## Tk

The Tk commands that are added to Tcl have to do with creating and using a set of widgets that appear in a window that appears whenever you invoke a Tcl script (unless you place the Tcl command, `TclOnly`, at the start of the script). The widgets include Labels, Text Boxes, Buttons, Menus, Frames, Scales, Radio Buttons, Check Boxes, List Boxes, etc. A widget is created by a command as follows:

```
<widget-type> <widget-name> <argument1> ...
```

The widget name traditionally starts with a “dot”. For example, a button is created as follows:

```
button .b -text “Push Me” -command { <action> }
```

This button with the text “Push Me” on it will appear in the GUI and will cause the Tcl command `<action>` to be executed when it is pushed.

A necessary Tk command in a Tcl script is the one which lays out the widgets. This pack command,

```
pack .b
```

lays out each of its widget arguments in the window, in the order in which the arguments are named. By default a vertical ordering is used.

## Hcl Embedding

Finally, Type 2 or Tcl/Tk scripts can all execute any of the HyperChem script commands in the HyperChem Script language (Hcl). Within a Tcl script these Hcl commands are divided into two types:

- Executable command - hcExec
- HSV Query - hcQuery

### hcExec

The Tcl command for executing a Hcl command is hcExec. For example,

```
hcExec "do-molecular-dynamics"
```

```
hcExec "window-color green"
```

```
hcExec "menu-file-open"
```

```
hcExec "query-value window-color"
```

This command can be used to execute any Hcl script, including menu activations, direct commands, and any HSV read or write. However, an HSV read such as, "query-value window-color", will place the return message into a HyperChem dialog on the screen as if it was a normal Hcl script and not return anything to the Tcl script. The string that is returned from any hcExec command is always the empty string, "". If one wants a Tcl script to manipulate and use HSV results from HyperChem then the hcQuery command should be used.

### hcQuery

The Tcl command for querying HyperChem for the value of an HSV is hcQuery. For example,

```
set x [hcQuery window-color]
```

or

```
set x [hcQuery "window-color"]
```

The quotes are optional in this context. The hcQuery command is a normal Tcl call and returns a string as it should. Thus, if one queries an integer value, a string is returned that represents the integer; everything is consistent. For example, if the number of atoms in molecule 3 is requested,

```
set number_atoms [hcQuery "atom-count 3"]
```

then this value can be later used as in the following:

```
set twice_number_atoms [expr $number_atoms * 2]
```

## Examples

While we have already seen a couple of very simple Tcl scripts in earlier chapters, this section describes three examples to help you get started in programming Tcl/Tk. The best way to learn the language is to use it. After understanding these examples, you should try a few scripts yourself. Obviously, if you are to become relatively expert in this subject you will need to obtain some books on Tcl or get assistance through the Internet in understanding the full syntax and semantics of the language.

### Calculating the Number of Atoms

You have seen in earlier chapters some simple TclOnly scripts. This script is our first example of a Tcl/Tk script that uses Tk to generate a GUI. The example is very simple and is used to show the number of atoms in the HyperChem workspace. The code is very short,

```
entry .en -textvariable natoms -width 20
button .b -text "Calculate Atoms" -width 20 -command {
  set natoms [hcQuery "atom-count"]
}
pack .b .en
```

This script has a GUI with two widgets. The first is a text entry widget, referred to in Tcl/Tk as an *entry*. It can be used to enter text or, as in this case, display text. We are going to use it to display the number of atoms in the HyperChem workspace. The text in an entry widget can be bound to a Tcl variable such that the text that appears always represent the current value of

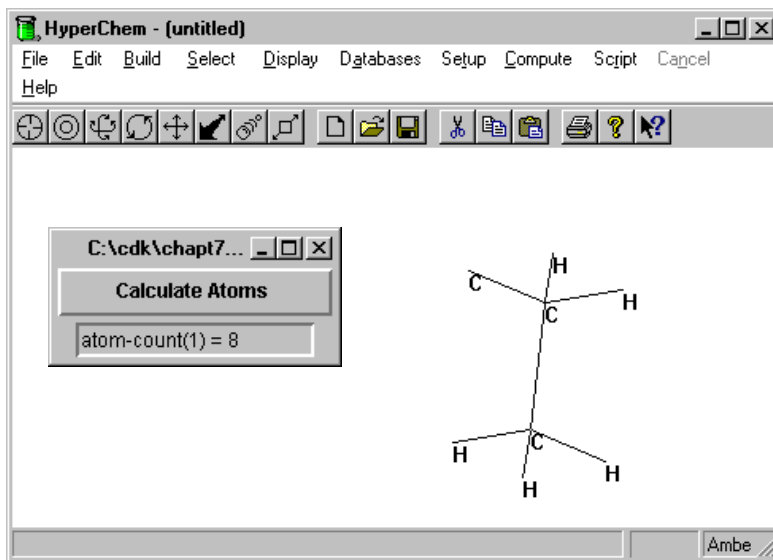
the variable. This is what we have done here; we have bound the variable *natoms* to the entry widget.

The next widget is just a button widget as we have discussed above. When the button is pushed, it executes the Tcl command,

```
set natoms [hcQuery atom-count]
```

which assigns a value to *natoms* that is the result of the query to HyperChem for the value of the HSV, *atom-count* (the elimination of quotation marks around *atom-count* is deliberate to show that they are optional). Now, *atom-count* is a vector and since we have left off the index describing the molecule number, i.e. the specific component of the vector, all vector components will be returned. If there are two molecules in the workspace, we will get two components returned, etc. Because *natoms* is bound to the entry widget, the value returned from pushing the button will appear in the entry widget.

Finally, the two widgets must be packed, i.e. they must be laid out on the window. The packing mentions the button first and then the entry so that this is the order in which they appear in the window. The default layout scheme is to have them arranged vertically. The result of pushing the button is thus:



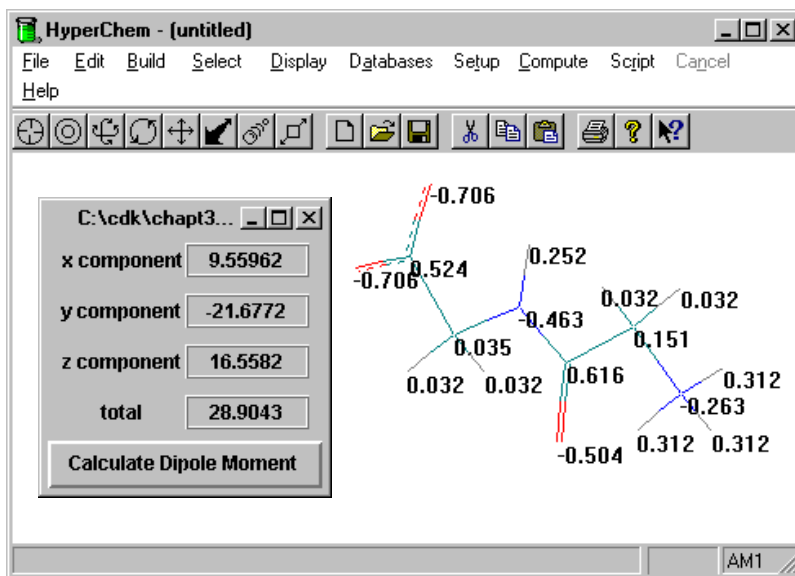
The title at the top of the Tcl/Tk window is the title of the script that has been executed. Only one molecule is present so only the first component of the vector is returned. If the Hcl script command, *query-response-has-tag false*,

had been executed in the Tcl script prior to the query for atom-count, then the Tcl/Tk entry in the window would only show “8” as the tag, *atom-count(1)=*, would have been eliminated.

To elaborate on this kind of script, you simply need to add more widgets and perform more complicated actions when GUI elements like buttons are pressed.

## Calculating a Dipole Moment

This example Tcl/Tk script can be used to calculate a dipole moment for any situation where there are atomic charges. The script is available on the HyperChem CD-ROM as *dipole.tcl*. The script makes the assumption that the dipole moment can be calculated from the partial atomic charges only. These are the charges you see when you ask for charge labels in HyperChem. This script would perhaps be useful for molecular mechanics where HyperChem currently does not calculate a dipole moment. If one uses the following Tcl script to calculate a dipole moment from the Amber template charges of the glycine dipeptide zwitterion, one gets,



The Tcl script for this calculation will now be described. The first part of the script is associated with setting up the eight label widgets for the text describing the four values and for the four values themselves.

## Labels

We are going to use labels to describe everything in this window apart from the button. The labels need to be laid out both horizontally and vertically. The easiest way to do this is to place labels horizontally into a frame and then arrange the frames vertically. We use a simple flat label for the text describing which dipole quantity we are talking about and a label with a relief ridge around it for the showing the value. Thus, the first frame is just:

```
frame .f1
label .f1.l1 -text "x component" -width 20
label .f1.l2 -textvariable xdipole -width 20 -relief ridge
pack .f1.l1 .f1.l2 -side left
```

The label `.l1`, for example, is a component of the frame `.f1` as indicated by the notation `.f1.l1`. The packing is done horizontally starting at the left of the window. This first frame is for the variable that we have called `xdipole`, the x-component of the dipole moment. The other three frames are identical:

```
frame .f2
label .f2.l1 -text "y component" -width 20
label .f2.l2 -textvariable ydipole -width 20 -relief ridge
pack .f2.l1 .f2.l2 -side left
```

```
frame .f3
label .f3.l1 -text "z component" -width 20
label .f3.l2 -textvariable zdipole -width 20 -relief ridge
pack .f3.l1 .f3.l2 -side left
```

```
frame .f4
label .f4.l1 -text "total" -width 20
label .f4.l2 -textvariable totdipole -width 20 -relief
ridge
pack .f4.l1 .f4.l2 -side left
```



Finally, we pack the four frames from top to bottom. We will use a slightly modified call to pack here, called the “configure” option which allows more formatting which, in this case, we use to add a little more padding between the frames.

```
pack configure .f1 .f2 .f3 .f4 -pady 10
```

## Button

The next part of the code describes the button and the action taken when the button is pressed. The button is called .b and the action is everything between the opening brace on the first line and the closing brace on the last line. The first thing that is done is to execute a Hcl command to see that the value we want comes back without a tag that would interfere with its direct use in Tcl commands. We then initialize the component values of the dipole moment we are going to calculate, and initialize the loop counter i. We then get the count of the number of atoms in molecule 1 for the loop over atoms. The script we are using here is limited to calculating only the dipole moment of molecule 1. It is an exercise for the reader to extend this to calculate the dipole moment of the whole molecular system when it has more than one molecule in it. The loop is then entered.

```
button .b -text "Calculate Dipole Moment" -command {
    hcExec "query-response-has-tag false"
    set xdipole 0
    set ydipole 0
    set zdipole 0
    set i 0
    set natoms [hcQuery "atom-count 1"]
    while { $i < $natoms } {
        incr i 1
```

The principal task inside the loop is to get the coordinates and charge of each atom and to multiple them together, accumulating them as we go through the loop. The charge is straight-forward as the first line below indicates. However, the coordinates of an atom come back as a string representing the three x,y,z components separated by commas. This string, referred to here as sz, must be parsed to extract the individual components which are placed in arrays x, y, and z. The parsing is accomplished with string commands that extract the length of a string (string length), look for particular substrings such as “,” starting at the beginning of the string (string first), extract a substring

out of a string (string range), search a string from the end to the beginning looking for a substring (string last), or trim leading blanks off of a string (string trimleft).

```
set charge($i) [hcQuery "atom-charge($i,1)"]
set sz [hcQuery "coordinates($i,1)"]
set sz_length [string length $sz]
set comma1 [string first "," $sz]
set x($i) [string range $sz 0 [expr $comma1 - 1] ]
set x($i) [string trimleft $x($i)]
set comma2 [string last "," $sz]
set y($i) [string range $sz [expr $comma1 + 1] [expr $comma2 - 1] ]
set y($i) [string trimleft $y($i)]
set z($i) [string range $sz [expr $comma2 + 1] $sz_length ]
set z($i) [string trimleft $z($i)]
```

This parsing may seem a bit difficult but this is about the worst that it ever gets. Finally, the charges and coordinates are multiplied and accumulated.

```
set xdipole [expr $xdipole + $charge($i) * $x($i)]
set ydipole [expr $ydipole + $charge($i) * $y($i)]
set zdipole [expr $zdipole + $charge($i) * $z($i)]
}
```

Once outside the loop, the dipole moment components are converted to Debyes and then the total is computed from the components.

```
set factor 4.8033
set xdipole [expr $xdipole * $factor]
set ydipole [expr $ydipole * $factor]
set zdipole [expr $zdipole * $factor]
```

```
set totdipole [expr $xdipole * $xdipole]
set totdipole [expr $totdipole + $ydipole * $ydipole]
set totdipole [expr $totdipole + $zdipole * $zdipole]
set totdipole [expr sqrt($totdipole)]
```

The last thing is to end the button code and to pack the button below the labels.

```
}
pack .b
```

This completes the calculation of the dipole moment. Additional Tcl scripts are available from the HyperChem CD-ROM.

# Chapter 8

## DDE Interface to HyperChem

### Introduction

This chapter describes the basics of a *lower-level* Dynamic Data Exchange (DDE) interface to HyperChem and illustrates it with interactions between Microsoft Word or Microsoft Excel and HyperChem.

### DDE versus HAPI

The DDE interface is referred to here as lower-level because Hypercube's new CDK has introduced a *higher-level* interface that sits on top of DDE and is referred to as the HyperChem Application Programming Interface (HAPI). The HAPI way to interface programs to HyperChem is fully described in Chapter 11. One point about making HAPI calls rather than DDE calls is that it has abstracted away from any machine or operating system dependency on DDE and can be used in a UNIX environment, etc. Any investment in HAPI is preserved since HAPI will be re-implemented by Hypercube if new lower levels of interprocess and interprocessor messaging replaces DDE.

### Use of DDE in Windows Applications

The DDE interface is more strictly *operating system dependent* than it really is very low-level. We will illustrate it here with a word processor and a spreadsheet. The next chapter illustrates its use within Visual Basic programs. These DDE interfaces are actually relatively high-level interfaces and are available for multiple word processors and spreadsheets. The DDE interface is very standard for Windows programs and many manufacturers have adopted it. It is almost non-existent outside the Microsoft Windows environment, however. Nevertheless, almost any Windows program will have a DDE interface and HyperChem can have conversations and exchange data with a large number of other standard Windows programs. We illustrate the CDK technology using DDE interactions between HyperChem and Word, Excel or

Visual Basic, because these are very prominent programs of their class not because identical interfaces could not be demonstrated for other word processors, spreadsheets, etc.

There are indications that Microsoft might eventually replace DDE with something referred to as OLE Automation. Essentially identical considerations would really apply to this new approach compared with the discussions given below for DDE. Nevertheless, a newer version of HyperChem might some day be needed to replace HyperChem's current use of DDE with OLE Automation. There are no indications that Microsoft will abandon its use of DDE anytime in the near future.

## Basic Properties of DDE

Dynamic Data Exchange is a mechanism for interprocess communication in Windows and NT. Two separate programs carry on a DDE conversation by sending messages to each other. The two programs are referred to as a client and a server. A DDE server generally has data or services that may be of interest to another program. A DDE client wishes to obtain such data or services. In our context, HyperChem is a DDE Server and makes its data and services available to other programs. A client such as Microsoft Word or Excel, or your program, initiates a communication with the HyperChem Server, exchanges data with it, and sends it messages to control its actions.

Two programs that engage in DDE conversations with each other need not be specifically coded to work with each other. A DDE Server, such as HyperChem, will publicly define its messaging protocol so that all clients know in advance how to obtain services from the server.

A protocol begins with a server defining the name of its *application*, a *topic* for the conversation, and an *item* that defines the exchange. In most cases the item is a piece of data to be read from the server. The client initiates the conversation by using the application and topic to establish a communication link. In our case, the application is *HyperChem*, the topic is always *System*, and the item is the name of an HSV. In addition, commands can be sent to the server and this is the way menu activations, direct commands and HSV writes are performed.

## DDE Message Types

Each DDE interaction or message passed between client and server is a member of a small number of types. The relevant message type for messages sent from your client program to the HyperChem server are as follows:

## **DDE\_INITIATE**

This message, broadcast by a client, requests a conversation with an application on a topic. HyperChem will respond if the application is “HyperChem” and the topic is “System”.

## **DDE\_EXECUTE**

This is the message sent by a client to HyperChem to have it execute a piece of text corresponding to a script command. Thus the content of such a message might be, “window-color green” or “do-molecular-dynamics” to cause the background HyperChem window to turn green or to initiate a molecular dynamics trajectory.

## **DDE\_REQUEST**

This message type is the traditional client request for a piece of data. The item named in the request is the data being requested, such as “window-color”, “scf-binding-energy”, etc.

## **DDE\_ADVISE**

This message type is sent when the client is requesting a hot link for an item. This means that the server will return an HSV automatically on the hot link whenever the data representing the HSV changes within HyperChem.

These ideas can be illustrate with simple Excel and Word macros or, as in the next chapter, by simple Visual Basic Programs.

# **DDE Interface to Microsoft Word**

Microsoft Word and other Windows word processing programs can talk DDE to other Windows programs. These conversations can be used to make a document come to life with illustrations, etc. For example, a tutorial on some aspect of chemistry could be written in Microsoft Word. The manuscript could contain buttons which the reader could push as they were reading about a topic. These would cause a HyperChem window on the same screen to perform calculations on, or illustrations of, the molecular systems being described in the manuscript.

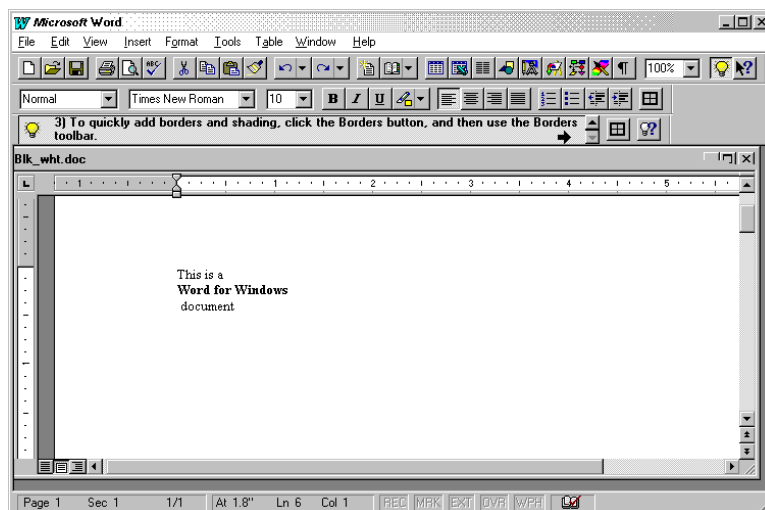
In this and following sections we will describe the topic of DDE communication with a very trivial illustration that just changes the color of the Hyper-

Chem screen. This example has very little chemical meaning but it is *visual* and easily seen when you try it. It is characteristic of hundreds of other available HSV's that do have chemical meaning and it is an example of an HSV that is both readable and writable. You should be able to extrapolate from this trivial example, that illustrates the basic idea, to examples that perform more meaningful chemistry in line with your own teaching or research interests.

## Red and Green Example

The following screen shows a Microsoft Word document.

42. Bring up a copy of Microsoft Word and type a few lines of text into it.



What we are going to do is add two buttons to the text that you can click on to effect actions in HyperChem. Let us first of all create two Word Macros. Word comes with a dialect of Word Basic that you can use to customize Word and write macros. These macros are then associated with a particular template file (\* .dot). When that template is in use, these macros are available.

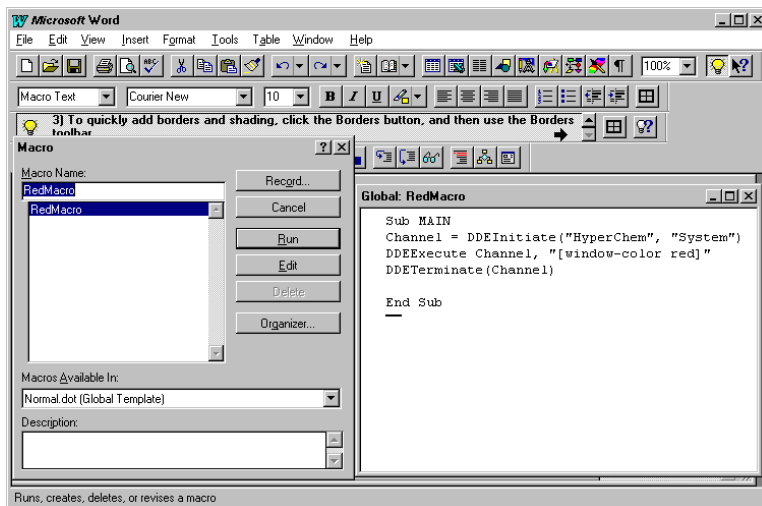
43. Select the <Tools/Macro...> menu item and create a macro with the name, RedMacro. That is, type RedMacro in the text box, "Macro Name", and choose a convenient template file to use for "Macros Available In:", and then push the "Create" button.

44. Add the following Word Basic code to create the macro,

```
Channel = DDEInitiate("HyperChem", "System")
```

```
DDEExecute Channel, "[window-color red]"
DDETerminate(Channel)
```

You should see a screen such as the following.

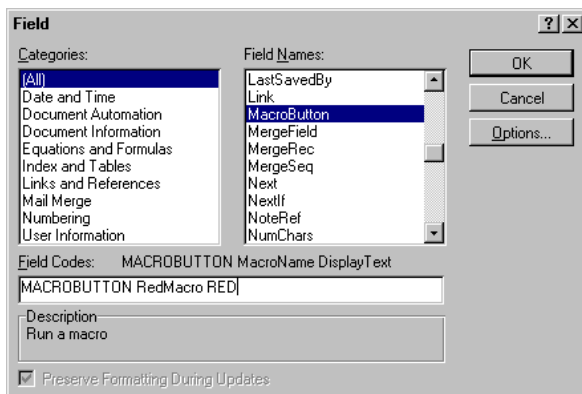


The macro code simply initiates communication with the application, HyperChem, on the topic of System and then sends a DDE\_EXECUTE message to HyperChem containing the text that corresponds to a HyperChem script command, which here sets the background window color to red.

Next you need to implement something in the Word document that triggers the Macro. Assuming the text cursor in the document is where you want it,

45. Select the menu item <Insert/Field...> to bring up the following dialog box.

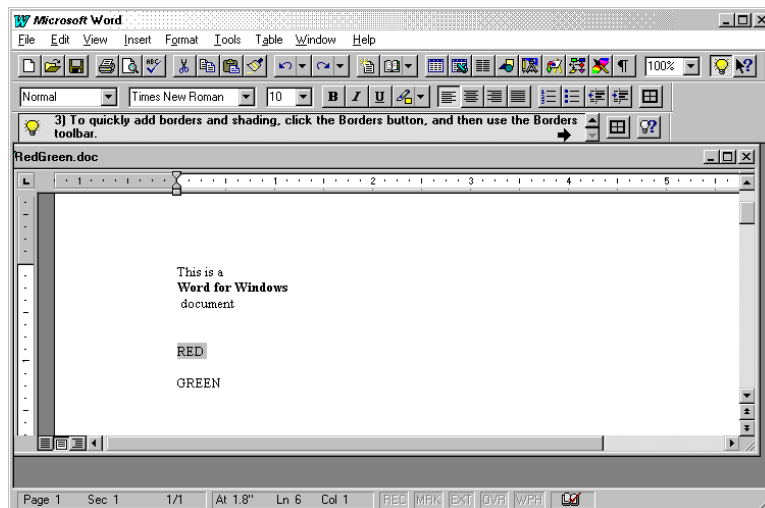




46. Select the “Field Name” corresponding to *MacroButton* as shown and then type arguments *RedMacro* and some arbitrary text, such as *RED*, into the bottom text entry box, as shown, prior to hitting OK.

47. Repeat the whole process for a “Green button”.

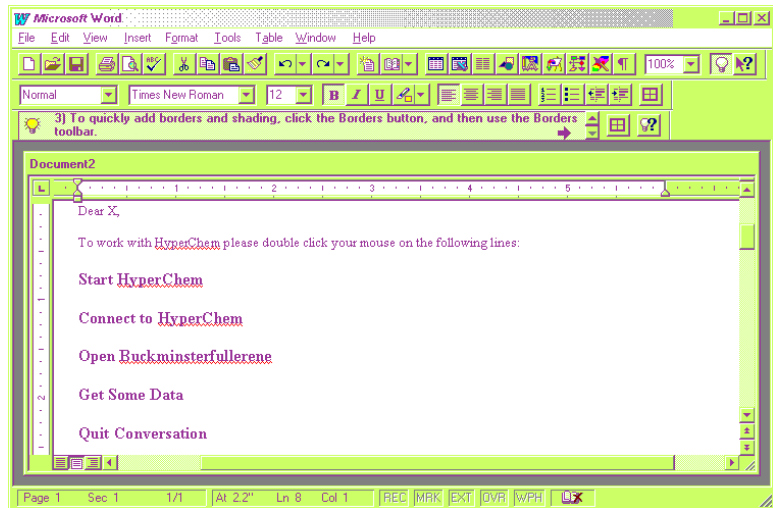
You should now see something resembling the following.



If you now bring up HyperChem and double click on either of the two “buttons”, i.e. the text RED or the text GREEN inside the Word document you will see these color changes in HyperChem.

## Extended Example

The first example of Word Basic above was about as simple as one can get. Word Basic has more programming capability than that example showed. A slightly more elaborate example can show some of this programming capability. The following five macros that we are about to describe were generated exactly as above and correspond to the large bold-faced type of the document shown below:



The five macros used here are:

### ActivateHC

```

Sub MAIN
If (AppIsRunning("HyperChem") = 0) Then
LineDown
Insert "Starting HyperChem ..." + Chr$(13)
Shell "c:\hyper\chem.exe", 0
LineUp
EndIf
End Sub

```

Each of these macros has a MAIN subroutine. You may use additional subroutines below, or above, the MAIN subroutine. This macro starts HyperChem on the assumption that it is in the directory `c:\hyper`. Your copy

may not be in this same location. The Insert command puts text at the current cursor and LineUp and LineDown move the cursor.

### ConnectHC:

```
Sub MAIN
channel = DDEInitiate("HyperChem", "System")
If (channel <> 0) Then
c$ = Str$(channel)
SetDocumentVar "channel", c$
LineDown
Insert "Document <-> HyperChem connection is ready ..."
Else
LineDown
Insert "Cannot connect to HyperChem !!!"
EndIf
End Sub
```

The above macro starts HyperChem and stores the communication channel in a variable that can be accessed by other macros.

### ExecuteCmd

```
Sub MAIN
c$ = GetDocumentVar$("channel")
channel = Val(c$)
name$="c:\hyper\c60.hin"
AppMaximize "HyperChem", 1
DDEExecute channel, "open-file " + name$
DDEExecute channel, "align-viewer z"
DDEExecute channel, "align-molecule primary, x, tertiary, z"
DDEExecute channel, "menu-display-scale-to-fit"
DDEExecute channel, "zoom 1.4"
DDEExecute channel, "menu-edit-copy-image"
LineDown
EditPaste
End Sub
```

The above macro is an example of one that opens a file containing Buckminsterfullerene (C<sub>60</sub>), manipulates it in HyperChem and then copies and pastes it into the document.

### GetData

```
Sub MAIN
c$ = GetDocumentVar$("channel")
```

```

channel = Val(c$)
atom_count$ = DDERequest$(channel, "atom-count")
atomic_symbol$=DDERequest$(channel, "atomic-symbol")
LineDown
Insert atom_count$
Insert atomic_symbol$
End Sub

```

The above macro first retrieves the text representation of the HyperChem vector variable, *atom-count*, which represents the number of atoms in each molecule. The second query retrieves the atomic symbols for the atoms in the workspace (C for Carbon, in this case).

### DisconnectHC

```

Sub MAIN
c$ = GetDocumentVar$("channel")
channel = Val(c$)
DDETerminate channel
End Sub

```

This last macro terminates the connection to HyperChem.

The macros above are stored as part of a Microsoft Word TEMPLATE, not as a part of the document. So, as long as you are using the same template, you do not need to worry about macros: they are always in place. However, if you move your document to another machine, or if you want to distribute it, you must remember to move the template also. You may save a template in a DOT file of Microsoft Word.

Install the macros and try them.

## DDE Interface to Microsoft Excel

Microsoft Excel is an example of a spreadsheet that has extensive capabilities for DDE conversations with HyperChem. You can, for example, set up tables of molecules and automate the computation of molecular properties so that they show up in tables within the spreadsheet. The capability for using a spreadsheet in conjunction with HyperChem will only be very briefly be illustrated here, as another example of the richness of the CDK and the open architecture of HyperChem. This capability has been there since HyperChem's inception and has been fairly commonly exploited so it will not be dwelled on here.

With Release 5 and the CDK, the spreadsheet capability has been enhanced because of the addition of enhanced scripting, i.e. now Tcl/Tk scripts in addi-

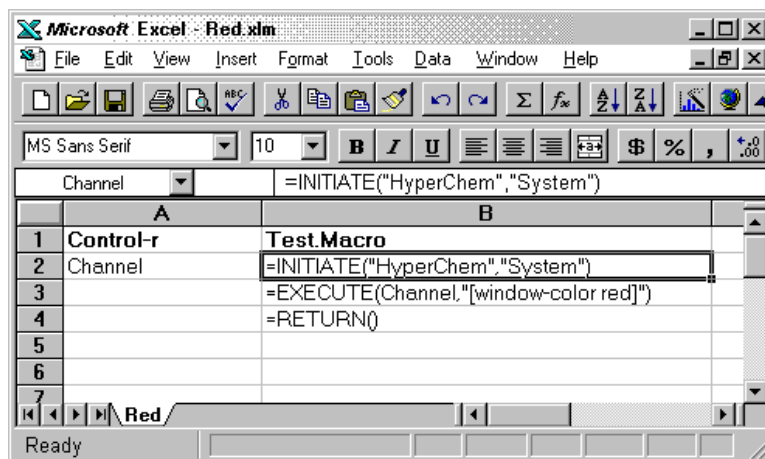
tion to Hcl scripts as with earlier versions of HyperChem. However, Excel macros have considerable programming and GUI capability so that if one prefers to program within a product like Excel itself, it might be that the enhanced functionality of HyperChem along with more Hcl scripts is more important to you than the addition of Tcl/Tk.

We emphasize Microsoft Excel here as our spreadsheet example but others spreadsheets such as Lotus 123, Quatro Pro, etc. also have DDE capability and could be used to communicate with and exchange data with HyperChem. The examples here are restricted to Excel, however.

We will illustrate the interface of HyperChem and Microsoft Excel using the same example as we began with in describing Microsoft Word, i.e. the trivial example that sets an HSV (window-color).

## Red (and Green)

A very simple Excel Macro is the following:



Running this macro, RED.XLM, will change the color of the HyperChem window to red. It can be run by simple hitting Ctrl-r on the keyboard (provided Excel is running) or from the <Tools/Macro...> menu item. As in earlier examples, we simply initiate a conversation with the *application*, HyperChem, on the *topic*, System. We then execute a DDE command that is the Hcl script command, *window-color red*. A similar macro could change the color to green but with a different keyboard accelerator such as Ctrl-g. Then, alter-

nately hitting Ctrl-r and Ctrl-g would flash the HyperChem window from red to green to red..., etc.

The macro DDE commands, EXECUTE and REQUEST are fundamentally all you need to communicate with HyperChem. Other macro programming is needed, of course, to deal with the data coming from or going to HyperChem.

## **Additional Macros**

The Excel macro language and Excel itself provide a powerful capability in combination with HyperChem but will not be described in detail here. The HyperChem Getting Started and Reference manual provide additional detail associated with the interaction between HyperChem and Excel and examples \*.XLM files have been distributed with HyperChem since its inception. The ChemPlus product includes the code for an extensive Excel macro that makes 3D Ramachandran-like plots of the energy of a molecule versus two independent structural variables. You should also check the Hypercube WWW site (<http://www.hyper.com>) where Excel macros, along with scripts, are made available to users of HyperChem.



# Chapter 9

## DDE and Visual Basic

### Introduction

This chapter describes the DDE interface between HyperChem and Microsoft's Visual Basic (VB). VB is chosen because it provides a very fast way of building extensions to HyperChem. While VB is not the only tool in its class, it is certainly one of the better ones. Some of the prejudice against Basic as a serious programming language remains, among scientific programmers and others. VB, however, is a serious modern tool that can allow you to very quickly put together applications, particularly applications involving graphical user interfaces, in a few hours - applications that used to take weeks or longer. For the demonstrations of the chapter, we will use Visual Basic 4.0 although earlier releases are also appropriate to use.

### VB for GUIs or Computation

While one can certainly write whole applications in Visual Basic, its object-oriented character and uniqueness are illustrated best when quickly putting together a graphical user interface(GUI) rather than scientific number crunching types of code. A potential compromise is to build dynamic link libraries (DLL's) with C, C++, or Fortran and just have the GUI built in Visual Basic. A VB program can call a DLL, and you can choose the best of both worlds by rapidly prototyping a user interface in VB and using legacy code in Fortran, for example.

### VB with DDE or HAPI Calls

Since VB can call a DLL, it can call the DLL that defines the HyperChem Application Programming Interface (HAPI.DLL). The subject of the HAPI library, its calls, and how to use this method of interfacing to HyperChem is the subject of Chapter 11 and the subsequent chapters which give examples



of the HAPI library being used in various contexts. Chapter 11 includes a brief example of using HAPI calls with VB.

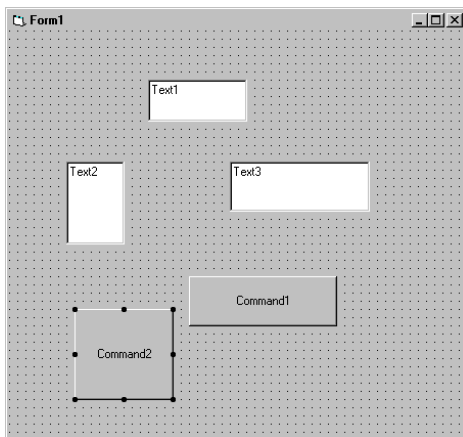
This chapter will focus on the low-level DDE interface to HyperChem for VB programs but we will also discuss very briefly the alternative HAPI way to build an interface between HyperChem and VB.

## Red and Green

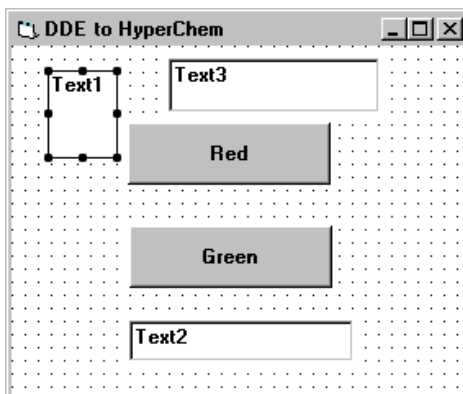
Our first example of a Visual Basic program communicating with HyperChem is, by now, our familiar and trivial example of a program that changes the background color of a HyperChem window. This particular example, however, also monitors the current color and displays the color, whether it is changed by the VB program or through the <Preferences> dialog box in HyperChem.

### Basic Form and Controls

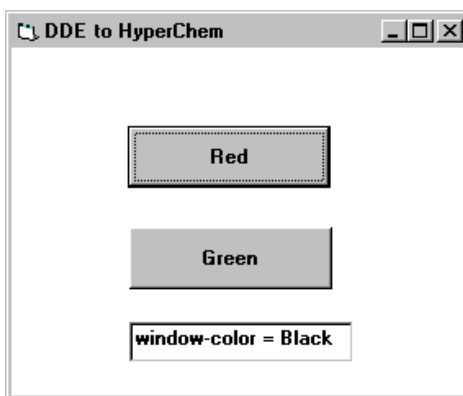
The VB project file for this example is called `REDGREEN.VBP` and is on the HyperChem CD-ROM. It was created by just opening Visual Basic and adding to the main form, three text boxes, `Text1`, `Text2`, and `Text3` plus two command buttons, `Command1` and `Command2`.



It is possible to lay out these linkcontrols and start from scratch or you can just read `REDGREEN.VBP` from the HyperChem installation directories,



What we have done here is to change the caption of Form1, the main form, to “DDE to HyperChem” and the caption of the two command buttons to “Red” and “Green”. We have also positioned the controls that have the Visible property set to “true”, i.e. Text2, Command1, and Command2. We have also changed the background color of Form1 to white and shrunk it a bit. A running version of the program might look as follows:



## Start Up (Load)

The most important code in this example is that which on start up establishes the basic link with HyperChem and retrieves the initial color of the window (which will be restored on exit). The following is the code that is executed when Form1 loads,

```
Private Sub Form_Load()  
If Text1.LinkMode = 0 Then  
    Text1.LinkTopic = "HyperChem|System"  
    Text1.LinkMode = 2  
End If  
  
    Text1.LinkItem = "window-color"  
    Text1.LinkRequest  
  
    Text2.Text = Text1.Text  
  
If Text3.LinkMode = 0 Then  
    Text3.LinkTopic = "HyperChem|System"  
    Text3.LinkItem = "window-color"  
    Text3.LinkMode = 1  
End If  
  
End Sub
```

With Visual Basic, a communication channel to HyperChem becomes associated with a control such as one of the text boxes on the form, i.e. Text1 or Text3. Once such a channel is set up, text being passed back and forth between the VB application and HyperChem is stored in VB, as the text field associated with the control, i.e. as Text1.Text or Text3.Text. These text fields are where Hcl script commands are placed prior to their execution and where any HSV messages coming from HyperChem end up.

## A Cold Link Request

The LinkMode field for a text object describes the type of channel that is established or exists with another application (DDE server) like HyperChem. A value of 0 indicates no link or channel. Setting the value to 2 establishes a cold link for the channel such that values will be returned from HyperChem only when requested. The first operation on start up is thus to use the Text1 object to establish a cold link with HyperChem on the generic topic, "System". Subsequently, a message request is made of HyperChem for the HSV, window-color. The return message from HyperChem containing the value of the HSV, *window-color*, is automatically placed into the text field of the Text1 object, i.e. as Text1.Text. The Text1 object has the visible property set to false so that it is not visible on the running application. Another text box, Text2, is used to display the value of the current window color. The appropriate operation above is to just pass Text1.Text into Text2.Text so that it is shown to the user.

## A Hot Link

The Text3 object in this example is used to establish a “hot link” to HyperChem, associated with LinkMode = 1. This means that whenever HyperChem detects that *window-color* changes its value, from whatever source, HyperChem will notify whoever is listening of the new value. Setting LinkMode equal to 1 establishes this hot link to HyperChem requesting a desire to listen to the LinkItem.

The value returned to the VB application by HyperChem goes into the text field of Text3 and this field is automatically updated whenever the value changes in HyperChem. This placing of the new value into Text3.Text can be detected using the routine, Text3\_Change(). The code for this routine in our example is,

```
Private Sub Text3_Change()  
Text2.Text = Text3.Text  
End Sub
```

This just takes the current value of the color and places it into the Text2 object so that it is visible to the user of the VB application as part of the VB window (form).

## Execute

Next, we look at the code behind the two buttons that change the color of the HyperChem window to Red or Green when they are “clicked”. The code for the red command button is,

```
Private Sub Command1_Click()  
Text1.LinkExecute ("window-color red")  
Text2.Text = "window-color = Red"  
End Sub
```

This code, simply sends a Hcl script command message to HyperChem. Here it changes the window color to red but it could be any script command from the HyperChem Command Language (Hcl) except that it should not be a query of an HSV (Use LinkRequest for these). In addition, the color displayed locally in Text2 is updated.

## Unload

The final portion of this example is the code that is executed on exit from the VB application. This is,

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Text1.LinkExecute (Text1.Text)  
End Sub
```

This code return the color of the HyperChem window to its original color prior to executing the VB application. This color is still stored in Text1.Text as per the original request on loading of the form.

## A HAPI Interface to VB

The Visual Basic example above showed how to make DDE calls to HyperChem, via an object such as a text box that provides the DDE channel or link. It is very object oriented but not an obvious way to do things for conventional C and Fortran programmers. Nor is it immediately obvious how to transfer binary data in this fashion.

The HAPI interface for VB consists of a set of straight-forward calls (either for text or binary data) that can be implemented in ordinary Basic code. The calls, such as *hbExecBin* which executes a binary form of a Hcl command, all begin with “hb”. Thus *hbExecTxt* is the text equivalent of the above. These and all the other HAPI calls that refer to Hcl script commands take as their first argument a long integer, defined in the file HSV .BAS. This integer maps to one of the many Hcl menu activations, direct commands, or HSV read/writes. For these calls, the name of the long integer is the normal Hcl name, e.g. *window-color*, but with hyphens replaced by underscores. That is, the relevant VB name to use for HAPI calls is *window\_color*. Thus,

```
Dim Value As Double  
Dim Result as Long  
Value = 3.5  
Result = hbSetReal( dipole_moment, value, 8)
```

is the code which would constitute a binary write of the value 3.5 to the HyperChem dipole moment. The integer variable “dipole\_moment” is defined in HSV .BAS. This call is really targetted at a corresponding C language routine, *hcSetReal*, contained in the dynamic link library, HAPI .DLL. The file HAPI .BAS makes the appropriate declaration to tie *hbSetReal* to this DLL and should be included in all your Visual Basic applications that wish to use the HyperChem API.

Thus, any of the HAPI calls of Chapter 11 or Appendix C can be used in Visual Basic programs. All that is necessary is to include two files, HSV .BAS and HAPI .BAS, in your project and to make calls to routines labelled hb... rather than to the hc... calls of the C/C++ language or to the hf... calls of For-

tran. Appendix C gives the details of each HAPI call including how to declare and use them from Visual Basic.



# Chapter 10

## External Tcl/Tk Interface

### Introduction

Hypercube has built a Tcl/Tk interpreter directly into HyperChem Release 5. This interface derives from Version 7.5 of Tcl and Version 4.1 of Tk. This *internal* interpreter is probably the most convenient way to use Tcl/Tk in conjunction with HyperChem, i.e. by either opening a \*.tcl file or by executing a Hcl script command, *read-tcl-script*. There is, however, an alternative way to use Tcl/Tk that is called the *external* interface.

The external use of Tcl/Tk implies that you use a Tcl/Tk program or interpreter that is completely separate from HyperChem and may even be of a different version than the one used in HyperChem. This external Tcl/Tk interpreter can then be augmented in a standard way to add the HyperChem Command Language (Hcl) as an embedded extension. The external Tcl/Tk program communicates ultimately with HyperChem via Dynamic Data Exchange (DDE). But, as a user, you need only to make “HAPI calls” just as with the internal interpreter. That is, hcExec and hcQuery are still how you invoke Hcl commands with external Tcl/Tk.

### Why External?

The interpreter for Tcl/Tk which is built into HyperChem offers an extremely powerful extension to the HyperChem Command Language. However there are certain situations where you might prefer external Tcl/Tk access to HyperChem. Such situations arise when:

1. You want to connect to HyperChem from another complex application already containing a Tcl/Tk interpreter as an extension to that application.



2. You want your Tcl program to react to changes in HyperChem via HSV notifications. Notifications cannot be easily defined with internal Tcl/Tk scripting.
3. A new version of Tcl/Tk becomes available and it is not yet incorporated into HyperChem. The new version of Tcl/Tk has enhancements that are necessary to you.
4. The internal implementation of Tcl/Tk does not perform correctly for some non-regular scripts or you encounter a devastating bug.

Hypercube has implemented a Tcl/Tk extension “package” (called the THAPI package) that you can load into a standard Tcl/Tk interpreter and use to communicate with HyperChem through regular HAPI calls. This chapter describes this extension.

## Invoking External Tcl/Tk

The external copy of Tcl/Tk is invoked in Microsoft Windows by executing the interpreter program, a Windows Shell called Wish. Associated with Version 4.1 of Tk, this program is called *wish41* and is included in your program directory along with your executable of HyperChem.



Wish41.exe

Executing *wish41.exe* gives a Console window, shown below after executing “?” to see all the possible Tcl/Tk commands.

```

Console
?
ambiguous command name "?": . Exit TclOnly after append array auto_execok auto_l
oad auto_mkindex auto_reset bell bind bindtags break button canvas case catch cd
checkbutton clipboard clock close concat console continue destroy entry eof err
or eval exec exit expr fblocked fconfigure file fileevent flush focus for foreach
h format frame gets glob global grab grid history if image incr info interp join
label lappend lindex linsert list listbox llength load lower lrange lreplace ls
earch lsort menu menubutton message open option pack package pid pkg_mkIndex pla
ce proc puts pwd radiobutton raise read regexp regsub rename return scale scan s
crollbar seek selection set socket source split string subst switch tclPkgSetup
tclPkgUnknown tell text time tk tkButtonDown tkButtonEnter tkButtonInvoke tkButt
onLeave tkButtonUp tkCancelRepeat tkCheckRadioInvoke tkEntryAutoScan tkEntryBack
space tkEntryButton1 tkEntryClipboardKeySyms tkEntryClosestGap tkEntryInsert tkE
ntryKeySelect tkEntryMouseSelect tkEntryPaste tkEntrySeeInsert tkEntrySetCursor
tkEntryTranspose tkFirstMenu tkListBoxAutoScan tkListBoxBeginExtend tkListBoxBeg
inSelect tkListBoxBeginToggle tkListBoxCancel tkListBoxDataExtend tkListBoxExten
dUpDown tkListBoxMotion tkListBoxSelectAll tkListBoxUpDown tkMbButtonUp tkMbEnte
r tkMbLeave tkMbMotion tkMbPost tkMenuButtonDown tkMenuEscape tkMenuFind tkMenuF
indName tkMenuFirstEntry tkMenuInvoke tkMenuLeave tkMenuLeftRight tkMenuMotion t
kMenuNextEntry tkMenuUnpost tkPostOverPoint tkSaveGrabInfo tkScaleActivate tkSca
leButton2Down tkScaleButtonDown tkScaleControlPress tkScaleDrag tkScaleEndDrag t
kScaleIncrement tkScreenChanged tkScrollButton2Down tkScrollButtonDown tkScrollB
uttonUp tkScrollByPages tkScrollByUnits tkScrollDrag tkScrollEndDrag tkScrollSel
ect tkScrollStartDrag tkScrollToPos tkScrollTopBottom tkTextAutoScan tkTextButto
n1 tkTextClipboardKeySyms tkTextClosestGap tkTextInsert tkTextKeyExtend tkTextKe
ySelect tkTextNextPara tkTextPaste tkTextPrevPara tkTextResetAnchor tkTextScroll
Pages tkTextSelectTo tkTextSetCursor tkTextTranspose tkTextUpDownLine tkTraverse
ToMenu tkTraverseWithinMenu tk_popup tk_textCopy tk_textCut tk_textPaste tkwait
toplevel trace unknown unset unsupported0 update uplevel upvar vwait while winfo
vnr
%

```

In addition to this console window, into which you type Tcl/Tk commands, you obtain another window where Tk widgets get placed after you request them from the Console window.

## The THAPI package

It is possible to add new functions, called packages, to the external Tcl/Tk interpreter without having to recompile Tcl/Tk. The set of commands that can be added to an external copy of Tcl/Tk are part of the Tcl Hyperchem Application Programming Interface (THAPI). There are 12 commands in total and they are all contained in a Dynamic Link Library (DLL) called THAPI.DLL. These commands are essentially a subset of the HAPI calls from HAPI.DLL.

To obtain these new commands in Tcl/Tk you use the Tcl *load* command giving it the file name of the appropriate package DLL. Thus, to augment Tcl/Tk and embed all the HyperChem API calls, you should just execute the following Tcl command in the Console window,

```
load thapi
```

If the THAPI.DLL is not available in the current path nor in Windows directory, you must specify a full path to the file. Remember that Tcl comes from the UNIX world and you must type a slash, “/”, rather than a back slash. “\” as the file folder separator in the Console window.

## Commands

THAPI, based on the full HAPI interface of the last chapters of this book, defines a dozen new Tcl/Tk commands that enable you to call the most important features of the HyperChem Application Programming Interface from your Tcl/Tk external program. Further details on these commands is available in conjunction with a description of the HAPI calls in Appendix C. THAPI is a subset of HAPI.

The THAPI commands divide up into commands associated with connecting to HyperChem, the execution of Hcl commands, a utility copy command, HSV notifications, time-outs, and error processing.

The THAPI commands, which are case sensitive, are:

### **hcConnect <instance>**

This command connects a Tcl/Tk program to HyperChem so that the other THAPI commands can be executed. It must be called before any other THAPI commands and after the command loading THAPI. The argument is optional but can be used to connect to a specific instance of HyperChem when multiple instances exist simultaneously.

### **hcDisconnect**

This command disconnects the Tcl/Tk program from HyperChem.

### **hcExec hcl\_script\_command**

This command passes its argument to HyperChem as a normal Hcl script command. The argument may need to be enclosed in quotes if it itself has arguments.

### **hcQuery hsv**

This command queries HyperChem for the value of an HSV and returns it as a string

### **hcCopy source\_file desination\_file**

This command copies a file.

**hcNotifyStart hsv**

This command requests a notification of the HSV corresponding to the argument. If the HSV changes in HyperChem its new value will be sent to the Tcl/Tk script. The new value can be made available to the script via `hcGetNotifyData`. No call-back routines are available in scripts so that the execution of `hcGetNotifyData` must be periodically scheduled via a Tcl command like *after* (see the monitor example).

**hcNotifyStop hsv**

This command requests that the notification of changes in an HSV be terminated.

**hcGetNotifyData notification\_data**

This command will place the result of the first notification from an internal buffer into the argument, *notification\_data*. The command returns the HSV corresponding to the original notification (or NULL) so that you can check whether a notification has happened or not and whether it is the right one.

**hcSetTimeouts exec\_timeout query\_timeout rest\_timeout**

This command controls the time-out for interaction with HyperChem if the default values (65 seconds) is inappropriate. The `exec_timeout` is the time-out for `hcExec` commands, the `query_timeout` is the time-out for `hcQuery` commands, and the `rest_timeout` is the time-out for the remaining Hcl commands, such as for notifications.

**hcLastError error\_text**

This command places text describing the last error in `error_text`. It returns the following values:

<code>errNO_ERROR = 0</code>	No error
<code>errFATAL = 1</code>	Fatal error of unknown origin
<code>errNON_FATAL = 2</code>	Non fatal error of unknown origin

**hcSetErrorAction action\_flag**

This command sets the behavior flag that will be used, upon recognition of an error. The arguments are as follows:

<code>errACTION_NO = 0</code>	No action on any error
-------------------------------	------------------------

<code>errACTION_MESS_BOX = 16</code>	Display message box with error message
<code>errACTION_DISCONNECT = 32</code>	Disconnect from HyperChem
<code>errACTION_EXIT = 64</code>	Immediately exit from application
<code>errDDE_REP = 1</code>	Report low-level DDE errors
<code>errDDE_NO_REP = 2</code>	Do not report low-level DDE errors

### **hcGetErrorAction**

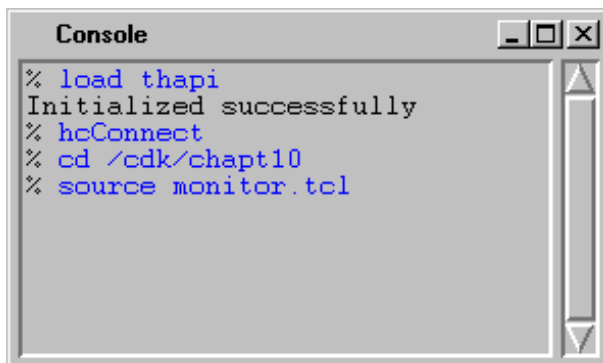
This command retrieves the behavior flag showing how errors will be acted upon. The values returned are the same as the values set by *hcSetErrorAction*.

## **A Notification Example**

Notification is a powerful capability associated with HyperChem. It allows you to request a live link to HyperChem such that you are notified of any change in an indicated variable or data structure. This allows many capabilities that would not be possible otherwise. In particular, this makes it possible to have very intimate connections between your custom capability and the “guts” of HyperChem without getting into source code details. It is not required to know the intimate details of how HyperChem operates but only that it must be operating in certain ways. Thus, for example, one would know that at the heart of optimizations are energy and gradient changes without worrying about what specific algorithm is being used by HyperChem. You can just ask to monitor these changes without having to dive into code, algorithms, etc.

Most external Tcl scripts are identical with internal Tcl scripts. A notable exception is that notifications cannot be processed by internal Tcl scripts (HyperChem does not send messages to itself!). Notifications are normally associated with external programs like Microsoft Excel, Word or Visual Basic or external applications build in C, C++, and Fortran. If you intend to build software in Tcl that requires notification, you will need to do it with external Tcl/Tk. The example of this section is one that monitors and plots values of any HSV that is of potential interest to you. A specific example would be to monitor the energy or rms gradient of a structure optimization.

The Tcl script is executed from the Console as follows:



```
Console
% load thapi
Initialized successfully
% hcConnect
% cd /cdk/chapt10
% source monitor.tcl
```

The `source` command begins the execution of `*.tcl` file as the next command. In this case it is `monitor.tcl` which sets up a Tk window that look as follows:



The Tcl code for setting up this window can be investigated by looking at the file installed from the CD-ROM but here we want to focus on the code associated with the notifications. The code behind the button, "Start New Plot" leaving out everything unessential to the notifications is,

## A Notification Example

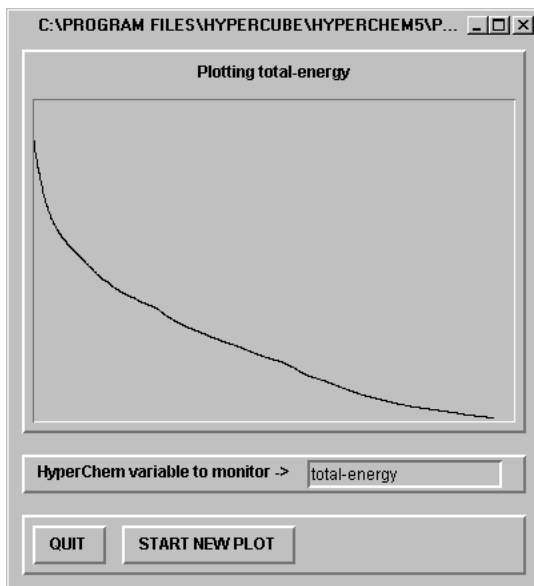
```
button .bts.plot -text "START NEW PLOT" -command {  
  
    if {$Prevhsv != -1} {  
        hcNotifyStop $Prevhsv  
    }  
    hcNotifyStart $hsv  
    set Prevhsv $hsv  
  
    if {$IsMonitor == -1} {  
        monitor  
        set IsMonitor 1  
    }  
  
}
```

When the button is pushed, any previous notifications are first cancelled. Then a new notification is requested for the new HSV which shows in the window as total-energy and which is stored in the value, \$hsv. A call is then made to a proc called monitor which will do the monitoring.

```
proc monitor {} {  
    global hsv interval  
    set name [hcGetNotifyData a]  
    if { $name == $hsv } {  
  
        <TAKE $A, THE TOTAL ENERGY POINT AND SAVE FOR PLOTTING>  
    }  
    # monitor restarts after plotting  
    after $interval [list monitor]  
}
```

The code associated with collecting and plotting the notification data has been abstracted away. The monitoring is done by executing the command, *hcGetNotifyData*., to see if there really is any notification data available, i.e. the command returns the name of the HSV being monitored. If there was a notification, it is dealt with. If not, or there was data and it has been dealt with, a call is made to re-schedule a return to monitor after \$interval milliseconds. In general, this means the process now goes to sleep and wakes up later to see if any notifications have arrived.

If you had connected the monitor example to HyperChem and then performed an optimization, the graph displayed in the TK window might look something like the following:







# Chapter 11

## The HAPI Interface to HyperChem

### Introduction

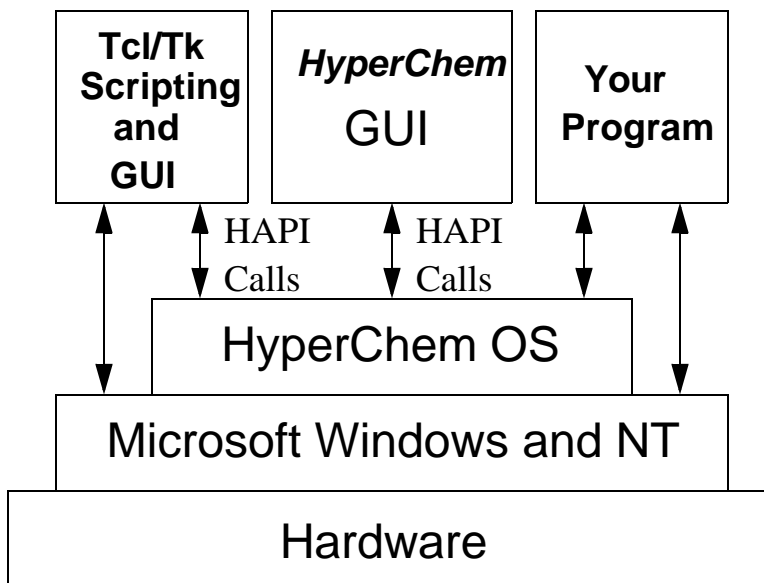
This chapter describes the HyperChem Application Programming Interface (HAPI or HyperChem API), the system library developed by Hypercube to simplify the task of communication with HyperChem. Instead of sending, posting and processing Windows DDE messages, an application makes HAPI calls and uses a set of functions provided by the API to manage the conversation between HyperChem and your external program. This is a high-level replacement for communicating via a lower-level DDE call.

The HyperChem API performs the task of packing and unpacking the data coming to and from messages so that arguments to HAPI calls can be in a form that you are familiar with and calls to the HAPI library become a simple extension to normal C or Fortran programming. The sophisticated error handling built into the HyperChem API guarantees robustness of the communication. While it is possible for applications to communicate with HyperChem without HAPI calls (as described in chapters 8 and 9, for example), it is not the recommended procedure. The HAPI library is meant for users intending to write their own sophisticated interfaces or back ends to the HyperChem front end. Moreover, some applications, such as most Fortran programs running under Windows 95 and NT, cannot easily interface to HyperChem without the HyperChem API.

Another reason for using the set of HAPI calls rather than DDE is the compatibility issue. There are indications that DDE may be replaced by another communication paradigm (OLE Automation) in future versions of Microsoft's operating systems. It thus might happen that Hypercube would need to change its underlying communication protocol from DDE to another messaging system in future releases of its HyperChem core product. Applications using the HAPI calls will preserve compatibility, while those that use a lower-level DDE protocol might lose future compatibility.

## Towards a Chemical Operating System

The richness of HyperChem, evident in its HyperChem Command Language (Hcl), its embedded Tcl/Tk interpreter, and its HAPI library of calls, is indicative of a direction Hypercube is making towards a Chemical Operating System (The HyperChem OS). A picture of this system is as follows:



In this picture we have divided HyperChem into the fundamental services (building molecules, performing calculations, visualizing results, etc.) that it supplies to its normal users (and can provide to your external program) and the graphical user interface (GUI) that normal users use to get at these services. These services are all available by making HAPI calls to the HyperChem “Operating System”.

## The Components

The components of the HyperChem Application Programming Interface are:

- hc.h - This is the header file for C and C++ interfaces that define each of the entry points for the HyperChem API.

- `hsv.h` - This header file defines an integer corresponding to each HSV and is needed if binary communication to HyperChem from C and C++ programs is to be used.
- `hload.c` - This file contains the C code for the LoadHAPI routine that C and C++ programs should call to load the HAPI calls before using them. It generally is placed after the `hc.h` and `hsv.h` files in a C/ C++ program.
- `HAPI.DLL` - This is the DLL that must get loaded to access the HyperChem API.
- `HAPI.LIB` - This is the file that could be linked with your C, C++ or Fortran application as an alternative to calling LoadHAPI.
- `hc.fi` - The Fortran equivalent of `hc.h` as an include file to define the HAPI calls of your program.
- `hsv.fi` - The Fortran equivalent of `hsv.h`. This file is a necessary include file for your Fortran program if it is to attempt binary communication with HyperChem.
- `hapi.bas` - The Visual Basic equivalent of `hc.h` as an include file to define the HAPI calls of your program.
- `hsv.bas` - The Visual Basic equivalent of `hsv.h`. This file is a necessary include file for your Visual Basic program if it is to attempt binary communication with HyperChem.

## The HAPI Calls

A complete and detailed documentation of each of the HAPI calls is given in Appendix C. Here we briefly list the calls and describe the kinds of calls that are in the library. These calls can be made from C, C++, Fortran, Visual, Basic, or external Tcl/Tk programs among others. In some situations, e.g. Tcl/Tk, the implemented set of calls is a subset of the full set.

In listing the calls below, we indicate their type and their arguments using a C-like syntax. Further details are again available in Appendix C. The calls for Fortran have the syntax, `hfHAPICALL`, the calls for Visual Basic have the syntax, `hbHAPICALL`, whereas the calls for other language situations begin with `hc` and have the syntax, `hcHAPICALL`.

## Initialization and Termination

### **BOOL hcInitAPI (void)**

This call is generally not needed in most contexts as initialization happens automatically.

### **BOOL hcConnect (LPSTR lszCmd)**

This call connects to HyperChem. A non-null string as an argument can be used to connect to a specific instance of HyperChem.

### **BOOL hcDisconnect (void)**

This call disconnects from HyperChem.

### **void hcExit(void)**

This causes immediate termination of the calling application.

### **Discussion**

The principal call of importance here is the one that connects to HyperChem. Prior to connecting, however, the HAPI library must have been loaded. See below how you should load the appropriate DLL or LIB file.

## Text-based Basic Communication Calls

### **BOOL hcExecTxt (LPSTR script\_cmd)**

This sends an Hcl script command to HyperChem.

### **LPSTR hcQueryTxt (LPSTR var\_name)**

This queries an HSV in HyperChem.

### **Discussion**

These are the text calls that send a command to HyperChem or perform a Read/Write of a HyperChem State Variable (HSV). They correspond to calls in the HyperChem Command Language (Hcl), i.e. they involve either a *menu invocation*, e.g. menu-file-open, a *direct command*, e.g. do-molecular - dynamics, an *HSV write*, e.g. window-color green, or an *HSV read*, e.g. win-

down-color ?. The reading of an HSV is performed by the call, hcQueryTxt, while the other Hcl script commands are invoked by hcExecTxt.

## Binary-based Basic Communication Calls

### **BOOL hcExecBin (int cmd, LPV args, DWORD args\_length)**

This is the binary form for sending an Hcl script command to HyperChem.

### **LPV hcQueryBin( int hsv, int indx1, int indx2, int\* length)**

This is the binary form for querying an HSV in HyperChem.

## Discussion

The HyperChem Command Language is basically a text form for communicating and exchanging data with HyperChem. For completeness and for efficiency there is an equivalent binary form for all the Hcl script commands. These can be much more effective and simpler to use in languages like Fortran where one has to use character data to invoke normal script commands. See the programming examples for examples of both text and binary based communication with HyperChem.

With text based communication you use strings, like “window-color” and “do-molecular-dynamics” to denote the HSV of interest or the direct command that you want to invoke. With binary versions of hcExec and hcQuery you use an integer to denote the operation. The name of this integer has the syntax, window\_color or do\_molecular\_dynamics, for example, where an underscore replaces the hyphen or minus sign of a Hcl text string. These integers are defined in hsv.h, hsv.fi, and hsv.bas for C and C++, Fortran, and Visual Basic applications. Only text-based communication is available for Tcl/Tk.

## Binary Format

The binary form of a HAPI call results in a binary message being sent to HyperChem rather than a simple text message. The format of these binary messages is

Length*	Binary Code	Arguments**
4 bytes	4 bytes	n-bytes

\*To make the binary message different from a text message it is required to set the highest bit of the first field to 1.

\*\* Arguments are also coded as binary data.

The binary code for each command is the 4-byte integer number found in `hsv.h`, `hsv.fi`, or `hsv.bas`. The codes may change between different versions of HyperChem. Three utility Tcl scripts are included on the HyperChem CD-ROM, called `cgenhsv.tcl`, `fgenhsv.tcl`, and `bgenhsv.tcl` that are capable of generating the correct `hsv.h`, `hsv.fi`, and `hsv.bas` files for the version of HyperChem that you are using.

To issue a binary command the user uses, for example:

```
result=hcExecBin(hsv,arg,length);
```

in a C/C++ program, or

```
result=hfExecBin(hsv,arg,length)
```

in a Fortran program.

## Binary-based Get Integer Calls

### **int hcGetInt (int hsv)**

This call gets the binary value of an integer HSV.

### **int hcGetIntVec( int hsv, int\* buff, int max\_length)**

This call gets all the binary values of an integer vector HSV into a buffer.

### **int hcGetIntArr (int hsv, int\* buff, int max\_length)**

This call gets all the binary values of an integer array HSV into a buffer.

### **int hcGetIntVecElm (int hsv, int index)**

This call gets a single binary element of an integer vector HSV.

### **int hcGetIntArrElm (int hsv, int atom\_index, int mol\_index)**

This call gets a single binary element of an integer array HSV.

## Discussion

These calls are specialized forms of `hcQueryBin` specific to integer variables, vectors, and arrays. One form gets the whole vector or array at once while the other form (Elm) gets only a single element at once.

## Binary-based Get Real Calls

### **double hcGetReal (int hsv)**

This call gets the binary value of a real HSV.

### **int hcGetRealVec( int hsv, double\* buff, int max\_length)**

This call gets all the binary values of a real vector HSV into a buffer.

### **int hcGetRealArr (int hsv, double\* buff, int max\_length)**

This call gets all the binary values of a real array HSV into a buffer.

### **double hcGetRealVecElm (int hsv, int index)**

This call gets a single binary element of a real vector HSV.

### **double hcGetRealArrElm (int hsv, int atom\_index, int mol\_index)**

This call gets a single binary element of a real array HSV.

### **int hcGetRealVecXYZ (int hsv, index, double\* x, double\* y, double\* z)**

This call gets the three real values of a single element of a real array HSV. The three real values are normally the Cartesian components of a real variable such as a dipole moment, etc.

### **int hcGetRealArrXYZ (int hsv, int atom\_index, int mol\_index, double\* x, double\* y, double\* z)**

This call gets the three real values of a single element of a real array HSV. The three real values are normally the Cartesian components of a real variable such as a coordinate, etc.



## Discussion

These calls are specialized forms of hcQueryBin specific to real variables, vectors, and arrays. One form gets the whole vector or array at once while the other form (Elm) gets only a single element at once. The XYZ forms are specific to Cartesian values.

## Binary-based Get String Calls

**int hcGetStr (int hsv, char\* buff, int max\_length)**

This call gets a string corresponding to all values of the HSV.

**int hcGetStrVecElm (int hsv, int index, char\* buff, int max\_length)**

This call gets a string corresponding to a particular element of a vector.

**int hcGetStrArrElm (int hsv, int atom\_index, int mol\_index, char\* buff, int max\_length)**

This call gets a string corresponding to a particular element of an array.

## Discussion

These calls correspond very closely to the text based calls in that they convey arguments as a string. However the HSV is represented as a binary integer not as a string.

## Binary-based Set Integer Calls

**int hcSetInt (int hsv, int value)**

This call sets the binary value of an integer HSV.

**int hcSetIntVec( int hsv, int\* buff, int length)**

This call sets length binary values of an integer vector HSV into a buffer.

**int hcSetIntArr (int hsv, int\* buff, int max\_length)**

This call sets length binary values of an integer array HSV into a buffer.

### **int hcSetIntVecElm (int hsv, int index, int value)**

This call sets a single binary element of an integer vector HSV.

### **int hcSetIntArrElm (int hsv, int atom\_index, int mol\_index, int value)**

This call sets a single binary element of an integer array HSV.

## **Discussion**

These calls are specialized forms of hcExecBin specific to writing HSV's for integer variables, vectors, and arrays. One form sets the whole vector or array at once while the other form (Elm) sets only a single element at once.

## **Binary-based Set Real Calls**

### **int hcSetReal (int hsv, double value)**

This call sets the binary value of a real HSV.

### **int hcSetRealVec( int hsv, double\* buff, int length)**

This call sets length binary values of a real vector HSV into a buffer.

### **int hcSetRealArr (int hsv, double\* buff, int length)**

This call sets length binary values of a real array HSV into a buffer.

### **int hcSetRealVecElm (int hsv, int index, double value)**

This call sets a single binary element of a real vector HSV.

### **int hcSetRealArrElm (int hsv, int atom\_index, int mol\_index, double value)**

This call sets a single binary element of a real array HSV.

### **int hcSetRealVecXYZ (int hsv, index, double x, double y, double z)**

This call sets the three real values of a single element of a real array HSV. The three real values are normally the Cartesian components of a real variable such as a dipole moment, etc.

**int hcSetRealArrXYZ (int hsv, int atom\_index, int mol\_index, double x, double y, double z)**

This call sets the three real values of a single element of a real array HSV. The three real values are normally the Cartesian components of a real variable such as a coordinate, etc.

### Discussion

These calls are specialized forms of hcExecBin specific to writing HSV's for real variables, vectors, and arrays. One form sets the whole vector or array at once while the other form (Elm) sets only a single element at once. The XYZ forms are specific to Cartesian values.

## Binary-based Set String Calls

**int hcSetStr (int hsv, char\* string)**

This call sets an HSV to the value of a string.

**int hcSetStrVecElm (int hsv, int index, char\* string)**

This call sets a particular element of an HSV vector to a string.

**int hcSetArrElm (int hsv, int atom\_index, int mol\_index, char\* string)**

This call sets a particular element of an HSV array to a string.

### Discussion

These calls correspond very closely to the text based calls in that they convey arguments as a string. However the HSV is represented as a binary integer not as a string.

## Get and Set Blocks

**int hcGetBlock (int hsv, char\* buff, int max\_length)**

This call gets all the data, irrespective of type, corresponding to an HSV.

### **int hcSetBlock (unt hsv, char\* buff, int length)**

This call sets an HSV, irrespective of its type from a buffer.

### **Discussion**

These calls do block copies of data associated with an HSV of any type.

## **Notification Calls**

### **int hcNotifyStart (LPSTR hsv)**

This call requests a notification for an HSV.

### **int hcNotifyStop (LPSTR hsv)**

This call terminates a request for notification of an HSV.

### **int hcNotifySetup (PFNB pCallback, int NotifyWithText)**

This call establishes how notifications are to be handled.

### **int hcNotifyDataAvail (void)**

This call determines whether a notification is available.

### **int hcGetNotifyData (char\* hsv, char\* buff, int max\_length)**

This call gets the data associated with a notification on an HSV.

### **Discussion**

These calls are associated with notifications.

## **Memory Allocation**

### **void \* hcAlloc (size\_t, n\_bytes)**

This allocates memory associated with the HAPI but is not recommended to replace the normal user memory allocation routines.

### **hcFree (void\* pointer)**

This frees memory allocated with hcAlloc or after processing data from hcQueryTxt and hcQueryBin which allocate memory for the result of the query.

### **Discussion**

These calls are for memory allocation and deallocation associated with the HAPI.

## **Auxiliary Calls**

### **void hcShowMessage (LPSTR message)**

This call displays a message box with the message provided.

### **void hcSetTimeouts (int ExecTimeout, int QueryTimeout, int OtherTimeout)**

This call sets timeouts for hcExec, hcQuery, and other HAPI calls.

### **int hcLastError (char\* LastErr)**

This call enquires about the last error.

### **int hcGetErrorAction (void)**

This call asks how errors are currently being handled.

### **void hcSetErrorAction (int err)**

This call sets how errors are to be handled.

### **Discussion**

Further details on these calls are available in Appendix C.

## **The HAPI Dynamic Link Library (HAPI.DLL)**

The HyperChem Application Programming Interface (HAPI), forms a set of system calls that can be utilized by an application to perform certain tasks in conjunction with HyperChem. It is somewhat analogous to the Microsoft

Windows API and is implemented in a similar way, through a dynamic Link Library (DLL).

The whole Microsoft Windows operating system is seen by a user program as an API - the Microsoft Windows 32-bit API for Windows 95 and NT. For example, this means that a user writing code in C can utilize all Microsoft's system calls that form the API. In addition to the generic Windows API there are specific API's, such as the WINSOCK API - the library that defines TCP/IP communication for Windows applications. Usually, a Microsoft API is provided to users in a form of a DLL.

A Dynamic Link Library is specific to Microsoft Windows and NT and as a library has one important feature: it can be linked to the user program at run-time. In most cases, it can also be linked while the application is loaded into memory (load time). At run-time, however, it can be loaded any time during program execution, as well as unloaded when it is no longer required. This feature is of great importance, as it provides both users and developers a painless method of improving upon both the standard Microsoft API and an application specific API.

The HyperChem API (HAPI) has been developed as a 32-bit DLL that is compatible with Microsoft Windows 95, Microsoft NT and Microsoft's W32S 32-bit subsystem for Windows 3.1. Thus, it can be used with all the programming tools that call 32-bit DLLs including almost all modern C/C++ and Fortran compilers, 32-bit versions of Microsoft Visual Basic, and tools like the Tcl/Tk interpreter, Microsoft Excel, Borland's Delphi, etc.

## **How to use the HyperChem API**

To make the HyperChem API available to your application, you must have the requisite header files, load the HAPI.DLL properly, and then make a proper connection of your program to HyperChem. Alternatively, it is possible to use a library file, HAPI.LIB, link it with your application and use load-time dynamic linking. Though it is possible for you to make use of the HyperChem API with a variety of compilers and development systems through intelligent use of these tools, inspecting our header files, etc., we believe it makes sense to illustrate the tools very explicitly with a very small variety of standard environments. Thus, we have chosen to provide standard interface source code along with the CDK so that at least for certain environments an interface to HyperChem becomes very straight-forward. We have chosen the following to illustrate the HyperChem API:

- - Microsoft 32-bit C/C++ compilers (from Visual C++ 4.0)

- - Microsoft Power Station Fortran compiler (version 4.0 and higher)
- - Microsoft Visual Basic 4.0
- - Tcl/Tk implemented as a 32-bit “wish41” Windows application

Nevertheless, if you must interface to the HyperChem API from another type of application, you can use the information in this section combined with each function’s header information to write your own code to call the API. Such a situation might happen when you want to use other tools such as Borland Delhi, for example.

## Accessing the HyperChem API from C/C++ code

### Run-Time Dynamic Linking

The method for accessing a function in the HAPI depends on the programming language used. For C/C++ programs, the easiest method is to include the provided `hclload.c` source code just after including the main header file, `hc.h`. The typical sequence is as follows:

```
#include "hc.h"  
#include "hclload.c"
```

Alternatively, instead of including `hclload.c` into each module of the larger application it may be convenient to link the application with `hclload.obj` during a static linkage phase. In this case the program only needs:

```
#include "hc.h"
```

The corresponding ‘*makefile*’ (or project workspace) would have the form:

```
#####  
#+   EXAMPLE make file   +  
#####  
  
userapp.exe : userapp.obj hclload.obj  
             $(LINKER) $(GUIFLAGS) -OUT:userapp.exe userapp.obj hclload.obj  
             $(GUILIBS)  
  
userapp.obj : userapp.c  
             $(CC) $(CFLAGS) step0.c  
  
hclload.obj : hclload.c  
             $(CC) $(CFLAGS) hclload.c
```

In the environment of the Microsoft Developer Studio, the corresponding action would be to insert `hclload.c` as another source file among the user's files, i.e. by using the menu item, <Insert/Files into Project...>.

Within your source code one of the first things to do is to dynamically load, or activate the library. Your program has to call the `LoadHAPI` function defined in `hclload.c`. A typical sequence would be:

```
/* loading HAPI.DLL */
if (!LoadHAPI("hapi.dll")) {
    MessageBox(hwnd, "Error loading HAPI.DLL",
        "Error", MB_OK | MB_ICONSTOP);
    exit(0);
}
```

The `LoadHAPI` call takes one parameter which is a file name, `HAPI.DLL`. Note that the DLL can be placed in the local directory, but the most appropriate place is the main Windows system directory (usually `C:\WINDOWS`). `LoadHAPI` activates each of the functions in the DLL by making repetitive calls to `GetProcAddress`, the function provided by Microsoft to get the address of the corresponding function in the DLL:

```
hcExecTxt=(T_hcExecTxt*)GetProcAddress(hinst, "hcExecTxt");
if ( hcExecTxt == NULL) res=_hcLoadError("hcExecTxt",szN);
```

This method utilized by `LoadHAPI` is called Run-Time Dynamic Linking.

## Load-Time Dynamic Linking

There is another method to link HAPI with a user application. It uses Load-Time Dynamic Linking and requires direct linking of the user program with the library file (`HAPI.LIB`) provided with the CDK. This file contains no code but contains all of the API function headers and proper initialization code. You can use load-time linking by inserting `HAPI.LIB` as one of the libraries into a project, i.e. by the menu command, <Insert/Files into Project>, or by writing an appropriate makefile when working outside of the Microsoft Developer Studio (using a DOS command shell for compilation).

In either cases the application still needs:

```
#include "hc.h"
```

However, with load-time linking, all the HAPI functions are available for calling as soon as the application starts. There is no need for the `LoadHAPI` call.



If you are making binary calls, it is necessary to have:

```
#include "hsv.h"
```

## Accessing the HyperChem API from Fortran code

To use the HAPI calls from Fortran programs you have to translate Fortran-style calls into the appropriate system calls to the HAPI. With Microsoft Power Station Fortran it suffices to just include the file, `hc.fi`, provided with the CDK and then statically link in `HAPI.LIB`. The code in `hc.fi` has to be included in all functions and subroutines that use HAPI calls. For example:

```
subroutine ExeTest
  character*100 cmd
  common /flags/flagfile,fconnected
  logical res,flagfile,fconnected
  include "hc.fi"

1  write(*,*)' Script command to execute (0 - returns) ->'
  read(*,'(A)')cmd
  if (cmd.eq.'0') return
  res=hfExecTxt(cmd)
  write(*,*)' hfQueryTxt returns: ',res
  goto 1

end
```

What does the source inside `hc.fi` look like and do? It provides an INTERFACE (a Fortran capability of Microsoft Fortran Power Station), so the Fortran compiler can type-check all function parameters and issue the proper calling sequences. The source looks as follows:

```
INTERFACE
C-----
C This is a Microsoft Fortran Power Station 4 Interface to CDK API
C-----
C-----
C Initialization & termination functions
C-----
  logical function hfInitAPI()
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hcInitAPI@0' :: hfInitAPI
  end function hfInitAPI
```

```

    logical function hfConnect(init_string)
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hfConnect@8' :: hfConnect
    character*(*) init_string
    end function hfConnect

    logical function hfDisconnect()
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hcDisconnect@0' :: hfDisconnect
    end function hfDisconnect

    subroutine hfExit()
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hcExit' :: hfExit
    end subroutine hfExit
C-----
C-----
C Text Query & Execute functions
C-----

    logical function hfExecTxt(script_cmd)
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hfExecTxt@8' :: hfExecTxt
    character*(*) script_cmd
!ms$ATTRIBUTES REFERENCE :: script_cmd
    end function hfExecTxt

    logical function hfQueryTxt(var_name, res)
    character*(*) var_name,res
!MS$ATTRIBUTES reference :: var_name
!MS$ATTRIBUTES reference :: res
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hfQueryTxt@16' :: hfQueryTxt
    end function hfQueryTxt

.....

C-----
    integer function hfLastError(error)
!ms$ATTRIBUTES DLLIMPORT,ALIAS: '_hcLastError@4' :: hfLastError
!ms$ATTRIBUTES C,REFERENCE :: error
    character *(*) error
    end function hfLastError

    integer function hfGetErrorAction()
!ms$ATTRIBUTES
DLLIMPORT,ALIAS: '_hcGetErrorAction@0'::hfGetErrorAction
    end function hfGetErrorAction

```

```
subroutine hfSetErrorAction(action)
!ms$ATTRIBUTES
DLLIMPORT,ALIAS: '_hcSetErrorAction@4'::hfSetErrorAction
!ms$ATTRIBUTES VALUE :: action
integer action

end subroutine hfSetErrorAction
```

```
C-----
c end of hc.fi
C-----
END INTERFACE
```

If you are making binary calls, it is also necessary to have:

```
include "hsv.fi"
```

## Accessing the HyperChem API from Visual Basic Code

While it is relatively straight forward for Visual Basic programs to communicate with HyperChem via DDE as described in Chapter 9, it may be desirable to use the HyperChem API and HAPI.DLL from Visual Basic for similar reasons to that described above for C, C++, and Fortran. It is possible to do so with the principal requirement being the need to have two files, `hapi.bas` and `hsv.bas` that define the interface to the HAPI.DLL. The following is a portion of the `hapi.bas` file,

```
Attribute VB_Name = "Module1"

Declare Function hbInitAPI Lib "hapi.dll" Alias "hcInitAPI" () As Long
Declare Function hbConnect Lib "hapi.dll" Alias "hcConnect" (ByVal
command As String) As Long
Declare Function hbDisconnect Lib "hapi.dll" Alias "hcDisconnect" ()
As Long
Declare Sub hbExit Lib "hapi.dll" Alias "hcExit" ()
Declare Function hbExecTxt Lib "hapi.dll" Alias "hcExecTxt" (ByVal
script_cmd as string) As Long

Declare Function hbExecBin Lib "hapi.dll" Alias "hcExecBin" (ByVal
cmd as Long, ByRef args as Long,
ByVal args_length as Long ) As Long
```

```

Declare Function hcQueryTxt Lib "hapi.dll" (ByVal command As String)
    As String
Declare Function hfQueryBinLib "hapi.dll" (ByVal var,indx1,
    indx2 as integer, ByVal result as integer,
    ByVal cbL as integer) As Integer

```

The Visual Basic application, DLA, which is on the HyperChem CD-ROM, is an example of a Visual Basic application that uses HAPI calls.

## Accessing the HyperChem API from Tcl/Tk code

A description of this interface is given in Chapter 10. Only text-based communication is supported. Notifications are processed by the Notification Agent.

## Considerations for Console-based Applications

A “Console application” is a new application type available for Windows 95 and Windows NT operating systems. It is an implementation of the popular text-based terminal environment well known in the UNIX world. Such a program uses a regular *main()* function as an entry point, rather than *WinMain()*. Instead of communicating with the user through a GUI, the console application simply *writes* to the screen using the regular *printf* call and *reads* using the *scanf* call. This is for C code. In Fortran the familiar *write*, *print*, and *read* calls are available.

But the real differences between GUI-based and Console-based applications are much larger than just input/output. All regular Windows application are *event-driven* applications, where all processing is performed by reacting to incoming messages or events, generated by the operating system. This makes writing Windows programs a difficult task for most traditional programmers with a so called top-down or serialized view of programming. This represents a new paradigm for programming that is not in the experience of most UNIX-style programmers.

The console interface makes the conversion of programs from UNIX to Windows easy and painless. Now, under Windows 95 and NT, almost all older UNIX C and FORTRAN programs can be compiled without problems and without severe modifications to the code. The underlying POSIX-compatible system calls layer, available for Windows NT, makes the porting to NT of a large number of “legacy” applications possible.

However, because console applications are not event-driven, Microsoft decided to insulate them from certain features of Windows. Particularly, there is no possibility for an application to be called via regular DDE callback functions. In practice, DDE communication can be used by the console application, as long as there is no need for the application *to be called* through a DDE Callback function. However, this “callback” feature is mandatory to receive notifications from HyperChem about the change in an HSV variable.

## The Notification Agent

To work around this, HyperCube, has implemented “The Notification Agent”. The agent is another thread of execution (both Win95 & NT are real multithreading applications) that does the following:

- Starts a regular Windows procedure (analogous to *WinMain*) by registering the window class and defining a *WndProc*-type of function that can process all windows messages.
- The agent then opens its own DDE communication with HyperChem.
- The agent registers and processes notifications and allocates its own buffers when necessary.
- Each notification results in buffer allocation up to the limit of memory resources.
- The notification buffers are freed when a user copies their contents into their own memory.
- The user application access the buffers which form very simple linked list using only two functions:

### From a C Program:

```
DWORD _stdcall hcNotifyDataAvail()  
  
DWORD _stdcall hcGetNotifyData(char* name, char *buffer, DWORD  
MaxBuffLength)
```

### From a Fortran program:

```
integer function hfNotifyDataAvail()  
integer function hfGetNotifyData(name,result,res_length)
```

- The first informs the caller if there is some data that has come in as a notification message. The second copies the top-most buffer into the user program, deallocating the buffer.
- The proper accessing by the notification agent and your user program of a critical section of storage (the buffer area and control area) is handled by use of semaphores, making the whole solution very robust. Tests made under both Windows 95 and NT have shown the proper handling of data without any loss of incoming messages, irrespective of their size.

The notification agent is useful also for those applications that cannot define a callback (like a regular Tcl/Tk scripting application). It is also the method used by HyperChem to avoid an event-driven programming paradigm in console applications, particularly.

## Examples of HAPI Calls

### C, C++

#### Text-based

```
int result;
result = hcExecTxt("do-molecular-dynamics");
result = hcExecTxt("menu-file-open");
result = hcExecTxt("dipole moment 2.5");
```

#### Binary-based

```
int result;
double value;
value = 2.5;
result = hcExecBin(do_molecular_dynamics);
result = hcExecBin(menu_file_open);
result = hcExecBin(dipole_moment, &value, 8);
```

### Fortran

#### Text-based

```
integer result
result = hfExecTxt('do_molecular_dynamics')
```

```
result = hfExecTxt ('menu-file-open')  
result = hfExecTxt ('dipole-moment 2.5')
```

### **Binary-based**

```
integer result  
double precision value  
value = 2.5  
result = hfExecBin (do_molecular_dynamics)  
result = hfExecBin (menu_file_open)
```

## **Visual Basic**

### **Text-based**

```
Dim result As Long  
result = hbExecTxt ("do_molecular_dynamics")  
result = hbExecTxt ("menu-file-open")  
result = hbExecTxt ("dipole-moment 2.5")
```

### **Binary-based**

```
Dim result As Long  
Dim value As Double  
value = 2.5  
result = hbExecBin (do_molecular_dynamics)  
result = hbExecBin (menu_file_open)  
result = hbExecBin (dipole_moment, value, 8)
```

# Chapter 12

## Development Using the Windows API

### Introduction

This chapter describes the development of “Standard” Windows programs that interface with HyperChem. By standard we mean that these programs are written in C and use the lower-level approach of calling the Windows Application Programming Interface (API) directly. Such programs are said to be developed with the System Developer Kit (SDK). This contrasts with the next chapter which describes development of Windows programs using C++ and the Microsoft Foundation Classes. The SDK is in some sense the *hard way* to do Windows programming but it is also the most basic and flexible way to build a Windows program and most of the commercial Windows applications you will encounter have been written in the following fashion.

### Microsoft Development Tools

The interfacing examples of these next few chapters assume that you have access to certain Windows development tools. While much of what we describe is generic to a selection of compilers and programming environments, we have specifically used Microsoft’s tools in all of our examples. Thus, this chapter and the next make use of the C and C++ compilers of Microsoft Visual C++, version 4.0, for 32-bit development. We have used this compiler for HyperChem Release 5.0. The information needed to interface your program to HyperChem is fundamentally independent of the specific Windows tools used, but there may be small changes necessary to adapt to your specific tool set if you are not using Visual C++ 4.0.

### Programming Assistance

This manual is certainly not a programming manual for Windows but you may need one. With the entering of any new area of endeavor, it is comforting to have the right information and assistance from experts. If you are new to



the type of programming illustrated in this chapter, we very strongly recommend that you get a copy of the following book,

***Programming Windows***

**Charles Petzold and Paul Yao**

**ISBN 1-55615-676-6**

**Microsoft Press, 1996**

This is a new version of the programming classic by Charles Petzold. Since Windows NT programming is very similar to that for Windows 95, it is an appropriate book for NT development as well. There is an assumption in this manual that you are probably a UNIX programmer. The Petzold book is an excellent way for you to approach programming for Windows and NT and we highly recommend it. A great deal has been written about programming for Windows and there exist other fine reference books as well.

## Language

This chapter uses the C language and the next chapter uses C++, rather than Fortran, which is generally much more familiar to scientific programmers and may be your language of choice. In Chapter 14 we describe how to write so-called *console* programs in Fortran that can be interfaced to HyperChem. However, for writing a *normal* Windows program that has a graphical user interface, visualization, etc., Fortran is somewhat problematic and C or C++ are much to be preferred for normal Windows programming. With the Chemist's Developer Kit, C, C++, and VB are the languages that we anticipate you will be using for true "Windows-like" development. Any of the four languages, C, C++, VB, and Fortran are appropriate for the "character-like" programming common to most large computational chemistry modules.

## A First Example

The first programming example will again be our "Red and Green" example which is the equivalent in this manual to the common "Hello World" programming examples that you probably have seen elsewhere.

This example is referred to as Colors and will be in its own Colors directory from the HyperChem CD-ROM. It is assumed that you have installed Visual C++ 4.0 and that you are working from a DOS box in Windows 95. The first thing that you must do, as with other programming examples from this chap-

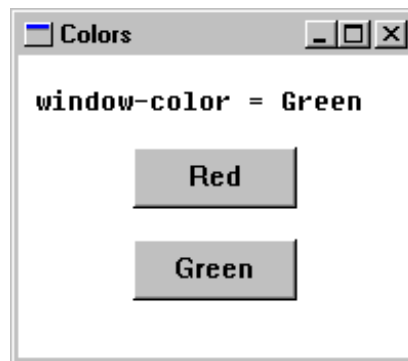
ter, is to go to the proper directory and execute MSC.BAT to configure your environment for the Microsoft compiler.

1. Change to the COLORS directory
2. Type MSC.BAT

If you have any trouble with running out of environment space, click on the Properties tool in the DOS-box tool bar to increase the amount of environment memory. You are now ready to compile the COLORS.C file and create the executable. Type NMAKE. You should see your program being compiled and the executable COLORS.EXE being created. Make sure HyperChem is running and then execute COLORS.EXE. That is,

3. Type NMAKE
4. Make sure HyperChem is running
5. Type COLORS to execute COLORS.EXE

You could do this last step by typing COLORS in your DOS box or by double clicking on the Colors icon inside the Explorer or by having HyperChem execute the script COLORS.SCR which simply has in it the Hcl script command, *execute-HyperChem-client colors.exe*. You should then see the following Windows Application on the screen:



If you push the Red button, the HyperChem background window color will turn red. If you push the Green button, it will turn green. The text above the buttons indicates the current color of the HyperChem window, if its last change came from the external application, Colors. If you change the window from the <File/Preferences...> dialog box within HyperChem, the external application will not know about the change (a notification operation is necessary for this).

The code for this Windows application is shown below. It is a completely standard “boiler-plate” windows code except for the text shown in bold.

```

/*
Window Colors - SDK program to connect and talk to HyperChem
*/
#include <windows.h>
#include "hc.h"
char cmd_line[120];
LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    PSTR szCmdLine, int iCmdShow)
{
    static char szAppName[] = "Colors" ;
    HWND        hwnd ;
    MSG         msg ;
    WNDCLASSEX  wndclass ;
    int         windowx,windowy;
    wndclass.cbSize       = sizeof (wndclass) ;
    wndclass.style        = CS_HREDRAW | CS_VREDRAW ;
    wndclass.lpfnWndProc  = WndProc ;
    wndclass.cbClsExtra   = 0 ;
    wndclass.cbWndExtra   = 0 ;
    wndclass.hInstance    = hInstance ;
    wndclass.hIcon        = LoadIcon (NULL, IDI_APPLICATION) ;
    wndclass.hCursor      = LoadCursor (NULL, IDC_ARROW) ;
    wndclass.hbrBackground = (HBRUSH) GetStockObject (WHITE_BRUSH) ;
    wndclass.lpszMenuName = NULL ;
    wndclass.lpszClassName = szAppName ;
    wndclass.hIconSm      = LoadIcon (NULL, IDI_APPLICATION) ;
    RegisterClassEx (&wndclass) ;
    lstrcpy(cmd_line,szCmdLine);
    windowx=200;
    windowy=175;
    hwnd = CreateWindow (szAppName, "Colors",
                        WS_OVERLAPPEDWINDOW,
                        CW_USEDEFAULT, CW_USEDEFAULT,
                        windowx,
                        windowy,
                        NULL, NULL, hInstance, NULL) ;
    ShowWindow (hwnd, iCmdShow) ;
    UpdateWindow (hwnd) ;
    while (GetMessage (&msg, NULL, 0, 0))
        {

```

```

        TranslateMessage (&msg) ;
        DispatchMessage (&msg) ;
    }
    return msg.wParam ;
}
LRESULT CALLBACK WndProc (HWND hwnd, UINT iMsg, WPARAM wParam,
    LPARAM lParam)
{
    static HWND  hwndButton[2] ;
    static RECT  rect ;
    static int   cxChar, cyChar ;
    HDC          hdc ;
    PAINTSTRUCT  ps ;
    TEXTMETRIC  tm ;
    static char  szBuff[200], szInitColor[50];
    char        *response;
    int         result;
    static int   connected=FALSE;
    switch (iMsg)
    {
        case WM_CREATE :
            hdc = GetDC (hwnd) ;
            SelectObject (hdc,GetStockObject(SYSTEM_FIXED_FONT));
            GetTextMetrics (hdc, &tm) ;
            cxChar = tm.tmAveCharWidth ;
            cyChar = tm.tmHeight + tm.tmExternalLeading ;
            ReleaseDC (hwnd, hdc) ;

            hwndButton[0]=CreateWindow("button","Red",
                WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
                7*cxChar, 3*cyChar, 10*cxChar, 2*cyChar,
                hwnd, (HMENU)0,
                ((LPCREATESTRUCT) lParam) -> hInstance,
                NULL) ;
            hwndButton[1]=CreateWindow("button","Green",
                WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
                7*cxChar, 6*cyChar, 10*cxChar, 2*cyChar,
                hwnd, (HMENU)1,
                ((LPCREATESTRUCT) lParam) -> hInstance,
                NULL) ;
            return 0;

        case WM_SHOWWINDOW :

```

```

    if (!connected)
    {
        /* loading CDK API DLL */
        if (!LoadHAPI("hapi.dll"))
        {
            MessageBox(hwnd,"Error loading CDK's API DLL !",
                "Error", MB_OK | MB_ICONSTOP);
            exit(0);
        }
        /* connecting to HyperChem */
        if (!hcConnect(cmd_line))
        {
            MessageBox(hwnd,"Error connecting with HyperChem !",
                "Error", MB_OK | MB_ICONSTOP);
            exit(0);
        }
        else
        {
            connected=TRUE;
        }
        /* obtaining initial window-color from HyperChem */
        response=hcQueryTxt("window-color");
        lstrcpy(szInitColor,response);
        hcFree(response);
        wsprintf(szBuff,"%s",szInitColor);
    }
    return 0;

case WM_PAINT :
    InvalidateRect (hwnd, &rect, TRUE) ;
    hdc = BeginPaint (hwnd, &ps) ;
    SelectObject(hdc,GetStockObject(SYSTEM_FIXED_FONT));
    SetBkMode (hdc, TRANSPARENT) ;
    TextOut(hdc,cxChar,cyChar,szBuff,lstrlen(szBuff)) ;
    EndPaint (hwnd, &ps) ;
    return 0 ;

case WM_DRAWITEM :
case WM_COMMAND :
    hdc = GetDC (hwnd) ;
    SelectObject(hdc,GetStockObject(SYSTEM_FIXED_FONT));
    switch (LOWORD (wParam)) {
        case 0 :

```

```

        wsprintf(szBuff,"window-color = Red");
        result=hcExecTxt(szBuff);
        break;
    case 1 :
        wsprintf(szBuff,"window-color = Green");
        result=hcExecTxt(szBuff);
        break;

    default : ;
}
/*TextOut (hdc, cxChar, cyChar,
           szBuff,lstrlen(szBuff));*/
ReleaseDC (hwnd, hdc) ;
InvalidateRect (hwnd, NULL, TRUE) ;
break ;

case WM_DESTROY :

    result=hcExecTxt(szInitColor);
    result=hcDisconnect();
    PostQuitMessage (0) ;
    return 0 ;
}
return DefWindowProc (hwnd, iMsg, wParam, lParam) ;
}

```

The changes to a standard version of the WinMain routine, as used by all Windows programs, are very minimal. The only changes are to include a header file, hc.h, to change the size of the window to a smaller than normal value, to save the invoking command line for potential use, and to change the name of the main window to Colors.

The changes to WndProc, the callback procedure, are more extensive. This is the routine that contains the code you get to execute when certain events occur. It is, of course, very application dependent although all Windows programs have a similar basic outline.

## Modification of a Molecule's Coordinates

The next example that we demonstrate and explain is one which has a very fundamental capability similar to many programs that you might wish to write. That is, this program gets a molecule and its coordinates from HyperChem, modifies these coordinates in some fashion and returns them to Hyper-

Chem for continuous display. A more elaborate example might extend this program to perform a geometry optimization, to execute a molecular dynamics, trajectory, etc. Once again the modifications from a standard Windows program are denoted in bold face type.

```

/*
 * C-API Examples*****
 * Rotation - program to demonstrate *
 * modification of coordinates in the HyperChem workspace *
*****
*/
#include <windows.h>
#include <math.h>
#include <stdio.h>
#include "hc.h"
#include "hsv.h"
char cmd_line[100];
static char *Label[] = {"Step Size",
                        "Total Steps"};
#define _CW(l) cxChar*(lstrlen(l)+1)
#define MAX_BUFF 200
typedef struct _ATM_COORDS { double x,y,z; } ATM_COORDS;
LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    PSTR szCmdLine, int iCmdShow)
{
    static char szAppName[] = "Rotate" ;
    HWND        hwnd ;
    MSG         msg ;
    WNDCLASSEX wndclass ;
    int windowx,windowy;
    wndclass.cbSize        = sizeof (wndclass) ;
    wndclass.style         = CS_HREDRAW | CS_VREDRAW ;
    wndclass.lpfWndProc    = WndProc ;
    wndclass.cbClsExtra    = 0 ;
    wndclass.cbWndExtra    = 0 ;
    wndclass.hInstance     = hInstance ;
    wndclass.hIcon         = LoadIcon (NULL, IDI_APPLICATION) ;
    wndclass.hCursor       = LoadCursor (NULL, IDC_ARROW) ;
    wndclass.hbrBackground=(HBRUSH)GetStockObject(COLOR_BACKGROUND);
    wndclass.lpszMenuName  = NULL ;
    wndclass.lpszClassName = szAppName ;
    wndclass.hIconSm       = LoadIcon (NULL, IDI_APPLICATION) ;

```

```

RegisterClassEx (&wndclass) ;
lstrcpy(cmd_line,szCmdLine);
windowx=280;
windowy=200;
hwnd = CreateWindow (szAppName, "Rotate",
                    WS_OVERLAPPEDWINDOW,
                    CW_USEDEFAULT, CW_USEDEFAULT,
                    windowx,
                    windowy,
                    NULL, NULL, hInstance, NULL) ;
ShowWindow (hwnd, iCmdShow) ;
UpdateWindow (hwnd) ;
while (GetMessage (&msg, NULL, 0, 0))
{
TranslateMessage (&msg) ;
DispatchMessage (&msg) ;
}
return msg.wParam ;
}

```

```

LRESULT CALLBACK WndProc(HWND hwnd,UINT iMsg,
                        WPARAM wParam,LPARAM lParam)
{
static HWND hwndButton[1],hwndLabel[2],hwndEdit[2] ;
static RECT rect ;
static int cxChar, cyChar ;
HDC hdc ;
TEXTMETRIC tm ;
HINSTANCE hins;
static char szBuff[MAX_BUFF],szInitColor[50];
int result,i;
static int connected=FALSE;
static ATM_COORDS *org_xyz,*new_xyz;
static int nMol,*nAtm,nAtmTot;
double dRot,rAng,rCos,rSin;
int nRot,ia;

switch (iMsg)
{
case WM_CREATE :
hdc = GetDC (hwnd) ;
SelectObject (hdc, GetStockObject (SYSTEM_FIXED_FONT)) ;
GetTextMetrics (hdc, &tm) ;

```



```
cxChar = tm.tmAveCharWidth ;
cyChar = tm.tmHeight + tm.tmExternalLeading ;
ReleaseDC (hwnd, hdc) ;
hins=((LPCREATESTRUCT) lParam) -> hInstance;

hwndLabel[0]=CreateWindow("static",
    Label[0],
    WS_CHILD | WS_VISIBLE,
    2*cxChar,
    3*cyChar,
    _CW(Label[0]),
    2*cyChar,
    hwnd,(HMENU)101, hins, NULL);
hwndEdit[0]=CreateWindow("edit",
    NULL,
    WS_CHILD | WS_VISIBLE | WS_BORDER | ES_LEFT,
    2*cxChar,
    (int)(4.5*cyChar),
    10*cxChar,
    7*cyChar/4,
    hwnd,
    (HMENU)200 ,hins, NULL);
hwndLabel[1]=CreateWindow("static",
    Label[1],
    WS_CHILD | WS_VISIBLE,
    22*cxChar,
    3*cyChar,
    _CW(Label[1]),
    2*cyChar,
    hwnd,(HMENU)102, hins, NULL);
hwndEdit[1]=CreateWindow("edit",
    NULL,
    WS_CHILD | WS_VISIBLE | WS_BORDER | ES_LEFT,
    22*cxChar,
    (int)(4.5*cyChar),
    10*cxChar,
    7*cyChar/4,
    hwnd,(HMENU)201 ,hins, NULL);
hwndButton[0]=CreateWindow("button",
    "Spin it !",
    WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
    12*cxChar,
    (int)(8.5*cyChar),
```

```

        10*cxChar,
        7*cyChar/4,
        hwnd,(HMENU)0,hins, NULL) ;

return 0;

case WM_SHOWWINDOW :
    if (!connected) {
        /* loading CDK API DLL */
        if (!LoadHAPI("hapi.dll")) {
            MessageBox(hwnd,
                "Error loading CDK's API DLL !",
                "Error", MB_OK | MB_ICONSTOP);
            exit(0);
        }
        /* connecting to HyperChem */
        if (!hcConnect(cmd_line)) {
            MessageBox(hwnd,
                "Error while connecting with HyperChem !",
                "Error", MB_OK | MB_ICONSTOP);
            exit(0);
        } else {
            connected=TRUE;
        }
        /* obtaining coordinates from HyperChem */
        nMol=hcGetInt(molecule_count); // getting number of molecules
        if (nMol < 1) {
            MessageBox(hwnd,
                "There are no molecules to play !",
                "Error", MB_OK | MB_ICONSTOP);
            hcDisconnect();
            exit(0);
        };
        nAtm=(int*)hcAlloc(nMol*sizeof(int));
        // getting vector specifying number of atoms in each
        // molecule
        if (!nAtm) {
            MessageBox(hwnd,
                "Memory allocation error",
                "Error", MB_OK | MB_ICONSTOP);
            hcDisconnect();
            exit(0);
        };
    }

```

```

    if (!hcGetIntVec(atom_count,nAtm,nMol)) {
        MessageBox(hwnd,
            "Error while getting data from HyperChem !",
            "Error", MB_OK | MB_ICONSTOP);
        hcDisconnect();
        exit(0);
    };
    nAtmTot=0;
    for (i=0;i<nMol;i++) nAtmTot += nAtm[i];
    org_xyz=(ATM_COORDS*)hcAlloc(nAtmTot*sizeof(ATM_COORDS));
    new_xyz=(ATM_COORDS*)hcAlloc(nAtmTot*sizeof(ATM_COORDS));
    if ( (!org_xyz) || (!new_xyz) ) {
        MessageBox(hwnd,
            "Memory allocation error",
            "Error", MB_OK | MB_ICONSTOP);
        hcDisconnect();
        exit(0);
    };
    if ( !hcGetRealArr(coordinates,
        (double*)org_xyz,nAtmTot*3) ) {
        MessageBox(hwnd,
            "Error getting atomic coordinates",
            "Error", MB_OK | MB_ICONSTOP);
        hcDisconnect();
        exit(0);
    };
}
SetWindowText(hwndEdit[0],"12");
// default step for rotation
SetWindowText(hwndEdit[1],"30");
// default number of rotations

return 0;

case WM_COMMAND :
    hdc = GetDC (hwnd) ;
    SelectObject (hdc, GetStockObject (SYSTEM_FIXED_FONT)) ;
    switch (LOWORD (wParam)) {
    case 0 :// ROTATE
        EnableWindow(hwndButton[0], FALSE);
        GetWindowText(hwndEdit[0],szBuff,MAX_BUFF);
        dRot=atof(szBuff);
        GetWindowText(hwndEdit[1],szBuff,MAX_BUFF);

```

```

nRot=atoi(szBuff);

dRot = dRot*3.14159256/180.0;
result=hcSetInt(cancel_menu,1);

for (i=0;i<nRot;i++) {
    rAng=dRot*(i+1);
    rSin = sin(rAng);
    rCos = cos(rAng);
    for (ia=0;ia<nAtmTot;ia++) {
        new_xyz[ia].x = rCos * org_xyz[ia].x
            + rSin * org_xyz[ia].y;
        new_xyz[ia].y =-rSin * org_xyz[ia].x
            + rCos * org_xyz[ia].y;
        new_xyz[ia].z=org_xyz[ia].z;
    }
    result=hcSetRealArr(coordinates,
        (double*)new_xyz,nAtmTot*3);
    if (!hcGetInt(cancel_menu)) {
        result=hcSetRealArr(coordinates,
            (double*)org_xyz,nAtmTot*3);
        hcDisconnect();
        PostQuitMessage(0);
        break;
    }
}
result=hcSetInt(cancel_menu,0);
EnableWindow(hwndButton[0], TRUE);

break;
default : ;
}

    ReleaseDC (hwnd, hdc) ;
    ValidateRect (hwnd, &rect) ;
    break ;

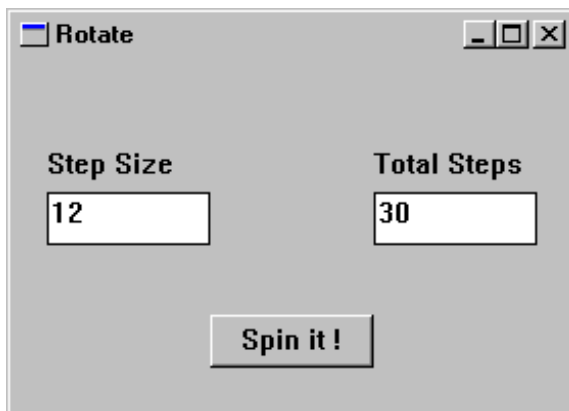
case WM_DESTROY :

    result=hcDisconnect();
    PostQuitMessage (0) ;
    return 0 ;
}
return DefWindowProc (hwnd, iMsg, wParam, lParam) ;

```

}

Executing this program (after invoking HyperChem) gives the following window:



Pushing the button will rotate the molecule by 12 degrees 30 times with the above parameters. In conjunction with this program the Cancel button functions appropriately. Pushing it will terminate the rotation and also terminate the rotate program.

If you are new to programming for Windows and NT, you should now have the basis for going on and beginning to build real Windows applications. However, development with the MFC, as described in the next chapter, will make your life much easier if you wish to develop serious graphical user interfaces.

# Chapter 13

## Development Using the MFC

### Introduction

This chapter describes the development of C++ Windows programs using the Microsoft Foundation Classes (MFC). We describe very briefly how to develop Windows applications of this kind and then how to have them interface and exchange data with HyperChem programs. Examples of such programs are given.

### Microsoft Development Tools

Once again, the appropriate development tool associated with this chapter is Microsoft Visual C++ which includes the Microsoft Foundation Classes. This set of C++ classes allow you to build a Windows graphical user interface in a very short period of time. The combination of the Integrated Development Environment (IDE), the C++ compiler, and the MFC classes, that are all part of Visual C++ 4.0 make for a powerful development tool. One can very quickly put together a Windows application, much faster than with the tools of the last chapter. It is still the case, however, that most of the commercial software, even that from Microsoft, does not yet use the MFC. As easy as it is, it does not provide quite the flexibility that making all your own lower-level API calls does, as in the SDK-style. In addition, it is still necessary to have a good appreciation of the Windows API, even if you are programming with the MFC. Indeed, the API calls are required for many things even within the higher level approach briefly described here. You should decide for yourself whether programming in C with direct API calls (Chapter 12) or programming in C++ with the MFC (this chapter) is for you.

### Programming Assistance

As stated before, this cannot be a programming manual for Windows development. We will illustrate the basic ideas for building Windows MFC pro-

grams that interface to HyperChem and provide some example code but a serious approach to this subject requires you to obtain additional resources. The first, of course, is Microsoft Visual C++ 4.0 itself. The ideas discussed here can certainly be implemented with other compilers and with other development environments, but you will have to make some adaptation of the descriptions given here. We will not attempt to describe development with alternative tools.

The second requirement, if you are new to Windows development, is access to good documentation and tutorial material. One recent book on the subject from Microsoft is,

**Programming Windows 95 with MFC**

Jeff Proise

Microsoft Press, 1996

ISBN 1-55615-902-1

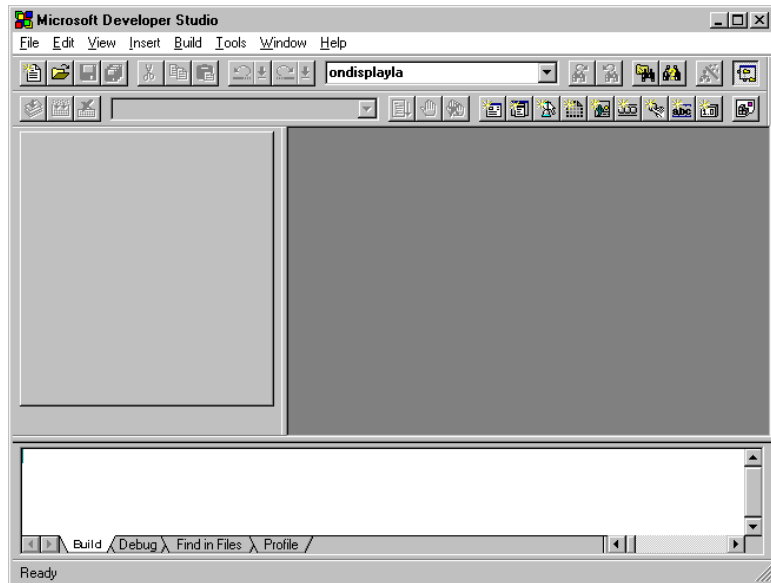
There are many other books that describe development using the MFC and Visual C++ 4.0 on the market, as well.

## **Language**

The C++ language of this chapter may be new to you. If this is the case you may need programming books and tools that specifically address programming in this language. In many ways the object-oriented flavor of C++ makes it more different from C, than C even is from Fortran. An advantage of Visual C++ and its Wizards is that very little code will need to be written for development of the GUI and you will be able to develop significant applications just writing C code for the chemical computation part of your application. An investment in learning C++ programming is probably a well-rewarded investment.

## **A First Example**

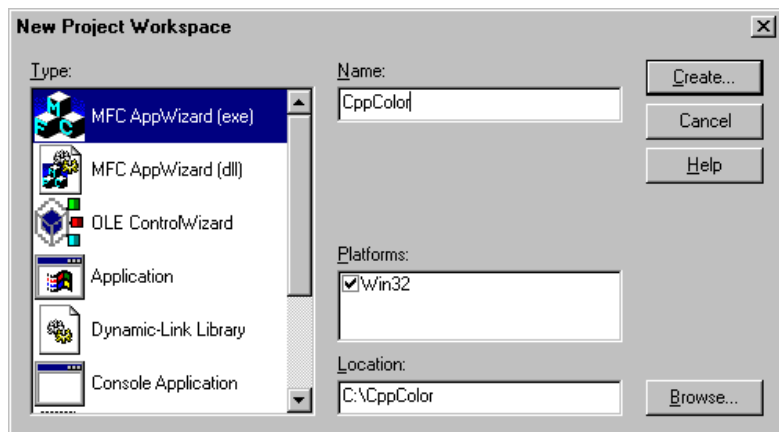
The first example of a C++ program that interfaces to HyperChem will be a simple one to illustrate the basic tools and concepts of Visual C++ 4.0. Bring up the Visual C++ program until it looks similar to the following:



Then,

1. Select <File/New...>

This will bring up a dialog box to select a Wizard to assist you in creating your application,



It is simplest to use the MFC AppWizard to create your application.



2. Type in a name for your application and hit <Create>.

You now are required to decide whether your application is a normal single document application with a menu bar, tool bar, etc., a corresponding document with a multiple document interface (MDI) or a simpler application that is essentially just a single dialog box. For this first application, choose the simpler approach.

3. Choose <Dialog based> and hit <Next>.

You are now asked to elaborate a little on the features of this dialog box.

4. Choose <About box> and <3D-controls> before hitting <Next>.

It is appropriate to choose a shared Dynamic Link Library for MFC and you can should comments that may assist you in understanding the files produced.

5. Choose <As a Shared DLL> and <Yes, Please> prior to hitting <Next>.

You can now complete the process of creating your first MFC application.

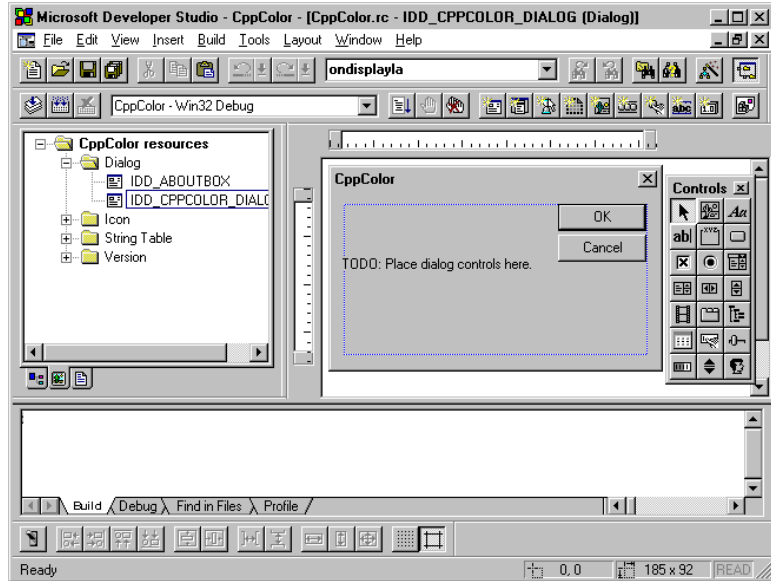
6. Choose <Finish> and <OK> to complete the AppWizard's work.

You are now left with an application that can be compiled and run but it will not do anything of significance yet or be able to interact with HyperChem.

## **Modifications**

To create a custom application, we want to modify the widgets on the dialog box and create code for the modified ones. Specifically, we are once again going to create two buttons - one to change the HyperChem screen color to red and one to change it to green. To accomplish this, we first of all need to modify the existing resources and create two new resources that are the two new buttons.

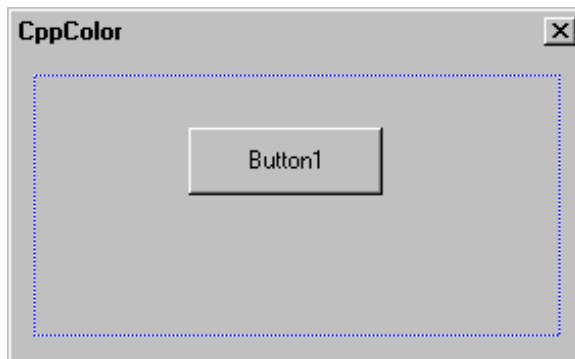
1. Click on the <ResourceView> tab at the bottom of the left window.
2. Double click on <IDD\_CPPCOLOR\_DIALOG> to place the current dialog box in the right window as shown below:



You will now want to delete the OK and Cancel buttons and the TODO label.

3. Click on <OK> to select the button and then <File/Delete> to delete it.
4. Repeat for the Cancel button.
5. Repeat for the TODO label.

You can now add whatever widgets you like to the dialog box. Select the Button control from the set of Controls and create a button on the dialog box by dragging with the mouse until you get the following:

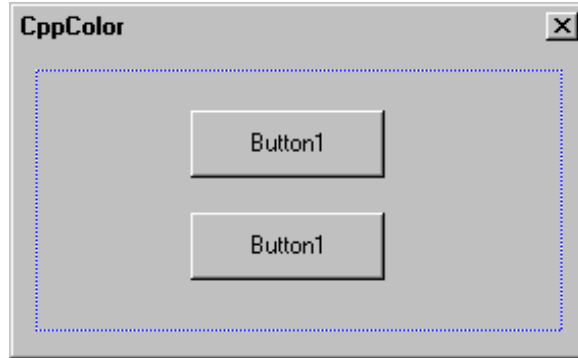


That is,

6. Create a Button on the dialog box.

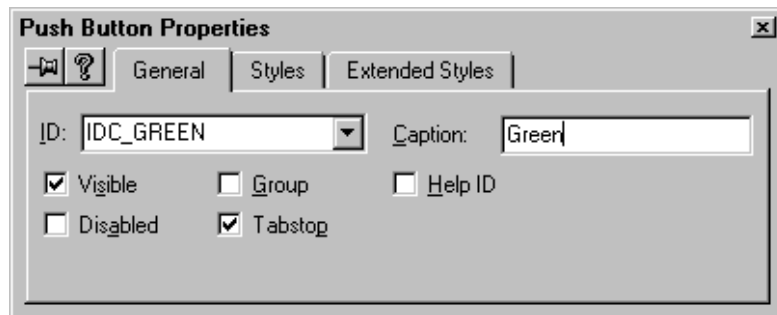
To create a second button select the first one shown above, copy it to the clipboard and then paste it back,

7. Copy and Paste the Button to create a second one.



You can now double click on each button in turn to modify its properties.

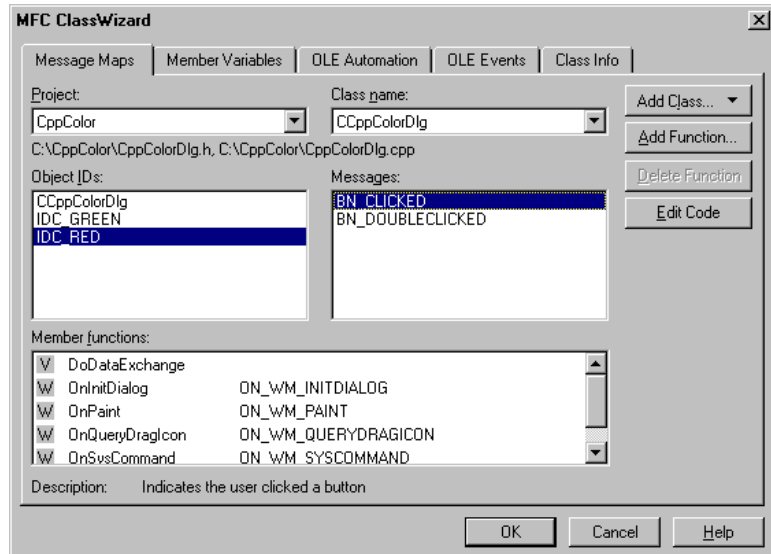
8. Double click on the first button to change its name to IDC\_RED and its Caption to Red.
9. Repeat for the second Green button.



The buttons should now be labelled Red and Green. To continue, we need to invoke the Class Wizard to allow us to create specific code for these buttons.

10. Select the menu item <View/Class Wizard...>

You should see the following:



11. Select <IDC\_RED> and <IDC\_GREEN> in turn with the Message field as <BN\_CLICKED> and hit <Add Function> to create functions OnRed and OnGreen.

Next you need to add code for these functions, so,

12. Hit <Edit Code> and type in the correct code for each button.

The appropriate code to add is:

```
void CcppColorDlg::OnRed()
{
    // TODO: Add your control notification handler code here
    hcExecTxt ("window-color red");
}

```

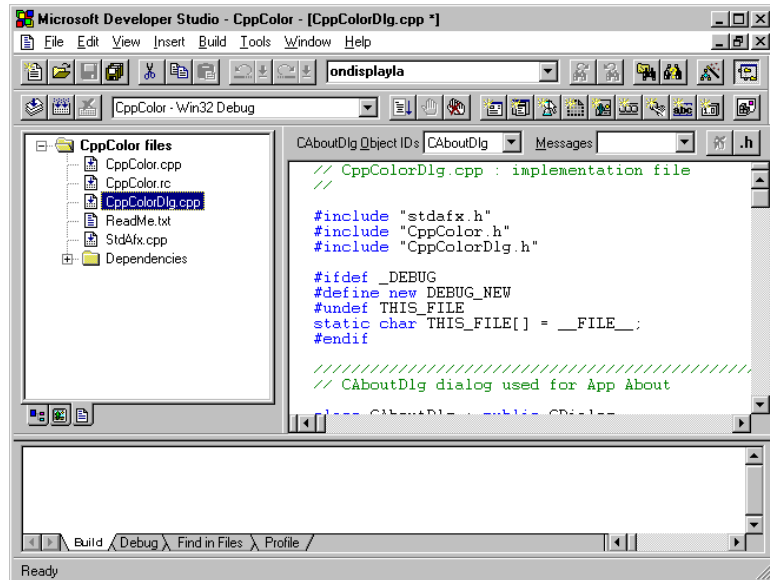
13. Repeat steps 11 and 12 for the green button.

```
void CcppColorDlg::OnGreen()
{
    // TODO: Add your control notification handler code here
    hcExecTxt ("window-color green");
}

```

This completes the use of the AppWizard and Control Wizard for creating this application. There are still a couple of things that have to be done, however, before this code will compile and run correctly. The first task is to add our include files to the set of *includes*. The second task is to load the dynamic link library so that the function `hcExec` can be found. Finally we have to make a connection between this program and HyperChem.

The MFC application includes a number of files, all of which are machine generated. In this case `CColor.cpp` is really identical to a generic application and no changes to it are necessary. The `CColorDlg.cpp` file contains all the code associated with the dialog box and we need to modify this code. To look at code you select the <FileView> tab of the left window, find the file of interest and double click on it. This will give you a view of `CColorDlg.cpp` as below:



## Included Files

At the beginning of the above file the following bold-faced code should be added to the include files already there:

```
// CppColorDlg.cpp : implementation file  
//  
#include "stdafx.h"
```

```
#include "CppColor.h"
#include "CppColorDlg.h"
#include "hc.h"
#include "hclload.c"
```

These are the include files that are needed to communicate with HyperChem. If you are making library calls, e.g. *hcExecBin*, you will need *hsv.h* also.

## Dynamic Link Library and Connecting to HyperChem

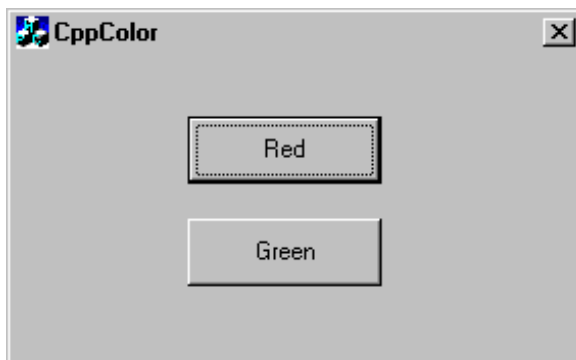
The code that is needed to initialize the loading of the HAPI.DLL and establish the connection to HyperChem should be placed in the *OnInitDialog* routine inside *CColorDlg.cpp* after the appropriate place where the AppWizard tells you it should go,

```
SetIcon(m_hIcon, TRUE); // Set big icon
SetIcon(m_hIcon, FALSE); // Set small icon

// TODO: Add extra initialization here

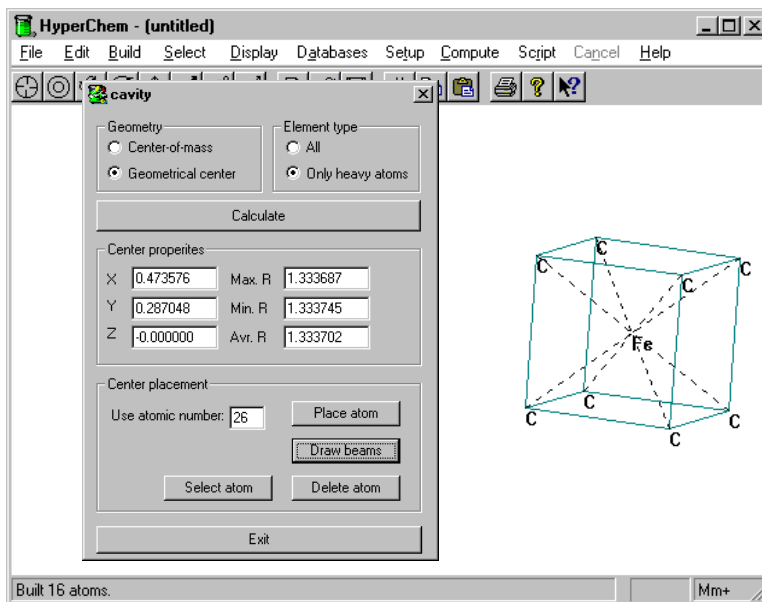
if (!LoadHAPI("hapi.dll"))
{
    MessageBox("Error loading HAPI.DLL", "Error",
              MB_OK|MB_ICONSTOP);
    exit(0);
}
if (!hcConnect(NULL))
    MessageBox("Error connecting to HyperChem", "Error",
              MB_OK|MB_ICONSTOP);
return TRUE; // return TRUE unless you set the focus to a control
```

This completes the first simple example. If everything went correctly, you should have created a Windows application that can talk to and control HyperChem. Remember that the code we have put into the new Windows application expects HyperChem to be there so that you need to start HyperChem before starting the new application. The CppColor application looks as follows:



## Cavity

This next example, called Cavity, is considerably more sophisticated than the last example. It is still a simple Dialog application but it collects significant information from HyperChem - the current selections, the current coordinates, plus it collects the values of quantities in the dialog box before calculating the center of mass of the selection. The application can then place an atom of arbitrary atomic number at the center of mass of the “cavity” and draw special dotted bonds between the center of the cavity and the rest of the molecule. It is essentially identical to the last example except for the more sophisticated interactions with HyperChem and the inclusion of entries made in the dialog box. The code for this example is on the HyperChem CD-ROM. The dialog box is shown below,



This application is too long to discuss all the code here. If you are interested, however, you should be able to follow every aspect of this relatively simple calculation using the full source code. All the calls in this program are text-based calls with no binary call that need `hsv.h`. We will discuss certain portions of the code as being instructive of how one generally interacts with HyperChem. The code, *OnCalc*, associated with pushing the Calculate button is where the coordinates are read. The first thing that is necessary, before reading the coordinates, is to characterize the atoms and molecules of the system in HyperChem,

```
// HyperChem won't send tags with OMSGs
hcExecTxt("query-response-has-tag = false");
// Gather information about molecular system
// start with number of molecules
resp=hcQueryTxt("molecule-count");
iMol=atoi(resp);hcFree(resp);
// allocate memory for iaAtomCount array
iaAtomCount=(int*)calloc(iMol,sizeof(int));
// next, get count of atom in each molecule
resp=hcQueryTxt("atom-count");
// parse string returned by CDK into integer array
```



```

pstr=resp;
i=0;iAtomTot=0;
while (ptok=strtok(pstr,_DELIMITERS)) {
    if (pstr) pstr=NULL;
    iAtomTot += iaAtomCount[i++]=atoi(ptok);
}
hcFree(resp);

```

As with many interfaces to HyperChem, the first thing to do is to eliminate the tag which comes, by default, with a returning message. Next, we get the number of molecules, *molecule-count*, which comes in as a string and gets converted to an integer. The memory allocation for the string occurs automatically within `hcQueryTxt` and this memory *must be freed* by you when the string is no longer needed. Forgetting to free memory is a common C programming error.

The HSV, atom-count, is a vector. Each element could have been read in turn but all the elements can also be read at once - the resulting string must then be parsed, however. The vector, `iaAtomCount`, is used to hold the result of this parsing.

The next step is to sort out which of the atoms and molecules are selected since we are only going to read the selected atoms coordinates. This result in vectors `iaSelectedAtom` and `iaSelectedMol` being allocated and filled in, plus a variable, `iSelected`, that represents the total number of atoms selected. The code for this can be inspected in the source files if you wish. We skip it for brevity. The next step is to read the coordinates for the selected atoms. The code for this is,

```

// get coordinates of atoms
faX=(float*)calloc(iSelected,sizeof(float));
faY=(float*)calloc(iSelected,sizeof(float));
faZ=(float*)calloc(iSelected,sizeof(float));
for (is=0;is<iSelected;is++) {
    wsprintf(buff,"coordinates(%d,%d)",
        iaSelectedAtom[is],iaSelectedMol[is]);
resp=hcQueryTxt(buff);
pstr=resp;
i=0;in=0;
while (ptok=strtok(pstr,_DELIMITERS)) {
    if (pstr) pstr=NULL;
    switch (in) {

```

```
        case 0: x=(float)atof(ptok); in++; break;
        case 1: y=(float)atof(ptok); in++; break;
        case 2: z=(float)atof(ptok); in=0; break;
    }
}
hcFree(resp);
faX[is] = x;
faY[is] = y;
faZ[is] = z;
}
```

The loop over selected atoms makes a request for the coordinates of that atom, as an element of the coordinate array. This element is a string of three values (x, y, and z coordinates) and has to be parsed.

The code for the remaining portions of this problem is available for your inspection, if you are interested.

*Cavity*

# Chapter 14

## Console C and Fortran Applications

### Introduction

This chapter describes how to develop “conventional” character-oriented or console applications in C or Fortran that can be interfaced with HyperChem.

It is relatively common in computational or theoretical chemistry to develop applications in Fortran (or as is becoming more common, in C) that have no graphical user interface but operate with input files creating output files. For example, many programs have been developed to calculate a molecular wave function. Such a program in its simplest form might need as input the Cartesian coordinates of a molecule and give as output the energies and coefficients describing the molecular orbitals. It is a significant effort to build the additional graphics program that would allow the molecule to be just sketched and the output to be presented as a 3D rendering of the molecular orbitals. The CDK, however allows the developer of such a molecular orbital program to use HyperChem to sketch the molecule and to render the molecular orbitals. HyperChem can act as a front end GUI and a back end visualizer to this character-oriented program by simply having it call on HyperChem whenever it needs graphical services.

### Console Applications

Microsoft refers to programs that have no graphical user interface as console applications. An example output from one of these console applications is shown below.

```

orbitals
Console Example
There are      8 atoms in      1 molecules
3.346881270408630E-001 -1.420232802629471E-001 -4.426935315132141E-001
2.884356863796711E-002  5.882838368415833E-001  8.781813979148865E-001
-5.350705385208130E-001 -7.251807451248169E-001 -7.452613115310669E-001
5.688869953155518E-001  5.892016887664795E-001 -1.216360688209534
      1.186713099479675 -8.069893717765808E-001 -3.013651371002197E-001
8.986021876335144E-001      1.171441197395325      1.180749177932739
-8.231729269027710E-001      1.253258109092712      7.368410229682922E-001
-2.053704410791397E-001 -1.429343074560165E-001      1.651850461959839

Press any key to continue_

```

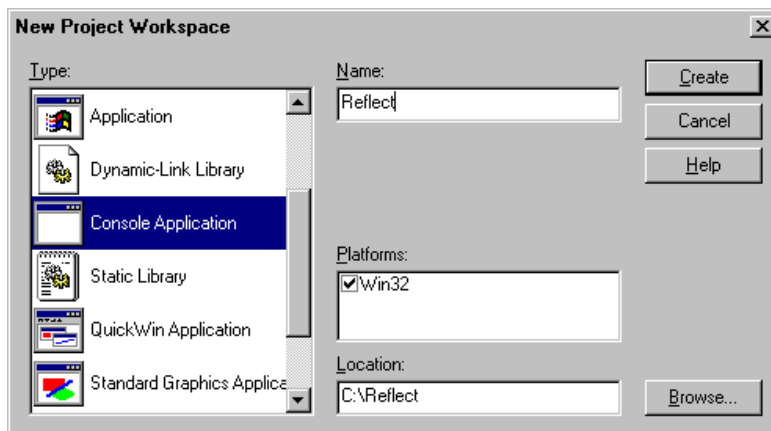
These applications look a lot like many UNIX applications or applications developed in DOS. Until recently it was very difficult to develop such console programs within the Windows environment, i.e. without a window and a graphical user interface. Now with Windows 95 or NT and Visual C++ 4.0 and Fortran Power Station it is relatively easy to port traditional non-graphical scientific applications to the Windows environment. With the Chemists Developer Kit, it is also easy to interface these to HyperChem.

## C or Fortran

Until recently, the Microsoft Windows environment and development tools have essentially been usable only with C (or C++). With the recent introduction of Fortran PowerStation, however, it has become easy to develop Windows applications in Fortran. For console applications, the tools for C and Fortran are essentially on the same basis and the choice can be based upon which language is your personal choice. For more graphical programs, Fortran (which requires calls to the Windows API - developed for C) is still more awkward to use than C. It is still possible, however, using tools that Microsoft has provided with Fortran PowerStation. In this chapter and in this manual we will not concern ourselves with development of graphical user interfaces in Fortran. Rather, we restrict our Fortran discussions to the console applications of this chapter. For such console applications, C and Fortran are interchangeable as to their ease of use within Windows.

## The Integrated Development Environment

The Visual C++ 4.0 Integrated Development Environment is where Fortran PowerStation programs are developed and this gives an identical development environment for both C and Fortran. Thus a console program is created, as before, by selecting <File/New>, choosing to create a new *Project Workspace* and then selecting a Console Application,



For console applications, one has to *Insert* C (\*.c), C++ (\*.cpp), or Fortran (\*.f) files into the project as needed since none are generated automatically as with the AppWizard.

## C Program

Our first example of a console program is a C program. The program looks like any C program beginning with main(), etc. The code for it is as follows:

```

/*
 * Console - UNIX like C program that talks to HyperChem *
 */
#include <stdio.h>
#include <windows.h>
#include "hc.h"
#include "hclload.c"
int main(int iArg, char **pArg)
{
char cmd_line[100],initColor[100],color[100],buffer[100];

```

```

char *response;
int result;
/* loading HAPI.DLL */
if (!LoadHAPI("hapi.dll")) {
    printf("Error loading HAPI.DLL !\n");
    exit(0);
};
/* connecting to HyperChem */
if (iArg!=2) {
    strcpy(cmd_line,"");
} else {
    strcpy(cmd_line,pArg[1]);
};
if (!hcConnect(cmd_line)) {
    printf("Error while connecting with HyperChem !\n");
    exit(0);
}

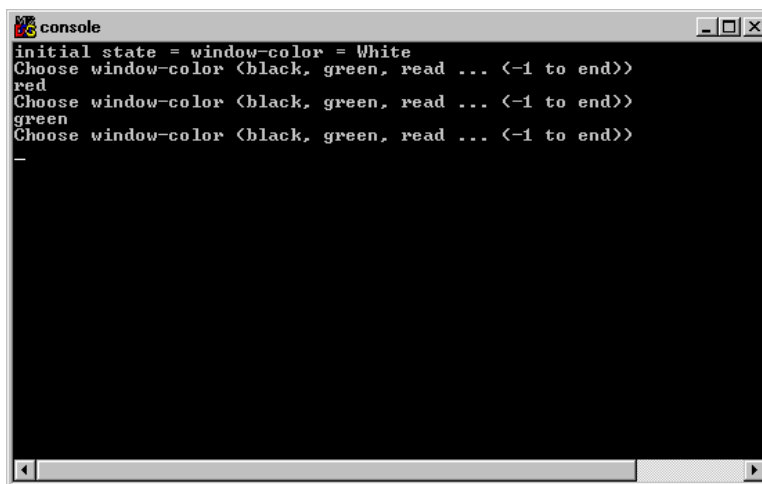
/* obtain initial window-color from HyperChem */
response=hcQueryTxt("window-color");
printf("initial state = %s\n",response);
strcpy(initColor,response);
hcFree(response);

for(;;) {
    printf("Choose window-color (black, green, red ... (-1 to end))\n");
    scanf("%s",color);
    if (atoi(color) == -1) {
        result=hcExecTxt(initColor);
        result=hcDisconnect();
        return 0 ;
    } else {
        sprintf(buffer,"window-color %s",color);
        result=hcExecTxt(buffer);
    }
}
return 0;

```

This code looks like any simple C program with the exception of calls to `hcConnect`, `hcQueryTxt`, `hcExecTxt`, and `hcDisconnect`. Any command line arguments are passed through to HyperChem. The header files, `hc.h` and `hload.c` are required as is the dynamic loading of the library of the HyperChem Application Programming Interface, `HAPI.DLL`. Subsequent to con-

necting to HyperChem, the program can just read and write HSV's to HyperChem or control HyperChem via direct commands. The program requires HyperChem to be running prior to its own invocation. The result of running this program is the following Console window.

A screenshot of a Windows-style console window titled "console". The window has a black background and white text. The text inside the window is:

```
initial state = window-color = White
Choose window-color <black, green, read ... <-1 to end>>
red
Choose window-color <black, green, read ... <-1 to end>>
green
Choose window-color <black, green, read ... <-1 to end>>
-
```

The window also shows standard window controls (minimize, maximize, close) in the top right corner and a scrollbar at the bottom.

The corresponding HyperChem window with its changing background colors is not shown. The console window contains any text that a normal C program outputs via “standard output” using `printf`, etc. It can be iconized, if desired, so that it is effectively running in the background while you interact with HyperChem. It behaves very much like HyperNewton, HyperNDO, etc., the standard back ends that come with HyperChem and perform compute intensive computations.

## Fortran Programs

We will now illustrate, in somewhat more detail, examples of Fortran programs that behave like the above. It is believed that HyperChem and the CDK provide an ideal opportunity to interface a wide variety of Fortran programs, such as could be available from the Quantum Chemistry Program Exchange (QCPE) at Indiana University. Many programs have been generated over many years, by many chemists, and these could very quickly have a “slick” graphical user interface.



## Reflect

Our first example is a program that simply collects the coordinates of all the atoms from HyperChem, performs some simple transformation of these coordinates, and sends them back to HyperChem for display. For simplicity we will use a reflection in the XY plane through the origin to illustrate the general process. That is, our program will replace all Z coordinates by -Z. If your molecule has chiral centers and their chirality label is displayed, you will see the conversions R->S and S->R from running this program.

This example, while trivial, has some of the characteristics of a “real” program that would perform significant computation on an “initial” structure leading to a “new” structure. Molecular dynamics programs or a structure optimization programs have this flavor. The whole program is,

```

program Reflect
c-----
c include header files - HAPI definitions and declarations
  include 'hc.fi'
  include 'hsv.fi'
  parameter (nDimensions=3)
  parameter (nMaxAtoms=2000)

  character*60 cmd_line
  logical result
  integer status

  integer nAtoms
  double precision XYZ(nDimensions, nMaxAtoms)

c
c Connect to HyperChem using current command line
c
  call getarg(1,cmd_line,status)
  result=hfConnect(cmd_line)
  if (.NOT. result) stop

c we do not want to have names in front of HSVs !
c query-response-has-tag=false does this for us
  result=hfExecTxt("query-response-has-tag=false")

  write(*,*)'Reflection Example'

c call subroutines to read in initial atom coordinates

```

```

c and write final atom coordinates
  call GetCoords(XYZ, nDimensions, nMaxAtoms, nAtoms)
  call PutCoords(XYZ, nDimensions, nMaxAtoms, nAtoms)

  end

  subroutine GetCoords(coords,nDim,nMaxAt,nAt)
c-----
c gets xyz coordinetes of all atoms in the system
c and stores them in 'coords' array
c-----
c include HAPI definitions and declarations
  include 'hc.fi'
  include 'hsv.fi'

  double precision coords(nDim,nMaxAt)

c get number of molecules from HyperChem (integer value)
c using hfGetInt "binary" function

  nMol=hfGetInt(molecule_count)

c get total number of atoms by scanning all molecules
  nAt=0
  do 1 i=1,nMol

c get number of atoms in the molecule
c hfGetIntVecElm is "binary" function ("atom_count" is Integer Vector)
  iatoms=hfGetIntVecElm(atom_count,i)
  nAt=nAt+iatoms
1 continue
  write(*,*)'There are ',nAt,' atoms in ',nMol,' molecules'

c get xyz coordinates of all atoms and place them into 'coords' array
c using hfGetRealArr "binary" function ("coordinates" is Real Array)
  lres=hfGetRealArr(coordinates,coords,nDim*nMaxAt)

  do 111 i=1,nAt
  write(*,*)(coords(k,i),k=1,3)
111continue
  write(*,*)'-----'
  return

```

```

end

subroutine PutCoords(coords,nDim,nMaxAt,nAt)
c-----
c takes the coords array and send it back to HyperChem
c as the cartesian coordinates of all atoms
c-----
c include HAPI definitions and declarations
  include 'hc.fi'
  include 'hsv.fi'

double precision coords(nDim,nMaxAt)

write(*,*)'There are ',nAt,' atoms'
do 111 i=1,nAt
  coords(3,i) = -coords(3,i)

  write(*,*)(coords(k,i),k=1,3)
111 continue
  write(*,*)'-----'

c get xyz coordinates of all atoms and place them into 'coords' array
c using hfGetRealArr "binary" function ("coordinates" is Real Array)
  lres=hfSetRealArr(coordinates,coords,nDim*nAt)
  return
end

```

The console output is,

```

Reflect
Reflection Example
There are 5 atoms in 1 molecules
 1.192092895507813E-007  0.000000000000000E+000  0.000000000000000E+000
 1.192092895507813E-007  1.360000014305115  0.000000000000000E+000
 1.659346342086792  -5.866602659225464E-001  -3.592139343491155E-017
-9.003690481185913E-001  -6.366597414016724E-001  -1.559519052505493
-9.899560809135437E-001  -7.000076174736023E-001  1.714635610580444
-----
There are 5 atoms
 1.192092895507813E-007  0.000000000000000E+000  0.000000000000000E+000
 1.192092895507813E-007  1.360000014305115  0.000000000000000E+000
 1.659346342086792  -5.866602659225464E-001  3.592139343491155E-017
-9.003690481185913E-001  -6.366597414016724E-001  1.559519052505493
-9.899560809135437E-001  -7.000076174736023E-001  -1.714635610580444
-----
Press any key to continue_

```

This example illustrates the use of both binary calls such as *hfSetRealArr* that require the include file `hsv.fi` as well as text calls such as *hfExecTxt* that do not. The binary calls require the include file `hsv.fi` to map integers such as “atom\_count” to the appropriate HyperChem variable. Note that in binary calls all “hyphens” are replaced by “underscores”. Thus the HSV, atom-count, maps to the integer `atom_count` through `hsv.fi`. Each of the HAPI calls is described in Appendix C.

A single call in the code above is all that is necessary to read or write the atom coordinates once the total number of atoms is known. HyperChem combines atom numbers within a molecule with molecule numbers to obtain a unique atom number. The code first of all has to query for the number of molecules and then for the number of atoms in each molecule to obtain the total number of atoms. Beyond that computation the rest of the above program is very straight forward.

## MiniGauss Orbitals

The next example is one which contains the elements of a number of potentially very significant uses for the CDK. It provides molecule creation and visualization for an *ab initio* wave function package. The *ab initio* package is based on a demonstration Fortran code that is an appendix to the book,

### Modern Quantum Chemistry

Attila Szabo and Neil S. Ostlund

Dover Publications, Inc.

New York, 1996

This program, only a couple of pages long, contains all the code for an *ab initio* STO-1G, STO-2G, or STO-3G calculation on 2-electron diatomics like  $\text{H}_2$ ,  $\text{HeH}^+$ ,  $\text{He}_2^{++}$ , etc. It has proved a useful educational tool for a number of young (and not-so-young) theoreticians. What we do here is offer a “front end” to this program to illustrate how HyperChem can provide molecular coordinates plus the visualization of the 3D shape of the calculated orbitals and charge density.

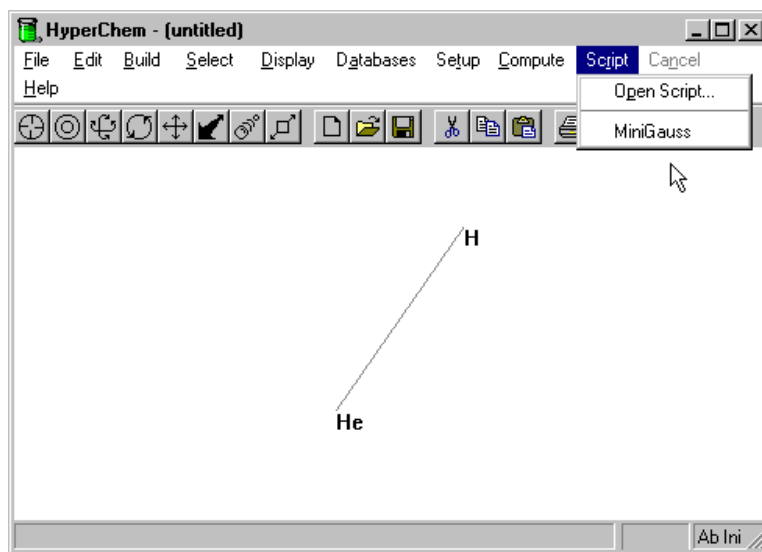
### Outline

The basic idea of this example is to mimic the actions of HyperGauss, a full fledged *ab initio* package that comes with HyperChem. We will operate the

MiniGauss back end by having a menu item and scripts to conveniently run the Fortran program. All the scripts for this example, plus the Fortran code, are on the HyperChem CD-ROM associated with the *Orbitals* directory. While we could easily use the idea of custom menus to run this program we will simply place an appropriate menu item in the <Script> menu. To do this, we first of all run a script, *orbitals.scr*, to set up a convenient way of running these calculations. This script is,

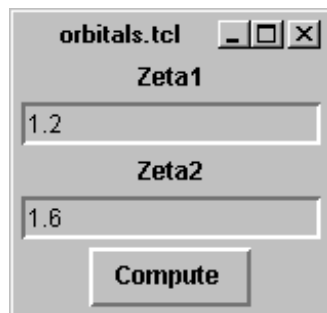
```
script-menu-caption(1) = "MiniGauss"
script-menu-enabled(1) = true
script-menu-command(1) = "read-tcl-script orbitals.tcl"
```

This sets up a new menu item, *MiniGauss*, that when invoked executes a Tcl script called *orbitals.tcl*. The *orbitals.scr* script could be placed into *chem.scr* so that it is always executed and the MiniGauss menu always appears - if you have developed an application that you would like to be more or less permanently installed. The HyperChem menu item looks as follows:



## A New GUI Element

The Tcl/Tk script, *orbitals.tcl* is going to give us an additional GUI element (dialog box) that makes for interactive use of MiniGauss. This dialog box, invoked by selecting the MiniGauss menu item above, is,



It collects the two Slater exponents for the minimal basis calculation. For the other input to the calculation, i.e. the N of STO-NG, we are going to use the standard facility of HyperChem to define a basis set. Thus, it is only the above orbital exponents, plus of course the molecule, that MiniGauss needs. The Tcl/Tk script, `orbitals.tcl`, is,

```

label .l1 -text "Zeta1"
entry .en1 -width 20 -textvariable inzeta1
label .l2 -text "Zeta2"
entry .en2 -width 20 -textvariable inzeta2
button .b -text "Compute" -command {
  hcExec "declare-string zeta1"
  hcExec "declare-string zeta2"
  hcExec "zeta1 $inzeta1"
  hcExec "zeta2 $inzeta2"
  hcExec "execute-client mini.exe"
Exit
}
pack .l1 .en1 .l2 .en2 .b

```

This Tcl script is basically just Tk code to create two labels, two text entry boxes, and a button to dismiss the dialog box and start the MiniGauss calculation.

The two text entry boxes are used to input the two orbital exponents, zeta1 and zeta2, as strings inzeta1 and inzeta2. The button calls HyperChem to create two new string variables (new HSV's) called appropriately zeta1 and zeta2. These new HSV's are then assigned the strings collected from the entry boxes. This is the first time we have seen the ability to create new HSV's and as you can see here it is very useful to provide a repository for any new values entered from a new GUI until they can be passed to their ultimate destination, which in this case will be the MiniGauss program.

## The Main Program

The last two things done by pushing the Compute button are:

```
hcExec "execute-client orbitals.exe"
Exit
```

The Exit, with a capital E, exits the Tcl script but not HyperChem [an exit with a small e would exit both the Tcl script and HyperChem!]. In concert with this, the MiniGauss program is invoked which is our main Fortran program and is here called orbitals.f (and orbitals.exe).

```
Program Orbitals
  implicit double precision(a-h,o-z)
c -----
c include header files - HAPI definitions and declarations
  include 'hc.fi'
  include 'hsv.fi'
  parameter (nDimension=3)
  parameter (nMaxAtoms=2000)

  character*60 cmd_line
  logical result
  integer status

  integer nAtoms
  double precision XYZ(nDimension, nMaxAtoms)

c
c Connect to HyperChem using current command line
c
  call getarg(1,cmd_line,status)
```

```

    result=hfConnect(cmd_line)
    if (.NOT. result) stop

c we do not want to have names in front of HSVs !
c query-response-has-tag=false does this for us
    result=hfExecTxt("query-response-has-tag=false")

    write(*,*)'Orbitals Example'

c call subroutine to read in initial atom coordinates
c
    call DoCalc(XYZ, nDimension, nMaxAtoms, nAtoms)
    call SendResults

end

```

The main program simply connects to HyperChem and makes sure that you don't get the tags along with the values when HSV's are queried. It then calls a routine (DoCalc) that is principally concerned with performing the calculation and one which sends back the results to be displayed graphically (SendResults).

## Get Molecule

The Subroutine DoCalc is primarily concerned with getting the coordinates of all the atoms (nAtoms) to be input to the minimal basis set *ab initio* calculation. This subroutine is,

```

    subroutine DoCalc(coords,nDim,nMaxAt,nAt)
    implicit double precision(a-h,o-z)
c-----
c gets xyz coordinates of all atoms in the system
c and stores them in 'coords' array
c-----
c include HAPI definitions and declarations
    include 'hc.fi'
    include 'hsv.fi'

    character *100 buffer
    double precision coords(nDim,nMaxAt)

c get number of molecules from HyperChem (integer value)
c using hfGetInt "binary" function

```



```

    nMol=hfGetInt(molecule_count)
c we can work only with one molecule
    if (nMol .gt. 1) then
        write(*,*)'This demo assumes that you have only ONE molecule'
c disconnect from HyperChem
    result=hfDisconnect()
    stop
    endif

c get total number of atoms by scanning all molecules
    nAt=0
    do 1 i=1,nMol

c get number of atoms in the molecule
c using hfGetIntVecElm "binary" function ("atom_count" is Integer
Vector)
        iatm=hfGetIntVecElm(atom_count,i)
        nAt=nAt+iatm
1 continue
c we can only work with a diatomic
    if (nAt .ne. 2) then
        write(*,*)'This demo assumes that you have TWO atoms'
c disconnect from HyperChem
    result=hfDisconnect()
    stop
    endif

c find out atomic numbers
    nAtNum1=hfGetIntArrElm(atomic_number, 1, 1)
    nAtNum2=hfGetIntArrElm(atomic_number, 2, 1)
    if (nAtNum1 .gt. 2 .or. nAtNum2 .gt. 2) then
        write(*,*)'This demo assumes that you have H or He atoms'
c disconnect from HyperChem
    result=hfDisconnect()
    stop
    endif
    za = dfloat(nAtNum1)
    zb = dfloat(nAtNum2)
    nCharge=hfGetInt(quantum_total_charge)
    nElectrons = nAtNum1 + nAtNum2 - nCharge
    if (nElectrons .ne. 2) then
        write(*,*)'This demo assumes that you have 2 electrons'
c disconnect from HyperChem

```

```

    result=hfDisconnect()
    stop
endif
c get the basis set
    result = hfQueryTxt("atom-basisset(1,1)", buffer)
    if (buffer(1:6).eq.'STO-3G') then
        n=3
    else if (buffer(1:6) .eq. 'STO-2G') then
        n=2
    else if (buffer(1:6) .eq. 'STO-1G') then
        n=1
    else
        n=3
    end if

c get the zeta
    result = hfQueryTxt("zeta1", buffer)
    read (buffer,*) zeta1
    result = hfQueryTxt("zeta2", buffer)
    read (buffer,*) zeta2

c get xyz coordinates of all atoms and place them into 'coords' array
c using hfGetRealArr "binary" function ("coordinates" is Real Array)
    lres=hfGetRealArr(coordinates,coords,nDim*nMaxAt)

    dx = coords(1,1)-coords(1,2)
    dy = coords(2,1)-coords(2,2)
    dz = coords(3,1)-coords(3,2)
    r = dsqrt( dx*dx + dy*dy + dz*dz) /0.52918

    call hfcalc(n,r,zeta1,zeta2,za,zb)
return
end

```

The above code contains a number of text and binary calls to HyperChem. A single call (hfGetRealArr) gets all the coordinates once we know the number of atoms. The two orbital exponents that we got from the Tk dialog box and stored in HyperChem are retrieved from HyperChem along with other information that HyperChem has about the calculation that is about to be performed. One aspect of this information is the basis set, as set by the user in HyperChem. If HyperChem says it is one of the STO-NG basis sets then that information is used, otherwise an STO-3G calculation is performed. The remaining code near the beginning is associated with getting basic information about the molecule and making sure it is simple enough for MiniGauss.

## Wave function Calculation

The call to the routine *hfcalc* is set up to be identical (apart from *iop* which is a printing option) to that in Szabo and Ostlund so that you might substitute their few pages of code if you so desire. A drastically simpler version of *hfcalc* used here is,

```
subroutine hfcalc(n,r,zeta1,zeta2,za,zb)
  implicit double precision(a-h,o-z)
  common/matrix/s(2,2),x(2,2),xt(2,2),h(2,2),f(2,2),g(2,2),c(2,2),
  $ fprime(2,2),cprime(2,2),p(2,2),oldp(2,2),tt(2,2,2,2),e(2,2)
  c(1,1) = .707
  c(2,1) = .707
  c(1,2) = .707
  c(2,2) = -0.707
  e(1,1) = -5.0
  e(2,2) = 5.0
  return
end
```

This just sets the eigenvalues to arbitrarily be -5.0 and +5.0 and the eigenvectors to be the standard in phase and out of phase orbitals for HOMO and LUMO. The assumption here is made that the overlap is small so that coefficients are just  $1/\sqrt{2}$ .

As discussed a better *hfcalc* could easily be programmed. The one shown here is perfectly satisfactory for illustrating the use of the CDK, however.

## Displaying Orbitals and ...

The routine that sends back the orbitals for display is,

```
subroutine SendResults
  implicit double precision(a-h,o-z)
  common/matrix/s(2,2),x(2,2),xt(2,2),h(2,2),f(2,2),g(2,2),c(2,2),
  $ fprime(2,2),cprime(2,2),p(2,2),oldp(2,2),tt(2,2,2,2),e(2,2)

c include HAPI definitions and declarations
  include 'hc.fi'
  include 'hsv.fi'

  character *80 buff

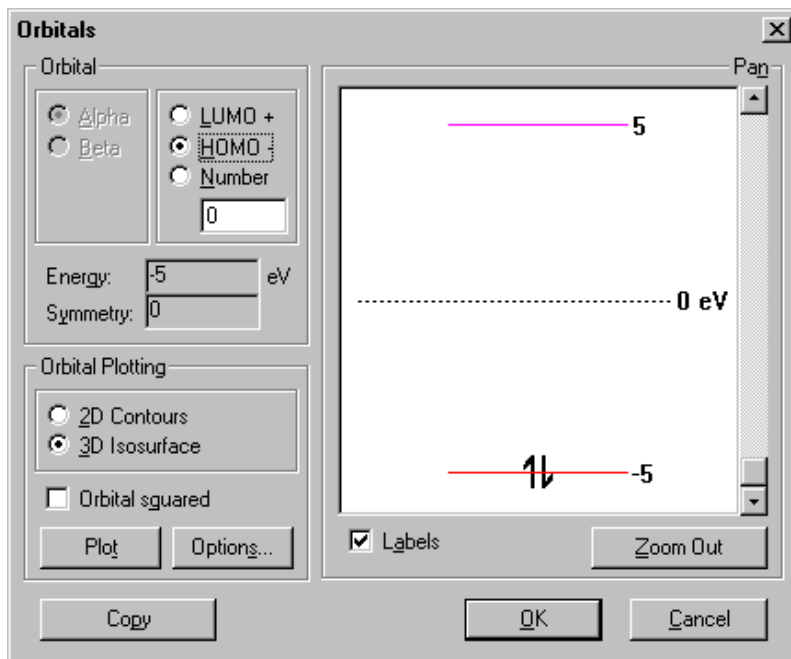
  result = hfExecTxt('orbital-count=2')
  write (unit = buff, fmt = '(2f8.4)') e(1,1), e(2,2)
```

```

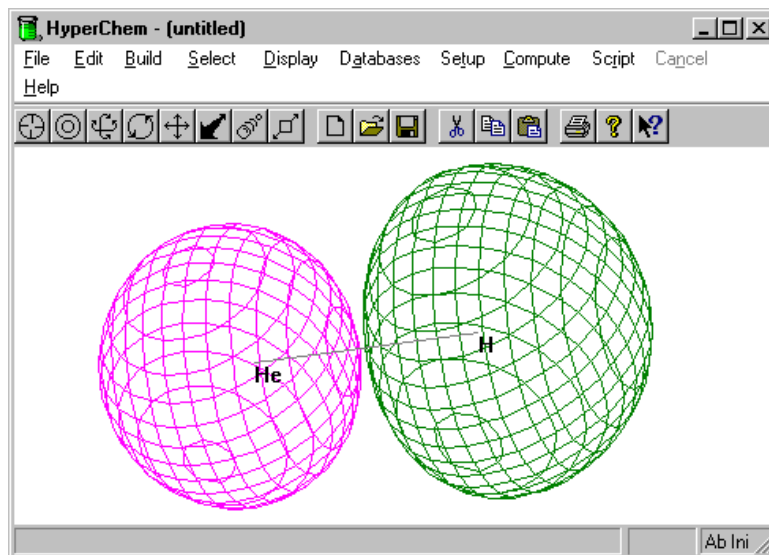
result = hfExecTxt('scf-orbital-energy='//buff)
write (unit = buff, fmt = '(2f8.4)') c(1,1), c(2,1)
result = hfExecTxt('alpha-scf-eigenvector(1)='//buff)
write (unit = buff, fmt = '(2f8.4)') c(1,2), c(2,2)
result = hfExecTxt('alpha-scf-eigenvector(2)='//buff)
write result = hfExecTxt('alpha-orbital-occupancy= 2 0')
return
end

```

There are a number of ways that the results might be returned. One way is to make binary HAPI calls such as to *hcSetRealVec* to return vectors of coefficients. We have chosen here to illustrate the return of text strings to HyperChem. The first thing that is done is to set the total count of the number of molecular orbitals which in the minimal basis illustrated here is 2. Next the orbital energies are returned and then the eigenvectors. Since this is only a closed-shell calculation, only the alpha (RHF) coefficients need be filled in. Finally the occupancy is set with 2 electrons in the first orbital (HOMO) and 0 electrons in the LUMO. The result of selecting the MiniGauss menu item with  $\text{HeH}^+$  on the screen is,



The orbitals or the charge density or the electrostatic potential can be plotted. The orbitals are somewhat artificial here but asking for the LUMO orbital gives a plot like the following:



## Diffusion Limited Aggregation

The final example to be mentioned is one of diffusion limited aggregation and is contained in the `dla` directory from the HyperChem CD-ROM. This is presented for those who wish to follow it as an example that has a Visual Basic program, `setup_dla.exe`, to create a number of new variables in HyperChem, the getting from HyperChem of the initial molecular system, the creation of new atoms and structures in the HyperChem workspace, and a more elaborated dialog box than we have used so far.

## Further Examples

Further examples of using the CDK that might be of interest to you can possibly be found in conjunction with your specific CD-ROM installation of HyperChem or possibly on the Hypercube WWW site, <http://www.hyper.com>

# Appendix A

## Classification of Hcl Commands

### The Classes

In this appendix we list all the HSV's and Hcl direct commands according to a set of classes that will hopefully assist you in finding the appropriate script command for your task. In the Reference Manual these script commands are classified differently - according to the related HyperChem menu. In Chapter 7 all HSV's and direct Hcl commands are listed in alphabetic order. These three listings complement each other and should assist you in getting familiar with the extensive set of script commands. The listings in the Reference Manual have the most explicit description of each of the following script commands and should be referred to if the use of the command listed here or in Chapter 6 is confusing to you.

The classes which we use to help classify all the script commands are:

- General Operations
- Cursors
- Selections
- File Operations
- Scripts
- Info
- Errors
- Logging
- Auxiliary
- Viewing
- Rendering
- Coloring and Labeling

- Images
- Model Building
- Stereochemistry
- Atom Properties
- Molecular Properties
- Backends
- Molecular Mechanics Calculations
- Amino Acids and Nucleic Acids
- Molecular Dynamics and Monte Carlo
- Optimization
- General Quantum Mechanics
- Semi-empirical Calculations
- Ab Initio Calculations
- Configuration Interaction
- Infrared Spectra
- UV Spectra
- Plotting

## General Operations

The following script commands involve the general operation of HyperChem including things that don't easily fit into other categories. Thus, in this category we have the script commands for a single-point calculation and for solvation using the periodic box. Other commands have to do with default or custom menus, printing, and the operation of the Cancel button.

### Single Point

**calculation-method:** Variable, Read/Write, Type: enum.

**do-single-point:** Command, Arg list: (void).

**total-energy:** Variable, Read/Write, Type: float in range (-1e+010 .. 1e+010).

## Solvation

**periodic-boundaries**: Variable, Read/Write, Type: boolean.

**periodic-box-size**: Variable, Readonly, Type: (unknown).

**solvate-system**: Command, Arg list: (void).

**solvate-system-in-this-box**: Command, Arg list: float, float, float.

## Fields

**field-direction**: Variable, Read/Write, Type: integer

**field-strength**: Variable, Read/Write, Type: float

**magnetic-field-direction**: Command, Arg list: unregistered.

**magnetic-field-strength**: Command, Arg list: unregistered.

**use-electric-field**: Command, Arg list: unregistered.

**use-magnetic-field**: Command, Arg list: unregistered.

## Customization

**hide-toolbar**: Variable, Read/Write, Type: boolean.

**load-default-menu**: Command, Arg list: (void).

**load-user-menu**: Command, Arg list: string.

**switch-to-user-menu**: Command, Arg list: (void).

**custom-title**: Variable, Read/Write, Type: string.

**factory-settings**: Command, Arg list: (void).

## Printing

**print**: Command, Arg list: (void).

**printer-background-white**: Variable, Read/Write, Type: boolean.

## Other

**cancel-menu**: Variable, Read/Write, Type: boolean.

**do-vibrational-analysis**: Command, Arg list: (void).

**help**: Command, Arg list: string.

**hide-messages**: Variable, Read/Write, Type: boolean.



**module-name:** Variable, Read/Write, Type: string.

## Cursors

The following script commands have to do with the operation of the various cursors apart from the drawing and selection cursor. The remaining cursors are associated with rotation, translation, zooming, and clipping operations. These have parameters that describe, for example, the unit of rotation when the rotation cursor is dragged in the work space (x-y-rotation-cursor). alternatively, there is a unit or rotation when the appropriate keyboard equivalent is used (x-y-rotation-icon-step). These parameters can be set via the <File/Pref-erences...> dialog box or via the following script commands.

### Mouse Mode

**mouse-mode:** Variable, Read/Write, Type: enum

### Clipping

**clip-cursor:** Variable, Read/Write, Type: float in range (0 .. 1000).

**clip-icon-step:** Variable, Read/Write, Type: float in range (0 .. 1000).

**back-clip:** Variable, Read/Write, Type: float

**front-clip:** Variable, Read/Write, Type: float.

### Rotation

**x-y-rotation-cursor:** Variable, R/W, Type: float angle in range (0 .. 3600).

**x-y-rotation-icon-step:** Variable, R/W, Type: float angle in range (0 .. 3600).

**z-rotation-cursor:** Variable, R/W, Type: float angle in range (0 .. 3600).

**z-rotation-icon-step:** Variable, R/W, Type: float angle in range (0 .. 3600).

### Translation

**x-y-translation-icon-step:** Variable, R/W, Type: float in range (0 .. 1000).

**z-translation-cursor:** Variable, R/W, Type: float in range (0 .. 1000).

**z-translation-icon-step:** Variable, R/W, Type: float in range (0 .. 1000).

### Zoom

**zoom-cursor:** Variable, Read/Write, Type: float in range (1 .. 1000).

**zoom-icon-step:** Variable, Read/Write, Type: float in range (1 .. 1000).

## Selections

The following script commands have to do with making selections, a fundamental operation. HyperChem generally operates on a selection rather than on the whole molecular system and these selection scripts are often used in conjunction with other operations.

### Select Options

**multiple-selections:** Variable, Read/Write, Type: boolean.

**select-sphere:** Variable, Read/Write, Type: boolean.

**selection-target:** Variable, Read/Write, Type: enum

### Select

**select-none:** Command, Arg list: (void).

**select-atom:** Command, Arg list: integer, integer.

**select-path:** Command, Arg list: integer, integer, integer, integer.

**select-residue:** Command, Arg list: integer, integer.

**un-select-atom:** Command, Arg list: integer, integer.

**un-select-residue:** Command, Arg list: integer, integer.

### Ask About Selection

**selected-atom-count:** Variable, Readonly, Type: integer.

**selected-atom:** Variable, Readonly, Type: vector of integer, integer.

**selection-value:** Variable, Readonly, Type: float.

### Operate on Selection

**delete-selected-atoms:** Command, Arg list: (void).

**reorder-selections:** Variable, Read/Write, Type: boolean.

### Named Selections

**name-selection:** Command, Arg list: string.

**delete-named-selection:** Command, Arg list: string.  
**named-selection-count:** Variable, Readonly, Type: integer.  
**named-selection-name:** Variable, Readonly, Type: vector of string.  
**named-selection-value:** Variable, Readonly, Type: vector of float.  
**select-name:** Command, Arg list: string.

## Other

**selection-color:** Variable, Read/Write, Type: enum

## File Operations

The following script commands deal with operations on the molecule files and the import/export files.

### Molecule File

**file-format:** Variable, Read/Write, Type: string.  
**path:** Variable, Read/Write, Type: string.  
**current-file-name:** Variable, Readonly, Type: string.  
**open-file:** Command, Arg list: string.  
**merge-file:** Command, Arg list: string.  
**write-file:** Command, Arg list: string.  
**delete-file:** Command, Arg list: string.

### Options

**file-needs-saved:** Variable, Read/Write, Type: boolean.  
**velocities-in-hin-file:** Variable, Read/Write, Type: boolean.  
**view-in-hin-file:** Variable, Read/Write, Type: boolean.

### PDB File

**connectivity-in-pdb-file:** Variable, Read/Write, Type: boolean.  
**hydrogens-in-pdb-file:** Variable, Read/Write, Type: boolean.

**non-standard-pdb-names:** Variable, Read/Write, Type: boolean.

## Import/Export

**import-dipole:** Variable, Read/Write, Type: boolean.

**import-ir:** Variable, Read/Write, Type: boolean.

**import-orbitals:** Variable, Read/Write, Type: boolean.

**import-property-file:** Command, Arg list: string.

**import-uv:** Variable, Read/Write, Type: boolean.

**export-dipole:** Variable, Read/Write, Type: boolean.

**export-ir:** Variable, Read/Write, Type: boolean.

**export-orbitals:** Variable, Read/Write, Type: boolean.

**export-property-file:** Command, Arg list: string.

**export-uv:** Variable, Read/Write, Type: boolean.

## Other

**file-diff-message:** Command, Arg list: string, string, string, string.

**write-atom-map:** Command, Arg list: string.

## Scripts

The following script commands have to do with the process of scripting itself. They are used to open script files, control execution of script commands, or control the process of notification. Certain other scripts control the OMSG output of a query or manage the menu items inserted under the script menu. Finally, values of HSVs can be pushed and popped with a stack.

## Script Files

**read-script:** Command, Arg list: string.

**read-tcl-script:** Command, Arg list: string.

**compile-script-file:** Command, Arg list: string, string.

**read-binary-script:** Command, Arg list: string.

## Execution

**query-value:** Command, Arg list: .

**execute-string:** Command, Arg list: string.

**pause-for:** Command, Arg list: integer in range (0 .. 32767).

**exit-script:** Command, Arg list: (void).

## Notifications

**notify-on-update:** Command, Arg list: string.

**cancel-notify:** Command, Arg list: string.

**notify-with-text:** Variable, Read/Write, Type: boolean.

**variable-changed:** Command, Arg list: string.

## OMSGs

**append-omsgs-to-file:** Command, Arg list: string.

**omsg-file:** Variable, Read/Write, Type: string.

**omsgs-not-to-file:** Command, Arg list: (void).

**omsgs-to-file:** Command, Arg list: string.

**query-response-has-tag:** Variable, Read/Write, Type: boolean.

## Menus

**change-user-menuitem:** Command, Arg list: integer, string, string.

**script-menu-caption:** Variable, Read/Write, Type: vector of string.

**script-menu-checked:** Variable, Read/Write, Type: vector of boolean.

**script-menu-command:** Variable, Read/Write, Type: vector of string.

**script-menu-enabled:** Variable, Read/Write, Type: vector of boolean.

**script-menu-help-file:** Variable, Read/Write, Type: vector of string.

**script-menu-help-id:** Variable, Read/Write, Type: vector of integer.

**script-menu-in-use:** Variable, Read/Write, Type: vector of boolean.

**script-menu-message:** Variable, Read/Write, Type: vector of string.

## Stack Operation

**pop-no-value:** Command, Arg list: string.

**pop-value:** Command, Arg list: string.

**push:** Command, Arg list: string.

## Other

**execute-client:** Command, Arg list: string.

**execute-hyperchem-client:** Command, Arg list: string.

**message:** Variable, Read/Write, Type: string.

**one-line-arrays:** Variable, Read/Write, Type: boolean.

## Info

The following script commands are part of a capability for enquiring and obtaining information about a specific HSV.

**info-access:** Variable, Readonly, Type: string.

**info-enum-id-of:** Variable, Readonly, Type: string.

**info-enum-list:** Variable, Readonly, Type: string.

**info-factory-setting:** Variable, Readonly, Type: string.

**info-id-of:** Variable, Readonly, Type: integer.

**info-type-of:** Variable, Readonly, Type: string.

**info-type-of-element:** Variable, Readonly, Type: string.

**info-variable-target:** Variable, Read/Write, Type: string.

## Errors

The following script commands deal with errors.

**no-source-refs-in-errors:** Command, Arg list: (void).

**source-refs-in-errors:** Command, Arg list: (void).

**script-refs-in-errors:** Variable, Read/Write, Type: boolean.

**error:** Variable, Read/Write, Type: string.

**errors-are-not-omsgs:** Command, Arg list: (void).

**errors-are-omsgs:** Command, Arg list: (void).

**ignore-script-errors:** Variable, Read/Write, Type: boolean.

**hide-errors:** Variable, Read/Write, Type: boolean.

## Logging

The following script commands have to do with the process of creating a log file.

**start-logging:** Command, Arg list: string, boolean.

**stop-logging:** Command, Arg list: (void).

**log-comment:** Command. Arg list: string.

**mechanics-print-level:** Variable, Read/Write, Type: integer in range (0 .. 9).

**quantum-print-level:** Variable, Read/Write, Type: integer in range (0 .. 9).

## Auxiliary

The following script commands defy simple classification.

### Declarations

**declare-integer:** Command, Arg list: string.

**declare-string:** Command, Arg list: string.

### Warnings

**warning:** Variable, Read/Write, Type: string.

**warning-type:** Variable, Read/Write, Type: enum(none, log, message).

**hide-warnings:** Variable, Read/Write, Type: boolean.

**warnings-are-not-omsgs:** Command, Arg list: void.

**warnings-are-omsgs:** Command, Arg list: void.

## Screen Output

**status-message:** Variable, Read/Write, Type: string.

**request:** Command, Arg list: string.

## Version

**version:** Variable, Readonly, Type: string.

**serial-number:** Variable, Readonly, Type: string.

## Other

**print-variable-list:** Command, Arg list: string.

**toggle:** Command, Arg list: string.

## Viewing

The following script commands are associated with the manipulations determining what one sees on the screen excluding the specific molecular rendering. Some of them are simply viewing transformations. Others move the molecules or show attributes of the molecules.

## Alignment

**align-molecule:** Command, Arg list: list of enums

**align-viewer:** Command, Arg list: enum.

## Redisplay

**global-inhibit-redisplay:** Variable, Readonly, Type: boolean.

**inhibit-redisplay:** Variable, Read/Write, Type: boolean.

## Rotation

**rotate-molecules:** Command, Arg list: enum, float.

**rotate-viewer:** Command, Arg list: enum, float.



## Translation

**translate-selection:** Command, Arg list: float, float, float.

**translate-view:** Command, Arg list: float, float, float.

**translate-whole-molecules:** Variable, Read/Write, Type: boolean.

**translate-merged-systems:** Variable, Read/Write, Type: boolean.

**use-fast-translation:** Variable, Read/Write, Type: boolean.

## Window

**window-height:** Variable, Read/Write, Type: integer.

**window-width:** Variable, Read/Write, Type: integer.

**workspace-height:** Variable, Read/Write, Type: integer

**workspace-width:** Variable, Read/Write, Type: integer

## Other

**show-perspective:** Variable, Read/Write, Type: boolean.

**wall-eyed-stereo:** Variable, Read/Write, Type: boolean.

**zoom:** Command, Arg list: float in range (0.01 .. 50).

**show-axes:** Variable, Read/Write, Type: boolean.

**show-dipoles:** Variable, Read/Write, Type: boolean.

## Rendering

The following script commands affect the molecular rendering of the molecule in the workspace.

### General Options

**bond-spacing-display-ratio:** Variable, R/W, Type: float in range (0 .. 1).

**cpk-max-double-buffer-atoms:** Variable, Read/Write, Type: integer.

**dot-surface-angle:** Variable, R/W, Type: float angle in range (-90 .. 90).

**double-buffered-display:** Variable, Read/Write, Type: boolean.

**render-method:** Variable, Read/Write, Type: enum.

## Specific Rendering Options

- balls-highlighted**: Variable, R/W, Type: boolean
- balls-radius-ratio**: Variable, R/W, Type: float in range (0 .. 1).
- balls-shaded**: Variable, R/W, Type: boolean
- cylinders-color-by-element**: Variable, R/W, Type: boolean
- cylinders-width-ratio**: Variable, R/W, Type: float in range (0 .. 1).
- spheres-highlighted**: Variable, R/W, Type: boolean
- spheres-shaded**: Variable, R/W, Type: boolean
- sticks-width**: Variable, R/W, Type: integer in range (0 .. 25)

## Show - Don't Show

- show-aromatic-rings-as-circles**: Variable, Read/Write, Type: boolean.
- show-gradient-vectors**: Variable, Read/Write, Type: boolean.
- show-hydrogen-bonds**: Variable, Read/Write, Type: boolean.
- show-hydrogens**: Variable, Read/Write, Type: boolean.
- show-isosurface**: Command, Arg list: boolean.
- show-multiple-bonds**: Variable, Read/Write, Type: boolean.
- show-periodic-box**: Variable, Read/Write, Type: boolean.
- show-ribbons**: Variable, Read/Write, Type: boolean.
- show-stereo**: Variable, Read/Write, Type: boolean.
- show-stereochem-wedges**: Variable, Read/Write, Type: boolean.
- show-vibrational-vectors**: Variable, Read/Write, Type: boolean.

## Coloring and Labeling

The following script commands affect the showing of labels for atoms or residues as well as the colors that appear on the screen for various objects.

### Color

- atom-color**: Variable, Read/Write, Type: array of enum.

**bond-color:** Variable, Read/Write, Type: enum.  
**color-element:** Command, Arg list: integer, enum.  
**color-selection:** Command, Arg list: string.  
**negatives-color:** Variable, Read/Write, Type: enum.  
**positives-color:** Variable, Read/Write, Type: enum.  
**revert-element-colors:** Command, Arg list: (void).  
**window-color:** Variable, Read/Write, Type: enum.

## Labels

**atom-label-text:** Variable, Readonly, Type: array of string.  
**atom-labels:** Variable, Read/Write, Type: enum.

## Images

The following script commands deal with getting bitmaps and metafiles (images) of molecules into the clipboard or into a file.

**image-color:** Variable, Read/Write, Type: boolean.  
**image-destination-clipboard:** Variable, Read/Write, Type: boolean.  
**image-destination-file:** Variable, Read/Write, Type: boolean.  
**image-destination-filename:** Variable, Read/Write, Type: string  
**image-file-bitmap:** Variable, Read/Write, Type: boolean.  
**image-file-bitmapRGB:** Variable, Read/Write, Type: boolean.  
**image-file-metafile:** Variable, Read/Write, Type: boolean.  
**image-include-cursor:** Variable, Read/Write, Type: boolean.  
**image-source-window:** Variable, Read/Write, Type: enum.

## Model Building

The following script commands deal with aspect of drawing and creating molecules with the model builder.

## Options

**allow-ions:** Variable, Read/Write, Type: boolean.

**explicit-hydrogens:** Variable, Read/Write, Type: boolean.

**default-element:** Variable, Read/Write, Type: integer in range (0 .. 103).

## Drawing

**create-atom:** Command, Arg list: integer in range (0 .. 103).

**set-bond:** Command, Arg list: integer, integer, integer, integer, enum.

**delete-atom:** Command, Arg list: integer, integer.

## Constraints

**constrain-geometry:** Command, Arg list: string.

**constrain-bond-length:** Command, Arg list: float in range (0 .. 100).

**constrain-bond-angle:** Command, Arg list: float angle in range (-360 .. 360).

**constrain-bond-torsion:** Command, Arg list: angle in range (-360 ... 360)

**constrain-drawing:** Variable, Read/Write. Type: boolean

**unconstrain-bond-length:** Command, Arg list: (void).

**unconstrain-bond-angle:** Command, Arg list: (void).

**unconstrain-bond-torsion:** Command, Arg list: (void).

## Other

**is-ring-atom:** Variable, Readonly, Type: array of boolean.

**neighbors:** Variable, Readonly, Type: array of (unknown).

## Stereochemistry

The following script commands deal with aspects of creating and showing specific stereochemistry.

**builder-enforces-stereo:** Variable, Read/Write, Type: boolean.

**change-stereochem:** Command, Arg list: integer, integer.

**chirality**: Variable, Read/Write, Type: array of string.  
**constrain-bond-down**: Command, Arg list: integer, integer, integer, integer.  
**constrain-bond-up**: Command, Arg list: integer, integer, integer, integer.  
**constrain-change-stereo**: Command, Arg list: integer, integer.  
**constrain-fix-stereo**: Command, Arg list: integer, integer.  
**cycle-atom-stereo**: Command, Arg list: integer, integer.  
**cycle-bond-stereo**: Command, Arg list: integer, integer, integer, integer.  
**remove-all-stereo-constraints**: Command, Arg list: (void).  
**remove-stereo-constraint**: Command, Arg list: integer, integer.  
**set-atom-charges-to-formal**: Command, Arg list: (void).

## Atom Properties

The following script commands deal with changing or displaying properties of individual atoms perhaps associated with labels.

### Labels

**atom-charge**: Variable, Read/Write, Type: array of float.  
**atom-mass**: Variable, Read/Write, Type: array of float.  
**atom-name**: Variable, Read/Write, Type: array of string.  
**atom-spin-density-at-nucleus**: Variable, Read/Write, array of float.  
**atom-spin-population**: Variable, Read/Write, Type: array of float.  
**atom-type**: Variable, Read/Write, Type: array of string.  
**atomic-number**: Variable, Read/Write, Type: array of integer.  
**atomic-symbol**: Variable, Readonly, Type: array of string.  
**set-atom-charge**: Command, Arg list: float in range (-100 .. 100).  
**set-atom-type**: Command, Arg list: string.

### Coordinates and Velocities

**coordinates**: Variable, Read/Write, Type: array of float, float, float.  
**gradient-x**: Variable, Reaonly, Type: float

**gradient-y:** Variable, Readonly, Type: float

**gradient-z:** Variable, Readonly, Type: float

**gradients:** Variable, Read/Write, Type: array of float, float, float

**set-velocity:** Command, Arg list: enum, float, float, float.

**velocities:** Variable, Read/Write, Type: array of float, float, float

**set-bond-angle:** Command, Arg list: float angle in range (0 .. 180).

**set-bond-length:** Command, Arg list: float in range (0 .. 3200).

**set-bond-torsion:** Command, Arg list: float angle in range (-360 .. 360).

## Other

**coordination:** Variable, Readonly, Type: array of integer.

## Molecule Properties

The following script commands deal with some properties of molecules or the atoms in molecules.

### Charge-Multiplicity

**multiplicity:** Variable, Read/Write, Type: integer in range (1 .. 6).

**quantum-total-charge:** Variable, R/W, Type: integer

### Counts

**atom-count:** Variable, Readonly, Type: vector of integer.

**molecule-count:** Variable, Readonly, Type: integer.

### Labels

**molecule-name:** Variable, Readonly, Type: vector.

**name-molecule:** Command, Arg list: string.

### Properties

**dipole-moment:** Variable, R/W , Type: float in range (-1e+010 .. 1e+010).

**dipole-moment-components:** Variable, R/W, Type: float, float, float.

**heat-of-formation:** Variable, R/W, Type: float in range (-1e+10 .. 1e+10).

**moments-of-inertia:** Variable, Readonly, Type: float float float.

## Back Ends

The following script commands deal with the operation of the computational back ends and communication with them.

### Basic

**backend-active:** Variable, Read/Write, Type: boolean.

**backend-communications:** Variable, Read/Write, Type: enum.

### Large Communication Structures

**atom-info:** Variable, Readonly, Type: (unknown).

**mechanics-info:** Variable, Readonly, Type: (unknown).

**mechanics-data:** Variable, Readonly, Type: (unknown).

### Remote Back Ends

**backend-host-name:** Variable, Read/Write, Type: string.

**backend-process-count:** Variable, R/W, Type: integer in range (1 .. 32).

**backend-user-id:** Variable, Read/Write, Type: string.

**backend-user-password:** Variable, Read/Write. Type: string.

## Molecular Mechanics Calculations

The following script commands deal with computations in molecular mechanics.

### Method

**molecular-mechanics-method:** Variable, Read/Write, Type: enum.

**is-extended-hydrogen:** Variable, Readonly, Type: array of boolean.

**keep-atom-charges:** Variable, Read/Write, Type: boolean.

## Energy Components

**bend-energy:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).

**stretch-energy:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).

**torsion-energy:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).

**nonbond-energy:** Variable, R only, Type: float in range (-1e+10 .. 1e+10).

**estatic-energy:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).

**hbond-energy:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).

## Cutoffs

**cutoff-type:** Variable, Read/Write, Type: enum.

**cutoff-inner-radius:** Variable, Read/Write, Type: float in range (0 .. 1e+10).

**cutoff-outer-radius:** Variable, Read/Write, Type: float in range (0 .. 1e+10).

## Scale Factors

**mechanics-dielectric:** Variable, R/W, Type: enum, enum, enum, enum.

**mechanics-dielectric-scale-factor:** Variable, R/W, Type: four floats.

**mechanics-electrostatic-scale-factor:** Variable, R/W, Type: four floats.

**mechanics-mmp-electrostatics:** Variable, Read/Write, Type: enum.

**mechanics-van-der-waals-scale-factor:** Variable, R/W, Type: four floats.

## Parameters

**parameter-set-changed:** Variable, Read/Write, Type: boolean.

**use-parameter-set:** Command, Arg list: string.

## Amino Acids and Nucleic Acids

The following script commands pertain to amino acid templates and the construction of peptides or to nucleic acid templates and the construction of DNA-like structures.

### Amino Acids

**add-amino-acid:** Command, Arg list: string.



**amino-alpha-helix:** Command, Arg list: (void).

**amino-beta-sheet:** Command, Arg list: (void).

**amino-isomer:** Variable, Read/Write, Type: enum.

**amino-omega:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**amino-phi:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**amino-psi:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

## Nucleic Acids

**add-nucleic-acid:** Command, Arg list: string.

**nucleic-a-form:** Command, Arg list: (void).

**nucleic-alpha:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**nucleic-b-form:** Command, Arg list: (void).

**nucleic-backwards:** Variable, Read/Write, Type: boolean.

**nucleic-beta:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**nucleic-chi:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**nucleic-delta:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

**nucleic-double-strand:** Variable, Read/Write, Type: boolean.

**nucleic-epsilon:** Variable, R/W, Type: float angle in range (-360 .. 360).

**nucleic-gamma:** Variable, R/W, Type: float angle in range (-360 .. 360).

**nucleic-sugar-pucker:** Variable, Read/Write, Type: enum.

**nucleic-z-form:** Command, Arg list: (void).

**nucleic-zeta:** Variable, Read/Write, Type: float angle in range (-360 .. 360).

## General Residue

**mutate-residue:** Command, Arg list: string.

**residue-charge:** Variable, Readonly, Type: array of float.

**residue-coordinates:** Variable, Readonly, Type: array of float, float, float.

**residue-count:** Variable, Readonly, Type: vector of integer.

**residue-label-text:** Variable, Readonly, Type: array of string.

**residue-labels:** Variable, Read/Write, Type: enum.

**residue-name:** Variable, Readonly, Type: array of string.

## Molecular Dynamics and Monte Carlo

The following script commands pertain to the two type of molecular dynamics (normal and Langevin) and the very closely related Monte Carlo calculations.

### Basic

**do-molecular-dynamics:** Command, Arg list: (void)

**do-langevin-dynamics:** Command, Arg list: (void)

**do-monte-carlo:** Command, Arg list: (void)

**dynamics-restart:** Variable, Read/Write, Type: boolean.

**dynamics-average-period:** Variable, R/W, Type: integer.

**dynamics-collection-period:** Variable, R/W, Type: integer.

**screen-refresh-period:** Variable, Read/Write, Type: integer.

### Run Parameters

**dynamics-bath-relaxation-time:** Variable, R/W, Type: float.

**dynamics-constant-temp:** Variable, Read/Write, Type: boolean.

**dynamics-cool-time:** Variable, Read/Write, Type: float.

**dynamics-heat-time:** Variable, Read/Write, Type: float.

**dynamics-final-temp:** Variable, R/W, Type: float.

**dynamics-friction-coefficient:** Variable, R/W, Type: float

**dynamics-run-time:** Variable, Read/Write, Type: float.

**dynamics-seed:** Variable, Read/Write, Type: integer.

**dynamics-simulation-temp:** Variable, Read/Write, Type: float.

**dynamics-starting-temp:** Variable, Read/Write, Type: float.

**dynamics-temp-step:** Variable, Read/Write, Type: float.

**dynamics-time-step:** Variable, Read/Write, Type: float.

## Averaging

**append-dynamics-average:** Command, Arg list: string.

**append-dynamics-graph:** Command, Arg list: string.

**dynamics-info-elapsed-time:** Variable, R only, Type: float.

**dynamics-info-kinetic-energy:** Variable, Readonly, Type: float.

**dynamics-info-last-update:** Variable, Readonly, Type: boolean.

**dynamics-info-potential-energy:** Variable, Readonly, Type: float.

**dynamics-info-temperature:** Variable, Readonly, Type: float.

**dynamics-info-total-energy:** Variable, Readonly, Type: float.

## Playback

**dynamics-playback:** Variable, Read/Write, Type: enum.

**dynamics-playback-end:** Variable, Read/Write, Type: integer.

**dynamics-playback-period:** Variable, Read/Write, Type: integer.

**dynamics-playback-start:** Variable, Read/Write, Type: integer.

**dynamics-snapshot-filename:** Variable, Read/Write, Type: string.

**dynamics-snapshot-period:** Variable, Read/Write, Type: integer.

## Monte Carlo Specific

**monte-carlo-cool-steps:** Variable, Read/Write, Type: float

**monte-carlo-heat-steps:** Variable, Read/Write, Type: float

**monte-carlo-info-acceptance-ratio:** Variable, Readonly, Type: float

**monte-carlo-max-delta:** Variable, R/W, Type: float

**monte-carlo-run-steps:** Variable, R/W, Type: float

## Optimization

The following script commands deal with facets of geometry optimization.

### Basic

**do-optimization:** Command, Arg list: (void).

**optim-algorithm:** Variable, Read/Write, Type: enum.  
**optim-converged:** Variable, Readonly, Type: boolean.  
**optim-convergence:** Variable, Read/Write, Type: float in range (0 .. 100).  
**optim-max-cycles:** Variable, Read/Write, Type: integer in range (1 .. +Inf).  
**rms-gradient:** Variable, Readonly, Type: float in range (-1e+10 .. 1e+10).  
**state-to-optimize:** Variable, Read/Write, Type: integer.

## Restrains

**restraint:** Command, Arg list: string, float, float.  
**restraint-tether:** Command, Arg list: complex.  
**use-no-restraints:** Command, Arg list: (void).  
**use-restraint:** Command, Arg list: string, boolean.

## General Quantum Mechanics

The following script commands deal with facets of quantum mechanical computations that are independent of the quantum mechanical method, whether it is semi-empirical or ab initio.

### Input Parameters

**do-qm-calculation:** Variable, Read/Write, Type: boolean.  
**uhf:** Variable, Read/Write, Type: boolean.  
**accelerate-scf-convergence:** Variable, Read/Write, Type: boolean.  
**alpha-orbital-occupancy:** Variable, Read/Write, Type: vector of float.  
**beta-orbital-occupancy:** Variable, Read/Write, Type: vector of float.  
**excited-state:** Variable, Read/Write, Type: boolean.  
**max-iterations:** Variable, Read/Write, Type: integer in range (1 .. 32767).  
**scf-convergence:** Variable, Read/Write, Type: float in range (0 .. 100).  
**polar-field-strength:** Variable, Read/Write, Type: float  
**polarizability-calculated:** Variable, Read/Write, Type: Boolean

## Output Results

**orbital-count:** Variable, Readonly, Type: integer.

**alpha-scf-eigenvector:** Variable, Read/Write, Type: vector of float-list.

**beta-scf-eigenvector:** Variable, Read/Write, Type: vector of float-list.

**scf-orbital-energy:** Variable, Read/Write, Type: vector of float.

**orbital-results:** Variable, Read/Write, Type: vector of float-list.

**scf-electronic-energy:** Variable, Read/Write, Type: float.

## Semi-empirical Calculations

The following script commands deal with the input parameters required to specifically perform semi-empirical calculations and with their results. Some of the input parameters for semi-empirical methods are essentially part of the \* .abp parameter files.

### General

**semi-empirical-method:** Variable, Read/Write, Type: enum.

**d-orbitals-on-second-row:** Variable, Read/Write, Type: boolean.

**scf-atom-energy:** Variable, Read/Write, Type: float.

**scf-binding-energy:** Variable, Read/Write, Type: float.

**scf-core-energy:** Variable, Read/Write, Type: float.

### Huckel

**huckel-constant:** Variable, Read/Write, Type: float in range (0 .. 10).

**huckel-scaling-factor:** Variable, R/W, Type: float in range (0 .. 100000).

**huckel-weighted:** Variable, Read/Write, Type: boolean.

### ZINDO

**zindo-1-pi-pi:** Variable, Read/Write, Type: float in range (0 .. 2).

**zindo-1-sigma-sigma:** Variable, Read/Write, Type: float in range (0 .. 2).

**zindo-s-pi-pi:** Variable, Read/Write, Type: float in range (0 .. 2).

**zindo-s-sigma-sigma:** Variable, Read/Write, Type: float in range (0 .. 2).

## Ab Initio Calculations

The following script commands pertain to performing ab initio calculations - their inputs and their results.

### Input Options

**abinitio-calculate-gradient:** Variable, Read/Write, Type: boolean.

**abinitio-d-orbitals:** Variable, Read/Write, Type: boolean.

**abinitio-f-orbitals:** Variable, Read/Write, Type: boolean.

**abinitio-direct-scf:** Variable, Read/Write, Type: boolean.

**abinitio-mo-initial-guess:** Variable, Read/Write, Type: enum.

**abinitio-mp2-correlation-energy:** Variable, Read/Write, Type: boolean.

**abinitio-mp2-frozen-core:** Variable, Read/Write, Type: boolean.

**abinitio-scf-convergence:** Variable, R/W, Type: float in range (0 .. 100).

**abinitio-use-ghost-atoms:** Variable, Read/Write, Type: boolean.

### Basis Set

**assign-basisset:** Command, Arg list: string.

**atom-basisset:** Variable, Read/Write, Type: array of string.

**atom-extra-basisset:** Variable, Read/Write, Type: array of string, float.

**auxilliary-basis:** Variable, Read/Write, Type: integer.

**basisset-count:** Variable, Readonly, Type: integer.

**use-auxilliary-basis:** Variable, Read/Write, Type: boolean.

### 2-electron Integrals

**abinitio-buffer-size:** Variable, Read/Write, Type: integer.

**abinitio-cutoff:** Variable, Read/Write, Type: float in range (0 .. 1e+10).

**abinitio-integral-format:** Variable, Read/Write, Type: enum.

**abinitio-integral-path:** Variable, Read/Write, Type: string.

### Results

**mp2-energy:** Variable, Read/Write, Type: float.

## TNDO

**tndo-version:** Variable, Read/Write, Type: integer.

## DFT

**correlation-functional:** Variable, Read/Write, Type: enum.

**exchange-functional:** Variable, Read/Write, Type: enum

**small-density-cutoff:** Variable, Read/Write, Type: float

**use-mullikan-approximation-to-density:** Variable, Read/Write, Type: boolean

**use-small-density-cutoff:** Variable, Read/Write, Type: boolean.

## Configuration Interaction

The following script commands are relevant to post-SCF configuration interaction calculations.

**ci-criterion:** Variable, Read/Write, Type: enum.

**ci-excitation-energy:** Variable, Read/Write, Type: float in range (0 .. 10000).

**ci-occupied-orbitals:** Variable, R/W, Type: integer in range (0 .. 32767).

**ci-state-to-optimize:** Command, Arg list: unregistered.

**ci-unoccupied-orbitals:** Variable, R/W, Type: integer in range (0 .. 32767).

**configuration-interaction:** Variable, Read/Write, Type: enum.

## Infrared Spectra

The following script commands are relevant to the calculation and display of vibrational spectra.

### Animations

**animate-vibrations:** Variable, Read/Write, Type: boolean.

**ir-animate-amplitude:** Variable, Read/Write, Type: float in range (0 .. 10).

**ir-animate-cycles:** Variable, Read/Write, Type: integer in range (0 .. +Inf).

**ir-animate-steps:** Variable, Read/Write, Type: integer in range (3 .. +Inf).

## Spectra

**ir-band-count:** Variable, Read/Write, Type: integer.

**vibrational-mode:** Variable, Read/Write, Type: integer.

**ir-frequency:** Variable, Read/Write, Type: vector of float.

**ir-intensity:** Variable, Read/Write, Type: vector of float.

**ir-intensity-components:** Variable, R/W, Type: vector of float, float, float.

**ir-normal-mode:** Variable, Read/Write, Type: vector of float-list.

## UV Spectra

The following script commands are relevant to the calculation of electronic spectra.

**configuration:** Variable, Read/Write, Type: integer.

**uv-band-count:** Variable, Read/Write, Type: integer.

**uv-dipole-components:** Variable, Read/Write, Type: vector of float-list.

**uv-energy:** Variable, Read/Write, Type: vector of float.

**uv-oscillator-strength:** Variable, Read/Write, Type: vector of float.

**uv-spin:** Variable, Read/Write, Type: vector of float.

**uv-total-dipole:** Variable, Read/Write. Type: vector of float.

**uv-transition-dipole:** Variable, R/W, Type: vector of float, float, float.

## Plotting

the following script commands are relevant to the plotting of 2D and 3D contours and renderings of orbitals, electron density, spin density and electrostatic potentials.

### General Options

**graph-beta:** Variable, Read/Write, Type: boolean.

**graph-data-type:** Variable, Read/Write, Type: enum.



**graph-orbital-selection-type:** Variable, Read/Write, Type: enum.

**graph-orbital-offset:** Variable, Read/Write, Type: integer in range (0 .. +Inf).

## 2D

**do-qm-graph:** Variable, Read/Write, Type: boolean.

**graph-contour-increment:** Variable, Read/Write, Type: float.

**graph-contour-increment-other:** Variable, Read/Write, Type: boolean.

**graph-contour-levels:** Variable, Read/Write, Type: integer.

**graph-contour-start:** Variable, Read/Write, Type: float.

**graph-contour-start-other:** Variable, Read/Write, Type: boolean.

## 3D

**do-qm-isosurface:** Variable, Read/Write, Type: boolean.

**isosurface-grid-step-size:** Variable, R/W, Type: float in range (0 .. 1e+10)

**isosurface-hide-molecule:** Variable, Read/Write, Type: boolean

**isosurface-map-function:** Variable, Read/Write, Type: boolean

**isosurface-map-function-display-legend:** Variable, R/W, Type: boolean

**isosurface-map-function-range:** Variable, Read/Write, Type: float, float

**isosurface-mesh-quality:** Variable, Read/Write, Type: enum

**isosurface-render-method:** Variable, Read/Write, Type: enum

**isosurface-threshold:** Variable, Read/Write, Type: float.

**isosurface-transparency-level:** Variable, R/W, Type: float in range (0 .. 1)

**isosurface-x-min:** Variable, Read/Write, Type: float.

**isosurface-x-nodes:** Variable, R/W, Type: integer in range (0 .. 32767).

**isosurface-y-min:** Variable, Read/Write, Type: float.

**isosurface-y-nodes:** Variable, R/W, Type: integer in range (0 .. 32767).

**isosurface-z-min:** Variable, Read/Write, Type: float.

**isosurface-z-nodes:** Variable, R/W, Type: integer in range (0 .. 32767).

## Grid

**graph-data-row:** Variable, Readonly, Type: vector of float-list.

**graph-horizontal-grid-size:** Variable, R/W, Type: integer in (2 .. 8192).

**graph-plane-offset:** Variable, Read/Write, Type: float.

**graph-vertical-grid-size:** Variable, R/W, Type: integer in range (2 .. 8192).

**grid-max-value:** Variable, Readonly, Type: float.

**grid-min-value:** Variable, Readonly, Type: float.

**isosurface-grid-step-size:** Variable, R/W, Type: float in range (0 .. 1e+10).



# Appendix B

## Listing of Tcl Commands

### The Tcl Commands

In this appendix we give a very brief alphabetic list of the most important Tcl commands and functions. They are listed with their arguments, with optional arguments in angular brackets. However, no detailed description of each command is given here and a reference book on Tcl is strongly suggested.

abs (x)  
acos (x)  
append varName value <value ...>  
asin (x)  
atan(x)  
atan2 (x,y)  
break  
catch command <varName>  
cd <dirName>  
ceil (x)  
close fileId  
concat <list list ...>  
continue  
cos (x)  
cosh (x)  
double (i)  
eof fileId

error message <info> <code>  
eval arg <arg arg ...>  
exec <-keepnewline> <- -> <arg ..>  
exit <code>  
exp (x)  
expr arg <arg arg ...>  
file atime name  
file dirname name  
file executable name  
file exists name  
file extension name  
file isdirectory name  
file isfile name  
file lstat name arrayName  
file mtime name  
file option name <arg arg ...>  
file owned name  
file readable name  
file readlink name  
file rootname name  
file size name  
file stat name arrayName  
file tail name  
file type name  
file writable name  
floor (x)  
flush file Id  
fmod (x,y)  
for init test reinit body

```

foreach varName list body
format formatString <value value ...>
gets file Id <varName>
glob <-nocomplain> <- -> pattern <pattern ...>
global name1 <name2 ...>
hypot (x,y)
if test1 body1 <elseif test2 body2 elseif ...> <else bodyn>
incr varName <increment>
int (x)
join list <joinString>
lappend varName value <value ...>
lindex list index
linsert list index value <value ...>
list <value value...>
llength list
log (x)
log10 (x)
lrange list first last
lreplace list first last <value value ...>
lsearch <-exact> <-glob> <-regexp> list pattern
lsort <-ascii> <-integer> <-real> <-command command> \ <-increasing> <-
decreasing> list
open l command <access>
open name <access>
pid <fileId>
pow (x,y)
proc name argList body
puts <-nonewline> <fileId> string
pwd

```

```
read <-nonewline> fileId
read fileId numBytes
regexp <-indices> <-nocase> <- -> exp string <matchVar> \ <subVar subVar
...>
regsub <-all> <-nocase> <- -> exp string subSpec varName
return <-code code> <-errorinfo info> <-errorcode code> <string>
return <options> <value>
round (x)
scan string format varName <varName varName ...>
seek fileId offset <origin>
set varName <value>
sin (x)
sinh (x)
source fileName
split string <splitChars>
sqrt (x)
string compare string1 string2
string first string1 string2
string index string charIndex
string last string1 string2
string length string
string match pattern string
string range string first last
string tolower string
string toupper ststring
string trim string <chars>
string trimleft string <chars>
string trimright string <chars>
switch <options> string pattern body <pattern body ...>
```

```
switch <options> string (pattern body <pattern body ...>)  
tan (x)  
tanh (x)  
tell fileId  
unset varName <varName varName ...>  
uplevel <level> arg <arg arg ...>  
upvar <level> otherVar1 my Var1 <otherVar2 myVar2 ...>  
while test body
```





# Appendix C

## Classification of HAPI Calls

This appendix lists each of the HAPI calls and gives details as to their use, their declaration and their parameters. The HAPI calls are classified according to their use. Visual Basic Arguments listed as N/A can be inferred from the corresponding C/C++ call.

### The API functions

#### Functions for Initialization and Termination

*The functions in this group are responsible for initialization of the HAPI library, establishing a connection with HyperChem and termination of the connection.*

## *hcInitAPI*

*The function performs initialization of the HAPI.*

#### API header

```
BOOL _stdcall hcInitAPI()
```

#### FORTRAN interface

```
logical function hfInitAPI()
```

#### VISUAL BASIC declaration

```
Declare Function hbInitAPI Lib "hapi.dll" Alias "hcInitAPI" () As Long
```

## Parameters

*The function takes no parameters.*

## Return Value

*The function returns TRUE if initialization was successful. The function returns FALSE if the user application is already connected with HyperChem.*

## Remarks

*There is no need for explicitly calling the function if you use any of the recommended methods described in the section "How to use the CDK API" of Chapter 11. Both LoadHAPI and automatic DLL initialization code call the function. However the call may be required if you perform your own Run-Time Load for an unusual application.*

# hcConnect

*The function establishes a link between the user application and HyperChem. It must be called before ANY other function is called except for auxiliary functions.*

## API header

```
BOOL _stdcall hcConnect(LPSTR lszCmd);
```

## FORTRAN interface

```
logical function hfConnect(init_string)  
character*(*) init_string
```

## VISUAL BASIC declaration

```
Declare Function hbConnect Lib "hapi.dll" Alias "hcConnect" (ByVal command As String) As Long
```

## Parameters

*C/C++:* LPSTR lszCmd - command line received from HyperChem or empty string ("" ). See Remarks.

**FORTTRAN:** *character\*(\*) init\_string - command line received from HyperChem or empty string (Fortran string). See Remarks.*

**VB:** *command - VB variable of string type*

## Return Value

The function returns TRUE(1) when it finds HyperChem and establishes a connection to it. When it fails, it returns FALSE(0).

## Remarks

The command line string `lszCmd` is received from HyperChem when the application is called via the 'execute-hyperchem-client' script command. The string contains the information required to connect the user application with a proper instance of HyperChem. When HyperChem initiates the user application after issuing a script command:

```
execute-hyperchem-client userapp.exe
```

*the user application receives a command line parameter. The parameter has the form:*

```
-hinst:ChemServer-xxxx
```

*For example:*

```
-hinst:ChemServer-510e
```

*The user application should read the command line and simply pass it as a parameter to `hcConnect`. If the parameter is not passed (`lszCmd` points to an empty string ""), the HAPI connects to the first HyperChem instance it finds. This may cause problems when there are more than one active HyperChem on the desktop.*

*If the flag `errACTION_MESS_BOX` is raised for error processing (see Auxiliary functions) the HAPI displays a message box if the function is not able to establish a connection with HyperChem. The action then depends on the user's choice (Abort/Retry/Ignore).*

*When the user ignores the message, `hcConnect` returns FALSE. See Error processing section for details.*

## See Also

*hcDisconnect*, *hcSetErrorAction*, *hcLastError*

# ***hcDisconnect***

*The function hcDisconnect closes the connection with HyperChem opened by hcConnect.*

## API header

```
BOOL hcDisconnect(void);
```

## FORTRAN interface

```
logical function hfDisconnect()
```

## VISUAL BASIC declaration

```
Declare Function hbDisconnect Lib "hapi.dll" Alias "hcDisconnect" () As Long
```

## Parameters

*The function takes no parameters*

## Return Value

*The function returns TRUE if the disconnecting was successful.*

## Remarks

*The function is called automatically when errors occurs and the user chooses ABORT as an action. The application should always close the connection before it exits, otherwise the number of open but not used DDE channels would grow unnecessarily.*

## See Also

*hcConnect*

# hcExit

*This function causes an immediate exit from the application that calls it.*

## API header

```
void hcExit(void);
```

## FORTRAN interface

```
subroutine hfExit()
```

## VISUAL BASIC declaration

```
Declare Sub hbExit Lib "hapi.dll" Alias "hcExit" ()
```

## Parameters

*The function takes no parameters.*

## Remarks

*The function should be treated as an emergency exit rather than a regular method of exiting program operation.*

## Functions for Text-based Communication

# hcExecTxt

*This function executes a HyperChem command in a text format. The function is also used for updating HSV values using a text format.*

## API header

```
BOOL hcExecTxt(LPSTR script_cmd);
```

## FORTRAN interface

```
logical function hfExecTxt(script_cmd)
character*(*) script_cmd
```

## VISUAL BASIC declaration

```
Declare Function hbExecTxt Lib "hapi.dll" Alias "hcExecTxt" (ByVal script_cmd  
As String) As Long
```

### Parameters

- C/C++:* The parameter is a NULL terminated string containing the command to be executed.
- FORTTRAN:* The parameter is Fortran string containing the command to be executed.
- VB:* The parameter is VB string containing the command to be executed

### Return Value

The function returns TRUE (1) upon successful completion. If there is an error, the user is notified by the appropriate message-box. If the error processing level is set to `errACTION_NO` (see `SetErrorAction`), the user may call `hcLastError` for info about the error.

### Remarks

The update of an HSV variable is conceptually similar to the execution of a command. Hence, to set the value of an HSV variable the user's application might use the following syntax:

```
result=hcExecTxt("coordinates(1,2)=0.1,0.2,0.3")
```

for setting the coordinates of atom 1 in molecule 2 to (0.1,0.2,0.3).

Some commands may take time to complete. The regular time-out value for command completion is set at about 65 seconds. The user may extend the time-out by calling `hcSetTimeouts`.

### See Also

`hcQueryTxt`, `hcExecBin`

## ***hcQueryTxt***

The function queries for a value of an HSV variable in text mode.

## API header

```
LPSTR hcQueryTxt(LPSTR hsv);
```

## FORTRAN interface

```
logical function hfQueryTxt(hsv, res)
character*(*) hsv,res
```

## VISUAL BASIC declaration

```
Declare Function hbQueryTxt Lib "hapi.dll" Alias "hcQueryTxt" (ByVal command
As String) As String
```

## Parameters

- C/C++:* Null terminated string containing HSV to be queried. May contain indices in the case of vector and array variables.
- FORTRAN:* 'hsv' is a Fortran string containing the HSV to be queried. 'res' is a Fortran string where the output is placed.
- VB:* VB String containing HSV name to be queried.

## Return Value

*In the C, C++, or VB version, the function allocates the memory required to hold the answer and returns a pointer or a NULL pointer if the answer is not valid. However, the Fortran interface receives logical .true. or .false. indicating successful or unsuccessful completion of the function.*

## Remarks

*When accessed from C/C++, the function returns a pointer to the newly allocated memory. Hence, the calling application is responsible for deallocating this particular memory block after it has been used, using the hcFree call.*

*The Fortran interface deallocates the memory block automatically. However, it is the calling application's responsibility to pass a reference to the Fortran string ('res') of sufficient length. If the answer is longer than provided string, the output is truncated.*

*When Querying for elements of vectors and arrays, the regular 'parenthesis' syntax is used. For example:*

```
txt_xyz=hcQueryTxt("coordinates(1,2)");
```

*for querying the coordinates of atom 1 in molecule 2.*



## See Also

*hcQueryBin*; see remarks for *hcExecTxt* for information about modifying HSV values.

## Functions for Binary Communication

### Binary Execute and Query

# *hcExecBin*

*This function executes a binary representation of a script command.*

### API header

```
BOOL hcExecBin(HSV cmd, LPV args, DWORD args_length);
```

### FORTRAN interface

```
logical function hfExecBin(cmd,args,args_length)  
integer cmd,args,args_length  
dimension args(args_length)
```

### VISUAL BASIC declaration

```
Declare Function hbExecBin Lib "hapi.dll" Alias "hcExecBin" (ByVal cmd As Long,  
ByRef args As IntBuff, ByVal args_length As Long) As Long
```

### Parameters

*C/C++:* *'cmd'* specifies the HSV code. The calling application should include an *'hsv.h'* file to have the codes available. *'args'* is a pointer to memory block containing possible arguments. *'args\_length'* is the total length of valid data pointed to by *'args'*.

*FORTRAN:* The calling application should include an *'hsv.fi'* file to have the relevant HSV codes available. The difference in the Fortran interface is that *'args'* is the reference to integer array. When transferring arguments of different types, the user has to use an equivalence instruction to pack the array with the required data type.

*VB:* The *'cmd'* and *'args\_length'* parameters have the same meaning as for C and

*Fortran. The 'args' parameter is of 'IntBuff' type. This user defined Visual Basic type is defined in 'hsv.bas' module file.*

## Return Value

*The function returns TRUE when successfully completed.*

## Remarks

*The function may be used to update the values of an HSV variable, similar to the update via the use of hcExecTxt. Data may need to be packed. However, it is highly recommended that you use instead the functions from the Binary Get/Set group, which perform all required packing internally.*

*Some commands may take time to complete. The regular time-out value for command completion is set at about 65 seconds. The user may extend the time-out by calling hcSetTimeouts*

## See Also

*hcQueryBin, hcGetxxx and hcSetxxx functions*

# hcQueryBin

*The function queries for the binary value of an HSV of any type.*

## API header

```
void* hcQueryBin(int hsv,int indx1,int indx2,int* resp_length);
```

## FORTRAN interface

```
logical function hfQueryBin(hsv,indx1,indx2,result,resp_length)
integer hsv,indx1,indx2,resp_length,result
dimension result(resp_length)
```

## VISUAL BASIC declaration

```
Declare Function hbQueryBin Lib "hapi.dll" Alias "hfQueryBin" (ByVal var As Long, ByVal indx1 As Long, ByVal indx2 As Long, ByRef result As IntBuff, ByRef cbL As Long) As Long
```

## Parameters

- C/C++:** *'hsv'* - the binary code for HSV to be queried. *'indx1'*, *'indx2'* the indexes for vector and array HSVs conforming the rules: *indx1=0, indx2=0* - queries scalar HSV, entire vector or entire array; *indx1=n, indx2=0* - queries *n*-th element of the vector HSV; *indx1=m, indx2=n* - get *m, n*-th element of the array HSV. *'resp\_length'* is the pointer to an integer variable where the length of the memory allocated by the function and returned as a pointer is stored.
- FORTTRAN:** The arguments: *'hsv'*, *'indx1'* and *'indx2'* have identical meaning as in C/C++ case. The fourth argument, *'result'* is a reference to the integer array that will receive the result of the query. The fifth argument, *'resp\_length'* should contain maximum number of bytes that *'result'* array can receive. Upon successful completion *'resp\_length'* will hold the number of bytes received.
- VB:** The *'var'*, *'indx1'* and *'indx2'* have the same meaning as for C/C++ and Fortran. *'result'* is a variable of the *'IntBuff'* type defined in *hsv.bas*. See remarks to *hcExecBin* for explanations. The fifth argument, *'resp\_length'* should contain maximum number of bytes that *'result'* array can receive.

## Return Value

In the C/C++ case the function returns the pointer to the memory allocated to hold the result of the query. A null value indicates failure during the query operation. In the Fortran and VB interface the function returns 1 if it was completed successfully and 0 otherwise.

## Remarks

In the C/C++ case the function allocates a required memory block and returns the resultant pointer. It is the calling application's responsibility to free the memory when it is no longer needed, by calling *hcFree*.

When querying for vector and array variable with indexes (*indx1* and/or *indx2* set to zero) the result of the query contains all of the elements of respective variable. If it is an array, the atom indices change faster than the molecule indices.

If, in the Fortran case, the size of *'result'* is smaller than the size of the query results, only the first *'resp\_length'* bytes are actually transferred to the *'result'* array.

## See Also

*hcExecBin, hcQueryTxt, hcGetxxx, hcSetxxx*

## Functions for Binary ‘Get’

*This group of HAPI functions provide the easiest way for querying (Get) and updating (Set) for HSV variables. The transfer is performed in binary mode, hence, it is the fastest method for exchanging data. The functions provide all the necessary low-level operations on the blocks of memory transferred between a user application and HyperChem. This isolates the user application from the tedious task of preparing and interpreting the binary messages used by a lower-level communication between HyperChem and the user's application.*

# hcGetInt

*The function gets the value of a scalar HSV of integer type and returns that value.*

### API header

```
int hcGetInt(int hsv);
```

### FORTTRAN interface

```
integer function hfGetInt(hsv)
integer hsv
```

### VISUAL BASIC declaration

```
Declare Function hbGetInt Lib "hapi.dll" Alias "hcGetInt" (ByVal hsv As Long)
As Long
```

### Parameters

<i>C/C++:</i>	<i>HSV code of the variable</i>
<i>FORTTRAN:</i>	<i>HSV code of the variable</i>
<i>VB:</i>	<i>HSV code of the variable</i>

### Return Value

*The function returns the value of the HSV variable of integer type.*

### Remarks

### See Also

*hcSetInt, Appendix A for the types of HSV variables*

## ***hcGetReal***

*The function gets the value of a scalar HSV of floating point (double precision) type and returns this value.*

### API header

```
double hcGetReal(HSV var);
```

### FORTRAN interface

```
double precision function hfGetReal (hsv)  
integer hsv
```

### VISUAL BASIC declaration

```
Declare Function hbGetReal Lib "hapi.dll" Alias "hcGetReal" (ByVal hsv As Long)  
As Double
```

### Parameters

*C/C++:* HSV code of the variable  
*FORTRAN:* HSV code of the variable  
*VB:* HSV code of the variable

### Return Value

*The function returns the double precision value of HSV variable.*

### Remarks

**See Also***hcSetReal*

# hcGetIntVec

The function gets the contents of the HSV variable of integer vector type .

**API header**

```
int hcGetIntVec(HSV var, int* buff, int max_length);
```

**FORTRAN interface**

```
integer function hfGetIntVec (hsv, result, res_length)
integer hsv, result, res_length
dimension result(res_length)
```

**VISUAL BASIC declaration**

```
Declare Function hbGetIntVec Lib "hapi.dll" Alias "hcGetIntVec" (ByVal hsv As Long, ByRef buff As IntBuff, ByVal max_length As Long) As Long
```

**Parameters**

- C/C++:** *'var'* - the HSV code, *'buff'* pointer to the buffer where the vector is copied, *'max\_length'* - the size of *'buff'*
- FORTRAN:** *'hsv'* - the HSV code for the vector variable, *'result'* - one dimensional integer array to receive the vector, *'res\_length'* - the size of *'result'* array.
- VB:** *The 'hsv' and 'max\_length' have the same meaning as in the C/C++ case. 'buff' is of IntBuff type defined in 'hsv.bas' module definition file.*

**Return Value**

*The function returns the number of integer words transferred to the buffer, if the operation was completed successfully, or 0 if it failed.*

**Remarks**

*The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the amount of integer words in the HSV is larger than the*

*buffer, only the first 'max\_length' words are copied into buffer. the same applies to the Fortran 'result' and 'res\_length' variables.*

### See Also

*hcSetIntVec, hcGetIntVecElm, hcGetIntArray, hcSetIntArray*

## hcGetIntArr

*The function gets the contents of the HSV variable of the integer array type.*

### API header

```
int hcGetIntArr(HSV var, int* buff, int max_length);
```

### FORTRAN interface

```
integer function hfGetIntArr (hsv, result, res_length)
integer hsv, result, res_length
dimension result(res_length)
```

### VISUAL BASIC declaration

```
Declare Function hbGetIntArr Lib "hapi.dll" Alias "hcGetIntArr" (ByVal var As Long, ByRef buff As IntBuff, ByVal max_length As Long) As Long
```

### Parameters

**C/C++:** 'var' - the HSV code, 'buff' pointer to the buffer where the array is copied, 'max\_length' - the size of 'buff'

**FORTRAN:** 'hsv' - the HSV code for the array variable, 'result' - one dimensional integer array to receive the HSV array, 'res\_length' - the size of the 'result' array.

**VB:** N/A

### Return Value

*The function returns the number of integer words transferred to the buffer, if the operation was completed successfully, or 0 if it failed.*

**Remarks**

The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the amount of integer words in the HSV is larger than the buffer, only first 'max\_length' words are copied into buffer. The same applies to the Fortran 'result' and 'res\_length' variables.

**See Also**

*hcGetIntArrElm, hcSetIntArr, hcSetIntArrElm*

# hcGetIntArrElm

The function gets the contents of the HSV variable of integer array type.

**API header**

```
int hcGetIntArrElm(HSV var, int atom_index, int molecule_index);
```

**FORTTRAN interface**

```
integer function hfGetIntArrElm( hsv, atom_index, mol_index)
integer hsv, atom_index, mol_index
```

**VISUAL BASIC declaration**

```
Declare Function hbGetIntArrElm Lib "hapi.dll" Alias "hcGetIntArrElm" (ByVal
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long) As Long
```

**Parameters**

<i>C/C++:</i>	'var' - the HSV code, 'atom_index' - index for the row of array, 'molecule_index' - index for the column of the array.
<b>FORTTRAN:</b>	'hsv' - the HSV code for the array variable, 'atom_index' - index for the row of array, 'mol_index' - index for the column of the array.
<i>VB:</i>	N/A

**Return Value**

The function returns the integer value of the element of the array HSV variable of type integer.



### Remarks

*HSV arrays are always the arrays with atom-in-molecule and molecule index. Both indices run from 1 to number of respective elements.*

### See Also

*hcSetIntArrElm, hcGetInt, hcGetIntArr, hcSetInt, hcSetIntArr*

## hcGetRealVec

*The function gets the contents of the HSV variable of the real (double precision) vector type.*

### API header

```
int hcGetRealVec(HSV var, double* buff,int max_length);
```

### FORTRAN interface

```
integer function hfGetRealVec (hsv, result, res_length)  
integer hsv, res_length  
double precision result  
dimension result(res_length)
```

### VISUAL BASIC declaration

```
Declare Function hbGetRealVec Lib "hapi.dll" Alias "hcGetRealVec" (ByVal var  
As Long, ByRef buff As DblBuff, ByVal max_length As Long) As Long
```

### Parameters

- C/C++:** 'var' - the HSV code, 'buff' pointer to the buffer where the vector is copied, 'max\_length' - the size of 'buff'
- FORTRAN:** 'hsv' - the HSV code for the vector variable, 'result' - one dimensional double precision array to receive the HSV vector, 'res\_length' - the size of 'result' array.
- VB:** N/A

**Return Value**

*The function returns the number of double precision words transferred to the buffer, if the operation was completed successfully, or 0 if it failed.*

**Remarks**

*The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the amount of double precision words in the HSV is larger than the buffer, only first 'max\_length' words are copied into buffer. the same applies to Fortran 'resul' and 'res\_length' variables.*

**See Also**

*hcGetRealVecElm, hcSetRealVec, hcGetReal, hcSetReal, hcGetRealArrXYZ, hcSetRealArrXYZ*

# hcGetRealArr

*The function gets the contents of the HSV variable of the real (double precision) array type.*

**API header**

```
int hcGetRealArr(HSV var, double* buff,int max_length);
```

**FORTRAN interface**

```
integer function hfGetRealVec (hsv, result, res_length)
integer hsv, res_length
double precision result
dimension result(res_length)
```

**VISUAL BASIC declaration**

```
Declare Function hbGetRealArr Lib "hapi.dll" Alias "hcGetRealArr" (ByVal var
As Long, ByVal buff As DblBuff, ByVal max_length As Long) As Long
```

**Parameters**

*C/C++:* 'var' - the HSV code, 'buff' pointer to the buffer where the array is copied, 'max\_length' - the size of 'buff'

**FORTTRAN:** 'hsv' - the HSV code for the vector variable, 'result' - one dimensional double precision array to receive the HSV vector, 'res\_length' - the size of 'result' array.

**VB:** N/A

### **Return Value**

*The function returns the number of double precision words transferred to the buffer, if the operation was completed successfully, or 0 if it failed.*

### **Remarks**

*The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the amount of double precision words in the HSV is larger than the buffer, only first 'max\_length' words are copied into buffer. The same applies to Fortran 'result' and 'res\_length' variables.*

### **See Also**

*hcGetRealArrElm, hcSetRealArr, hcSetRealArrElm, hcSetReal, hcGetReal*

## **hcGetIntVecElm**

*The function gets the integer value of an element of the vector-type HSV variable.*

### **API header**

```
int hcGetIntVecElm(HSV var, int index);
```

### **FORTTRAN interface**

```
integer function hfGetIntVecElm (hsv, index)  
integer hsv, index
```

### **VISUAL BASIC declaration**

```
Declare Function hbGetIntVecElm Lib "hapi.dll" Alias "hcGetIntVecElm" (ByVal  
var As Long, ByVal index As Long) As Long
```

**Parameters**

<i>C/C++:</i>	<i>'var'</i> - the HSV code, <i>'index'</i> - points out the element to be retrieved.
<b>FORTTRAN:</b>	'hsv' - the HSV code for the vector variable, 'index' - points out the element to be retrieved.
<i>VB:</i>	<i>N/A</i>

**Return Value**

*The function returns the integer value of the element of an HSV variable of integer type.*

**Remarks**

*The first element of the vector is referred to as an element with index equal to one.*

**See Also**

*hcGetInt, hcSetInt, hcSetIntVecElm, hcGetIntVecElm*

# **hcGetRealVecElm**

*The function gets the value of a double-precision element of a vector HSV variable..*

**API header**

```
double hcGetRealVecElm(HSV var,int index);
```

**FORTTRAN interface**

```
double precision function hfgetRealVecElm (hsv, index)
integer hsv, index
```

**VISUAL BASIC declaration**

```
Declare Function hbGetRealVecElm Lib "hapi.dll" Alias "hcGetRealVecElm" (ByVal
var As Long, ByVal index As Long) As Double
```

### Parameters

C/C++:	<i>'var'</i> - the HSV code, <i>'index'</i> - points out the element to be retrieved.
FORTTRAN:	<i>'hsv'</i> - the HSV code for the vector variable, <i>'index'</i> - points out the element.
VB:	N/A

### Return Value

*The function returns the double precision value of the element of the HSV vector of real type.*

### Remarks

*The first element of the vector is referred to as an element with index equal to one.*

### See Also

*hcSetRealVecElm, hcGetReal, hcSetReal, hcGetRealVec, hcSetRealVec*

## ***hcGetRealArrElm***

*The function gets an element of an array HSV of real (double precision type).*

### API header

```
double hcGetRealArrElm(HSV var,int atom_index,int molecule_index);
```

### FORTTRAN interface

```
integer function hfgetRealArrElm( hsv, atom_index, molecule_index)  
integer hsv, atom_index, molecule_index
```

### VISUAL BASIC declaration

```
Declare Function hbGetRealArrElm Lib "hapi.dll" Alias "hcGetRealArrElm" (ByVal  
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long) As Double
```

**Parameters**

C/C++:	'var' - the HSV code, 'atom_index' - index for the row of array, 'molecule_index' - index for the column of the array
FORTTRAN:	'hsv' - the HSV code for the array variable, 'atom_index' - index for the row of array, 'molecule_index' - index for the column of the array.
VB:	N/A

**Return Value**

The function returns the double precision value of the element of the array HSV variable of real type.

**Remarks**

HSV arrays are always arrays with atom-in-molecule and molecule index. Both indices run from 1 to number of respective elements.

**See Also**

*hcSetRealArrElm, hcGetReal, hcGetRealArr, hcSetReal, hcSetRealArr*

# hcGetRealArrXYZ

The function gets the three reals (double precision) that form an element of the array HSV variable. It's designed almost exclusively for getting Cartesian coordinates of an atom.

**API header**

```
int hcGetRealArrXYZ(HSV var,int atom_index,int molecule_index, double* x,
double* y, double* z);
```

**FORTTRAN interface**

```
logical function hfGetrealArrXYZ (hsv, atom_index, molecule_index, x, y, z)
integer hsv, atom_index, molecule_index
double precision x, y, z
```

**VISUAL BASIC declaration**

```
Declare Function hbGetRealArrXYZ Lib "hapi.dll" Alias "hcGetRealArrXYZ" (ByVal
```

```
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByRef x  
As Double, ByRef y As Double, ByRef z As Double) As Long
```

### Parameters

C/C++: 'var' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'x', 'y' and 'z' pointers to double precision words where the results will be placed.

FORTTRAN: 'hsv' - the HSV code, 'atom\_index' - index for the row of the array, 'molecule\_index' - index for the column of the array, 'x', 'y', and 'z' - double precision variables that receive the results.

VB: N/A

### Return Value

The function returns 1 if the operation was successful and 0 otherwise.

### Remarks

The coordinates of atoms are represented in HyperChem as an array indexed by the atom-in-molecule number and the molecule number. However the element of the array is not a number but a triple of numbers representing three cartesian components of the position of the atom in space. Both atom and molecule indices run from 1 to the respective number of elements.

### See Also

*hcSetRealArrXYZ, hcGetRealArr, hcSetRealArr*

## **hcGetRealVecXYZ**

The function gets the three components of the elements of a vector HSV variable.

### API header

```
int hcGetRealVecXYZ(HSV var, int atom_index, double* x, double* y, double* z);
```

### FORTTRAN interface

```
logical function hfGetRealVex\cXYZ (hsv, index, x, y, z)  
integer hsv, index
```

double precision x, y, z

### VISUAL BASIC declaration

```
Declare Function hbGetRealVecXYZ Lib "hapi.dll" Alias "hcGetRealVecXYZ" (ByVal
var As Long, ByVal index As Long, ByRef x As Double, ByRef y As Double, ByRef
z As Double) As Long
```

### Parameters

C/C++:	<i>'var' - the HSV code, 'index' - index for the element of the vector, 'x', 'y' and 'z' pointers to double precision words where the results will be placed.</i>
FORTRAN:	<i>'hsv' - the HSV code, 'index' - index of the vector element, 'x', 'y', and 'z' - double precision values that receive the results.</i>
VB:	<i>N/A</i>

### Return Value

*The function returns 1 if the operation was successful and 0 otherwise.*

### Remarks

*Some properties in HyperChem are represented as a vector of triples usually representing three cartesian components of the property. The index runs from 1 to the respective number of elements.*

### See Also

*hcSetRealVecXYZ, hcGetReal, hcSetReal, hcGetRealVec, hcSetRealVec*

## hcGetStr

*This function retrieves the contents of the HSV variable of string type.*

### API header

```
int hcGetStr(HSV var, char* buff, int max_length);
```

### FORTRAN interface

```
integer function hfGetStr( hsv, result, res_length)
cahracter*(*) result
```



## VISUAL BASIC declaration

```
Declare Function hbGetStr Lib "hapi.dll" Alias "hcGetStr" (ByVal var As Long,  
ByVal buff As String, ByVal max_length) As Long
```

### Parameters

C/C++:	'var' - the HSV code, 'buff' pointer to the buffer where the string is copied, 'max_length' - the size of 'buff'
FORTTRAN:	'hsv' - the HSV code, 'result' - character array where the string is copied, 'res_length' - the length of the 'result' string.
VB:	N/A

### Return Value

The function returns the number of characters copied into the buffer if the operation was successful or zero otherwise.

### Remarks

The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the number of characters in the HSV is larger than the buffer, only first 'max\_length' bytes are copied into buffer.

### See Also

*hcSetStr, hcSetBlock, hcGetBlock*

## hcGetStrVecElm

The function retrieves the contents of the element of the vector HSV variable of string type.

### API header

```
int hcGetStrVecElm(HSV var, int index, char* buff, int max_length);
```

### FORTTRAN interface

```
integer function hfGetStrVecElm( hsv, index, result, res_length)  
integer hsv, index, res_length
```

```
character*(*) result
```

### VISUAL BASIC declaration

```
Declare Function hbGetStrVecElm Lib "hapi.dll" Alias "hcGetStrVecElm" (ByVal  
var As Long, ByVal index As Long, ByVal buff As String, ByVal max_length As  
Long) As Long
```

### Parameters

C/C++:	<i>'var'</i> - the HSV code, <i>'index'</i> - points out the element to be retrieved, <i>'buff'</i> pointer to the buffer where the string is copied, <i>'max_length'</i> - the size of <i>'buff'</i>
FORTTRAN:	<i>'hsv'</i> - the HSV code, <i>'index'</i> - points out the element to be retrieved, <i>'result'</i> - character array where the string is copied, <i>'res_length'</i> - the length of <i>'result'</i> array.
VB:	N/A

### Return Value

*The function returns the number of characters copied into the buffer if the operation was successful or zero otherwise.*

### Remarks

*The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the number of characters in the HSV is larger than the buffer, only first 'max\_length' bytes are copied into buffer.*

*The first element of the vector has index equal to one.*

### See Also

*hcSetStrVecElm, hcGetStr, hcSetStr*

## hcGetStrArrElm

*The function retrieves the contents of the element of the array HSV variable of string type.*

### API header

```
int hcGetStrArrElm(HSV var, int atom_index, int molecule_index, char* buff,
int max_length);
```

### FORTTRAN interface

```
integer function hfGetStrArrElm(hsv,atom_index,mol_index,result,res_length)
integer hsv, atom_index, mol_index, res_length
character*(*) result
```

### VISUAL BASIC declaration

```
Declare Function hbGetStrArrElm Lib "hapi.dll" Alias "hcGetStrArrElm" (ByVal
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByVal buff
As String, ByVal max_length As Long) As Long
```

### Parameters

- C/C++: 'var' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'buff' pointer to the buffer where the string is copied, 'max\_length' - the size of 'buff'
- FORTTRAN: 'hsv' - the HSV code, 'atom\_index' - index for the row of array, 'mol\_index' - index for the column of array, 'result' - character array where the string is copied, 'res\_length' - the length of 'result' array.
- VB: N/A

### Return Value

*The function returns the number of characters copied into buffer if the operation was successful or zero otherwise.*

### Remarks

*The function does not allocate memory for 'buff' - it assumes that the appropriate memory block pointed out by 'buff' was allocated and is of 'max\_length' size. If the number of characters in the HSV is larger than the buffer, only first 'max\_length' bytes are copied into buffer.*

*Both indices start with one.*

### See Also

*hcSetStrArrElm, hcSetStr, hcGetStr*

# hcGetBlock

The function retrieves the contents of the whole HSV variable irrespectiv of its type.

## API header

```
int hcGetBlock(HSV var, char* buff, int max_length);
```

## FORTTRAN interface

```
integer function hfgetBlock( hsv, result, res_length)
integer*1 result
```

## VISUAL BASIC declaration

```
Declare Function hbGetBlock Lib "hapi.dll" Alias "hcGetBlock" (ByVal var As Long, ByVal buff As String, ByVal max_length) As Long
```

## Parameters

C/C++:	'var' - the HSV code, 'buff' pointer to the buffer where the data is copied, 'max_length' - the size of 'buff'
FORTTRAN:	'hsv' - the HSV code, 'result' - the integer*1 (byte) array where the data is copied, 'res_length' - the size of the result.
VB:	N/A

## Return Value

The function returns the number of bytes copied into the buffer.

## Remarks

Some HSV variables, particularly some vectors and arrays, have the type of element which is not just an integer, real or string. The hcGetBlock was provided to get access to that kind of variable. However, the user's application is responsible for the interpretation of the data in the block and the proper "sorting-out" of individual elements of the block.

## See Also

*hcSetBlock, hcQueryBin*

## Functions for Binary 'Set'

# *hcSetInt*

*The function updates the scalar HSV of integer type.*

### API header

```
int hcSetInt(int var,int value);
```

### FORTRAN interface

```
logical function hfSetInt( hsv, value)  
integer hsv, value
```

### VISUAL BASIC declaration

```
Declare Function hbSetInt Lib "hapi.dll" Alias "hcSetInt" (ByVal hsv As Long,  
ByVal value As Long) As Long
```

### Parameters

- C/C++:* 'var' - HSV code of the variable to be modified, 'value' - new value for the variable.
- FORTRAN:* 'hsv' - HSV code of the variable to be modified, 'value' - new value for the variable.
- VB:* 'var' - HSV code of the variable to be modified, 'value' - new value for the variable.

### Return Value

*The function returns 1 if the update was successful or 0 if it failed.*

### Remarks

### See Also

*hcGetInt, Appendix A for the types of HSV variables*

# hcSetReal

*The function updates the scalar HSV of floating point (double precision) type.*

## API header

```
int hcSetReal(int var,double value);
```

## FORTTRAN interface

```
logical function hfSetReal( hsv, value)
integer hsv
double precision value
```

## VISUAL BASIC declaration

```
Declare Function hbSetReal Lib "hapi.dll" Alias "hcSetReal" (ByVal hsv As Long,
ByVal value As Double) As Long
```

## Parameters

- C/C++:* 'var' - HSV code of the variable to be modified, 'value' - new value for the variable.
- FORTTRAN:* 'hsv' - HSV code of the variable to be modified, 'value' - new value for the variable.
- VB:* 'var' - HSV code of the variable to be modified, 'value' - new value for the variable.

## Return Value

*The function returns 1 if the update was successful or 0 if it failed.*

## Remarks

## See Also

*hcGetReal*

# hcSetIntVec

The function updates the vector HSV variable of the integer element type.

## API header

```
int hcSetIntVec(HSV var, int* buff, int length);
```

## FORTRAN interface

```
logical function hfSetIntVec( hsv, result, res_length)  
integer hsv, result, res_length  
dimension result(res_length)
```

## VISUAL BASIC declaration

```
Declare Function hbSetIntVec Lib "hapi.dll" Alias "hcSetIntVec" (ByVal hsv As  
Long, ByRef buff As IntBuff, ByVal max_length As Long) As Long
```

## Parameters

- C/C++:** 'var' - the HSV code, 'buff' pointer to the buffer with the new contents for the vector, 'length' - the number of elements in the 'buff' buffer.
- FORTRAN:** 'hsv' - the HSV code for the vector variable, 'result' - one dimensional integer array containing data for the vector, 'res\_length' - the size of 'result' array.
- VB:** N/A

## Return Value

The function returns 1 if the variable was updated successfully, or 0 if it was not updated.

## Remarks

The user's application is responsible for the appropriate number of elements in the buffer. If the number was incorrect, HyperChem would signal an error.

## See Also

*hcGetIntVec, hcSetIntVecElm, hcSetIntArray, hcGetIntArray, hcSetIntVecElm*

# hcSetIntArr

*The function updates the array HSV variable of the integer element type.*

## API header

```
int hcSetIntArr(int var, int* buff, int length);
```

## FORTTRAN interface

```
logical function hfSetIntArr( hsv, result, res_length)
integer hsv, result, res_length
dimension result(res_length)
```

## VISUAL BASIC declaration

```
Declare Function hbSetIntArr Lib "hapi.dll" Alias "hcSetIntArr" (ByVal var As Long, ByRef buff As IntBuff, ByVal max_length As Long) As Long
```

## Parameters

- C/C++:** 'var' - the HSV code, 'buff' pointer to the buffer with the new contents for the array, 'length' - the number of elements in the 'buff' buffer.
- FORTTRAN:** 'hsv' - the HSV code for the array variable, 'result' - one dimensional integer array containing data for the array, 'res\_length' - the size of 'result' array.
- VB:** N/A

## Return Value

*The function returns 1 if the variable was updated successfully, or 0 if it was not updated.*

## Remarks

*The user's application is responsible for the appropriate number of elements in the buffer. If the number was incorrect, HyperChem would signal an error.*

## See Also

*hcSetIntArrElm, hcGetIntArr, hcGetIntArrElm*



# hcSetIntArrElm

*The function updates the element of the array HSV variable of the integer array type.*

## API header

```
int hcSetIntArrElm(HSV var, int atom_index, int molecule_index, int value);
```

## FORTRAN interface

```
logical function hfSetIntArrElm( hsv, atom_index, mol_index, value)  
integer hsv, atom_index, mol_index, value
```

## VISUAL BASIC declaration

```
Declare Function hbSetIntArrElm Lib "hapi.dll" Alias "hcSetIntArrElm" (ByVal  
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByVal  
value As Double) As Long
```

## Parameters

- C/C++: 'var' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'value' - the new value for the variable.
- FORTRAN: 'hsv' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'value' - the new integer value for the variable.
- VB: N/A

## Return Value

*The function returns 1 if the variable was updated successfully, or 0 if it was not updated.*

## Remarks

*HSV arrays are always the arrays with atom-in-molecule and molecule index. Both indices run from 1 to number of respective elements.*

## See Also

*hcGetIntArrElm, hcSetInt, hcSetIntArr, hcGetInt, hcGetIntArr*

# hcSetRealVec

*The function updates the contents of the HSV variable of the real (double precision) type.*

## API header

```
int hcSetRealVec(int var, double* buff,int length);
```

## FORTRAN interface

```
logical function hfSetRealVec( hsv, result, res_length)
integer hsv, res_length
double precision result
dimension result(res_length)
```

## VISUAL BASIC declaration

```
Declare Function hbSetRealVec Lib "hapi.dll" Alias "hcSetRealVec" (ByVal var
As Long, ByRef buff As DblBuff, ByVal max_length As Long) As Long
```

## Parameters

- C/C++:** 'var' - the HSV code, 'buff' pointer to the buffer with the new contents for the vector, 'length' - the number of elements (double precision words) in the 'buff' buffer.
- FORTRAN:** 'hsv' - the HSV code for the vector variable, 'result' - one dimensional double precision array containing the data for the vector, 'res\_length' - the size of the 'result' array.
- VB:** N/A

## Return Value

*The function returns 1 if the variable was updated successfully, or 0 if it was not updated.*

## Remarks

*The user's application is responsible for the appropriate number of elements in the buffer. If the number was incorrect, HyperChem would signal an error.*

### See Also

*hcSetRealVecElm, hcGetRealVec, hcSetReal, hcGetReal, hcSetRealArrXYZ, hcGetRealArrXYZ*

# hcSetRealArr

*The function updates the HSV variable of the real (double precision) type.*

### API header

```
hcSetRealArr(int var, double* buff,int length);
```

### FORTRAN interface

```
logical function hfSetRealArr( hsv, result, res_length)  
integer hsv, res_length  
double precision result  
dimension result(res_length)
```

### VISUAL BASIC declaration

```
Declare Function hbSetRealArr Lib "hapi.dll" Alias "hcSetRealArr" (ByVal var  
As Long, ByRef buff As DblBuff, ByVal max_length As Long) As Long
```

### Parameters

**C/C++:** *'var' - the HSV code, 'buff' pointer to the buffer with the new contents for the array, 'length' - the number of elements (double precision words) in the 'buff' buffer.*

**FORTRAN:** *'hsv' - the HSV code for the array variable, 'result' - one dimensional double precision array containing the data for the array, 'res\_length' - the size of the 'result' array.*

**VB:** *N/A*

### Return Value

*The function returns 1 if the variable was updated successfully, or 0 if it was not updated.*

**Remarks**

*The user's application is responsible for the appropriate number of elements in the buffer. If the number was incorrect, HyperChem would signal an error.*

**See Also**

*hcSetRealArrElm, hcGetRealArr, hcGetRealArrElm, hcGetReal, hcSetReal*

# hcSetIntVecElm

*The function updates the element of the vector HSV variable of integer type.*

**API header**

```
int hcSetIntVecElm(int var,int index,int value);
```

**FORTRAN interface**

```
logical function hfSetIntVecElm( hsv, index, value)
integer hsv, index, value
```

**VISUAL BASIC declaration**

```
Declare Function hbSetIntVecElm Lib "hapi.dll" Alias "hcSetIntVecElm" (ByVal
var As Long, ByVal index As Long, ByVal value As Long) As Long
```

**Parameters**

- C/C++:** 'var' - the HSV code, 'index' - points out the element to be updated, 'value' - new value for the element.
- FORTRAN:** 'hsv' - the HSV code for the vector variable, 'index' - points out the element to be updated, 'value' - new value for the element.

**VB:** N/A

**Return Value**

*The function returns 1 if the variable was updated successfully, or 0 if it was not updated.*

### Remarks

*The first element of the vector has an index equal to one.*

### See Also

*hcSetInt, hcGetInt, hcGetIntVecElm, hcSetIntVecElm*

## **hcSetRealVecElm**

*The function updates the element of the vector HSV variable of double-precision type.*

### API header

```
int hcSetRealVecElm(int var,int index,double value);
```

### FORTRAN interface

```
logical function hfSetRealVecElm( hsv, index, value)  
integer hsv, index  
double precision value
```

### VISUAL BASIC declaration

```
Declare Function hbSetRealVecElm Lib "hapi.dll" Alias "hcSetRealVecElm" (ByVal  
var As Long, ByVal index As Long, ByVal value As Long) As Long
```

### Parameters

C/C++:	<i>'var' - the HSV code, 'index' - points out the element to be updated, 'value' - new value for the element.</i>
FORTRAN:	<i>'hsv' - the HSV code for the vector variable, 'index' - points out the element to be updated, 'value' - new double precision value for the element.</i>
VB:	<i>N/A</i>

### Return Value

*The function returns 1 if the variable was updated successfully, or 0 if it was*

*not updated.*

### Remarks

*The first element of the vector has index equal to one.*

### See Also

*hcGetRealVecElm, hcSetReal, hcGetReal, hcSetRealVec, hcGetRealVec*

## hcSetRealArrElm

*The function updates the element of the array HSV variable of real (double precision type).*

### API header

```
int hcSetRealArrElm(int var,int atom_index,int molecule_index,double value);
```

### FORTRAN interface

```
logical function hfSetRealVecElm( hsv, atom_index, mol_index, value)
integer hsv, atom_index, mol_index
double precision value
```

### VISUAL BASIC declaration

```
Declare Function hbSetRealArrElm Lib "hapi.dll" Alias "hcSetRealArrElm" (ByVal
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByVal
value As Double) As Double
```

### Parameters

<i>C/C++:</i>	<i>'var' - the HSV code, 'atom_index' - index for the row of array, 'molecule_index' - index for the column of the array, value - the new value for the variable</i>
<i>FORTRAN:</i>	<i>'hsv' - the HSV code, 'atom_index' - index for the row of array, 'mol_index' - index for the column of the array, value - the new value for the variable</i>
<i>VB:</i>	<i>N/A</i>

## Return Value

The function returns 1 if the variable was updated successfully, or 0 if it was not updated.

## Remarks

HSV arrays are always arrays with an 'atom-in-molecule' and a 'molecule' index. Both indices run from 1 to the number of respective elements.

## See Also

*hcGetRealArrElm*, *hcSetReal*, *hcSetRealArr*, *hcGetReal*, *hcGetRealArr*

# hcSetRealArrXYZ

The function updates an element of array HSV variable that has three real numbers as the element type. It's designed almost exclusively for updating the cartesian coordinates for an atom

## API header

```
int hcSetRealArrXYZ)(int var,int atom_index,int molecule_index,double x,
double y, double z);
```

## FORTRAN interface

```
logical function hfSetRealArrXYX( hsv, atom_index, mol_index, x, y, z)
integer hsv, atom_index, mol_index
double precision x, y, z
```

## VISUAL BASIC declaration

```
Declare Function hbSetRealArrXYZ Lib "hapi.dll" Alias "hcSetRealArrXYZ" (ByVal
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByVal x
As Double, ByVal y As Double, ByVal z As Double) As Long
```

## Parameters

C/C++: 'var' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'x', 'y' and 'z' are double precision new values for the element.

**FORTTRAN:** *'hsv' - the HSV code, 'atom\_index' - index for the row of array, 'mol\_index' - index for the column of the array, 'x', 'y' and 'z' are double precision new values for the element.*

**VB:** *N/A*

### **Return Value**

*The function returns 1 if the operation was successful and 0 otherwise.*

### **Remarks**

*The coordinates of atoms are represented in HyperChem as an array indexed by the atom-in-molecule number and the molecule number. However the element of the array is not a number but a triple of numbers representing three Cartesian components of the position of the atom in space. Both atom and molecule indices run from 1 to the respective number of elements.*

### **See Also**

*hcGetRealArrXYZ, hcSetRealArr, hcGetRealArr*

## **hcSetRealVecXYZ**

*The function updates an element of vector HSV variable that has three real numbers as the element type.*

### **API header**

```
int hcSetRealVecXYZ(int var,int index,double x, double y, double z);
```

### **FORTTRAN interface**

```
logical function hfSetRealVecXYX( hsv, index, x, y, z)
integer hsv, index
double precision x, y, z
```

### **VISUAL BASIC declaration**

```
Declare Function hbSetRealVecXYZ Lib "hapi.dll" Alias "hcSetRealVecXYZ" (ByVal
var As Long, ByVal index As Long, ByVal x As Double, ByVal y As Double, ByVal
z As Double) As Long
```



### Parameters

C/C++:	<i>'var'</i> - the HSV code, <i>'index'</i> - index for the element of the vector, <i>'x'</i> , <i>'y'</i> and <i>'z'</i> double precision new values for the element.
FORTTRAN:	<i>'hsv'</i> - the HSV code, <i>'index'</i> - index for the element of the vector, <i>'x'</i> , <i>'y'</i> and <i>'z'</i> double precision new values for the element.
VB:	N/A

### Return Value

The function returns 1 if the operation was successful and 0 otherwise.

### Remarks

Some properties in HyperChem are represented as the vector of triples, usually representing three Cartesian components of the property. The index runs from 1 to the respective number of elements.

### See Also

*hcGetRealVecXYZ*, *hcSetReal*, *hcGetReal*, *hcSetRealVec*, *hcGetRealVec*

## hcSetStr

The function updates the content of the HSV variable of string type.

### API header

```
int hcSetStr(int var,char* string);
```

### FORTTRAN interface

```
logical function hfSetStr( hsv, buff)  
integer hsv  
character*(*) buff
```

### VISUAL BASIC declaration

```
Declare Function hbSetStr Lib "hapi.dll" Alias "hcSetStr" (ByVal var As Long,  
ByVal buff As String) As Long
```

**Parameters**

C/C++:	<i>'var'</i> - the HSV code, <i>'string'</i> pointer to the NULL-terminated string containing a new string.
FORTTRAN:	<i>'hsv'</i> - the HSV code, <i>'buff'</i> - Fortran character array containing the new value of the string.
VB:	N/A

**Return Value**

The function returns 1 if the operation was successful and 0 otherwise.

**Remarks****See Also**

*hcGetStr, hcGetBlock, hcSetBlock*

# hcSetStrVecElm

The function updates the content of the element of the vector HSV variable of string type.

**API header**

```
int hcSetStrVecElm(int var, int index, char* string);
```

**FORTTRAN interface**

```
logical function hfSetStrVecElm( hsv, index, buff)
integer hsv, index
character*(*) buff
```

**VISUAL BASIC declaration**

```
Declare Function hbSetStrVecElm Lib "hapi.dll" Alias "hcSetStrVecElm" (ByVal
var As Long, ByVal index As Long, ByVal buff As String, ByVal max_length As
Long, ByVal buff As String) As Long
```

**Parameters**

C/C++:	<i>'var'</i> - the HSV code, <i>'index'</i> - points out the element of the variable to be
--------	--

updated, 'string' pointer to the NULL-terminated string containing a new string.

FORTRAN: 'hsv' - the HSV code, 'index' - points out the element of the variable to be updated, 'buff' - character array containing the Fortran string.

VB: N/A

### Return Value

The function returns 1 if the operation was successful and 0 otherwise.

### Remarks

The first element of the vector has index equal to one.

### See Also

*hcGetStrVecElm, hcSetStr, hcGetStr*

## hcSetStrArrElm

The function updates the element of the array HSV variable of string type.

### API header

```
int hcSetStrArrElm(int var,int atom_index, int molecule_index, char* string);
```

### FORTRAN interface

```
logical function hfSetStrArrElm( hsv, atom_index, mol_index, buff)
integer hsv, atom_index, mol_index
character*(*) buff
```

### VISUAL BASIC declaration

```
Declare Function hbSetStrArrElm Lib "hapi.dll" Alias "hcSetStrArrElm" (ByVal
var As Long, ByVal atom_index As Long, ByVal molecule_index As Long, ByVal buff
As String, ByVal max_length As Long, ByVal buff As String) As Long
```

### Parameters

C/C++: 'var' - the HSV code, 'atom\_index' - index for the row of array, 'molecule\_index' - index for the column of the array, 'string' pointer to the

*new, NULL-terminated string.*

**FORTRAN:** *'hsv' - the HSV code, 'atom\_index' - index for the row of array, 'mol\_index' - index for the column of the array, 'buff' - character array containing Fortran string.*

**VB:** *N/A*

### **Return Value**

*The function returns 1 if the operation was successful and 0 otherwise.*

### **Remarks**

*The both atom and molecule indices start with one.*

### **See Also**

*hcGetStrArrElm, hcGetStr, hcSetStr*

## **hcSetBlock**

*The function updates the contents of the whole HSV variable irrespevtively of its type.*

### **API header**

```
int hcSetBlock(int var, char* buff, int length);
```

### **FORTRAN interface**

```
logical function hfSetBlock( hsv, buff, res_length)
integer*1 buff
```

### **VISUAL BASIC declaration**

```
Declare Function hbSetBlock Lib "hapi.dll" Alias "hcSetBlock" (ByVal var As Long, ByVal buff As String, ByVal max_length) As Long
```

### **Parameters**

**C/C++:** *'var' - the HSV code, 'buff' pointer to the buffer where the data is copied, 'length' - the size of 'buff' (in bytes)*

**FORTRAN:** *'hsv' - the HSV code, 'buff' - integer\*1 (byte) array containing the data,*

'res\_length' - the size of 'buff' (in bytes)

VB: N/A

### Return Value

The function returns 1 if the operation was successful and 0 otherwise.

### Remarks

Some HSV variables, particularly, some vectors and arrays have the type of element which is not scalar. The `hcGetBlock` was provided to get access to that kind of variable. However, the user's application is responsible for the interpretation of the data in the block and proper setting all of the individual elements of the block.

### See Also

`hcGetBlock`, `hcQueryBin`

## Functions for Processing Notifications

# *hcNotifyStart*

The function requests for notifications about any change of the HSV variable.

### API header

```
int hcNotifyStart(LPSTR var_name);
```

### FORTRAN interface

```
logical function hfNotifyStart(hsv_name)  
character*(*) hsv_name
```

### VISUAL BASIC declaration

```
Declare Function hbNotifyStart Lib "hapi.dll" Alias "hcNotifyStart" (ByVal  
var_name As String) As Long
```

**Parameters**

C/C++: 'var\_name' - HSV name (text) for which notification is requested.  
 FORTRAN: 'hsv\_name' - HSV name (text) for which notification is requested.  
 VB: N/A

**Return Value**

The function returns 1 if the notification request is accepted.

**Remarks**

The function simply request for notification. How the notification will be processed is specified by the `hcNotifySetup` function.

**See Also**

`hcNotifyStop`, `hcNotifySetup`

# ***hcNotifyStop***

The function cancels the request for notifications about the change of the HSV variable.

**API header**

```
int hcNotifyStop(LPSTR var_name);
```

**FORTRAN interface**

```
logical function hfNotifyStop( hsv_name)  
character*(*) hsv_name
```

**VISUAL BASIC declaration**

```
Declare Function hbNotifyStop Lib "hapi.dll" Alias "hcNotifyStop" (ByVal  
var_name As String) As Long
```

**Parameters**

C/C++: 'var\_name' - HSV name (text) for which notification request is cancelled.

FORTRAN: *'hsv\_name'* - HSV name (text) for which notification request is cancelled  
VB: N/A

### Return Value

The function returns 1 if the notification request was canceled.

### Remarks

The function stops the notification irrespective of the method for notification processing.

### See Also

*hcNotifyStart, hcNotifySetup*

## hcNotifySetup

The function establishes how the notifications have to be processed.

### API header

```
int hcNotifySetup(PFNB pCallback,int NotifyWithText);
```

### FORTRAN interface

```
logical function hfNotifySetup (FnCallback, TextAdviseFlag)  
logical TextAdviseFlag
```

### VISUAL BASIC declaration

```
Declare Function hbNotifySetup Lib "hapi.dll" Alias "hcNotifySetup" (ByVal clb  
As Long, ByVal NotifyWithText As Long) As Long
```

### Parameters

C/C++: *'pCallback'* - pointer to the callback function designed to process notifications. However, the user may provide NULL parameter for *pCallback* and in this case the Notification Agent will be used. The last parameter *'NotifyWithText'* orders text notifications if is 1 and binary if it is 0.

**FORTTRAN:** *'FnCallBack'* - pointer to the callback function designed to process notifications. However, the user may provide 0 as a parameter for *FnCallBack* and in this case the Notification Agent will be used. The last parameter *'TextAdviseFlag'* orders text notifications if it is 1 and binary if it is 0.

**VB:** N/A

### **Return Value**

*The function returns 1 if the notification request was cancelled.*

### **Remarks**

*The user application may define its own function to process notification in one of the forms:*

```
typedef VOID (*PFNB)(int var, char* data, int length);  
for the binary notifications, or:
```

```
typedef VOID (*PFNX)(char* name, char* data);  
for the text notifications.
```

In both cases the notification agent will not be used and the application is responsible for processing, storing or buffering the incoming data through the callback function. This is the best method for processing notifications.

However, as it was noted in Chapter 11, many types of applications cannot receive or properly process notifications. This include all console based applications, FORTRAN programs, external Tcl/Tk programs etc. Providing NULL as the callback address parameter automatically starts the Notification Agent (0 in Fortran).

### **See Also**

*hcNotifyStart, hcNotifyStop*

## ***hcNotifyDataAvail***

*The functions checks if the Notification Agent has any not-processed notifications in its buffers.*



### API header

```
int hcNotifyDataAvail(void);
```

### FORTRAN interface

```
integer function hfNotifyDataAvail()
```

### VISUAL BASIC declaration

```
Declare Function hbNotifyDataAvail Lib "hapi.dll" Alias "hcNotifyDataAvail"  
(ByVal var_name As String) As Long
```

### Parameters

*The function has no parameters*

### Return Value

*The function returns 1 if there is any unprocessed notification or 0 otherwise.*

### Remarks

*The application may call the function as often as required; a call to the function deschedules the time slicing of the Windows operating system, so the application does not consume much processing time.*

### See Also

*hcNotifyStart, hcNotifySetup*

## ***hcGetNotifyData***

*The functions gets data arriving from a notification previously checked.*

### API header

```
int hcGetNotifyData(char* name, char *buffer, DWORD MaxBuffLength);
```

## FORTRAN interface

```
integer function hfGetNotifyData( name, result, res_length)
integer res_length
character*(*) name, result
```

### VISUAL BASIC declaration

```
Declare Function hbGetNotifyData Lib "hapi.dll" Alias "hcGetNotifyData" (ByVal
name As String, ByVal buffer As String, ByVal MaxBuffLength As Long) As Long
```

### Parameters

- C/C++:** *'name'* - is the address of the buffer where the name of the variable is placed. *'buffer'* - is the address of the buffer where incoming data will be copied, *'MaxBufferLength'* is the maximum size of the data block that *'buffer'* can accept.
- FORTRAN:** *'name'* - is the character array where the name of the variable is placed. *'result'* - is the character array where incoming data will be copied, *'res\_length'* is the maximum size of the data block that *'result'* can accept (in bytes).
- VB:** N/A

### Return Value

*The function returns the number of bytes copied by the function to the buffer 'buffer'.*

### Remarks

*Each call to hcGetNotifyData copies the notification message to the provided buffers and discards that message, deleting the appropriate buffers in the Notification Agent area. This means that the user's application listening for notifications must properly process all discarded (and copied) messages.*

### See Also

*hcNotifyStart, hcNotifySetup, hcNotifyDataAvail*

## Functions For Memory Allocation

# *hcAlloc*

*The function allocates a memory block.*

### API header

```
void* hcAlloc(size_t n_bytes);
```

### FORTRAN interface

This function is unavailable for Fortran programs

### VISUAL BASIC declaration

This function is unavailable for Visual Basic programs

### Parameters

C/C++:	<i>'n_bytes' number of bytes to allocate.</i>
FORTRAN:	N/A
VB:	N/A

### Return Value

*The function returns a pointer to the allocated block, or NULL if the allocation was not successful.*

### Remarks

*The 'hcAlloc' allocates memory for both internal HAPI needs and possible user requirements. However, the user may use regular C/C++ allocation routines.*

### See Also

*hcFree*

# hcFree

*The function deallocates a block of memory previously allocated by a call to hcAlloc.*

## API header

```
void hcFree(void* pointer);
```

## FORTRAN interface

This function is unavailable for Fortran programs

## VISUAL BASIC declaration

This function is unavailable for Visual Basic programs

## Parameters

<i>C/C++:</i>	<i>'pointer' is the pointer obtained by a previous call to hcAlloc.</i>
<i>FORTRAN:</i>	<i>N/A</i>
<i>VB:</i>	<i>N/A</i>

## Return Value

*The function does not return any data.*

## Remarks

*The main use for hcFree is after processing the data returned by hcQueryTxt and hcQueryBin. See description for these functions.*

## See Also

*hcAlloc, hcQueryBin, hcQueryTxt*

## Auxiliary Functions

# *hcShowMessage*

*The function displays message box with provided string.*

### API header

```
void hcShowMessage(LPSTR message);
```

### FORTRAN interface

```
subroutine hfShowMessage( str)  
character*(*) str
```

### VISUAL BASIC declaration

This function is useless for Visual Basic programs

### Parameters

- C/C++: *'message' points to NULL terminated string containing message to be displayed.*
- FORTRAN: *'str' - Fortran string containing the message to display.*
- VB: *N/A*

### Return Value

*The function does not return any data.*

### Remarks

*The function may be useful for debugging programs that cannot easily display regular Windows messages (like most console-based programs).*

### See Also

# hcSetTimeouts

*The function sets new timeout values for execution, querying and other types of communication.*

## API header

```
void hcSetTimeouts(int ExcTimeOut,int QryTimeOut,int RstTimeOut);
```

## FORTRAN interface

```
subroutine hfSetTimeouts(t_exc, t_qry, t_other)
integer t_exc, t_qry, t_other
```

## VISUAL BASIC declaration

```
Declare Function hbSetTimeouts Lib "hapi.dll" Alias "hcSetTimeouts" (ByVal
ExcTimeOut As Long, ByVal QryTimeOut As Long, ByVal RstTimeOut As Long) As Long
```

## Parameters

- C/C++:** *'ExcTimeOut' is the new time-out value (in miliseconds) for the execution of commands sent to HyperChem, 'QryTimeOut' is the new time-out value (in miliseconds) for the processing requests for HSV variables, 'RstTimeOut' is the new time-out value for processing controlling commands, like notification requests etc.*
- FORTRAN:** *'t\_exc' is the new time-out value (in milliseconds) for the execution of commands sent to HyperChem, 't\_qry' is the new time-out value (in milliseconds) for processing requests for HSV variables, 't\_other' is the new time-out value for process controlling commands, like notification requests, etc.*
- VB:** *N/A*

## Return Value

*The function does not return any data.*

## Remarks

*The default value for all time-outs is about 65 seconds (0xFFFF0.) After this time expires and any command, query or other operation has not finished, the error condition is invoked. The user may increase the time-out value for any*

of these three types. The most common situation where the time expires is associated with execution of commands controlled by 'ExcTimeOut' and *t\_exc*'.

## See Also

# hcLastError

This function retrieves code and messages associated with the last error associated with an HAPI operation.

## API header

```
int hcLastError(char* LastErr);
```

## FORTRAN interface

```
integer function hfLastError( error)  
character*(*) error
```

## VISUAL BASIC declaration

```
Declare Function hbLastError Lib "hapi.dll" Alias "hcLastError" (ByVal  
last_error As String) As Long
```

## Parameters

**C/C++:** The 'LastErr' pointer to string that will receive the text message associated with the last error. The string should be of 'hcMaxMessSize', with 'hcMaxMessSize' defined in 'hc.h'

**FORTRAN:** 'error' is the Fortran string that receives the last error message.

**VB:** N/A

## Return Value

The function returns a value indicating how severe the error was. There are three possibilities:

*errNO\_ERROR* - last operation was completed successfully

*errNON\_FATAL* - last operation has not completed successfully, but the

*program can continue.*

*errFATAL - last operation caused a severe error and the application cannot continue.*

*The flags `errNO_ERROR`, `errNON_FATAL` and `errFATAL` are defined in 'hc.h'.*

## Remarks

*The `hcLastError` function is most useful when the user sets the error action flag to `errACTION_NO` using `hcSetErrorAction`. In this case the error does not invoke messages on the screen asking for user intervention.*

## See Also

*`hcSetErrorAction`, `hcGetErrorAction`*

# hcGetErrorAction

*This function retrieves the flag informing you how HAPI processes errors.*

## API header

```
int hcGetErrorAction(void);
```

## FORTRAN interface

```
integer function hfGetErrorAction()
```

## VISUAL BASIC declaration

```
Declare Function hbGetErrorAction Lib "hapi.dll" Alias "hcGetErrorAction" ()  
As Long
```

## Parameters

*C/C++:*            *The function takes no parameters.*

*FORTRAN:*

*VB:*



## Return Value

The function returns the flag that may be a sum of the following flags defined in 'hc.h':

*errACTION\_NO* - do not perform any action on any error

*errACTION\_MESS\_BOX* - display message box with error message

*errACTION\_DISCONNECT* - disconnect the application from HyperChem

*errACTION\_EXIT* - immediately exit from application

*errDDE\_REP* - report low level DDE error messages

*errDDE\_NO\_REP* - do not report low level DDE error messages

## Remarks

The function should be called before the user changes the error processing method by a call to *hcSetErrorAction*, and its value stored for later use in restoring the original error processing method.

## See Also

*hcLastError*, *hcSetErrorAction*

# hcSetErrorAction

This function changes the way errors are processed.

## API header

```
void hcSetErrorAction(int err);
```

## FORTRAN interface

```
subroutine hfSetErrorAction(action)  
integer action
```

## VISUAL BASIC declaration

```
Declare Function hbSetErrorAction Lib "hapi.dll" Alias "hcSetErrorAction"  
(ByVal action As Integer) As Long
```

## Parameters

C/C++:	<p><i>'err'</i> - the value indicating how errors are to be processed. Must be a sum of the following flags defined in <i>'hc.h'</i>:</p> <p><i>errACTION_NO</i> - do not perform any action on any error</p> <p><i>errACTION_MESS_BOX</i> - display message box with error message</p> <p><i>errACTION_DISCONNECT</i> - disconnect the application from HyperChem</p> <p><i>errACTION_EXIT</i> - immediately exit from application</p> <p><i>errDDE_REP</i> - report low level DDE error messages</p> <p><i>errDDE_NO_REP</i> - do not report low level DDE error messages</p>
FORTTRAN:	<p><i>'action'</i> -indicates how errors are processed. See above for symbolic names.</p>
VB:	<p>N/A</p>

## Return Value

*The function does not return any data.*

## Remarks

*There are situations when the user does want to change the default error processing. By combining the value of flags in the appropriate sum it is possible to get different actions on errors, ranging from no action (*errACTION\_NO*) to full information ( *errACTION\_MESS\_BOX* | *errDDE\_REP*) and, possibly, exiting from current application (*errACTION\_EXIT* ).*

## See Also

*hcGetErrorAction, hcLastError*



# **HyperChem<sup>®</sup> Computational Chemistry**

**Part 1  
Practical Guide**

**Part 2  
Theory and Methods**

# Copyright © 2002 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperTNDO, HyperDFT, HyperGauss, HyperNMR, HyperChem Data and ChemPlus are trademarks of Hypercube, Inc.

## Third Party Trademarks

MM2 is a trademark of Dr. Norman Allinger, University of Georgia.

CHARMm is a registered trademark of Accelrys Inc.

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows, Windows 95, Windows 98, Windows ME, Windows NT, and Windows XP are trademarks of Microsoft Corporation.

IBM is a registered trademark of International Business Machines, Inc.

ChemDraw is a trademark of CambridgeSoft Corporation.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

<b>Preface</b>	<b>1</b>
Who Should Read this Guide? . . . . .	1
Annotated Bibliography . . . . .	2

## Part 1: Practical Guide

<b>Chapter 1 Practical Guide Introduction</b>	<b>7</b>
What is Computational Chemistry? . . . . .	7
What is HyperChem? . . . . .	8
Building and Displaying Molecules . . . . .	8
Optimizing the Structures of Molecules . . . . .	8
Investigating the Reactivity of Molecules . . . . .	8
Generating and Viewing Orbitals and Electronic Plots . . . . .	9
Evaluating Chemical Pathways and Mechanisms . . . . .	9
Studying the Dynamic Behavior of Molecules . . . . .	10
<b>Chapter 2 HyperChem Calculations</b>	<b>11</b>
Exploring Potential Energy Surfaces . . . . .	11
Complexity of Potential Energy Surfaces . . . . .	14
Types of Calculations . . . . .	16
Single Point . . . . .	16
Geometry Optimization . . . . .	16
Transition State Search . . . . .	17
Molecular Dynamics . . . . .	18
Langevin Dynamics . . . . .	18
Monte Carlo Simulations . . . . .	19

## Chapter 3 Calculation Methods 21

Molecular Mechanics . . . . .	21
Bonds and Angles . . . . .	23
Torsions . . . . .	25
van der Waals Interactions and Hydrogen Bonding . . . . .	26
Electrostatic Potential . . . . .	27
United versus All Atom Force Fields . . . . .	28
Cutoffs . . . . .	29
Quantum Mechanics. . . . .	31
Background . . . . .	33
Molecular Geometry . . . . .	33
Calculating Electronic Potential Energy . . . . .	33
Range of Quantum Mechanics Methods. . . . .	35
Exclusion Principle . . . . .	35
Simplified Wave Functions . . . . .	37
Hartree-Fock Wave Functions . . . . .	38
Extending the Wave Function Calculation . . . . .	39
Extending the Wave Function Calculation . . . . .	39
Configuration Interaction . . . . .	39
Møller-Plesset Perturbation Theory . . . . .	41
Density Functional Theory (DFT) . . . . .	42
Kohn-Sham Orbitals . . . . .	43
Molecular Orbitals and Electronic Structure. . . . .	43
Orbital Occupancy. . . . .	43
Atomic Orbitals and Their Interactions . . . . .	44
SCF Technique . . . . .	45
Virtual Orbitals . . . . .	46
RHF and UHF . . . . .	46
Multiplicity Considerations . . . . .	47
Bond Breaking . . . . .	47
RHF Half-Electron Technique . . . . .	48
SCF Convergence . . . . .	49
Applying an External Field . . . . .	50
Calculation Results . . . . .	50
Quantitative Results . . . . .	52

## Chapter 4 Single Point Calculations 53

Dipole Moment . . . . .	54
Electric Polarizabilities . . . . .	54
Total Electron Density . . . . .	54
Total Spin Density . . . . .	55

Electrostatic Potential . . . . .	55
Examples of Single Point Calculations . . . . .	55

## **Chapter 5 Geometry Optimizations 59**

Geometry Optimizations . . . . .	59
Steepest Descent . . . . .	60
Conjugate Gradient . . . . .	61
Block Diagonal . . . . .	62
Eigenvector Following . . . . .	62
Conjugate Directions . . . . .	62
Setting Convergence Criteria . . . . .	63
Examples of Geometry Optimizations . . . . .	64
Solvation and Periodic Boundary Conditions . . . . .	65
Transition State Searching . . . . .	67
Eigenvector Following . . . . .	68
Synchronous Transit . . . . .	69

## **Chapter 6 Molecular Dynamics 71**

Integration Algorithm . . . . .	72
Length of Simulations . . . . .	73
Conservation of Energy . . . . .	73
Temperature Control . . . . .	74
Translational and Rotational Quenching . . . . .	75
Simulation Periods . . . . .	75
Initial Conditions and Heating . . . . .	75
Equilibration and Data Collection . . . . .	76
Tests for Equilibration . . . . .	77
Effect of Solvent on Equilibration . . . . .	77
Collecting Data . . . . .	78
Cooling . . . . .	78
Examples of Molecular Dynamics Simulations . . . . .	78
Strategies . . . . .	79
Constant Temperature versus Constant Energy . . . . .	80
Conformational Searching . . . . .	80
Quenched Dynamics . . . . .	80
Simulated Annealing . . . . .	81
Randomization During Molecular Dynamics . . . . .	81
Sampling Frequency . . . . .	82
When is Conformational Space Adequately Sampled? . . . . .	82
Using Geometric Restraints . . . . .	83



Using Experimental Data as Restraints . . . . .	84
Crossing Energy Barriers . . . . .	84
Limiting Conformational Changes during High Temperature Simulations . . . . .	84
Docking Molecules . . . . .	86
Freezing Part of a System . . . . .	86
Solvent Simulations . . . . .	87
Choice of Dielectric Constant . . . . .	87
Effects on Dynamic Motion . . . . .	88
Collecting Averages from Simulations . . . . .	88
Evaluating Stability and Equilibration . . . . .	89
Constant Energy Simulations . . . . .	89
Constant Temperature Simulations . . . . .	90
Conformational Searches . . . . .	90
Setting Up a Molecular Dynamics Simulation . . . . .	91
Heating Time . . . . .	91
Simulation or Run Time . . . . .	91
Cooling (Annealing) Time . . . . .	92
Step Size . . . . .	92
Temperature . . . . .	93
Scripts . . . . .	93
Bond Breaking . . . . .	93

## **Chapter 7 Langevin Dynamics and Monte Carlo 95**

Langevin Dynamics . . . . .	95
Integration Algorithm . . . . .	96
Setting Up a Langevin Dynamics Simulation . . . . .	97
Heating Time, Run Time, and Cooling Time . . . . .	97
Time Step . . . . .	97
Temperature . . . . .	98
Constant Temperature vs. Constant Energy . . . . .	98
Friction coefficient . . . . .	98
Additional Procedures . . . . .	99
Monte Carlo Simulations . . . . .	99
Introduction . . . . .	99
Background and Method . . . . .	100
Monte Carlo Trajectories and Simulation Parameters . . . . .	101
Initial Configurations . . . . .	101
Temperature . . . . .	101
Step Size and Acceptance Ratio . . . . .	102
Initial Phase and Equilibration . . . . .	102
Equilibration and Statistical Averaging . . . . .	102

Heating and Cooling . . . . .	103
-------------------------------	-----

## **Chapter 8 Using Molecular Mechanics Methods 105**

Availability of Parameters . . . . .	105
Force Field Features . . . . .	105
MM+ . . . . .	105
Electrostatic Interactions . . . . .	107
Accuracy of Force Fields . . . . .	107
Previous Experiences . . . . .	107
Choosing Force Field Options . . . . .	107
Dielectric Function . . . . .	107
1–4 Nonbonded Scale Factors . . . . .	108
Nonbonded Cutoffs . . . . .	108
Restraints . . . . .	109
References . . . . .	110

## **Chapter 9 Using Quantum Mechanics Methods 111**

Obtaining a Starting Structure . . . . .	111
Calculating Part of a Molecular System . . . . .	112
Setting Up Options . . . . .	113
Selecting Options for an <i>Ab Initio</i> or DFT Method . . . . .	113
Choosing a Basis Set . . . . .	113
Add Extra Basis Function . . . . .	115
Applying a Basis Set . . . . .	115
Charge and Spin Multiplicity . . . . .	115
Convergence Criteria . . . . .	116
UHF versus RHF . . . . .	116
Convergence Acceleration . . . . .	116
Calculate Gradient . . . . .	117
Calculate MP2 Correlation Energy . . . . .	117
Two-Electron Repulsion Integrals . . . . .	117
Two-electron Integral Cutoff . . . . .	117
Two-electron Integral Buffer Size . . . . .	118
Regular Two-Electron Integral Format . . . . .	118
Raffenetti Two-Electron Integral Format . . . . .	118
Direct SCF Calculation . . . . .	119
Initial Guess of MO Coefficients . . . . .	119
Number of d Orbitals . . . . .	119
Ghost-Atoms . . . . .	120

Configuration Interaction . . . . .	121
Selecting Options for the Extended Hückel Method . . . . .	121
Charge and Spin Multiplicity . . . . .	121
Hückel Constant . . . . .	121
d Orbitals. . . . .	122
Selecting Options for NDO Methods . . . . .	122
Convergence Criteria . . . . .	122
Charge, Spin, and Excited State. . . . .	123
UHF versus RHF. . . . .	123
Convergence Acceleration . . . . .	123
Configuration Interaction . . . . .	123
Log File for Results . . . . .	124
Log File for Results . . . . .	124
Types of Calculations . . . . .	125
Single Point Calculations and CI . . . . .	125
Optimization Methods . . . . .	126
Transition State Search . . . . .	127
Molecular Dynamics . . . . .	127
Energy Conservation in Molecular Dynamics Calculations . . . . .	128
Vibrations . . . . .	128
Contour Plots and Orbitals . . . . .	128
Vibrational Spectrum. . . . .	129
Electronic Spectrum . . . . .	129
Saving Information in a Log File . . . . .	129
Extended Hückel Method . . . . .	130
Limitations of Extended Hückel . . . . .	130
NDO Methods. . . . .	131
Defining Electron-Electron Interactions . . . . .	131
Treatment of Electron-Electron Interactions . . . . .	131
CNDO, INDO, MINDO/3, ZINDO/1, and ZINDO/S Methods . . . . .	132
MNDO, MNDO/d, AM1, and PM3 Methods . . . . .	132
TNDO Methods. . . . .	133
Practical Uses of NDO Methods . . . . .	133
Parameterization . . . . .	134
Results of Semi-Empirical Calculations . . . . .	135
Energies of Molecules. . . . .	135
Geometries of Molecules. . . . .	138
Energies of Transition States . . . . .	138
Molecular Orbital Energies and Ionization Potentials. . . . .	139
Dipole Moments . . . . .	140
Electrostatic Potential . . . . .	141

Atomic Charges . . . . .	143
Chemical Reactivity . . . . .	144
Atomic Charges and Reactivity . . . . .	146
Frontier Molecular Orbitals. . . . .	146
Vibrational Analysis and Infrared Spectroscopy . . . . .	149
Experimental Characteristic	
IR Fundamental Frequencies . . . . .	150
UV-visible Spectra . . . . .	152
Choosing a Semi-Empirical Method . . . . .	153
Extended Hückel. . . . .	153
CNDO . . . . .	154
INDO . . . . .	155
MINDO/3 . . . . .	155
MNDO . . . . .	155
MNDO/d . . . . .	156
AM1. . . . .	156
PM3 . . . . .	156
ZINDO/1 . . . . .	157
ZINDO/S . . . . .	157
TNDO . . . . .	157
Further Reading . . . . .	158

## Part 2: Theory and Methods

<b>Chapter 10 Theory and Methods Introduction</b>	<b>161</b>
HyperChem Architecture . . . . .	161
The Back Ends . . . . .	162
HyperChem Philosophy . . . . .	163
Background on Computational Chemistry . . . . .	164
Potential Energy Surfaces . . . . .	164
Single Point . . . . .	167
Geometry Optimization . . . . .	168
Molecular Dynamics . . . . .	168
The Born-Oppenheimer Approximation . . . . .	168
The Hamiltonian of a Collection of	
Nuclei and Electrons . . . . .	169
The Electronic Hamiltonian . . . . .	170
The Nuclear Hamiltonian . . . . .	170
Molecular Mechanics versus Quantum Mechanics . . . . .	171

Classical Mechanics on a Potential Energy Surface . . . . .	172
Force-Energy Generators. . . . .	173

## **Chapter 11 Molecular Mechanics 175**

Background . . . . .	176
Atom Types . . . . .	177
Definition of Atom Type. . . . .	177
The Typing Rules . . . . .	178
Redefining or Adding Types. . . . .	180
Force Fields. . . . .	181
Background . . . . .	182
Energetic Terms in the Potential . . . . .	182
Bond Stretching . . . . .	183
Bond Angle Bending . . . . .	183
Dihedrals . . . . .	183
Improper Dihedrals. . . . .	184
van der Waals . . . . .	184
Electrostatic . . . . .	187
Hydrogen Bonding . . . . .	188
Effective Dielectric Constants . . . . .	188
Cutoffs . . . . .	189
Terms Involving Vicinal 1-4 Interactions. . . . .	190
MM+ . . . . .	190
Units . . . . .	191
Bond Stretching. . . . .	191
Bond Dipoles . . . . .	192
Angle Bending . . . . .	193
Bond Stretch and Angle Bending Cross Term . . . . .	194
Out-of-Plane Bending . . . . .	195
Dihedrals. . . . .	195
van der Waals . . . . .	196
AMBER . . . . .	197
Bond Stretching. . . . .	198
Angle Bending . . . . .	198
Dihedrals. . . . .	198
Improper Dihedrals . . . . .	198
van der Waals . . . . .	199
Electrostatic . . . . .	199
Hydrogen Bonding. . . . .	200
Lone Pairs . . . . .	200
OPLS . . . . .	200
van der Waals . . . . .	201

Electrostatic . . . . .	201
BIO+. . . . .	202
Bond Stretching . . . . .	202
Angle Bending. . . . .	203
Dihedrals . . . . .	203
Improper Dihedrals . . . . .	203
van der Waals . . . . .	204
Electrostatic . . . . .	205
Hydrogen Bonding . . . . .	205
Urey-Bradley . . . . .	205
Parameter Sets . . . . .	205
Text or DBF Form for Parameters. . . . .	206
Modifying Existing Parameter Sets . . . . .	207
Creating New Parameter Sets . . . . .	207
Compiling Parameters . . . . .	208
Registry Setup of Force Field Options . . . . .	208
Periodic Boundary Conditions and Solvent . . . . .	210
The Periodic Boundary Conditions . . . . .	210
Equilibrated Water Box . . . . .	211
Details of Solvation Methodology . . . . .	212
Restraints . . . . .	212
Definition of a Restraint . . . . .	212
Adding Restraints . . . . .	213
The Default (MM+) Force Field . . . . .	214
The Wild Card Approach . . . . .	215
The Default Force Field . . . . .	216
Approximate Hybridization . . . . .	216
Bond Stretching . . . . .	218
Parameters for Butadiene and Related Molecules . . . . .	220
Angle Bending. . . . .	221
Stretch-Bends . . . . .	221
Out-Of-Plane Bends . . . . .	221
Torsions. . . . .	221
Single Bonds . . . . .	221
Double Bonds . . . . .	222
Electrostatic . . . . .	222
van der Waals . . . . .	222

## **Chapter 12 Quantum Mechanics 225**

Background . . . . .	227
Approximate Solutions of the Schrödinger equation . . . . .	227

Charge and Multiplicity . . . . .	228
Independent Electron Methods . . . . .	229
The Independent Electron Approximation . . . . .	229
Molecular Orbitals. . . . .	230
Orbital Energy Diagrams. . . . .	230
The MO-LCAO Approximation. . . . .	231
The Matrix equations for the Molecular Orbitals . . . . .	232
Solving for the Molecular Orbitals. . . . .	233
Self-Consistent Field Methods . . . . .	234
Hartree-Fock Method . . . . .	234
The Roothaan equations. . . . .	235
Spin Pairing. . . . .	236
Pople-Nesbet Unrestricted equations . . . . .	237
Convergence . . . . .	239
Spin Pairing—Restricted or Unrestricted? . . . . .	240
Electronic States . . . . .	242
Post Self-Consistent Field Calculations . . . . .	245
Configuration Interaction . . . . .	245
MP2 Correlation Energy . . . . .	246
The Neglect of Differential Overlap Approximation . . . . .	248
Characterizations of the Wave function. . . . .	250
Contour Plots . . . . .	250
Total Electron Density . . . . .	251
Spin Density . . . . .	253
Orbital Plots . . . . .	253
Electrostatic Potential . . . . .	254
Mixed Quantum/Classical Model. . . . .	256
Choosing the Classical–Quantum Boundary . . . . .	256
Capping Atoms and their Parameters. . . . .	258
Supported Methods . . . . .	260
<i>Ab Initio</i> Method . . . . .	261
Normalized Primitive Gaussian Functions . . . . .	263
Contracted Gaussian Functions. . . . .	264
Minimal Basis Sets: STO-3G . . . . .	265
Split-Valence Basis Sets . . . . .	268
Polarized Basis Sets. . . . .	271
Two-Electron Integrals . . . . .	272
Regular Integral Format . . . . .	273
Raffenetti Integral Format. . . . .	274
Two-Electron Integral Cutoff. . . . .	275
Direct SCF Calculation . . . . .	275
Initial Guess of MO Coefficients . . . . .	276
Projected Hückel . . . . .	277

Projected CNDO/INDO . . . . .	277
Mixed Model . . . . .	277
Ghost-Atoms . . . . .	278
Extended Hückel Theory . . . . .	278
Basic Method . . . . .	279
Valence Atomic Orbitals. . . . .	279
Hamiltonian Matrix Elements. . . . .	281
Choosing a Hückel Constant . . . . .	282
Mixed Model . . . . .	283
CNDO . . . . .	284
The CNDO equations . . . . .	284
Expectation Values . . . . .	287
Mixed Model (CNDO and INDO). . . . .	287
INDO . . . . .	288
Exchange Phenomena . . . . .	288
Differences Between INDO and CNDO . . . . .	290
Spin Interactions in INDO . . . . .	291
MINDO/3 . . . . .	292
Two-Center Two-Electron Integrals . . . . .	293
One-Center One-Electron Integral $H_{\mu\mu}$ . . . . .	293
Two-Center One-Electron Integral $H_{\mu\nu}$ (Resonance Integral) . . . . .	294
One-Center Two-Electron Integral . . . . .	294
Core-Core Repulsion Integrals. . . . .	295
Mixed Model . . . . .	295
MNDO . . . . .	296
Two-Center Two-Electron Integrals . . . . .	298
One-Center One-Electron Integral $H_{\mu\mu}$ . . . . .	301
Two-Center One-Electron Integral $H_{\mu\nu}$ . . . . .	302
One-Center Two-Electron Integrals . . . . .	302
Core-Core Repulsion Integrals. . . . .	303
Mixed Model (MNDO, AM1, and PM3) . . . . .	304
MNDO/d . . . . .	304
AM1 and PM3 . . . . .	305
ZINDO/1 . . . . .	306
Overlap Weighting Factors . . . . .	307
Mixed Model . . . . .	308
ZINDO/S . . . . .	308
Mixed Model . . . . .	309
TNDO . . . . .	309



<b>Chapter 13 Density Functional Theory</b>	<b>311</b>
Wavefunctions versus Electron Density . . . . .	311
The Hohenberg-Kohn Theorem . . . . .	312
The Kohn-Sham Equations. . . . .	312
Overview . . . . .	312
The Electron Density and Kohn-Sham Orbitals . . . . .	313
The Kohn-Sham Equations . . . . .	314
The Exchange-Correlation Functional . . . . .	316
A Simple Example: The Slater Functional . . . . .	317
The Uniform Electron Gas and Local Density Approximations. . . . .	317
Exchange versus Correlation Functional. . . . .	317
Uniform Electron Gas (UEG) . . . . .	318
LDA Correlation Functionals . . . . .	318
Beyond the LDA - Gradient Corrected Functionals . . . . .	318
Energy Gradients . . . . .	319
Hybrid Functionals . . . . .	320
A Set of Functionals . . . . .	320
Exchange Functionals . . . . .	320
Hartree-Fock. . . . .	320
Slater . . . . .	321
Becke-88 . . . . .	321
Perdew-Wang-91 . . . . .	322
Gill-96. . . . .	322
PBE-96 . . . . .	322
HCTH-98 . . . . .	323
Correlation Functionals . . . . .	323
VWN#5 . . . . .	323
Perdew-Zunger-81 . . . . .	324
Perdew-86 . . . . .	324
LYP. . . . .	324
Perdew-Wang-91 . . . . .	325
PBE-96 . . . . .	325
HCTH-98 . . . . .	325
Custom and Hybrid Functionals . . . . .	326
B3-LYP . . . . .	326
B3-PW91 . . . . .	326
EDF1 . . . . .	327
Becke-97 . . . . .	327
Numerical Integration Schemes . . . . .	327
Angular Integrations . . . . .	328
Radial Integrations . . . . .	328

## Chapter 14 Computational Options 331

Single Points on a Potential Energy Surface . . . . .	331
Local Minima on a Potential Energy Surface . . . . .	333
Unconstrained Geometry Optimization . . . . .	334
Optimization Methods . . . . .	335
Steepest Descent . . . . .	335
Conjugate Gradient Methods . . . . .	336
Fletcher-Reeves . . . . .	337
Polak-Ribiere . . . . .	338
Newton-Raphson Methods . . . . .	338
Block Diagonal Newton-Raphson . . . . .	338
Powell's Conjugate Directions . . . . .	339
Transition Structures on a Potential Energy Surface. . . . .	339
Transition States Search Methods . . . . .	340
Eigenvector Following Method . . . . .	340
Synchronous Transit Method . . . . .	341
Molecular Dynamics on a Potential Energy Surface. . . . .	342
Temperature . . . . .	343
Statistical Averaging. . . . .	343
Background . . . . .	343
Newton's Equations of Motion . . . . .	344
Leap-frog Algorithm . . . . .	344
Statistical Mechanical Averages . . . . .	345
Random Velocities or Restart Velocities . . . . .	345
The Basic Phases of a Trajectory . . . . .	346
The Fundamental Time Step . . . . .	346
Heating and Cooling . . . . .	347
Equilibration at a Temperature T . . . . .	348
Collecting Data . . . . .	349
Free Dynamics or Constant Temperature Dynamics . . . . .	349
Data, Averaging, Snapshot, and Screen Refresh Periods . . . . .	350
The Data Collection Period . . . . .	351
The Statistical Averaging Period . . . . .	351
The Snapshot Collection Period . . . . .	351
The Screen Refresh Period . . . . .	351
Averaging Energetic and Structural Data . . . . .	351
Averaging Energetic Values . . . . .	352
Averaging Named Selections . . . . .	353
Deviations from the Average . . . . .	354
The CSV File . . . . .	355
Plotting Instantaneous Values Along the Trajectory . . . . .	355
Obtaining and Understanding MD Graphs . . . . .	355
Placing Graphs into Other Documents . . . . .	357

Collecting Trajectory for Subsequent Playback . . . . .	357
Creating a Snapshot (SNP) file . . . . .	358
Reading a (HIN, SNP) File for Playback. . . . .	358
Running a Script at Playback . . . . .	359
Global Minima on a Potential Energy Surface . . . . .	359
Simulated Annealing . . . . .	360
Simple Reactions on a Potential Energy Surface . . . . .	360
Trajectory Analysis. . . . .	361
Setting Initial Coordinates and Velocities . . . . .	362
Temperature Considerations. . . . .	362
RHF/UHF Considerations. . . . .	363
UV Visible Spectroscopy . . . . .	363
Vibrational Analysis and IR Spectroscopy . . . . .	365
Vibrational Calculation . . . . .	365
Normal Coordinate Analysis . . . . .	366
Infrared Absorption . . . . .	369

## Index

375

# Preface

*HyperChem Computational Chemistry* contains two parts. Part 1, the *Practical Guide*, contains an overview and introduction to the types of calculations that you can perform with HyperChem™. Part 2, *Theory and Methods*, provides detailed information on the specific implementation of calculations in HyperChem.

This *Practical Guide* first explores the discipline of computational chemistry and the nature of HyperChem calculations. Next, it examines the concept of a potential energy surface and three calculations of the potential energy surface: single point, geometry optimization, and molecular dynamics. These calculations provide the energies and energy derivatives you need to construct and examine potential energy surfaces. Finally, the *Practical Guide* illustrates strategies for specific calculations.

*Theory and Methods* includes the equations, analytical descriptions, and data you need to understand the calculations. It deals with the science behind HyperChem calculations. Information on parameters and settings lets you modify and customize calculations.

## Who Should Read this Guide?

*HyperChem Computational Chemistry* is for both novice computational chemists and for scientists with advanced knowledge and experience.

Novice users can benefit from this book in several ways:

- It gives an overview of the computational chemistry methods that you'll find in HyperChem. The overview acquaints you with the program and the power of methods.
- It helps you choose computational techniques and protocols.

- It includes general references to aid you in more detailed study. Although these references are not comprehensive, some are key references. Other references provide examples of research problems using these computational methods.

Advanced users can also benefit from reading this guide. Many people use a limited number of algorithms and methods for chemical calculations. This book compares of the different methods in HyperChem and helps you determine the most appropriate method for your research problems. The book discusses strengths and weaknesses of the methods and algorithms.

## Annotated Bibliography

Although you can investigate many research problems after reading the *HyperChem Reference Manual* and *HyperChem Computational Chemistry*, you may also need information from textbooks and current journals. The following list of selected texts can supply the background necessary for understanding the calculations in HyperChem.

Allen, M.P.; Tildesley, D.J. *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.

An important though demanding book. Topics include statistical mechanics, Monte Carlo simulations, equilibrium and non-equilibrium molecular dynamics, analysis of calculational results, and applications of methods to problems in liquid dynamics. The authors also discuss and compare many algorithms used in force field simulations. Includes a microfiche containing dozens of Fortran-77 subroutines relevant to molecular dynamics and liquid simulations.

Berkert, U.; Allinger, N.L. *Molecular Mechanics*, American Chemical Society Monograph 177, Washington, D.C., 1982.

The classical introduction to molecular mechanics calculations. The authors describe common components of force fields, parameterization methods, and molecular mechanics computational methods. Discusses the application of molecular mechanics to molecules common in organic and biochemistry. Several chapters deal with thermodynamic and chemical reaction calculations.

Brooks III, C.L.; Karplus, M.; Pettitt, B.M. *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics*, in: *Advances in Chemical Physics*, Vol. 71. John Wiley and Sons, New York, 1988.

This text is similar to that of McCammon and Harvey (see below), but also provides a background for force field-based calculations and a more sophisticated discussion. Includes numerous examples of computing the structure, dynamics, and thermodynamics of proteins. The authors provide an interesting chapter on the complementary nature of molecular mechanics calculations and specific experimental techniques.

Clark, T. *A Handbook of Computational Chemistry*, John Wiley and Sons, New York, 1985.

A practical introduction to molecular mechanics and semi-empirical quantum mechanics calculations, with extensive examples from the MMP2 (not in HyperChem), MINDO/3, and MNDO methods. One of the more accessible books for new computational chemists.

Dewar, M.J.S. *The Molecular Orbital Theory of Organic Chemistry*. McGraw-Hill, New York, 1969.

Provides a survey of quantum mechanics, semi-empirical computational methods, and the application of molecular orbital theory to organic chemistry. The concepts explored in this book should be easy for most readers to understand.

Hehre, W.J.; Radom, L.; Schleyer, P.v.R.; Pople, J.A. *Ab Initio Molecular Orbital Theory*, John Wiley and Sons, New York, 1986

Covers theory and applications of *ab initio* quantum mechanics calculations. The discussions are useful for understanding the differences between *ab initio* and semi-empirical methods. Although both sections are valuable, the discussion of the applications of *ab initio* theory fills a void. It includes comparisons between experiment and many types and levels of calculation. The material is helpful in determining strategies for, and the validity of, *ab initio* calculations.

Lipkowitz, K.B.; Boyd, D.B., Eds., *Reviews in Computational Chemistry*, VCH Publishers, New York, 1990.

Contains nine reviews in computational chemistry by various experts. This book is particularly useful for beginning computational chemists. Six chapters address issues relevant to HyperChem, including semi-empirical quantum mechanics

methods, using the results of semi-empirical calculations to obtain molecular properties, and a brief review of empirical calculations (particularly molecular dynamics). The chapter by D. B. Boyd, "Aspects of Molecular Modeling," contains an excellent basic review of quantum mechanics calculations and summaries of the accuracy of these methods.

McCammon, J.A.; Harvey, S.C. *Dynamics of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, 1987.

Focuses on force field calculations for understanding the dynamic properties of proteins and nucleic acids. Provides a useful introduction to several computational techniques, including molecular mechanics minimization and molecular dynamics. Includes discussions of research involving structural changes and short time scale dynamics of these biomolecules, and the influence of solvent in these processes.

Szabo, A.; Ostlund, N.S. *Modern Quantum Chemistry*, McGraw-Hill, New York, 1989.

A textbook describing the theory associated with calculations of the electronic structure of molecular systems. While the book focuses on *ab initio* calculations, much of the information is also relevant to semi-empirical methods. The sections on the Hartree-Fock and Configuration Interactions methods, in particular, apply to HyperChem. The self-paced exercises are useful for the beginning computational chemist.

Pople, J.A.; Beveridge, D.L. *Approximate Molecular Orbital Theory* McGraw-Hill, New York, 1970.

Presents the basic theory of quantum mechanics, particularly, semi-empirical molecular orbital theory. The authors detail and justify the approximations inherent in the semi-empirical Hamiltonians. Includes useful discussions of the applications of these methods to specific research problems.

Stewart, J.J.P. MOPAC: A Semiempirical Molecular Orbital Program. *J. Computer-Aided Molecular Design* 4:1-105, 1990.

A valuable review of the MOPAC program and the semi-empirical methods MNDO, MINDO/3, AM1, and PM3. Of particular use are theoretical discussions of these semi-empirical methods and many tables validating the accuracy of the MOPAC program and its associated Hamiltonians.

# **HyperChem® Computational Chemistry**

**Part 1  
Practical Guide**





# Chapter 1

## Practical Guide Introduction

### What is Computational Chemistry?

Computational chemistry is a new discipline. Its advent and popularity have paralleled improvements in computing power during the last several decades. As with other disciplines in chemistry, computational chemistry uses tools to understand chemical reactions and processes. Scientists use computer software to gain insight into chemical processes. Although computational chemists frequently develop and refine software tools, their primary interest is in applying software tools to enhance chemical knowledge.

The challenges for computational chemistry are to characterize and predict the structure and stability of chemical systems, to estimate energy differences between different states, and to explain reaction pathways and mechanisms at the atomic level. Meeting these challenges could eliminate time-consuming experiments.

Software tools for computational chemistry are often based on empirical information. To use these tools, you need to understand how the technique is implemented and the nature of the database used to parameterize the method.<sup>1</sup> You use this knowledge to determine the most appropriate tools for specific investigations and to define the limits of confidence in results.

To ensure that computational chemistry develops in an orderly way, researchers must provide certain information so that others can reproduce and analyze their results. Gund *et al.* proposed guidelines for reporting molecular modeling results.<sup>2</sup> You should consider these guidelines for your publications.

1. Dearing, A., Computer-aided Molecular Modelling: Research Study or Research Tool *J. Computer-Aided Molecular Design* 2:179–189, 1988.

2. Gund, P.; Barry, D.C.; Blaney, J.M.; Cohen, N.C. Guidelines for Publications in Molecular Modeling Related to Medicinal Chemistry *J. Med. Chem.* 31:2230–2234, 1988.

## What is HyperChem?

HyperChem is a versatile molecular modeler and editor and a powerful computational package. It offers many types of molecular and quantum mechanics calculations.

### Building and Displaying Molecules

Submitting a structure to a calculation can be expensive in terms of human time and effort. HyperChem lets you build and display molecules easily. Since HyperChem contains a graphical interface, you can monitor the construction of molecules.

Using the Drawing tool, you can draw a two-dimensional (2D) representation of a molecule, and then use the Model Builder to generate a three-dimensional (3D) structure. The Model Builder adds implicit hydrogens to the molecule at your request. You can also manipulate individual bonds, bond geometries, angles, torsions, and atomic charges during model building (using the Constrain commands on the Build menu) or after model building (using the Set commands on the Edit menu).

HyperChem contains a database of amino and nucleic acid residues so you can quickly build polymers containing these subunits. You can also read in structures in files from standard databases, such as the Brookhaven Protein Data Bank (see the *HyperChem Reference Manual*).

### Optimizing the Structures of Molecules

To calculate the properties of a molecule, you need to generate a well-defined structure. A calculation often requires a structure that represents a minimum on a potential energy surface. HyperChem contains several geometry optimizers to do this. You can then calculate single point properties of a molecule or use the optimized structure as a starting point for subsequent calculations, such as molecular dynamics simulations.

### Investigating the Reactivity of Molecules

You can use HyperChem to investigate the reactivity of molecules and their functional groups. One method is to use Frontier Molec-

ular Orbital Theory<sup>3</sup>. You can use molecular orbital energies, coefficients, and nodal properties from single point quantum mechanics calculations to investigate issues such as the relative reactivity of different molecular substituents, regioselectivity of reactions, and site-selectivity of nucleophiles and electrophiles.

## Generating and Viewing Orbitals and Electronic Plots

HyperChem can plot orbital wave functions resulting from semi-empirical and *ab initio* quantum mechanical calculations. It is interesting to view both the nodal properties and the relative sizes of the wave functions. Orbital wave functions can provide chemical insights.

You can also plot the electrostatic potential, the total charge density, or the total spin density determined during a semi-empirical or *ab initio* calculation. This information is useful in determining reactivity and correlating calculational results with experimental data. These examples illustrate uses of these plots:

- If you calculate the electrostatic potential for cyclopropane, three minima occur in regions that bisect the carbon-carbon bonds. This result is consistent with protonation of cyclopropane occurring along the bond bisector.
- Spin densities help to predict the observed coupling constants in electron spin resonance (ESR) spectroscopy. From spin density plots you can predict a direct relationship between the spin density on a carbon atom and the coupling constant associated with an adjacent hydrogen.

## Evaluating Chemical Pathways and Mechanisms

Calculating single point properties and energies provides information about chemical pathways and mechanisms.

*Example:* Researchers have used MNDO and AM1 semi-empirical methods to calculate possible reaction pathways for the interaction of glycine and cocaine.<sup>4</sup> In choosing possible interaction sites,

3. Fleming, I. *Frontier Orbitals and Organic Chemical Reactions* John Wiley & Sons, New York, 1976.

4. Wu, C.S.; Neely, W.C.; Worley, S.D. A Semiempirical Theoretical Study of the Molecular Interaction of Cocaine with the Biological Substrate Glycine. *J. Comput. Chem.* 12:862-867, 1991.

they first calculated atomic charges and HOMO-LUMO (highest occupied and lowest unoccupied molecular orbitals) electron densities for the two molecules. Using this information, they proposed two possible reaction pathways and computed the heats of formation for geometry-optimized structures.

## **Studying the Dynamic Behavior of Molecules**

A molecular system at room temperature is accurately characterized by its motion. Molecular dynamics simulations calculate the future positions and velocities of atoms based upon their current values. You can obtain qualitative and quantitative data from HyperChem molecular dynamics simulations.

The greatest value of molecular dynamic simulations is that they complement and help to explain existing data for designing new experiments. The simulations are increasingly useful for structural refinement of models generated from NMR, distance geometry, and X-ray data.

The results are also important for investigating macromolecules, where concerted motions may occur during ligand binding, enzymatic activity, and molecular recognition. Using constant temperature molecular dynamics, you can investigate an enzyme that has a loop of residues flanking the active site. You can store snapshots of a molecular dynamic trajectory (the set of positions and velocities of atoms over time) and use them to detect possible conformational changes of the loop region that could change the interaction with the active site. You can further investigate conformational change during the course of the simulation by averaging and plotting the dihedral angle values associated with the loop.

# Chapter 2

## HyperChem Calculations

### Exploring Potential Energy Surfaces

HyperChem calculations examine potential energy surfaces with single point calculations, optimizations, and molecular dynamic simulations. Energies and derivatives of energy, such as the forces on atoms, are necessary to “construct” a potential energy surface. This concept gives the molecular and quantum mechanics methods a unifying purpose: they generate the energies and energy derivatives necessary to produce and examine potential energy surfaces.

Both molecular and quantum mechanics methods rely on the Born-Oppenheimer approximation. In quantum mechanics, the Schrödinger equation (1) gives the wave functions and energies of a molecule.

$$H\Psi = E\Psi \quad (1)$$

where  $H$  is the molecular Hamiltonian,  $\Psi$  is the wave function, and  $E$  is the energy. The molecular Hamiltonian is composed of these operators: the kinetic energy of the nuclei ( $N$ ) and electrons ( $E$ ), nuclear-nuclear ( $NN$ ) and electron-electron repulsion ( $EE$ ), and the attraction between nuclei and electrons ( $NE$ ) (equation 1).

$$H = (\text{kinetic energy})_N + (\text{kinetic energy})_E + (\text{repulsion})_{NN} \quad (2) + (\text{repuls}$$

Nuclei have many times more mass than electrons. During a very small period of time when the movement of heavy nuclei is negligible, electrons are moving so fast that their distribution is smooth. This leads to the approximation that the electron distribution is dependent only on the fixed positions of nuclei and not on their velocities. This approximation allows two simplifications

of the molecular Hamiltonian. The nuclear kinetic energy term drops out (equation 3).

$$H = (\text{kinetic energy})_E + (\text{repulsion})_{NN} + (\text{repulsion})_{EE} + (\text{attraction})_{NE} \quad (3)$$

Since the nuclear-nuclear repulsion is constant for a fixed configuration of atoms, this term also drops out. The Hamiltonian is now purely electronic.

$$H_{\text{electronic}} = (\text{kinetic energy})_E + (\text{repulsion})_{EE} + (\text{attraction})_{NE} \quad (4)$$

After solving the electronic Schrödinger equation (equation 4), to calculate a potential energy surface, you must add back nuclear-nuclear repulsions (equation 5).

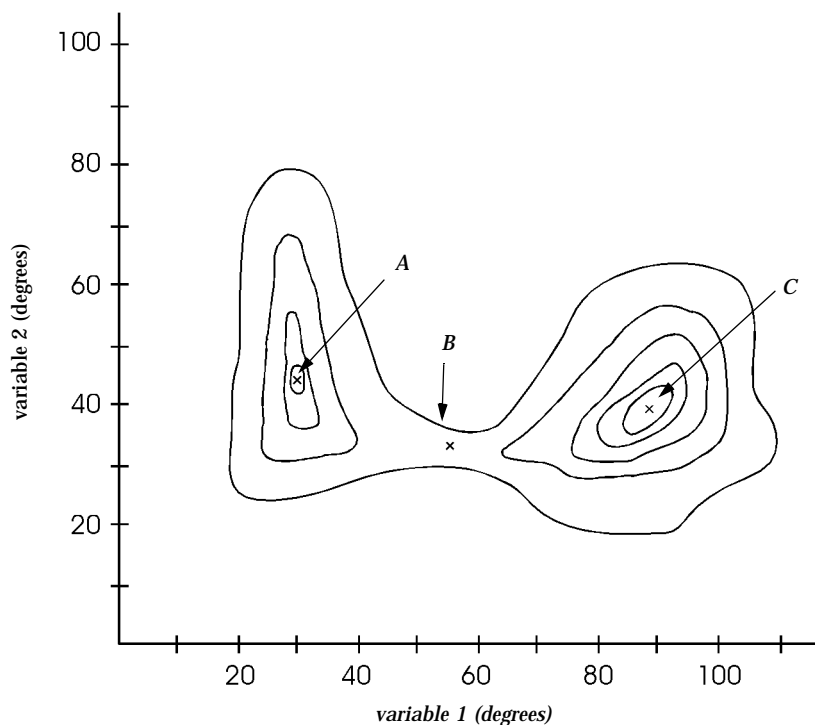
$$H_{\text{electronic}} \Psi_{\text{electronic}} = E_{\text{electronic}} \Psi_{\text{electronic}}$$

$$V_{\text{PES}} = E_{\text{electronic}} + (\text{repulsion})_{NN} \quad (5)$$

Generating the potential energy surface (PES) using this equation requires solutions for many configurations of nuclei. In molecular mechanics, the electronic energy is not evaluated explicitly. Instead, these methods solve the potential energy surface by using a force field equation (see “Molecular Mechanics” on page 21). The force field equation represents electronic energy implicitly through parameterization.

The dimensionality of a potential energy surface depends on the number of degrees of freedom in a molecule. If  $V_{\text{PES}}$  is a function of two variables, then a plot of the potential energy surface represents a 3D space.

*Example:* In this plot of a potential energy surface with two variables, points A and C are both minima.



This means that the forces on the atoms are zero for the structures found at A and C. The structure at C is more stable than the structure at A. In fact, C is the *global* minimum for this example. Geometry optimizations seek minima such as A and C.

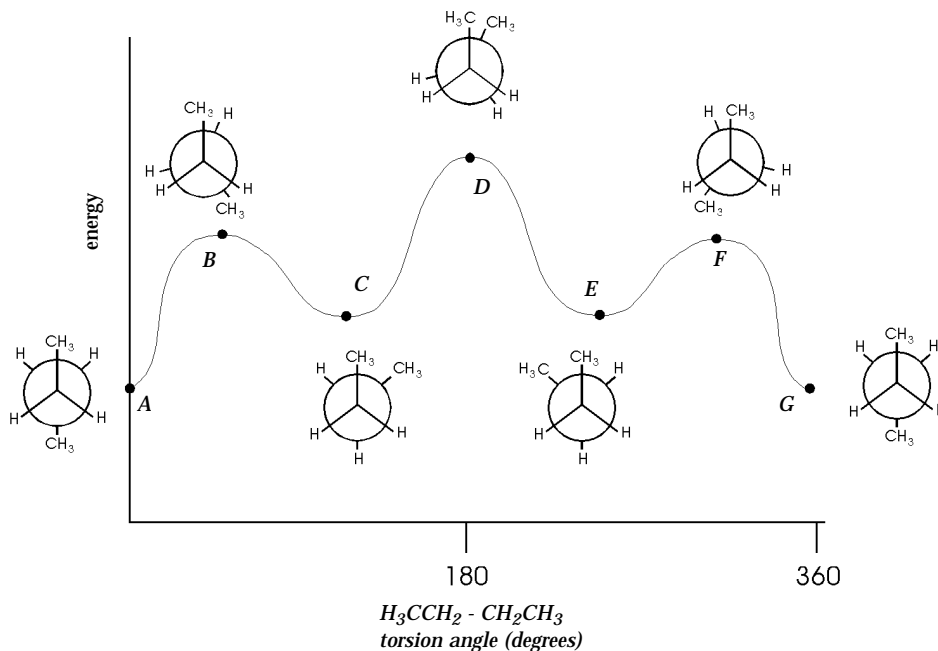
Point B is a maximum along the path from A to C (saddle point). The forces on the atoms are also zero for this structure. Point B represents a transition state for the transformation of A to C.

Convergence of a geometry optimization calculation does not necessarily mean that the calculation found a minimum on the potential energy surface. To categorize a point definitively on the potential energy surface, you must calculate the second derivatives of the potential energy with respect to atomic coordinates. You can then calculate vibrational frequencies from the second derivative matrix. A minimum configuration has six vibrational frequencies with a value of zero. For linear molecules the minimum is five with zero values. A transition state has one vibrational frequency, and this frequency is an imaginary number.



The path of minimum potential energy that connects reactants and products is known as the *reaction coordinate*.

*Example:* The reaction coordinate for rotation about the central carbon-carbon bond in *n*-butane has several stationary points. A, C, E, and G are minima and B, D, and F are maxima. Only the structures at the minima represent stable species and of these, the *anti* conformation is more stable than the *gauche*.



## Complexity of Potential Energy Surfaces

Most potential energy surfaces are extremely complex. Elber and Karplus analyzed a 300 psec molecular dynamics trajectory of the protein myoglobin<sup>5</sup>. They estimate that 2000 thermally accessible minima exist near the native protein structure. The total number of conformations is even larger. Dill derived a formula to calculate the upper bound of thermally accessible conformations in a protein.<sup>6</sup> Using this formula, a protein of 150 residues (the approximate size of myoglobin) has an upper limit of  $4 \times 10^{34}$  conforma-

5. Elber, R.; Karplus, M. Multiple conformational states of proteins: a molecular dynamics analysis of myoglobin. *Science* 235:318-321, 1987.

tions for the backbone atoms. Here are two of the consequences of this complexity:

- A single stable structure may not adequately represent the properties of a molecule. You should investigate a Boltzmann distribution of thermally accessible potential structures.
- For all but the smallest of molecules, a search for the global minimum may be impossible.

United atom force fields (see “United versus All Atom Force Fields” on page 28) are sometimes used for biomolecules to decrease the number of nonbonded interactions and the computation time. Another reason for using a simplified potential is to reduce the dimensionality of the potential energy surface. This, in turn, allows for more samples of the surface.

*Example:* Crippen and Snow reported their success in developing a simplified potential for protein folding<sup>7</sup>. In their model, single points represent amino acids. For the avian pancreatic polypeptide, the native structure is not at a potential minimum. However, a global search found that the most stable potential minimum had only a 1.8 Ångstrom root-mean-square deviation from the native structure.

*Example:* Solvation can have a profound effect on the potential energy profile for a reaction. Jorgensen’s research group provided important insights into the role of solvation<sup>8</sup>. Consider the nucleophilic addition of the hydroxide anion to formaldehyde:

*Ab initio*, gas phase calculations found two minima and one maximum for the reaction coordinate, leading to an exothermic formation of the tetrahedral complex. One minimum is an ion-dipole complex at a  $\text{H}_2(\text{O})\text{C}-\text{OH}^-$  distance of 2.74 Ångstroms. The second is the tetrahedral complex at a  $\text{H}_2(\text{O})\text{C}-\text{OH}^-$  distance of 1.47 Ångstroms. A potential-of-mean-force calculation was performed using a Monte Carlo simulation of the reactants in TIP4P water.

6. Dill, K.A. Theory for the folding and stability of globular proteins. *Biochemistry* 24:1501-1509, 1985.

7. Crippen, G.M.; Snow, M.E. A 1.8Å resolution potential function for protein folding. *Biopolymers* 29:1479-1489, 1990.

8. Madura, J.D.; Jorgensen, W.L. Ab initio and Monte Carlo calculations for a nucleophilic addition reaction in the gas phase and in aqueous solution. *J. Am. Chem. Soc.* 108:2517-2527, 1986.

The reaction coordinate showed no ion-dipole complex. In contrast to the gas phase study, the tetrahedral intermediate was 10–14 kcal/mol less stable than the reactants and there was a 24–28 kcal/mol activation barrier to the formation of the tetrahedral intermediate.

## Types of Calculations

There are multiple types of calculations in HyperChem. The include: single point, geometry optimization or minimization, transition state searching, molecular dynamics, and Monte Carlo.

### Single Point

A single point calculation gives the static properties of a molecule. The properties include potential energy, derivatives of the potential energy, electrostatic potential, molecular orbital energies, and the coefficients of molecular orbitals for ground or excited states. The input molecular structure for a single point calculation usually reflects the coordinates of a stationary point on the potential energy surface, typically a minimum or transition state. Use a geometry optimization to prepare a molecule for a single point calculation.

One type of single point calculation, that of calculating vibrational properties, is distinguished as a *vibrations* calculation in HyperChem. A *vibrations* calculation predicts fundamental vibrational frequencies, infrared absorption intensities, and normal modes for a geometry optimized molecular structure.

You can use a single point calculation that determines energies for ground and excited states, using configuration interaction, to predict frequencies and intensities of an electronic ultraviolet-visible spectrum.

### Geometry Optimization

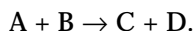
To carry out a geometry optimization (minimization), HyperChem starts with a set of Cartesian coordinates for a molecule and tries to find a new set of coordinates with a minimum potential energy. You should appreciate that the potential energy surface is very

complex, even for a molecule containing only a few dihedral angles.

Since minimization calculations cannot cross or penetrate potential energy barriers, the molecular structure found during an optimization may be a local and not a global minimum. The minimum represents the potential energy closest to the starting structure of a molecule. Researchers frequently use minimizations to generate a structure at a stationary point for a subsequent single point calculation or to remove excessive strain in a molecule, preparing it for a molecular dynamics simulation.

## Transition State Search

As mentioned earlier, a potential energy surface may contain *saddle points*; that is, *stationary points* where there are one or more directions in which the energy is at a maximum. A saddle point with one negative eigenvalue corresponds to a *transition structure* for a chemical reaction of changing isomeric form. Transition structures also exist for reactions involving separated species, for example, in a bimolecular reaction



Activation energy, *i.e.*, the energy of the transition structure relative to reactants, can be observed experimentally. However, the only way that the geometries of transition structures can be evaluated is from theory. Theory also can give energetics and geometry parameters of short-lived reaction intermediates.

Transition state search algorithms rather climb up the potential energy surface, unlike geometry optimization routines where an energy minimum is searched for. The characterization of even a simple reaction potential surface may result in location of more than one transition structure, and is likely to require many more individual calculations than are necessary to obtain equilibrium geometries for either reactant or product.

Calculated transition structures may be very sensitive to the level of theory employed. Semi-empirical methods, since they are parametrized for energy minimum structures, may be less appropriate for transition state searching than *ab initio* methods are. Transition structures are normally characterized by weak “partial” bonds, that is, being broken or formed. In these cases UHF calcu-

lations are necessary, and sometimes even the inclusion of electron correlation effects.

## **Molecular Dynamics**

A molecular dynamics simulation samples phase space (the region defined by atomic positions and velocities) by integrating numerically Newton's equations of motion. Unlike single point and geometry optimization calculations, molecular dynamics calculations account for thermal motion. Molecules may contain enough thermal energy to cross potential barriers. Molecular dynamics calculations provide information about possible conformations, thermodynamic properties, and dynamic behavior of molecules. (See also "Studying the Dynamic Behavior of Molecules" on page 10.)

## **Langevin Dynamics**

Molecules in solution undergo collisions with other molecules and experience frictional forces as they move through the solvent. Using Langevin dynamics, you can model these effects and study the dynamical behavior of a molecular system in a liquid environment. These simulations can be much faster than molecular dynamics. Instead of explicitly including solvent molecules, the Langevin method models their effect through random forces applied to the molecule of interest to simulate collisions, and frictional forces are added to model the energy losses associated with these collisions.

Langevin dynamics simulations can be used to study the same kinds of problems as molecular dynamics: time dependent properties of solvated systems at non-zero temperatures. Because of the implicit treatment of the solvent, this method is particularly well-suited for studying large molecules in solution. It is possible to decouple the time scales of molecular motion and focus on just the slow modes associated with conformational changes, for example, or to follow the rapid bond stretching motions leading to chemical reaction, while incorporating the influence of molecular collisions. Langevin dynamics has been used to study solvent effects on the structural dynamics of the active site of the enzyme lysozyme, conformational equilibria and the dynamics of conformational transitions in liquid alkanes, and temperature effects on a system of interacting quantum harmonic oscillators.

## Monte Carlo Simulations

The Monte Carlo method samples phase space by generating random configurations from a Boltzmann distribution at a given temperature. Averages computed from a properly equilibrated Monte Carlo simulation correspond to thermodynamic ensemble averages. Thus, the Monte Carlo method can be used to find average energies and equilibrium structural properties of complex interacting systems.

A sequence of successive configurations from a Monte Carlo simulation constitutes a trajectory in phase space; with HyperChem, this trajectory may be saved and played back in the same way as a dynamics trajectory. With appropriate choices of setup parameters, the Monte Carlo method may achieve equilibration more rapidly than molecular dynamics. For some systems, then, Monte Carlo provides a more direct route to equilibrium structural and thermodynamic properties. However, these calculations can be quite long, depending upon the system studied.

Monte Carlo simulations provide an alternate approach to the generation of stable conformations. As with HyperChem's other simulation methods, the effects of temperature changes and solvation are easily incorporated into the calculations.



# Chapter 3

## Calculation Methods

HyperChem uses two types of methods in calculations: molecular mechanics and quantum mechanics. The quantum mechanics methods implemented in HyperChem include semi-empirical, *ab initio*, and density functional quantum mechanics methods. The molecular mechanics and semi-empirical quantum mechanics methods have several advantages over *ab initio* and density functional methods. Most importantly, these methods are fast. While this may not be important for small molecules, it is certainly important for biomolecules. Another advantage is that for specific and well-parameterized molecular systems, these methods can calculate values that are closer to experiment than lower level *ab initio* and density functional techniques.

The accuracy of a molecular mechanics or semi-empirical quantum mechanics method depends on the database used to parameterize the method. This is true for the type of molecules and the physical and chemical data in the database. Frequently, these methods give the best results for a limited class of molecules or phenomena. A disadvantage of these methods is that you must have parameters available before running a calculation. Developing parameters is time-consuming.

The *ab initio* or density functional methods may overcome this problem. However they are slower than any molecular mechanics and semi-empirical methods.

### Molecular Mechanics

Molecular mechanical force fields use the equations of classical mechanics to describe the potential energy surfaces and physical properties of molecules. A molecule is described as a collection of atoms that interact with each other by simple analytical functions. This description is called a *force field*. One component of a force field is the energy arising from compression and stretching a bond.



This component is often approximated as a harmonic oscillator and can be calculated using Hooke's law.

$$V_{\text{spring}} = \frac{1}{2}K_r(r - r_0)^2 \quad (7)$$

The bonding between two atoms is analogous to a spring connecting two masses. Using this analogy, equation 7 gives the potential energy of the system of masses,  $V_{\text{spring}}$ , and the force constant of the spring,  $K_r$ . The equilibrium and displaced distances of the atoms in a bond are  $r_0$  and  $r$ . Both  $K_r$  and  $r_0$  are constants for a specific pair of atoms connected by a certain spring.  $K_r$  and  $r_0$  are force field *parameters*.

The potential energy of a molecular system in a force field is the sum of individual components of the potential, such as bond, angle, and van der Waals potentials (equation 8). The energies of the individual bonding components (bonds, angles, and dihedrals) are functions of the deviation of a molecule from a hypothetical compound that has bonded interactions at minimum values.

$$E_{\text{Total}} = \text{term}_1 + \text{term}_2 + \dots + \text{term}_n \quad (8)$$

The absolute energy of a molecule in molecular mechanics has no intrinsic physical meaning;  $E_{\text{Total}}$  values are useful only for comparisons between molecules. Energies from single point calculations are related to the enthalpies of the molecules. However, they are not enthalpies because thermal motion and temperature-dependent contributions are absent from the energy terms (equation 8).

Unlike quantum mechanics, molecular mechanics does not treat electrons explicitly. Molecular mechanics calculations cannot describe bond formation, bond breaking, or systems in which electronic delocalization or molecular orbital interactions play a major role in determining geometry or properties.

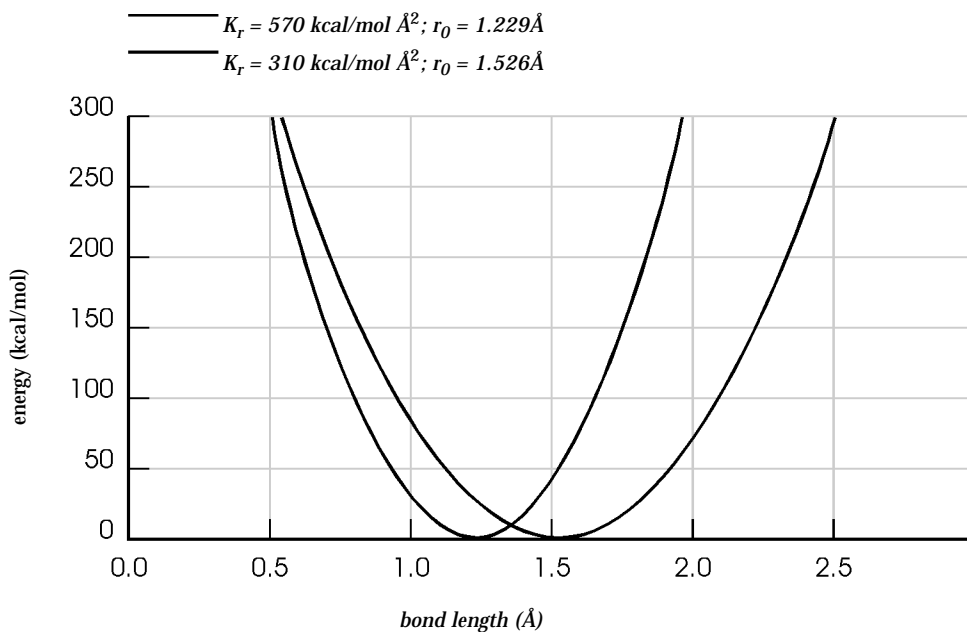
This discussion focuses on the individual components of a typical molecular mechanics force field. It illustrates the mathematical functions used, why those functions are chosen, and the circumstances under which the functions become poor approximations. Part 2 of this book, *Theory and Methods*, includes details on the implementation of the MM+, AMBER, BIO+, and OPLS force fields in HyperChem.

## Bonds and Angles

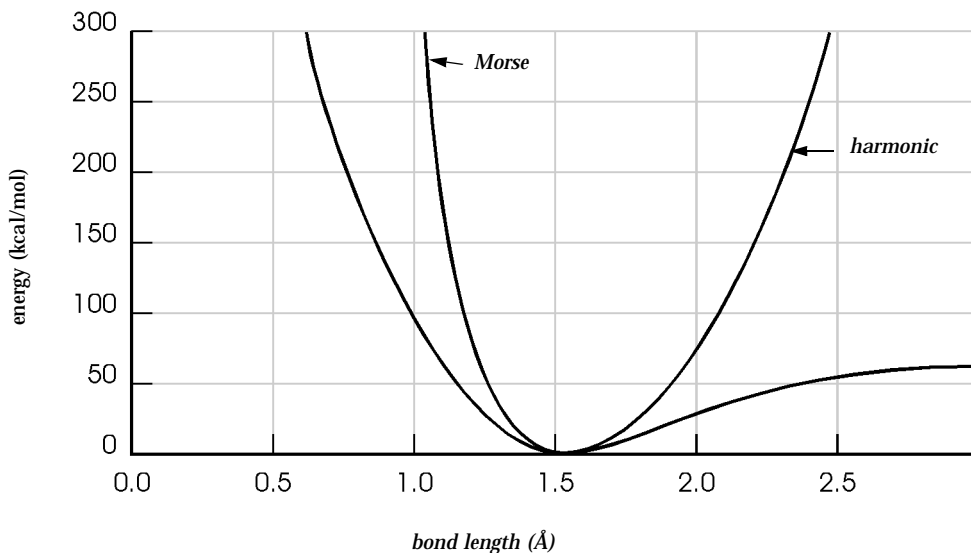
HyperChem uses harmonic functions to calculate potentials for bonds and bond angles (equation 9).

$$V_{\text{stretch}} = \sum_{\text{bond}} K_r (r - r_0)^2 \quad V_{\text{bend}} = \sum_{\text{angle}} K_\theta (\theta - \theta_0)^2 \quad (9)$$

*Example:* For the AMBER force field, a carbonyl C–O bond has an equilibrium bond length of 1.229 Å and a force constant of 570 kcal/mol Å<sup>2</sup>. The potential for an aliphatic C–C bond has a minimum at 1.526 Å. The slope of the latter potential is less steep; a C–C bond has a force constant of 310 kcal/mol Å<sup>2</sup>.



A Morse function best approximates a bond potential. One of the obvious differences between a Morse and harmonic potential is that only the Morse potential can describe a dissociating bond.



The Morse function rises more steeply than the harmonic potential at short bonding distances. This difference can be important especially during molecular dynamics simulations, where thermal energy takes a molecule away from a potential minimum.

In light of the differences between a Morse and a harmonic potential, why do force fields use the harmonic potential? First, the harmonic potential is faster to compute and easier to parameterize than the Morse function. The two functions are similar at the potential minimum, so they provide similar values for equilibrium structures. As computer resources expand and as simulations of thermal motion (See “Molecular Dynamics”, page 71) become more popular, the Morse function may be used more often.

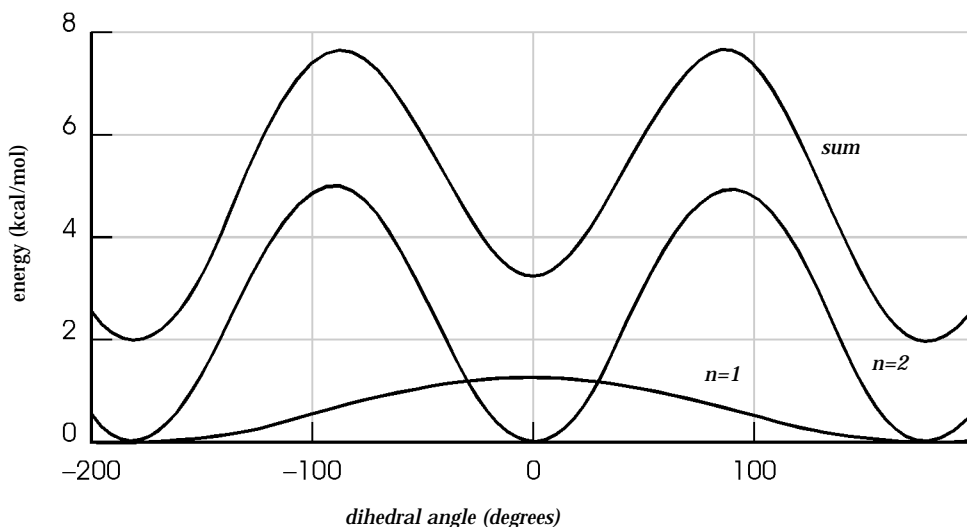
## Torsions

In molecular mechanics, the dihedral potential function is often implemented as a truncated Fourier series. This periodic function (equation 10) is appropriate for the torsional potential.

$$V_{\text{dihedrals}} = \sum_{\text{dihedrals}} \frac{V_n}{2} [1 + \cos(n\phi - \phi_0)] \quad (10)$$

In this representative dihedral potential,  $V_n$  is the dihedral force constant,  $n$  is the periodicity of the Fourier term,  $\phi_0$  is the phase angle, and  $\phi$  is the dihedral angle.

*Example:* This example of an HN-C(O) amide torsion uses the AMBER force field. The Fourier component with a periodicity of one ( $n = 1$ ) also has a phase shift of 0 degrees. This component shows a maximum at a dihedral angle of 0 degrees and minima at both  $-180$  and  $180$  degrees. The potential uses another Fourier component with a periodicity of two ( $n = 2$ ).



The relative sizes of the potential barriers indicate that the  $V_2$  force constant is larger than the  $V_1$  constant. The phase shift is  $180$  degrees for the Fourier component with a two-fold barrier. Minima occur at  $-180$ ,  $0$ , and  $180$  degrees and maxima at  $-90$  and  $90$

degrees. Adding the two Fourier terms results in potential with minima at  $-180$ ,  $0$ , and  $180$  degrees and maxima at  $-90$  and  $90$  degrees. (The “sum” potential is shifted by  $2$  kcal/mol to make this illustration legible.) Note that the addition of  $V_1$  to  $V_2$  shows that the *cis* conformation (dihedral<sub>hncO</sub> =  $0$  degrees) is destabilized relative to the *trans* conformation (dihedral<sub>hncO</sub> =  $180$  degrees).

## van der Waals Interactions and Hydrogen Bonding

A 6–12 function (also known as a Lennard-Jones function) frequently simulates van der Waals interactions in force fields (equation 11).

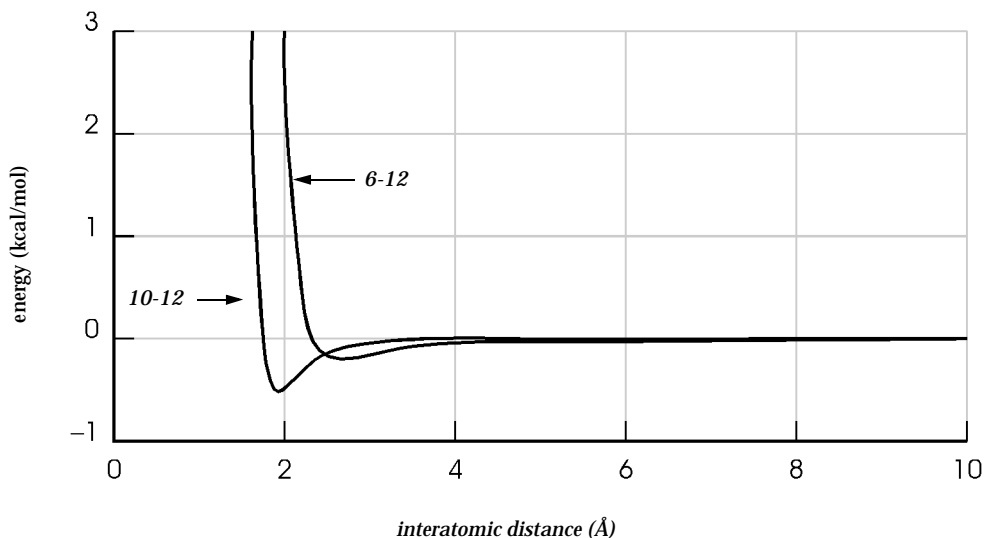
$$V_{\text{VDR}} = \sum_{i < j} \left[ \frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} \right] \quad (11)$$

$R_{ij}$  is the nonbonded distance between two atoms.  $A_{ij}$ , and  $B_{ij}$  are van der Waals parameters for the interacting pair of atoms. The  $R^{-6}$  term describes the attractive London dispersion interaction between two atoms, and the  $R^{-12}$  term describes the repulsive interaction caused by Pauli exclusion. The 6–12 function is not as appropriate physically as a 6-exponential function, but it is faster to compute.

The AMBER force field replaces the van der Waals by a 10–12 potential for pairs of atoms that can participate in hydrogen bonding (equation 12). The hydrogen bond potential does not contribute significantly to the hydrogen bonding attraction between two atoms; rather, it is implemented to fine-tune the distances between these atoms.

$$V_{\text{Hbonds}} = \sum_{\text{Hbonds}} \left[ \frac{C_{ij}}{R_{ij}^{12}} - \frac{D_{ij}}{R_{ij}^{10}} \right] \quad (12)$$

**Example:** In this example, the van der Waals (6-12) and hydrogen bond (10-12) potentials are quickly damped.



The attraction for two neutral atoms separated by more than four Ångstroms is approximately zero. The depth of the potential wells is minimal. For the AMBER force field, hydrogen bonds have well depths of about 0.5 kcal/mol; the magnitude of individual van der Waals well depths is usually less.

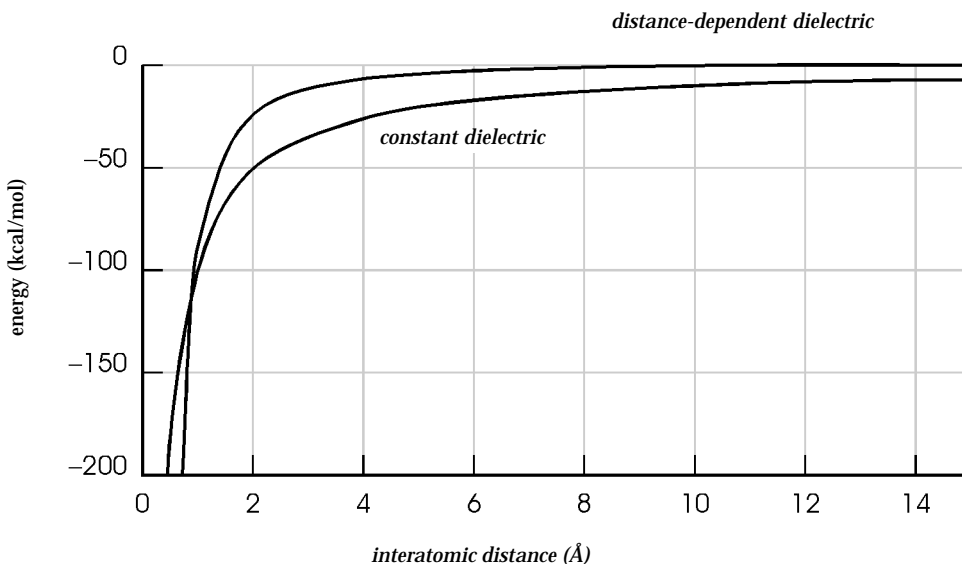
## Electrostatic Potential

This is a typical function for electrostatic potential (equation 13).

$$V_{\text{EEL}} = \sum_{i < j} \frac{q_i q_j}{\epsilon R_{ij}} \quad (13)$$

In this model of electrostatic interactions, two atoms ( $i$  and  $j$ ) have point charges  $q_i$  and  $q_j$ . The magnitude of the electrostatic energy ( $V_{\text{EEL}}$ ) varies inversely with the distance between the atoms,  $R_{ij}$ . The effective dielectric constant is  $\epsilon$ . For *in vacuo* simulations or simulations with explicit water molecules, the denominator equals  $\epsilon R_{ij}$ . In some force fields, a distance-dependent dielectric, where the denominator is  $\epsilon R_{ij} \cdot R_{ij}$ , represents solvent implicitly.

**Example:** For nonbonded interaction between two atoms having point charges of 0.616 and  $-0.504$  e, a distance-dependent dielectric, compared to a constant dielectric, reduces long range electrostatic interactions. Unlike van der Waals and hydrogen bond potentials, the magnitude of electrostatic potential energy between two atoms is large and remains significant over long nonbonded distances.



## United versus All Atom Force Fields

Because of the restricted availability of computational resources, some force fields use United Atom types. This type of force field represents implicitly all hydrogens associated with a methyl, methylene, or methine group. The van der Waals parameters for united atom carbons reflect the increased size because of the implicit (included) hydrogens.

United Atom force fields are used often for biological polymers. In these molecules, a reduced number of explicit hydrogens can have a notable effect on the speed of the calculation. Both the BIO+ and OPLS force fields are United Atom force fields. AMBER contains both a United and an All Atom force field.

**Caution:** If you are new to computational chemistry, do not use United Atoms for AMBER calculations. This HyperChem option is available for researchers who want to alter atom types and parameters for this force field.

## Cutoffs

Force field calculations often truncate the nonbonded potential energy of a molecular system at some finite distance. Truncation (nonbonded cutoff) saves computing resources. Also, periodic boxes and boundary conditions require it. However, this approximation is too crude for some calculations. For example, a molecular dynamic simulation with an abruptly truncated potential produces anomalous and nonphysical behavior. One symptom is that the solute (for example, a protein) cools and the solvent (water) heats rapidly. The temperatures of system components then slowly converge until the system appears to be in equilibrium, but it is not.

A relatively simple method for alleviating some of the nonphysical behaviors caused by imposing a nonbonded cutoff is to use a potential switching function (equation 14).

$$\text{When } R_{ij} \leq R_{\text{on}}, \quad E'_{\text{EEL}} = E_{\text{EEL}} \geq 1$$

$$\text{When } R_{\text{on}} < R_{ij} \leq R_{\text{off}}$$

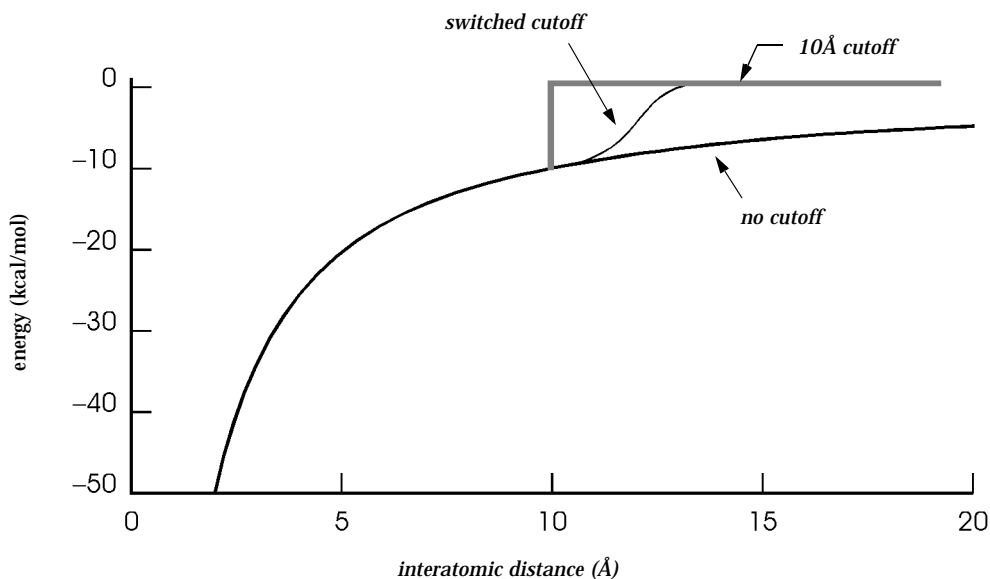
$$E'_{\text{EEL}} = E_{\text{EEL}} \cdot \frac{(R_{\text{off}} - R_{ij})^2 (R_{\text{off}} + 2R_{ij} - 3R_{\text{on}})}{(R_{\text{off}} - R_{\text{on}})^3} \quad (14)$$

$$\text{When } R_{ij} > R_{\text{off}}, \quad E'_{\text{EEL}} = E_{\text{EEL}} \geq 0$$

These functions allow the nonbonded potential energy to turn off smoothly and systematically, removing artifacts caused by a truncated potential. With an appropriate switching function, the potential function is unaffected except in the region of the switch.

*Example:* For two atoms having point charges of 0.616 and -0.504 e and a constant dielectric function, the energy curve shows a switching function turned on ( $R_{\text{on}}$ ) at a nonbonded distance of 10 Å and off ( $R_{\text{off}}$ ) at a distance of 14 Å. Compare the switched potential with the abruptly truncated potential.

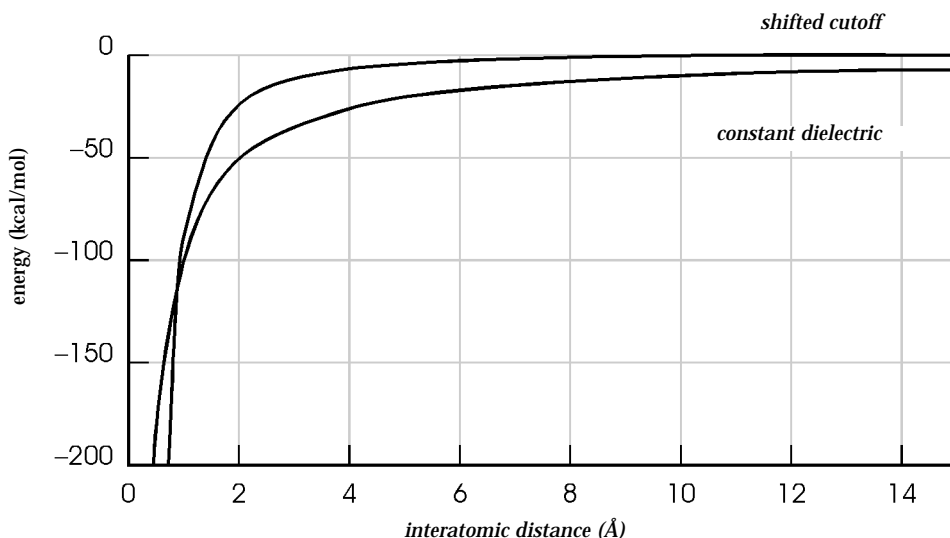




HyperChem also provides a shifting potential for terminating nonbonded interactions (equation 15).

$$E'_{\text{EEL}} = E_{\text{EEL}} \cdot \left( 1 - \frac{2R_{\text{ij}}^2}{R_{\text{off}}^2} + \frac{R_{\text{ij}}^4}{R_{\text{off}}^4} \right) \quad (15)$$

In an attempt to remedy truncation problems, a shifting potential multiplies the nonbonded electrostatic potential by a function that goes to zero. That is, the potential is shifted to zero at the cutoff  $R_{\text{off}}$ . Unlike the switching function, the shifted potential does not apply to van der Waals interactions.



**Caution:** Comparing the shifted constant dielectric to a constant dielectric function without a cutoff shows that the shifted dielectric, unlike a switching function, perturbs the entire electrostatic energy curve, not only the region near the cutoff.

## Quantum Mechanics

This section provides an overview and review of quantum mechanics calculations. The information can help you use HyperChem to solve practical problems. For quantitative details of quantum mechanics calculations and how HyperChem implements them, see the second part of this book, *Theory and Methods*.

*Ab initio* quantum mechanics methods have evolved for many decades. The speed and accuracy of *ab initio* calculations have been greatly improved by developing new algorithms and introducing better basis functions.

Density functional methods are newer methods that have a lot in common with *ab initio* methods. They use density functional theory (DFT) to predict how the energy depends on the density rather than the wavefunction. In particular, they replace the Hartree-Fock potential of *ab initio* calculations with an exchange-correlation potential that is a functional of the electron density. They include the effects of electron correlation and hence can, in prin-

cipal, be more accurate than *ab initio* Hartree-Fock calculations. Their results might be compared with *ab initio* Hartree-Fock plus MP2 correlation.

Semi-empirical quantum mechanics methods have evolved over the last three decades. Using today's microcomputers, they can produce meaningful, often quantitative, results for large molecular systems. The roots of the methods lie in the theory of  $\pi$  electrons, now largely superseded by all-valence electron theories.

## Background

Molecules consist of electrons and nuclei. Most applications of quantum chemistry separate the motion of the nuclei from the motion of electrons (the Born-Oppenheimer approximation). The approximation results in a model of nuclei moving on a potential energy surface, with electrons adjusting instantly to changes in nuclear positions. Nuclear motion is constrained by the interaction of nuclei and electrons. At any fixed positions of the nuclei, potential energy is the sum of repulsions among the positively charged nuclei and attractions arising from the electrons. Electrons are the “glue” holding nuclei together.

## Molecular Geometry

The most important chemical structures are stable, equilibrium molecular geometries and transition states. Equilibrium geometry of a molecule (bond lengths and angles) describes the coordinates of a deep minimum on a potential energy surface (see page 11). A set of atoms may have a number of potential energy minima, each corresponding to a different isomer of the molecular system. Other, less-deep minima may correspond to intermediates in chemical reactions.

Saddle points serve as models for transition states of chemical reactions. The potential energy for saddle points is a maximum along one direction, called the reaction coordinate, and a minimum along all others. You can think of saddle points as mountain passes: they are the lowest barriers separating reactant and product potential energy valleys. For a discussion of the potential energy surface, see “Exploring Potential Energy Surfaces” on page 11.

Using the coordinates of special geometries, minima, and saddle points, together with the nearby values of potential energy, you can calculate spectroscopic properties and macroscopic thermodynamic and kinetic parameters, such as enthalpies, entropies, and thermal rate constants. HyperChem can provide the geometries and energy values for many of these calculations.

## Calculating Electronic Potential Energy

How is electronic potential energy computed? Electrons, which are more than three orders of magnitude lighter than nuclei, are too small for classical mechanics calculations. Electronic energy must

be computed by solving the quantum mechanical Schrödinger equation. The lowest energy solution to this equation is called the ground-state wave function. Molecular orbital methods provide approximate solutions of the Schrödinger equation, the ground-state wave function, and certain low energy excited states.

Quantum mechanics methods have several attractive features:

- They require no information about the location or geometry of bonds in a molecular system.
- Parameters for elements (basis functions in *ab initio* and DFT-methods usually derived from experimental data and empirical parameters in semi-empirical methods usually obtained from empirical data or *ab initio* calculations) are independent of the chemical environment. In contrast, parameters used in molecular mechanics methods often depend on the chemical environment.
- With simple precautions, quantum mechanics methods can describe bond breaking.
- HyperChem can perform quantum mechanics MO calculations on molecules containing 100 or more atoms. There is no restriction on the number of atoms, but larger structures may require excessive computing times and computer main memory.
- You can interpret results, including dipole moments and atomic charges, using the simple concepts and familiar vocabulary of the Linear Combination of Atomic Orbitals (LCAO)-molecular orbital (MO) theory.

Once HyperChem calculates potential energy, it can obtain all of the forces on the nuclei at negligible additional expense. This allows for rapid optimization of equilibrium and transition-state geometries and the possibility of computing force constants, vibrational modes, and molecular dynamics trajectories.

In addition to computing molecular geometries, energies, and forces, HyperChem computes and displays charge and spin density functions, atomic charges, dipole moments, and electrostatic potential. Individual molecular orbital contributions, important in visualizing and interpreting chemical results, are calculated and displayed graphically.

## Range of Quantum Mechanics Methods

The quantum mechanics methods in HyperChem differ in how they approximate the Schrödinger equation and how they compute potential energy. The *ab initio* and DFT methods expand molecular orbitals into a linear combination of atomic orbitals (LCAO) and do not immediately introduce any further approximation. *Ab initio* Hartree-Fock calculations then approximate the form of the final wavefunction determining the energy while DFT calculations approximate the relationship of the energy to the electron density.

The Extended Hückel method, for example, does not explicitly consider the effects of electron-electron repulsions but incorporates repulsions into a single-electron potential. This simplifies the solution of the Schrödinger equation and allows HyperChem to compute the potential energy as the sum of the energies for each electron.

Neglect of Differential Overlap (NDO) methods, such as CNDO and INDO, include electron repulsions, but the resulting equations are nonlinear. The energy calculation includes terms for pairs of electrons. These methods include nuclear repulsions, but only by reducing the charge on each nucleus by the number of core electrons shielding it. The other NDO methods, MINDO/3, MNDO, AM1, and PM3 replace nuclear repulsion terms in the potential energy by parameterized, core-repulsion terms. The terms compensate for considering only valence electrons in the electronic Schrödinger equation, and they incorporate effects of electron correlation (see “Configuration Interaction” on page 123). For more about these methods see “Using Quantum Mechanics Methods” on page 111 “Using Quantum Mechanics Methods” on page 111.

## Exclusion Principle

Because of the quantum mechanical Uncertainty Principle, quantum mechanics methods treat electrons as indistinguishable particles. This leads to the Pauli Exclusion Principle, which states that the many-electron wave function—which depends on the coordinates of all the electrons—must change sign whenever two electrons interchange positions. That is, the wave function must be antisymmetric with respect to pair-wise permutations of the electron coordinates.

**Example:** If two functions are  $\sin$  and  $\cos$  and two variables are  $x$  and  $y$ , then  $\sin(x)\cos(y)$  has the same sign if you interchange  $x$  and  $y$  to give  $\sin(y)\cos(x)$ . The antisymmetrized version,  $\sin(x)\cos(y) - \cos(x)\sin(y)$ , does change sign. This is the determinant:

$$\begin{vmatrix} \sin(x) & \cos(x) \\ \sin(y) & \cos(y) \end{vmatrix}$$

**Example:** If  $a(1)$  is a function of the coordinates of electron 1 and  $b(2)$  is a function of the coordinates of electron 2, then the product function,  $a(1)b(2)$ , is not antisymmetric. The function  $a(1)b(2) - b(1)a(2)$ , an expansion of a  $2 \times 2$  determinant, is antisymmetric.

The Exclusion Principle is fundamentally important in the theory of electronic structure; it leads to the picture of electrons occupying distinct molecular orbitals. Molecular orbitals have well-defined energies and their shapes determine the bonding pattern of molecules. Without the Exclusion Principle, all electrons could occupy the same orbital.

The Exclusion Principle also gives rise to the “exchange” phenomenon. This phenomenon is important in understanding the energy differences between electronic states of different spin symmetries; for example, singlet-triplet splittings and related properties. Exchange is also important in understanding the nature of the covalent bond. In the simplest description of bonds for the hydrogen molecule, almost all of the binding energy is due to exchange. (Binding energy is the energy of the two hydrogen atoms separated by infinity minus the energy of the stable hydrogen molecule at its equilibrium bond length.)

The Exclusion Principle is quantum mechanical in nature, and outside the realm of everyday, “classical” experience. Think of it as the inherent tendency of electrons to stay away from one another; to be mutually excluded. Exclusion is due to the antisymmetry of the wave function and not to electrostatic coulomb repulsion between two electrons. Exclusion exists even in the absence of electrostatic repulsions.

Consider what happens to the many-electron wave function when two electrons have identical coordinates. Since the electrons have the same coordinates, they are indistinguishable; the wave function should be the same if they trade positions. Yet the Exclusion Principle requires that the wave function change sign. Only a zero

value for the wave function can satisfy these two conditions, identity of coordinates and an antisymmetric wave function. For the hydrogen molecule, the antisymmetric wave function is  $a(1)b(1) - b(1)a(1)$ , which is zero. Since the wave function is a probability function, the probability of two electrons having the same coordinates is also zero. In this sense they are mutually excluded.

## Simplified Wave Functions

The simplest many-electron wave function that satisfies the Exclusion Principle is a product of  $N$  different one-electron functions that have been antisymmetrized, or written as a determinant. Here,  $N$  is the number of electrons (or valence electrons) in the molecule. HyperChem uses this form of the wave function for most semi-empirical and *ab initio* calculations. Exceptions involve using the Configuration Interaction option (see page 123). HyperChem computes one-electron functions, termed molecular spin orbitals, by relatively simple integration and summation calculations. The many-electron wave function, which has  $N!$  terms (the number of terms in the determinant), never needs to be evaluated.

$N$  molecular spin orbitals must be different from one another in a way that satisfies the Exclusion Principle. Because the wave function is written as a determinant, interchanging two rows of the determinant corresponds to interchanging the coordinates of the two electrons. The determinant changes sign according to the antisymmetry requirement. It also changes sign when two columns are interchanged; this corresponds to interchanging two spin orbitals.

To avoid having the wave function zero everywhere (an unacceptable solution), the spin orbitals must be fundamentally different from one another. For example, they cannot be related by a constant factor. You can write each spin orbital as a product of a space function which depends only on the  $x$ ,  $y$ , and  $z$  coordinates of the electron—and a spin function. The space function is usually called the molecular orbital. While an infinite number of space functions are possible, only two spin functions are possible: alpha and beta.

To use HyperChem for calculations, you specify the total molecular charge and spin multiplicity (see “Charge, Spin, and Excited State” on page 123). The calculation selects the appropriate many-electron wave function with the correct number of alpha or beta



electrons. You don't need to specify the spin function of each orbital.

## Hartree-Fock Wave Functions

Quantum mechanics calculations use either of two forms of the wave function: Restricted Hartree-Fock (RHF) or Unrestricted Hartree-Fock (UHF). Use the RHF wave function for singlet electronic states, such as the ground states of stable organic molecules.

Spin orbitals are grouped in pairs for an RHF calculation. Each member of the pair differs in its spin function (one alpha and one beta), but both must share the same space function. For  $N$  electrons,  $N/2$  different molecular orbitals (space functions) are doubly occupied, with one alpha (spin up) and one beta (spin down) electron forming a pair.

The UHF wave function is most often used for multiplicities greater than singlets. In this case, the space orbitals for alpha electrons need not be the same as for beta electrons. For open-shell systems, some of the spins are unpaired — alpha and beta electrons occupy different orbitals. This is because there are more alpha electrons than beta electrons, and they tend to repel each other more than they do beta electrons. This effect is known as exchange repulsion. It is a quantum mechanical effect with no classical analog and would persist even without electrostatic coulomb forces. Essentially, it is due to the requirement that the wave function must be antisymmetric (see “Exclusion Principle” on page 35). Exchange repulsion affects the choice of the semi-empirical method (see “Choosing a Semi-Empirical Method” on page 153).

The UHF wave function can also apply to singlet molecules. Usually, the results are the same as for the faster RHF method. That is, electrons prefer to pair, with an alpha electron sharing a molecular space orbital with a beta electron. Use the UHF method for singlet states only to avoid potential energy discontinuities when a covalent bond is broken and electrons can unpair (see “Bond Breaking” on page 47).

## Extending the Wave Function Calculation

### Extending the Wave Function Calculation

#### Configuration Interaction

You can extend the calculation of the Hartree-Fock semi-empirical wave function by choosing Configuration Interaction (CI) in the Semi-empirical Options dialog box and you can also extend the calculation of the Hartree-Fock *ab initio* wave function by choosing CI in the Ab Initio Options dialog box. Use CI for these electron configurations:

- closed-shell singlet ground states (for both semi-empirical and *ab initio* methods)
- half-electron, excited singlet states (for semi-empirical methods only)
- half-electron, doublet and triplet open-shell ground states (for semi-empirical methods only)

Configuration Interaction (or electron correlation) adds to the single determinant of the Hartree-Fock wave function a linear combination of determinants that play the role of atomic orbitals. This is similar to constructing a molecular orbital as a linear combination of atomic orbitals. Like the LCAO approximation, CI calculations determine the weighting of each determinant to produce the lowest energy ground state (see “SCF Technique” on page 45).

CI calculations can be used to improve the quality of the wavefunction and state energies. Self-consistent field (SCF) level calculations are based on the one-electron model, wherein each electron moves in the *average* field created by the other  $n-1$  electrons in the molecule. Actually, electrons interact instantaneously and therefore have a natural tendency to avoid each other beyond the requirements of the Exclusion Principle. This correlation results in a lower average interelectronic repulsion and thus a lower state energy. The difference between electronic energies calculated at the SCF level versus the exact nonrelativistic energies is the *correlation energy*.

There are two types of electron correlations: static and dynamic. Static correlation refers to a near degeneracy of a given state; a

dynamic correlation refers to the instantaneous avoidance of electrons with each other.

CI calculations are perhaps the most widely used method of going computationally beyond an SCF description. An SCF computation yields a configuration describing the orbital occupancy of the electrons. Other configurations may be generated from this reference configuration by exciting electrons from the set of occupied orbitals to the set of virtual (*unoccupied*) orbitals. A CI calculation yields a set of improved states, each of which is represented by a linear combination of these configurations.

There are two types of CI calculations implemented in HyperChem — singly excited CI and microstate CI. The singly excited CI which is available for both *ab initio* and semi-empirical calculations may be used to generate UV spectra and the microstate CI available only for the semi-empirical methods in HyperChem is used to improve the wave function and energies including the electronic correlation. Only single point calculations can be performed in HyperChem using CI.

You can use CI calculations to do the following:

- Calculate UV spectra
- Calculate the energy of excited states
- Study the making or breaking of bonds, and change of spin couplings (e.g. dissociation of H<sub>2</sub>)
- Capture the effects of London dispersion forces
- Describe a nearly degenerate state
- Study singlet-triplet splittings more accurately

You can use CI to predict electronic spectra. Since the CI wave function provides ground state and excited state energies, you can obtain electronic absorption frequencies from the differences between the energy of the ground state and the excited states.

You can also use a RHF wave function with CI for calculations involving bond breaking, instead of using a UHF wave function (see also “Bond Breaking” on page 47).

The Microstate CI Method lowers the energy of the uncorrelated ground state as well as excited states. The Singly Excited CI Method is particularly appropriate for calculating UV visible spectra, and

does not affect the energy of the ground state (Brillouin's Theorem).

Be careful when you use the Orbital Criterion for symmetrical systems. To get correct results, you must include all or none of any set of degenerate orbitals in the CI, not just some of them. Carrying out an RHF calculation first and studying the Orbitals dialog box will help you to spot degenerate orbitals and avoid this pitfall.

In large systems there can be many orbitals in a small energy range, and the size of the CI matrix can be very sensitive to the value of the maximum excitation if you use Energy Criterion. Since calculation time depends heavily on the size of the CI matrix, you can end up with very long calculations, especially if you use the *ab initio* methods or the MNDO, AM1, or PM3 semi-empirical methods. This could exhaust the memory of your system. Again, inspecting the results of an RHF (no CI) calculation will help you avoid these pitfalls.

Using CI may not necessarily improve the calculation of ground state energies. Parameters for the MINDO/3, MNDO, AM1, and PM3 methods already include the effects of CI. CI calculations require more computing time.

*Note:* When you use CI, HyperChem does not calculate forces (gradient).

## Møller-Plesset Perturbation Theory

Configuration interaction is a systematic procedure for going beyond the Hartree-Fock approximation. A different systematic approach for finding the correlation energy is perturbation theory (Rayleigh-Schrödinger many-body perturbation theory — RSPT). In this approach, the total Hamiltonian of the system is divided or partitioned into two parts: a zeroth-order part,  $H_0$  (which has known eigenfunctions and eigenvalues), and a perturbation,  $V$ . The exact energy is then expressed as an infinite sum of contributions of increasing complexity. The expressions for these contributions contain the eigenvalues of  $H_0$  and matrix elements of the perturbation between the eigenfunctions of  $H_0$ . By introducing the ordering parameter  $\lambda$ , the total Hamiltonian can be written as<sup>9</sup>

$$H = H_0 + \lambda V$$

9. Szabo, A.; Ostlund, N. *Modern Quantum Chemistry*. Macmillan, New York, 1985

The exact eigenfunctions and eigenvalues can now be expanded in a Taylor series in  $\lambda$ .

$$\psi_\lambda = \psi^{(0)} + \lambda\psi^{(1)} + \lambda^2\psi^{(2)} + \dots$$

$$E_\lambda = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \dots$$

If  $H_0$  is chosen wisely, then the perturbation is small and the perturbation expansion (i.e., the sum of the 1st, 2nd, ..., nth-order energies) converges quickly. To obtain a perturbation expansion for the correlation energy, the best way is to choose the Hartree-Fock Hamiltonian as the zeroth-order Hamiltonian. The application to N-electron molecular systems is sometimes called Møller-Plesset (MP) perturbation theory. These methods, which can be terminated at second (MP2), third (MP3), or fourth order (MP4), with these three being the most frequently used in different *ab initio* programs, calculate the correlation energy and rely on a good description of the virtual orbitals in the original SCF function. The calculated total correlation energy is therefore quite dependent on the quality of the basis set.

HyperChem supports MP2 (second order Møller-Plesset) correlation energy calculations using *ab initio* methods with any available basis set. In order to save main memory and disk space, the HyperChem MP2 electron correlation calculation normally uses a so called “frozen-core” approximation, i.e. the inner shell (core) orbitals are omitted. A Registry setting allows excitations from the core orbitals to be included if necessary (melted core). Only the single point and geometry optimization calculations are available for this option.

## Density Functional Theory (DFT)

Density functional calculations appear, at first glance, to be almost identical to *ab initio* Hartree-Fock calculations apart from having to choose a specific functional for the exchange and correlation to replace the Hartree-Fock functional. One solves a set of SCF equations in the same way as one does in *ab initio* Hartree-Fock calculations.

## Kohn-Sham Orbitals

These similarities are somewhat of an illusion, however, and the results of *ab initio* and DFT calculations may have to be interpreted quite differently. For example DFT results in orbitals that are referred to as Kohn-Sham orbitals. These should not be used to perform Configuration Interaction calculations in the way one uses the Hartree-Fock orbitals. HyperChem allows one, in practice, to use the computed orbitals of DFT in exactly the same ways that one can use Hartree-Fock orbitals (once the orbitals are computed via either of the two methods no further distinction is made). Kohn-Sham orbitals, however, do not satisfy a theorem called Brueckner's Theorem (as Hartree-Fock orbitals do) and hence their interpretations requires explicit consideration of the subtle differences between Hartree-Fock and DFT calculations.

## Molecular Orbitals and Electronic Structure

The molecular orbitals computed by HyperChem have several important properties. Each molecular orbital has a uniquely defined orbital energy. Interpretations of orbital energies give useful predictions. For example, the highest energy-occupied molecular orbital (HOMO) provides a remarkably accurate estimate of the first ionization potential (Koopmans' theorem<sup>10</sup>). Orbital energies are also important in interpreting molecular geometry (Walsh's rules) and chemical reactivity (Woodward-Hoffmann rules<sup>11</sup>).

### Orbital Occupancy

Because of the LCAO-MO approximation, *ab initio*, DFT, and semi-empirical calculations produce occupied and unoccupied (virtual) orbitals. The Aufbau or "building up" principle determines the occupancy of orbitals for the ground-state wave function. This first arranges the orbitals in order of increasing energy. Then it assigns electrons to orbitals beginning with the lowest energy. For example, in an RHF calculation for a molecule with ten electrons and twelve molecular orbitals, the five lowest energy orbitals receive two electrons each. The remaining seven virtual orbitals remain unoccupied and do not appear in the many-electron wave func-

10. Koopmans, T. On the Relation of Wave Functions and Eigenvalues to Individual Electrons of Atoms *Physica (Utrecht)* 1:104, 1933.

11. Woodward, R.B. and Hoffmann, R. *Conservation of Orbital Symmetry* Verlag Chemie, Weinheim, FRG, 1970.

tion. Some of these virtual orbitals are important in discussions of chemical reactivity and excited electronic states (see “Chemical Reactivity” on page 144).

In an Extended Hückel calculation, the Aufbau population of molecular orbitals is unambiguous. The calculation method is non-iterative and the total energy is proportional to the sum of the energies of occupied orbitals. The Aufbau guarantees the lowest energy wave function.

For NDO calculations (see “NDO Methods” on page 131), the Aufbau also produces the lowest energy wave function. However, in some cases, a calculation may converge to a result that is not the ground state (lowest energy), even though the Aufbau is satisfied (for example, RHF calculations on singlet biradicals or molecules with highly distorted bonds).

In extreme cases, a calculation may not converge on a lowest energy solution because of the iterative nature of the SCF procedure. The highest energy occupied (HOMO) and lowest energy unoccupied (virtual) orbitals (LUMO) may switch occupancies from one SCF cycle to the next when the two orbitals are close in energy (nearly degenerate). It is even possible, though unlikely, that the ground state would not satisfy the Aufbau—there is a virtual orbital with an orbital energy lower than an occupied orbital.

HOMO and LUMO, also known as Frontier orbitals, are important in interpreting results of a calculation (see “Frontier Molecular Orbitals” on page 146). You can use these molecular orbitals to compute the lowest excited electronic singlet state of molecules and the ground states of radicals.

## Atomic Orbitals and Their Interactions

HyperChem uses the Linear Combination of Atomic Orbitals-Molecular Orbital (LCAO-MO) approximation for all of its *ab initio*, DFT, and semi-empirical methods. If  $\psi_i$  represents a molecular orbital and  $\phi_\mu$  an atomic orbital, then the LCAO representation of the molecular orbital is

$$\psi_i = \sum_{\mu} C_{\mu i} \phi_{\mu} \quad (16)$$

where  $C_{\mu_i}$  is the coefficient or “weight” of the  $\mu$ th atomic orbital in the  $i$ th molecular orbital.

This equation is important in interpreting the results of calculations. In *ab initio*, DFT, and semi-empirical calculations, atomic orbitals are functions of the  $x$ ,  $y$ , and  $z$  coordinates of the electron that closely resemble the valence orbitals of the isolated atoms.

These atomic orbitals, called Slater Type Orbitals (STOs), are a simplification of exact solutions of the Schrödinger equation for the hydrogen atom (or any one-electron atom, such as  $\text{Li}^{+2}$ ). HyperChem uses Slater atomic orbitals to construct semi-empirical molecular orbitals. The complete set of Slater atomic orbitals is called the basis set. Core orbitals are assumed to be chemically inactive and are not treated explicitly. Core orbitals and the atomic nucleus form the atomic core.

Because the calculation of multi-center integrals that are inevitable for *ab initio* and DFT methods is very difficult and time-consuming, HyperChem uses Gaussian Type Orbital (GTO) for *ab initio* and DFT methods. In truly reflecting a atomic orbital, STO may be better than GTO, so HyperChem uses several GTOs to construct a STO. The number of GTOs depends on the basis sets. For example, in the minimum STO-3G basis set HyperChem uses three GTOs to construct a STO.

## SCF Technique

All molecular orbitals are combinations of the same set of atomic orbitals; they differ only by their LCAO expansion coefficients. HyperChem computes these coefficients,  $C_{\mu_i}$ , and the molecular orbital energies by requiring that the ground-state electronic energy be at a minimum. That is, any change in the computed coefficients can only increase the energy.

The technique for this calculation involves two steps. The first step computes the Hamiltonian or energy matrix. The elements of this matrix are integrals involving the atomic orbitals and terms obtained from the Schrödinger equation. The most important contributions to this matrix involve the overlap integrals, which are mainly responsible for the bonding between atoms.

The second step determines the LCAO coefficients by standard methods for matrix diagonalization. In an Extended Hückel calculation, this results in molecular orbital coefficients and orbital



energies. *Ab initio*, DFT, and NDO calculations repeat these two steps iteratively because, in addition to the integrals over atomic orbitals, the elements of the energy matrix depend upon the coefficients of the occupied orbitals. HyperChem ends the iterations when the coefficients or the computed energy no longer change; the solution is then “self-consistent.” The method is known as Self-Consistent Field (SCF) calculation.

*Note:* You should be aware that the length of semi-empirical calculations increases as the cube of the number of atomic orbitals, because this is the number of operations required for matrix diagonalization; and the length of *ab initio* and DFT calculations increases as the power of four of the number of atomic orbitals (basis functions), because of the calculation of two-electron repulsion integrals. A molecule with 60 atomic orbitals takes about eight times as long to compute as a molecule containing 30 atomic orbitals by using a semi-empirical method and sixteen times as long to compute by using the *ab initio* or DFT method.

## Virtual Orbitals

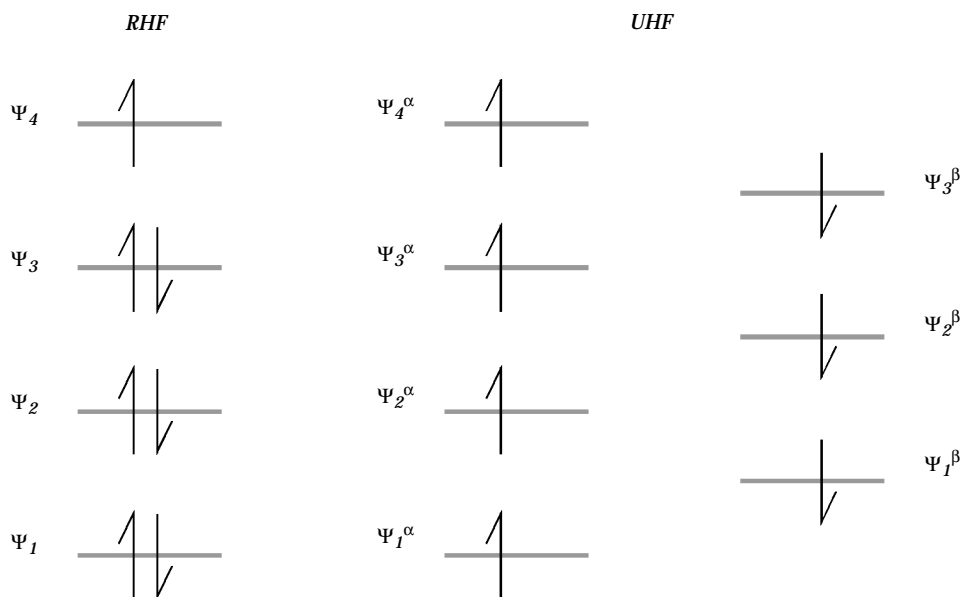
Even with the minimal basis set of atomic orbitals used in most semi-empirical calculations, the number of molecular orbitals resulting from an SCF calculation exceeds the number of occupied molecular orbitals by a factor of about two. The number of virtual orbitals in an *ab initio* or DFT calculation depends on the basis set used in this calculation.

## RHF and UHF

HyperChem quantum mechanics calculations must start with the number of electrons (N) and how many of them have alpha spins (the remaining electrons have beta spins). HyperChem obtains this information from the charge and spin multiplicity that you specify in the Semi-empirical Options dialog box or Ab Initio and DFT Options dialog box. N is then computed by counting the electrons (valence electrons in semi-empirical methods and all electrons in *ab initio* or DFT methods) associated with each (assumed neutral) atom and subtracting the charge. Multiplicity, which is related to the allowed values of the total spin angular momentum, is one for a singlet state, two for a doublet, three for a triplet, and four for a quartet.

## Multiplicity Considerations

Most stable organic molecules have an even number of electrons and a singlet ground-state wave function. Half the electrons are alpha. For a radical with an odd number of electrons, the ground state is almost always a doublet with one more alpha than beta electron. For some highly symmetrical molecules, such as oxygen, the ground state is a triplet with two more alpha than beta electrons. Biradicals (metastable species formed, for example, by breaking a bond in a ring molecule) also frequently have triplet ground states.



If you specify a multiplicity of one (singlet), then you would most often choose the RHF method, unless the reactions result in bond breaking (see page 47). If the selected multiplicity is greater than one, then the system is open-shell and the usual choice is the UHF method, which uses different orbitals for electrons with different spins.

## Bond Breaking

Applications of semi-empirical, *ab initio*, and DFT methods frequently follow the course of chemical reactions that break and

form covalent bonds. You must take care in choosing a method for these applications.

If a covalent bond is broken, as in the simple case of dissociation of the hydrogen molecule into atoms, then the RHF wave function without the Configuration Interaction option (see “Extending the Wave Function Calculation” on page 39) is inappropriate. This is because the doubly occupied RHF molecular orbital includes spurious terms that place both electrons on the same hydrogen atom, even when they are separated by an infinite distance.

An RHF wave function is also inappropriate for a reaction where reactants and products share different occupied orbitals, as in a Woodward-Hoffmann orbital symmetry forbidden thermal reaction. For example, in the absence of symmetry, the disrotatory ring opening of cyclobutene to butadiene must involve a change in the orbital occupancy between reactants and products. The lowest virtual orbital in reactants becomes the highest occupied in the product. This gives rise to an artificial and undesirable cusp in the potential energy surface along the reaction path, plus an unrealistically high activation energy. You can prevent this problem and the breaking of a covalent bond by using a UHF wave function for the singlet state. A minor disadvantage of the UHF method is that the space and spin symmetry of the molecular orbitals are broken. For example, the UHF wave function is a mixture of singlet and triplet states. To preserve spin symmetry, use RHF plus the Configuration Interaction option instead of UHF.

For Woodward-Hoffman allowed thermal reactions (such as the conrotatory ring opening of cyclobutane), orbital symmetry is conserved and there is no change in orbital occupancy. Even though bonds are made and broken, you can use the RHF wave function.

## RHF Half-Electron Technique

With all semi-empirical methods, HyperChem can also perform pseudo-RHF calculations for open-shell systems. For a doublet state, all electrons except one are paired. The electron is formally divided into two “half electrons” with paired spins. Each half electron carries half the charge of a whole electron. The calculation produces a set of molecular orbitals appropriate for this pseudo-wave function. HyperChem then assigns the unpaired electron its proper spin (alpha), substitutes this electron in the orbital for-

merly occupied by the half electrons, and calculates energy and other properties.

You can think of the half-electron technique as a device to get reasonable orbitals to populate later with whole electrons.

The half-electron method can also apply to triplet states. For this calculation, HyperChem populates selected molecular orbitals with pairs of half electrons. The final energy is computed by assigning the proper spins.

You can study the lowest excited singlet state of a system with a closed-shell ground state. The difference in energy between this state and the singlet ground state is related to the lowest electronic absorption frequency. Here, the calculation is carried out for the half-electron triplet state. The final energy is computed using the appropriate singlet wave function (which involves two determinants) for this singly excited state.

An advantage of the half-electron technique is its simplicity. HyperChem can carry it out with only minor modifications of the usual calculation. A disadvantage is that forces may not be accurate because of the half electron approximation.

Although UHF is often a better theoretical treatment of open-shell systems than the RHF (half-electron) methods, it takes longer to compute. Separate matrices for electrons of each spin roughly double the length of the calculation.

## SCF Convergence

Not all iterative semi-empirical, *ab initio*, or DFT calculations converge for all cases. For SCF calculations of electronic structure, systems with a small energy gap between the highest occupied orbital and the lowest unoccupied orbital may not converge or may converge slowly. (They are generally poorly described by the Hartree-Fock method.)

Some systems converge poorly, particularly those with multiple bonds or weak interactions between open-shell systems. HyperChem includes two convergence accelerators. One is the default convergence accelerator, effective in speeding up normally convergent calculations. The other, called the Direct Inversion in the Iterative Subspace (DIIS) method, is available for all the SCF methods in HyperChem. You can turn on this accelerator in the

Semi-empirical Options dialog box or in the Ab Initio and DFT Options dialog box. It uses more memory and computing time.

If you observe convergence problems, you might try a different semi-empirical method or first try optimizing the geometry using another method. Convergence problems occur more often away from optimized geometries than close to them. Often, a fast geometry optimization with a molecular mechanics calculation may solve convergence difficulties with a semi-empirical optimization.

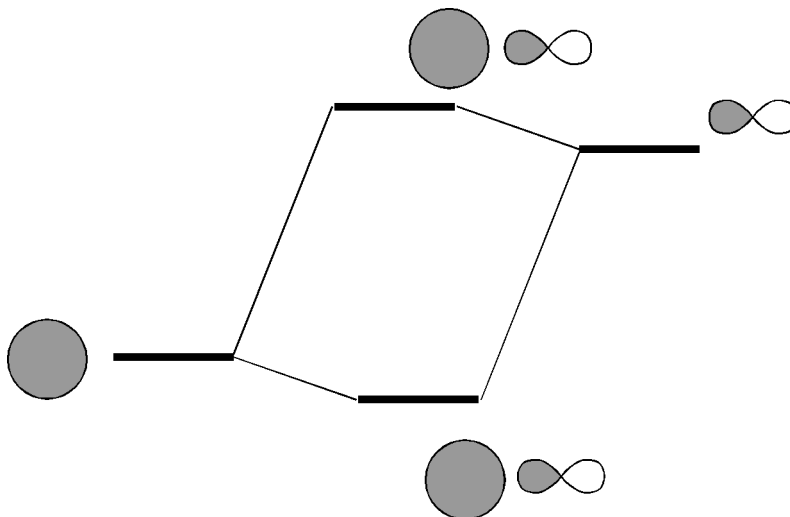
## **Applying an External Field**

An external electric or magnetic field can be applied to quantum mechanical calculations. The external field is part of the workspace and affects any result computed for the molecular system while the field is in effect.

Because of complexities in implementing magnetic fields, the external magnetic field is only available for TNDO semi-empirical calculations. The electric field is available for any quantum mechanical method.

## **Calculation Results**

The results of calculations are often interpreted in terms of familiar orbital interaction diagrams. The simplest example is two one-electron atoms combining to form a molecule. This diagram shows a two-orbital interaction forming a bonding (lower energy) and antibonding (higher energy) combination.



*Note:* This simple orbital interaction picture is useful for interpreting results, but neglects many aspects of a calculation, such as electron-electron interactions. These diagrams are closely related to the results from Extended Hückel calculations.

A larger interaction energy (the off-diagonal element of an energy matrix, proportional to the overlap integral between two molecular orbitals) causes a larger energy separation between two atomic orbitals, and the lower orbital is doubly occupied according to the Aufbau principle (see “Orbital Occupancy” on page 43). The magnitude of this additional splitting is proportional to the magnitude of the overlap integral between the two atomic orbitals. The greater the splitting, the greater the strength of the covalent bond. The occupied molecular orbital results from the in-phase or positive combination of the two atomic orbitals. This is shown for an s-type orbital on the left atom and a sigma p-type on the right. The virtual orbital is the negative or out-of-phase combination. The occupied or “bonding” molecular orbital shows a buildup of electron density between the two atoms. The virtual orbital, on the other hand, has a node (a value of zero) between the atoms.

A simple example is the formation of the hydrogen molecule from two hydrogen atoms. Here the original atomic energy levels are degenerate (they have equal energy), but as the two atoms approach each other, they interact to form two nondegenerate molecular orbitals, the lowest of which is doubly occupied.

In the general heteronuclear case, individual atomic orbital and overlap energies determine the magnitude of energy difference between two molecular orbitals. Atomic orbital energies are the diagonal elements of the Hamiltonian matrix, and the interaction energy between atomic orbitals are the off diagonal elements.

HyperChem can display molecular orbitals and the electron density of each molecular orbital as contour plots, showing the nodal structure and electron distribution in the molecular orbitals.

## **Quantitative Results**

HyperChem provides quantitative results for all calculations. The status line shows the binding energy and gradient (only binding energy if you use the Configuration Interaction option, see page 123). The log file for a calculation contains this information, plus more, including heats of formation (for all the SCF semiempirical methods in HyperChem), orbital coefficients, dipole moments, and atomic charges.

# Chapter 4

## Single Point Calculations

A single point calculation determines molecular properties, such as energy or spin density, of a defined molecular structure. Normally, these calculations are for stationary points on a potential energy surface. Occasionally, you may want to characterize the potential energy surface by calculating the energies of a grid of points on the surface. You can use those results to generate a contour plot of the surface.

With HyperChem, you can use either molecular or quantum mechanical methods for single point calculations. The calculation provides an energy and the gradient of that energy. The gradient is the root-mean-square of the derivative of the energy with respect to Cartesian coordinates. At a minimum the forces on atoms (the gradient) are zero. The size of the gradient can provide qualitative information to determine if a structure is close to a minimum.

In addition to total energy and gradient, HyperChem can use quantum mechanical methods to calculate several other properties. The properties include the dipole moment, polarizabilities, total electron density, total spin density, electrostatic potential, heats of formation, orbital energy levels, vibrational normal modes and frequencies, infrared spectrum intensities, and ultraviolet-visible spectrum frequencies and intensities. The HyperChem log file includes energy, gradient, dipole and polarizability values, while HIN files store atomic charge values.



## Dipole Moment

This describes the classical dipole moment:

$$\sum q_i \mathbf{r}_i$$

where  $q_i$  are point charges located at positions  $\mathbf{r}_i$ .

In the quantum mechanical description of dipole moment, the charge is a continuous distribution that is a function of  $\mathbf{r}$ , and the dipole moment is an average over the wave function of the dipole moment operator,  $\mu$ :

$$\mu = \sum_i (-\mathbf{r}_i) + \sum_A Z_A \mathbf{R}_A \quad (17)$$

where  $Z_A$  is the charge of the nuclear core and  $\mathbf{R}_A$  is the distance between the origin and nucleus A. Note that the sign convention used in the quantum mechanical calculation of dipoles is opposite to that used in molecular mechanics dipole calculations; this reflects the differing sign conventions of physics and chemistry.

## Electric Polarizabilities

The electric polarizabilities are computed as a tensor  $\alpha_{ij}$ ,

$$\alpha_{ij} = \frac{\partial \mu_i}{\partial F_j}$$

by differentiating the dipole moment in an electric field,  $\mathbf{F}$ . The  $ij$  element of the polarizability tensor is the dipole moment in the  $i$ th direction induced by an electric field in the  $j$ th direction.

## Total Electron Density

Electron density represents the probability of finding an electron at a point in space. It is calculated from the elements of the density matrix. The total electron density is the sum of the densities for alpha and beta electrons. In a closed-shell RHF calculation, electron densities are the same for alpha and beta electrons.

## Total Spin Density

Total spin density reflects the excess probability of finding  $\alpha$  versus  $\beta$  electrons in an open-shell system. For a system in which the  $\alpha$  electron density is equal to the  $\beta$  electron density (for example, a closed-shell system), the spin density is zero.

## Electrostatic Potential

The classical electrostatic potential for  $q_n$  point charges is the potential energy at a position  $\mathbf{R}$  (equation 18).

$$V = \sum \frac{q_i}{|\mathbf{R} - \mathbf{r}_i|} \quad (18)$$

For a molecule with a continuous electron charge distribution and nuclear point charges, the expression becomes:

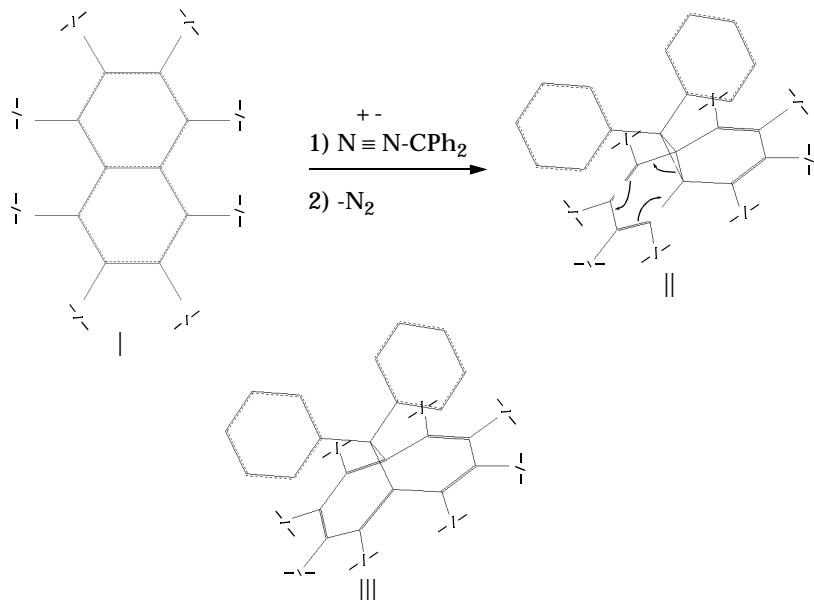
$$V = \sum \frac{Z_A}{|\mathbf{R} - \mathbf{R}_A|} - \int \frac{|\Psi|^2 d\tau}{|\mathbf{R} - \mathbf{r}|} \quad (19)$$

The potential indicates what a reactant 'sees' as it approaches a molecule. You can use a contour plot of electrostatic potential to visually determine possible trajectories for the approach of a reactant.

## Examples of Single Point Calculations

*Example:* Suzuki et. al. used a reaction strategy to expand the  $C_{60}$  molecule, buckminsterfullerene, by adding divalent carbon equivalents.<sup>11</sup> Adding phenyl diazomethane to  $C_{60}$ , (I), followed by the loss of molecular nitrogen, results in a  $C_{61}$  compound.

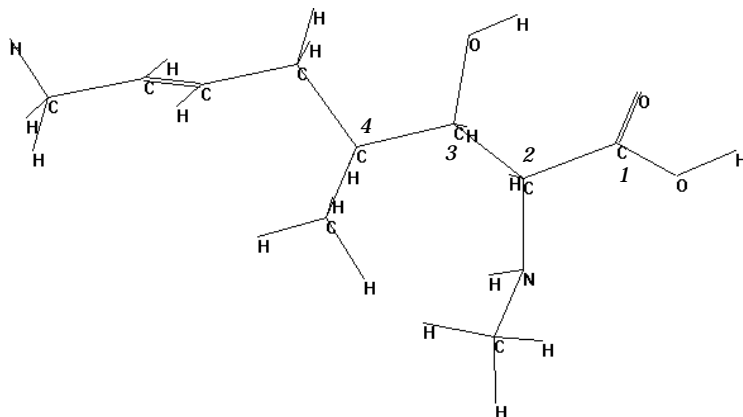
11. Suzuki, T.; Li, Q.; Khemani, K.C.; Wudl, F.; Almarsson, Ö. Systematic Inflation of Buckminsterfullerene  $C_{60}$ : Synthesis of Diphenyl Fullerooids  $C_{61}$  to  $C_{66}$  *Science*, 254:1186-1188, 1991.



By analogy to additions of divalent carbon to the C<sub>10</sub> aromatic framework, the molecule C<sub>61</sub> was expected to have the norcaradiene (II) or the cycloheptatriene (III) structure. Although an X-ray structure was not available, the UV-visible spectrum, <sup>13</sup>C NMR spectrum, and cyclic voltammetry supported the cycloheptatriene (III) structure. The researchers then calculated the relative molecular mechanics energies of II and III and found the cycloheptatriene structure stabilized by 31 kcal/mol with respect to the norcaradiene structure. Although the calculations do not confirm the structures, they provide additional supporting evidence.

*Example:* Miller and Rich investigated the conformational consequences of substitutions on an amino acid in cyclosporin A, an important immunosuppressive drug<sup>12</sup>. One of the amino acids in this cyclic undecapeptide is (2*s*, 3*r*, 4*r*, 6*e*)-3-Hydroxy-4-methyl-2-(methylamino)-6-octenoic acid (MeBmt). It is essential for biological activity.

12. Miller, K.E.; Rich, D.H. Molecular Mechanics Calculations of Cyclosporin A Analogues. Effect of Chirality and Degree of Substitution on the Side-Chain Conformations of (2*s*, 3*r*, 4*r*, 6*e*)-3-Hydroxy-4-methyl-2-(methylamino)-6-octenoic Acid and Related Derivatives. *J. Am. Chem. Soc.*, 111:8351-8362, 1989.



The researchers investigated three MeBmt analogs: the C-4 epimer ((4*s*)-MeBmt), the C-4 des methyl analog (MeBth), and the C-4 dimethyl (MeBm<sub>2</sub>t) analog. The immunosuppressive activity of the analogs follows the order MeBmt > MeBm<sub>2</sub>t > MeBth > (4*s*)-MeBmt.

Starting with the crystal and solution structures of cyclosporin A, Miller and Rich generated conformations of the parent structure and its analogs through a grid search of C<sub>1</sub>-C<sub>4</sub> torsional angles.

They minimized the conformations and calculated their potential energies, using the AMBER force field. They calculated a Boltzmann distribution for the conformations.

Miller and Rich assumed that cyclosporin A and its MeBmt analogs share a common bioactive conformation. Their molecular mechanics calculations found such a conformation. The Boltzmann population of the proposed bioactive conformation correlates with the immunosuppressive activities of the compounds.



# Chapter 5

## Geometry Optimizations and Transition State Searching

### Geometry Optimizations

HyperChem can calculate geometry optimizations (minimizations) with either molecular or quantum mechanical methods. Geometry optimizations find the coordinates of a molecular structure that represent a potential energy minimum.

These are excellent references for optimization calculations:

- Fletcher, R. *Practical Methods of Optimization*, John Wiley & Sons, New York, 1980
- Gill, P.E.; Murray, W.; Wright, M.H. *Practical Optimization*, Academic Press, Inc., New York, 1981

For a potential energy  $V$  and Cartesian coordinates  $\mathbf{r}_i$ , the optimized coordinates satisfy this equation:

$$\frac{\partial V}{\partial \mathbf{r}_i} = 0 \quad (20)$$

You might perform a geometry optimization calculation with one of these goals in mind:

- Characterize a potential energy minimum. A geometry optimization results in a new structure at a minimum. You can examine atomic coordinates and energy of this structure.
- Obtain a new stable structure as a starting point for a single point, quantum mechanical calculation, which provides a large set of structural and electronic properties.

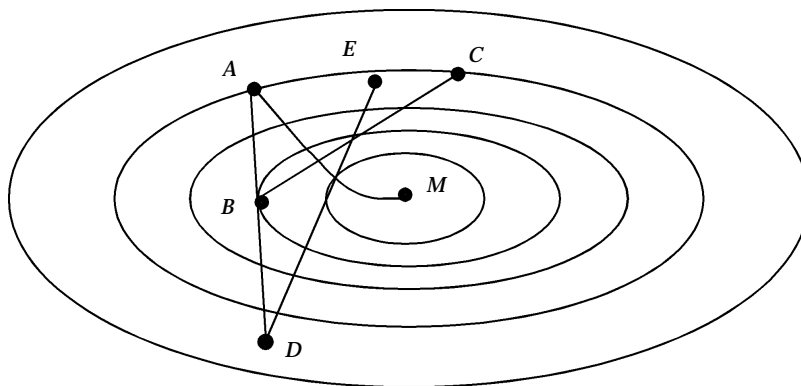
- Prepare a molecule for a molecular dynamics simulation. If the forces on atoms are too large, the integration algorithm may fail during a molecular dynamics calculation.

HyperChem supplies three types of optimizers or algorithms: steepest descent, conjugate gradient (Fletcher-Reeves and Polak-Ribiere), and block diagonal (Newton-Raphson).

## Steepest Descent

The steepest descent method is a first order minimizer. It uses the first derivative of the potential energy with respect to the Cartesian coordinates. The method moves down the steepest slope of the interatomic forces on the potential energy surface. The descent is accomplished by adding an increment to the coordinates in the direction of the negative gradient of the potential energy, or the force.

*Example:* A potential energy surface has a minimum at M. If the minimization begins at point A and proceeds with infinitesimally small steps, the structure follows the overall path A-M during a steepest descent optimization. If the first step is larger, it might proceed along A-B, then the next step would proceed along B-C. A larger initial step might place the system at D. A second step could proceed along the path D-E, which diverges from the minimum.



The steepest descent method rapidly alleviates large forces on atoms. This is especially useful for eliminating the large non-bonded interactions often found in initial structures. Each step in

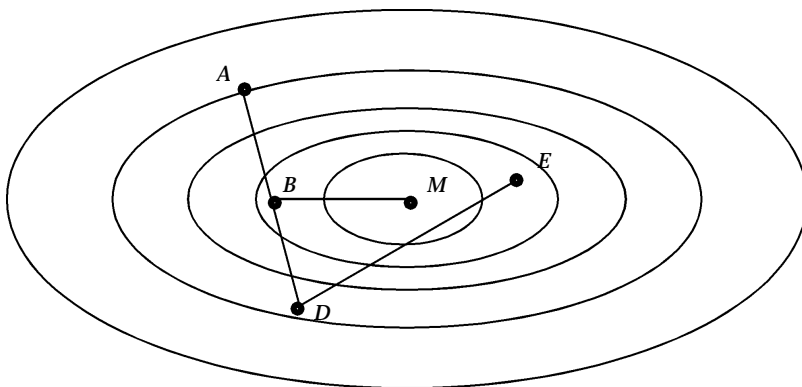
a steepest descent requires minimal computing time. Its disadvantage is that convergence toward a minimum is very slow.

## Conjugate Gradient

A conjugate gradient method differs from the steepest descent technique by using both the current gradient and the previous search direction to drive the minimization. A conjugate gradient method is a first order minimizer.

The advantage of a conjugate gradient minimizer is that it uses the minimization history to calculate the search direction, and converges faster than the steepest descent technique. It also contains a scaling factor,  $b$ , for determining step size. This makes the step sizes optimal when compared to the steepest descent technique.

*Example:* Compare the steps of a conjugate gradient minimization with the steepest descent method. A molecular system can reach a potential minimum after the second step if the first step proceeds from A to B. If the first step is too large, placing the system at D, the second step still places the system near the minimum (E) because the optimizer remembers the penultimate step.



HyperChem provides two versions of the conjugate gradient method, Fletcher-Reeves and Polak-Ribiere. Polak-Ribiere is more refined and is the default choice in HyperChem.

*Caution:* Geometry optimizations of large molecules may take longer than you expect. The number of computing cycles required for a conjugate gradient calculation is approximately proportional



to the number of atoms (N), and the time per cycle is proportional to  $N^2$ .

## Block Diagonal

The Newton-Raphson block diagonal method is a second order optimizer. It calculates both the first and second derivatives of potential energy with respect to Cartesian coordinates. These derivatives provide information about both the slope and curvature of the potential energy surface. Unlike a full Newton-Raphson method, the block diagonal algorithm calculates the second derivative matrix for one atom at a time, avoiding the second derivatives with respect to two atoms.

This technique is available only for the MM+ force field. As is true for the conjugate gradient methods, you should *not* use this algorithm when the initial interatomic forces are very large (meaning, the molecular structure is far from a minimum).

*Note:* Because of its neglect of off-diagonal blocks, this optimizer can sometimes oscillate and fail to converge. In this case, use a conjugate gradient method.

## Eigenvector Following

The Eigenvector Following method is in some ways similar to the Newton-Raphson method. Instead of explicitly calculating the second derivatives, it uses a diagonalized Hessian matrix to implicitly give the second derivatives of energy with respect to atomic displacements. The initial guess is computed empirically.

This method is available for all semi-empirical methods except Extended Hückel, and for *ab initio* calculations. This algorithm may be used if the structure is far from a minimum.

## Conjugate Directions

The Conjugate Direction method like the Conjugate Gradient method seeks to define directions of search that are conjugate to each other, i.e. optimum. Unlike the Conjugate Gradient optimizers, however, a conjugate direction optimizer only uses values of the independent variable itself without information about the gradient.

This makes such an optimizer much more generic since the value of the variable (energy) is almost always available while the gradients of that energy (or forces) may be more difficult to compute. Thus, for example, the MP2 correlation energy is relatively easy to compute while all the gradients for an MP2 calculation are more difficult to compute. Thus, a conjugate direction optimizer is the easiest to use or implement and sometimes only this optimizer is available for situations like the geometry of a ground state from a Hartree-Fock plus MP2 model.

On the other hand, a Conjugate Direction optimizer is not as good as one that has available the first derivatives of the energy. It is given less information to enable it to find a minimum. In particular it may not have a good idea how close to an energy minimum it is. The gradient which goes to zero at an energy minimum provides a good measure of convergence and this is unavailable for the conjugate directions method. Only when it is the only method available should the Conjugate Directions optimizer commonly be used.

## Setting Convergence Criteria

In setting up an optimization calculation, you can use two convergence criteria: the root-mean-square gradient and the number of optimization cycles.

You should terminate a geometry optimization based upon the root-mean-square gradient, because the number of cycles needed to minimize a molecule varies according to the initial forces on the atoms and the type of optimizer used. There is no way of predicting how many cycles are needed for an adequate geometry optimization.

HyperChem automatically calculates a default number of cycles as 15 times the number of atoms. Use the cycle number as a termination condition when mixing optimization algorithms. Suppose you are preparing a protein for a molecular dynamics simulation. The initial structure of the protein comes from X-ray crystallography studies. A molecular mechanics force field or a quantum mechanics method would undoubtedly find some large forces associated with the protein. An efficient minimization scheme would first subject the protein to 500–1000 cycles of steepest

descent minimization. Then it would minimize the structure using a conjugate gradient or Newton-Raphson optimizer, with the root-mean-square gradient as a termination condition.

## Examples of Geometry Optimizations

*Example:* Saunders used conformational searching and molecular mechanics to investigate the structure of a fully reduced buckminsterfullerene,  $C_{60}H_{60}$ .<sup>13</sup> By putting one of the hydrogens inside the reduced buckminsterfullerene he found that the molecule stabilized by 53 kcal/mol. After geometry optimization, the molecule with one hydrogen inside had three less eclipsed interactions and more favorable C-C-C angles than the isomer with all hydrogens outside. A search for other isomers, with different numbers of inside hydrogens, resulted in the finding that the most stable arrangement has ten hydrogens inside. This isomer, after minimization, had  $C_1$  symmetry, with an energy 402 kcal/mol more stable than the all-outside isomer. Each of the six-membered rings had one inside hydrogen, and the closest H...H nonbonded distance for the inside hydrogens was greater than 3 Ångstroms.

*Example:* Jensen and Gordon calculated the potential energy surface of glycine using *ab initio* and semi-empirical methods.<sup>14</sup> This study is of special interest to developers of molecular mechanics force fields. They frequently check their molecular mechanics methods by comparing their results with *ab initio* and semi-empirical calculations for small amino acids.

The researchers established that the potential energy surface is dependent on the basis set (the description of individual atomic orbitals). Using an *ab initio* method (6-31G\*), they found eight  $C_s$  stationary points for the conformational potential energy surface, including four minima. They also found four minima of  $C_s$  symmetry. Both the AM1 and PM3 semi-empirical methods found three minima. Only one of these minima corresponded to the 6-31G\* conformational potential energy surface.

13. Saunders, M. Buckminsterfullerene: The inside story. *Science* 253:330-331, 1991

14. Jensen, J.H.; Gordon, M.S. The conformational potential energy surface of glycine: A theoretical study. *J. Am. Chem. Soc.* 113:7917-7924, 1991.

## Solvation and Periodic Boundary Conditions

Solvation can have a profound effect on the results of a chemical calculation. This is especially true when the solute and solvent are polar or when they can participate in hydrogen bonding. The solvent effect is expressed in several ways, including these:

- The solvent can strongly influence the energies of different solute conformations or configurations of atoms.
- The nature of solute-solute and solute-solvent interactions is dependent on the solvent environment. Solvent influences the hydrogen-bonding pattern, solute surface area, and hydrophilic and hydrophobic group exposures.

Be cautious when projecting the results of calculations in one solvent to those using a different environment.

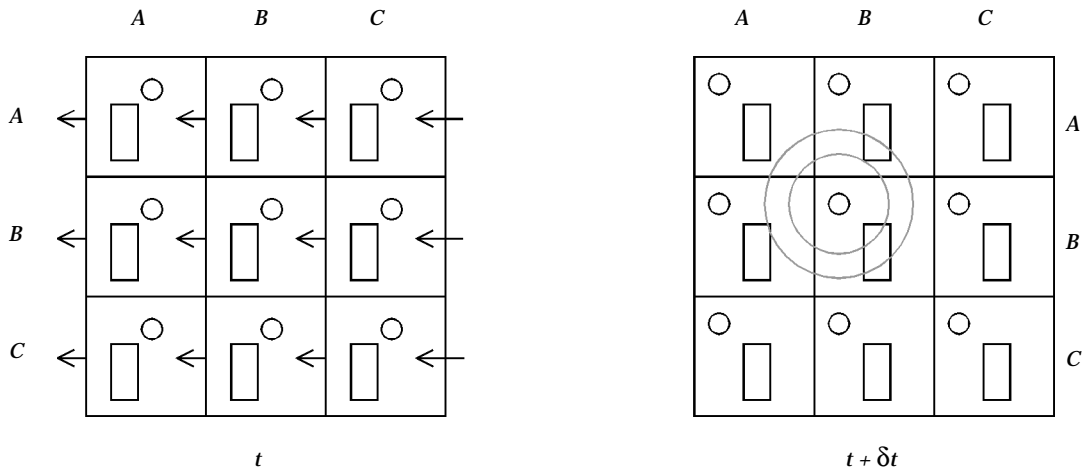
HyperChem uses the TIP3P water model for solvation.<sup>15</sup> You can place the solute in a box of TIP3P water molecules and impose periodic boundary conditions. You may then turn off the boundary conditions for specific geometry optimization or molecular dynamics calculations. However, this produces undesirable edge effects at the solvent-vacuum interface.

Periodic boundary conditions place the molecular system in a box. Virtual identical images surround the initial box. The atoms farthest from the center of the original box no longer suffer from interactions with an edge, since the original box now meets an image of itself in all directions. HyperChem uses 26 virtual images. The solute in the original box interacts only with its nearest neighbor solute images.

Since the periodic images are exact copies of the original box, movements of atoms in the original box are duplicated in the images. When a particle leaves a box, its periodic image enters from the opposite box side and replaces it. This conserves the number of atoms in each box.

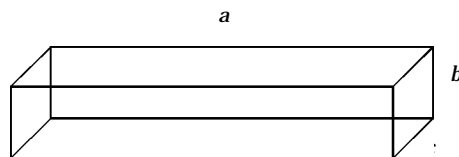
15. Jorgensen, W.L.; Chandrasekhas, J.; Madura, J.D.; Impey, R.W.; Klein, M.L. Comparison of simple potential functions for simulation liquid water. *J. Chem. Phys.* 79:926-935, 1983.

Choose the nonbonded cutoff carefully when using periodic boundary conditions. The cutoff must be small enough to prevent an atom from interacting simultaneously with another atom and with that atom's virtual image.



This example shows the round particle in cell {B,B} with two possible nonbonded cutoffs. With the outer cutoff, the round particle interacts with both the rectangle and its periodic image. By reducing the nonbonded cutoff to an appropriate radius (the inner circle), the round particle can interact with only one rectangle—in this case, the rectangle also in cell {B,B}.

For a rectangular box, the nonbonded cutoff should be less than one-half the smallest box dimension.



$$\text{cutoff} < 1/2b$$

## Transition State Searching

HyperChem can calculate transition structures with either semi-empirical quantum mechanics methods or the *ab initio* quantum mechanics method. A transition state search finds the maximum energy along a reaction coordinate on a potential energy surface. It locates the first-order saddle point; that is, the structure with only one imaginary frequency, having one negative eigenvalue.

There are excellent references for transition state search calculations and methods:

1. Baker, J., "An Algorithm for the Location of Transition States," *J. Comp. Chem*, Vol. 7(4), 385-395 (1986)
2. Peng, C. and Schlegel, H.B., "Combining Synchronous Transit and Quasi-Newton Methods to Find Transition States", *Israel Journal of Chemistry*, Vol. 33, 449-454 (1993)

Computational strategies and many examples can also be found in: Hehre, W.J., Radom, L., Schleyer, P. von R., and Pople, J. A., *Ab Initio Molecular Orbital Theory*, John Wiley & Sons, New York, 1986.

To find a first order saddle point (*i.e.*, a transition structure), a maximum must be found in one (and *only* one) direction and minima in all other directions, with the Hessian (the matrix of second energy derivatives with respect to the geometrical parameters) being varied. So, a transition structure is characterized by the point where all the first derivatives of energy with respect to variation of geometrical parameters are zero (as for geometry optimization) and the second derivative matrix, the Hessian, has one and only one negative eigenvalue.

You might perform a transition geometry optimization calculation with one of these goals in mind:

1. Characterize a potential energy maximum along the reaction coordinate. Transition state searching results in a new structure. You can examine the atomic coordinates and energy of this structure.
2. Characterize a potential energy surface for a certain number of atoms, *i.e.*, detect all the local energy minima, the global minimum on the surface, and all the transition states between different minima.

3. Calculate activation energies for chemical processes and energy barriers for isomerization between different isomers of the same molecule.

HyperChem supplies two different types or algorithms for transition state search: eigenvector following and synchronous transit (linear and quadratic search).

## Eigenvector Following

The eigenvector-following (or Hessian mode) method implemented in HyperChem is based on an efficient quasi-Newton like algorithm for locating transition states, which can locate transition states for alternative rearrangement/dissociation reactions, even when starting from the wrong region on the potential energy surface.

The algorithm consists of several steps. The first one involves making an initial guess at the position of the transition state. It will calculate the gradient vector  $\mathbf{g}$  and the Hessian matrix  $\mathbf{H}$ , at the initial point. The second step involve diagonalization of the Hessian and determination of local surface characteristics (number of negative eigenvalues). The next step depends on the structure of the Hessian. If the Hessian has the wrong number of negative eigenvalues, it will determine which Hessian mode has the greatest overlap with the eigenvector followed. If mode following has not been switched on, this algorithm will follow the lowest mode. The next step will determine SCF convergence. If the criteria are satisfied, it will stop at this point as the position of the transition state. If convergence criteria are not satisfied, it will calculate the energy and gradient vector at the new point, provided that maximum number of steps has not been exceeded.

If a transition state has not been reached after a certain number of steps, the user may need to upgrade the Hessian and repeat the calculation. It may be necessary if many calculation steps have been done, and the current geometry differs considerably from the starting point. The Hessian calculated at the starting point and updated at each new point may not be appropriate at the geometry reached by the search. This procedure can also help to get to a good starting point where the Hessian has a correct structure with only one negative eigenvalue.

HyperChem uses the eigenvector following method described in Baker, J., "An Algorithm for the Location of Transition States," *J. Comp. Chem.*, Vol. 7(4), 385-395 (1986).

## Synchronous Transit

The synchronous transit method is also combined with quasi-Newton methods<sup>16</sup> to find transition states. Quasi-Newton methods are very robust and efficient in finding energy minima. Based solely on local information, there is no unique way of moving uphill from either reactants or products to reach a specific reaction state, since all directions away from a minimum go uphill. HyperChem has two synchronous transit methods implemented. The linear synchronous transit method (LST) searches for a maximum along a linear path between reactants and products. It may happen that this method will end up with a structure having two or more negative eigenvalues. The quadratic synchronous transit method (QST) is an improvement of the LST approach and searches for a maximum along a parabola connecting reactants and products, and for a minimum in all directions perpendicular to the parabola.

HyperChem offers a Reaction Map facility under the Setup menu. This is needed for the synchronous transit method to match reactants and products, and depending on  $\lambda$  (a parameter having values between 0 and 1, determining how far away from reactants structures a transition structure can be expected) will connect atoms in reactants and products and give an estimated or expected transition structure. This procedure can also be used if the eigenvector following method is later chosen for a transition state search method, *i.e.*, if you just want to get an estimate of the transition state geometry.

HyperChem uses the synchronous transit method described in Peng, C., and Schlegel, H.B., *Israel Journal of Chemistry*, **33**, 449-454 (1993).

16. Banerjee, A., Adams, N., Simons, J., and Shepard, R., *J. Phys. Chem.* **87**, 52(1985); Simons, J., Nichols, J., *Int. J. Quantum Chem.*, Quantum Chem. Symp. **24**, 263(1990)





# Chapter 6

## Molecular Dynamics

Molecular dynamics simulations calculate future positions and velocities of atoms, based on their current positions and velocities.<sup>16</sup> A simulation first determines the force on each atom ( $\mathbf{F}_i$ ) as a function of time, equal to the negative gradient of the potential energy (equation 21).

$$\mathbf{F}_i = -\frac{\partial V}{\partial \mathbf{r}_i} \quad (21)$$

where  $V$  = potential energy function and  $\mathbf{r}_i$  = position of atom  $i$ .

You can then determine the acceleration,  $\mathbf{a}_i$ , of each atom by dividing the force acting on it by the mass of the atom (equation 22).

$$\mathbf{a}_i = \frac{\mathbf{F}_i}{m_i} \quad (22)$$

The change in velocities,  $\mathbf{v}_i$ , is equal to the integral of acceleration over time. The change in the position,  $\mathbf{r}_i$ , is equal to the integral of velocity over time. Kinetic energy ( $K$ ) is defined in terms of the velocities of the atoms (equation 23).

$$K = \frac{1}{2} \sum_{i=1}^N m_i \mathbf{v}_i^2 \quad (23)$$

The total energy of the system, called the Hamiltonian, is the sum of the kinetic and potential energies (equation 24).

16. See van Gunsteren, W.F.; Berendsen, H.J.C. Computer simulation of molecular dynamics – methodology, applications, and perspectives in chemistry *Angewandte Chemie, International Edition in English*, 29:992-1023, 1990, and Karplus, M.; Petsko, G.A. Molecular dynamics simulations in biology *Nature* 347:631-639, 1990.

$$H(\mathbf{r}, \mathbf{p}) = K(\mathbf{p}) + V(\mathbf{r}) \quad (24)$$

where  $\mathbf{r}$  = the set of Cartesian coordinates and  $\mathbf{p}$  = the momenta of the atoms.

## Integration Algorithm

If the atomic positions at a particular time,  $t$ , are  $\mathbf{r}_i(t)$ , the velocities,  $\mathbf{v}_i(t)$ , are the first derivative of the positions with respect to time (equation 25).

$$\mathbf{v}_i(t) = \frac{d}{dt}\mathbf{r}_i(t) \quad (25)$$

HyperChem employs the leap frog algorithm to integrate the equations of motion. This algorithm updates the positions of atoms and the velocities for the next time step by this calculation (equation 26).

$$\mathbf{v}_i\left(t + \frac{1}{2}\Delta t\right) = \mathbf{v}_i\left(t - \frac{1}{2}\Delta t\right) + \Delta t\mathbf{a}_i(t) \quad (26)$$

In the first time step ( $\Delta t$ ), the velocities advance from time  $t=0$  to  $(t + 1/2 \Delta t)$ . In doing so, they “leap” over the positions at time  $t$ . The current velocities are then calculated using equation 27. This equation supplies only approximate velocities used to calculate energies at time  $t$ .

$$\mathbf{v}_i(t) = \frac{1}{2}\left[\mathbf{v}_i\left(t + \frac{1}{2}\Delta t\right) + \mathbf{v}_i\left(t - \frac{1}{2}\Delta t\right)\right] \quad (27)$$

The acceleration is determined and the cycle renewed. In the subsequent steps, the velocities advance from  $t-1/2 \Delta t$  to  $t+1/2 \Delta t$ . The resulting set of positions and velocities is the molecular dynamics trajectory.

During a molecular dynamics simulation, HyperChem stores the current positions,  $\mathbf{r}_i(t)$ , and the mid-step velocities,  $\mathbf{v}_i(t - 1/2 \Delta t)$ . Since the algorithm provides mid-step velocities, but not velocities,  $\mathbf{v}_i(t)$ , for the positions at time  $t$ , HyperChem calculates approximate values of  $E_{\text{TOT}}(t)$ . This results in slightly larger fluctuations in  $E_{\text{TOT}}(t)$  than an algorithm that calculates exact values of  $\mathbf{v}_i(t)$ .

## Length of Simulations

HyperChem integrates the equations of motion using very small time steps ( $\Delta t$ ). At each step, the algorithm evaluates energy and forces of the molecular system. Use a time step of about 0.5 to 1.0 femtoseconds (fs) for an All Atom system or 1 to 2 fs for a United Atom system. Small time steps allow the simulation to adequately integrate the highest frequency motions of the system, usually bond stretching vibrations on the order of several picoseconds. Adjust  $\Delta t$  for each molecular system to obtain energy conservation (see the next section).

One drawback to a molecular dynamics simulation is that the trajectory length calculated in a reasonable time is several orders of magnitude shorter than any chemical process and most physical processes, which occur in nanoseconds or longer. This allows you to study properties that change within shorter time periods (such as energy fluctuations and atomic positions), but not long-term processes like protein folding.

## Conservation of Energy

Molecular dynamics calculations use equations 25–27. HyperChem integrates equations 26 and 27 to describe the motions of atoms. In the absence of temperature regulation, there are no external sources or depositories of energy. That is, no other energy terms exist in the Hamiltonian, and the total energy of the system is constant.

One way to test for success of a dynamics simulation and the length of the time step is to determine the change in kinetic and potential energy between time steps. In the microcanonical ensemble (constant number, volume, and energy), the change in kinetic energy should be of the opposite sign and exact magnitude as the change in potential energy. Though the total energy of the system should not change during the simulation, molecular dynamics simulations only approximate this condition. Energy conservation is difficult to achieve using computers with finite precision.

Because of limitations in computer power and time, it is frequently impractical to run a constant energy molecular dynamics simulation. Several approximations to the energy (usually to the potential energy) are possible, which require modifying the Hamilto-

nian. The most extreme approximation is to cut off intermolecular nonbonded interactions. Another common modification is temperature scaling: changing the atomic velocities to maintain a desired amount of kinetic energy.

## Temperature Control

In a molecular dynamics calculation, you can add a term to adjust the velocities, keeping the molecular system near a desired temperature.<sup>17</sup> During a constant temperature simulation, velocities are scaled at each time step. This couples the system to a simulated heat bath at  $T_0$ , with a temperature relaxation time of  $\tau$ . The velocities are scaled by a factor  $\lambda$ , where

$$\lambda = \left[ 1 + \frac{\Delta t}{\tau} \left( \frac{T_0}{T} - 1 \right) \right]^{1/2} \quad (28)$$

If the coupling parameter (the Bath relaxation constant in HyperChem),  $\tau$ , is too “tight” (<0.1 ps), an isokinetic energy ensemble results rather than an isothermal (microcanonical) ensemble. The trajectory is then neither canonical or microcanonical. You cannot calculate true time-dependent properties or ensemble averages for this trajectory. You can use small values of  $\tau$  for these simulations:

- To obtain a minimum energy structure at 0K.
- To reach equilibrium temperature quickly before starting the equilibration phase of a simulation (see “Equilibration and Data Collection” on page 76).

If the Bath relaxation constant,  $\tau$ , is greater than 0.1 ps, you should be able to calculate dynamic properties, like time correlation functions and diffusion constants, from data in the SNP and/or CSV files (see “Collecting Averages from Simulations” on page 88).<sup>17</sup>

**Note:** This method of temperature regulation does not give all properties of the canonical ensemble. In particular, you cannot calculate  $C_V$ , heat capacity at constant volume.

17. Berendsen, H.J.C., Postma, J.P.M.; van Gunsteren, W.F.; DiNola, A.; Haak, J.R. Molecular Dynamics with coupling to an external bath *J. Chem. Phys.* 81:3684, 1984.

For a stable trajectory, use a Bath relaxation constant greater than 0.1 ps. A constant of 0.01 ps is too small and causes disturbances in a simulation.<sup>17</sup>

## **Translational and Rotational Quenching**

During a molecular dynamics trajectory, energy that resides in internal degrees of freedom can be transformed into translational or rotational energy. This may be undesirable if one prefers to focus on the internal degrees of freedom and not have the system rotate or translate off the screen, for example. Dialog box items are available to measure rotational and translation energy and to quench either or both of them as desired.

## **Simulation Periods**

A molecular dynamics simulation can have three distinct time and temperature periods: heating, simulation (run), and cooling. If you want to measure equilibrium properties of a molecular system, you can divide the simulation period into two parts: equilibration and data collection.

## **Initial Conditions and Heating**

A molecular dynamics simulation usually starts with a molecular structure refined by geometry optimization, but without atomic velocities. To completely describe the dynamics of a classical system containing  $N$  atoms, you must define  $6N$  variables. These correspond to  $3N$  geometric coordinates ( $x$ ,  $y$ , and  $z$ ) and  $3N$  variables for the velocities of each atom in the  $x$ ,  $y$ , and  $z$  directions.

To begin a molecular dynamics simulation from this static structure, HyperChem assigns velocity values that are realistic for the molecular system at a designated temperature.

Since HyperChem-constructed molecular systems are near 0 K (the atoms have zero velocity), a simulation usually begins by adjusting the system to a higher temperature during a heating step. Heating can take place in one step (from near 0 K to simulation temperature), but it is better to heat to the simulation temperature slowly, in small temperature increments. Slow heating allows the system

to approach equilibrium during each heating step, so the system requires less time at the simulation temperature to reach equilibrium.

HyperChem can either use initial velocities generated in a previous simulation or assign a Gaussian distribution of initial velocities derived from a random number generator. Random numbers avoid introducing correlated motion at the beginning of a simulation.

HyperChem immediately scales (adjusts) the velocities of each atom so that the total kinetic energy is equal to  $3/2kT$ , where  $k$  is the Boltzmann constant and  $T$  is the specified starting temperature. Scaling creates a Gaussian distribution of velocities. After 100 to 300 steps of a simulation, this becomes a Boltzmann distribution.

*Caution:* For molecular dynamics simulations, use molecules that are optimized to an energy minimum or to a small gradient value. For room temperature simulations, the gradient should be  $\leq 3$ . At lower temperatures, the gradient should be smaller. This avoids artifacts caused by huge local forces in a high energy region. If a region of the molecule is highly strained, the molecular dynamics simulation tries to release the strain during the first few steps, leading to an explosion of the molecule or to structural distortion. Adequate geometry optimization is particularly important for molecular dynamics simulations of peptides or of large, flexible molecules.

## Equilibration and Data Collection

In many molecular dynamics simulations, equilibration is a separate step that precedes data collection. Equilibration is generally necessary to avoid introducing artifacts during the heating step and to ensure that the trajectory is actually simulating equilibrium properties. The period required for equilibration depends on the property of interest and the molecular system. It may take about 100 ps for the system to approach equilibrium, but some properties are fairly stable after 10–20 ps.<sup>18</sup> Suggested times range from 5 ps to nearly 100 ps for medium-sized proteins.

Equilibration corrects the velocities of atoms. Velocities resulting from heating do not simulate the type of motion found in a real molecular system. Instead, these velocities depend on a random

18. Levitt, M.; Sharon, R. *Proc. Nat. Acad. Sci. (U.S.)* 85:7557–7561, 1989.

distribution of values corresponding to a given temperature and on the forces in a partially minimized structure.

To generate characteristic velocities and bring a molecular system to equilibrium at the simulation temperature, atoms are allowed to interact with each other through the equations of motion. For isothermal simulations, a “temperature bath” scales velocities to drive the system towards the simulation temperature. Scaling occurs at each step of a simulation, according to equation 28.

## Tests for Equilibration

To determine when a molecular system reaches equilibrium, you can monitor fluctuations in temperature, kinetic energy, total energy, number of hydrogen bonds or nonbond contacts, or the number of times a particular configuration occurs. If you plot potential energy against time during a constant temperature run, then equilibration is close when average potential energy is constant. Other properties of the molecular system may also become constant unless a major conformational change occurs.

## Effect of Solvent on Equilibration

In systems with solvent molecules, many properties may still show fluctuations even after equilibration periods lasting several tens of picoseconds.

Molecular dynamics simulations of proteins often begin with a known structure (such as an X-ray diffraction structure) that you want to maintain during equilibration. Since the solvent may contain high energy hot spots, equilibration of the protein and solvent at the same time can change the protein conformation. To avoid this, select only the water molecules and run a molecular dynamics equilibration. This relaxes the water while fixing the protein structure. Then deselect the water and equilibrate the whole system.

*Caution:* During a simulation, solvent temperature may increase while the solute cools. This is particularly true of small solvent molecules, such as water, that can acquire high translational and rotational energies. In contrast, a macromolecule, such as a peptide, retains most of its kinetic energy in vibrational modes. This problem remains unsolved, and this note of caution is provided to advise you to give special care to simulations using solvent.



## Collecting Data

After initial heating and equilibration, the trajectory may be stable for thousands of time points. During this phase of a simulation, you can collect data. Snapshots and CSV files (see “Collecting Averages from Simulations” on page 88) store conformational and numeric data that you can later use in thermodynamic calculations.

In general, the longer the simulation, the more reliable the calculated properties. You can also average the data from several simulations.

Generating several different trajectories for the same molecular system can lead to more accurate and more reliable simulations of real physical systems and their properties. To increase the sampling of conformational states, you can run several medium-length simulations from the same molecular geometry but with different initial velocities or velocity vectors. You can change the distribution of velocities—but not their initial directions—by changing the number and size of heating steps. To change the directions of the initial velocities, you can rotate the molecular system in the Molecular Coordinate System, changing atomic coordinates.

## Cooling

Cooling a molecular system after heating or equilibration can serve these functions:

- Reduce stress on molecules caused by a simulation at elevated temperatures. The cooling process, called simulated annealing, takes new, high energy conformational states toward stable conformations.
- Overcome potential energy barriers and force a molecule into a lower energy conformation than the one you might obtain using geometry optimization alone.

## Examples of Molecular Dynamics Simulations

*Example:* Brady investigated classical dynamics of  $\alpha$ -d-glucose in water.<sup>19</sup> In this simulation, 207 water molecules surrounded one

$\alpha$ -d-glucose. The system was in a cubic box with periodic boundary conditions. During the simulation, several hydroxyl group transitions occurred. These transitions are normally unlikely with an *in vacuo* simulation.

You can detect hydroxyl group transitions by plotting dihedral angles versus time over the course of the simulation. This is the “distance history.” Brady investigated the distance history of water near the hydroxyl groups and found that the water structure for some conformations of the C-2 hydroxyl were different than for the anomeric hydroxyl group. He suggested that this difference might be related to the free energy differences between the  $\alpha$ - and  $\beta$ -anomers.

*Example:* Paulsen and Ornstein investigated the molecular dynamics of a 414-amino acid protein, cytochrome p-450<sub>cam</sub>.<sup>20</sup> The 175-psec simulation, with 40 psec used to equilibrate the protein, was carried out at constant temperature. The simulation was performed *in vacuo*, although 204 crystallographic waters were included in the structure. The trajectory analysis included the calculation of root-mean-square deviations and fluctuations, dihedral angle changes, and a hydrogen bond analysis. The authors analyzed the dynamic motion near the protein binding site with respect to a ‘gate-opening’ mechanism for substrate binding. In this example, the simulation does not model the motions of the protein.

## Strategies

Molecular dynamics simulations can overcome energy barriers and provide information about the time-dependent motion of molecular systems. You can use various strategies to set up and run a molecular dynamics simulation, depending on your objective. This section defines many of these strategies and discusses specific considerations in setting up a simulation.

19. Brady, J.W. Molecular dynamics simulations of  $\alpha$ -d-glucose in aqueous solution. *J. Am. Chem. Soc.* 111:5155–5165, 1989.

20. Paulsen, M.D.; Ornstein, R.L. A 175-psec molecular dynamics simulation of camphor-bound cytochrome p-450<sub>cam</sub> *Proteins* 11:184–204, 1991.

## Constant Temperature versus Constant Energy

If there is no external temperature control (using a simulated constant temperature bath), molecular dynamics simulations are constant energy.

For a constant temperature simulation, a molecular system is coupled to a heat bath via a Bath relaxation constant (see “Temperature Control” on page 74). When setting this constant, remember that a small number results in tight coupling and holds the temperature closer to the chosen temperature. A larger number corresponds to weaker coupling, allowing more fluctuation in temperature and possibly an unstable trajectory. However, weak coupling leads to a more natural trajectory.

Most molecular dynamics simulations, and particularly those for proteins, use constant temperature conditions.

## Conformational Searching

Molecular dynamics simulations are efficient for searching the conformational space of medium-sized molecules and peptides. Different protocols can increase the efficiency of the search and reduce the computer time needed to sample adequately the available conformations.

### Quenched Dynamics

Quenched dynamics is a combination of high temperature molecular dynamics and energy minimization. This process determines the energy distribution of conformational families produced during molecular dynamics trajectories. To provide a better estimate of conformations, you should combine quenched dynamics with simulated annealing.

For a conformation in a relatively deep local minimum, a room temperature molecular dynamics simulation may not overcome the barrier and search other regions of conformational space in reasonable computing time. To overcome barriers, many conformational searches use elevated temperatures (600–1200 K) at constant energy. To search conformational space adequately, run simulations of 0.5–1.0 ps each at high temperature and save the molecular structures after each simulation. Alternatively, take a snapshot of a simulation at about one picosecond intervals to store

the structure. Run a geometry optimization on each structure and compare structures to determine unique low-energy conformations.

*Note:* A molecular dynamics simulation cannot overcome constraints imposed by covalent bonds, such as disulfide bonds and rings. Check that such constraints are acceptable. Search other possible structures in separate simulations.

High temperature simulations require special consideration in choosing the sampling interval (see “Step size” on page 92).

The potential energy functions used in molecular mechanics calculations often cannot simulate the behavior of molecules at high temperature. This is true for United Atom representations of alpha carbons in amino acids; high temperatures can cause a change in chirality. In some cases, you can use restraints to control the behavior of molecules at high temperature (see “Using Geometric Restraints” on page 83).

Researchers report that high temperature molecular dynamics searches of many different starting conformations are much more efficient than using one starting structure and longer simulations.

## Simulated Annealing

Quenched dynamics can trap structures in local minima. To prevent this problem, you can cool the system slowly to room temperature or some appropriate lower temperature. Then run room temperature molecular dynamics simulations to search for conformations that have lower energies, closer to the starting structure. Cooling a structure slowly is called simulated annealing.

## Randomization During Molecular Dynamics

A problem in searching conformational space using molecular dynamics simulations is repeating a trajectory that generates the same structures. To reduce this possibility, you can randomize the velocities of the atoms.

To increase the possibility of finding new conformers, try one of these techniques:

- Run a molecular dynamics simulation, then rotate the molecular system in the Molecular Coordinate System. This changes

the coordinates of all atoms, but not the velocity vectors present at the end of the last molecular dynamics simulation.

- Run a molecular dynamics simulation. Start another simulation with the same molecular system and with Restart off in the Molecular Dynamics dialog box. HyperChem assigns a new set of velocities at random.

## Sampling Frequency

One of the most important considerations in using molecular dynamics for a conformational search is determining the sampling interval. HyperChem lets you sample the simulation in two ways:

- Snapshots at regular time intervals that store atomic coordinates and velocities. You can play back these snapshots to inspect the simulated structures or to average values. You specify a Snapshot period in the Molecular Dynamics Snapshots dialog box.
- Averages or plotted values at regular time intervals. You specify an Average/Graph period in the Molecular Dynamics Averages dialog box.

If the sampling interval is too short, the total number of structures that belong to unique structural families is small compared to the total number available. If the sampling interval is too large, the total number of available structures is small, and you can miss unique conformations.

High temperature molecular dynamics simulations add another consideration to setting the length of the sampling interval. Low potential energy regions in a molecule can develop high kinetic energies and velocities. If you are sampling a molecular dynamics simulation, you have a higher probability of observing the slow-moving conformations that have higher potential energy. Longer sampling periods accentuate this effect.

You must consider these factors when deciding on sampling interval. There is no ideal interval for all molecules; however, intervals of 0.5–1.0 ps are often used.

## When is Conformational Space Adequately Sampled?

One of the most important questions for a conformational search strategy is, “When have I found all of the energetically interesting

conformers?” This is an area of active research and the ideal answer seems to be, “When you find all of the local minima.” However, this answer is not always reasonable, because medium to large molecules have a large number of minima (see “Complexity of Potential Energy Surfaces” on page 14).

Sometimes you need only a quick survey of the available low energy structures near a known or suspected minimum. When you need a broader search, you can assess the completeness of a search by observing the amount of variation between families of unique structures. To obtain this information you can monitor torsion angles, nonbonded distances, or other properties. You can also make a histogram of the number of times a search found each conformation. If each structure occurs many times, you can be confident that the search was adequate for that region of the potential energy surface. If each conformation occurs only one or a few times, the search may not have covered all possible structures and more simulations are probably necessary.

## Using Geometric Restraints

You can include geometric restraints—for interatomic distances, bond angles, and torsion angles—in any molecular dynamics calculation or geometry optimization. Here are some applications of restraints:

- Include experimental data in a geometry optimization or molecular dynamics search.
- Force a geometric parameter to cross a barrier during a geometry optimization or molecular dynamics simulation.
- Reduce the possibility of undesired conformational changes during a molecular dynamics simulation.
- Dock two different molecules by restraining intermolecular (interatomic) distances.

Restraints add potential terms to a force field calculation, favoring the value that you specify in a restraint. The larger the value of the harmonic force constant, the more tightly the calculation restrains the value.

For all restraints, HyperChem uses named selections that contain two, three, or four atoms each. You use Name Selection on the

Select menu to assign names to groups of selected atoms. Then you can apply named selections as restraints for a calculation in the Restraint Forces dialog box from Restraints on the Setup menu.

**Caution:** Restraints may significantly affect the property you are observing. When using restraints, check their effects on measured values.

## Using Experimental Data as Restraints

You can often use experimental data, such as Nuclear Overhauser Effect (NOE) signals from 2D NMR studies, as restraints. NOE signals give distances between pairs of hydrogens in a molecule. Use these distances to limit distances during a molecular mechanics geometry optimization or molecular dynamics calculation. Information on dihedral angles, deduced from NMR, can also limit a conformational search.

When using NOE signals as restraints, choose the distance between the nonbonded atoms. A typical distance is about 4 to 5 Ångstroms. A typical force constant is 0.5 to 5.0 kcal/Å<sup>2</sup>, depending on the flexibility of the rest of the molecule and the strength of the NOE signal.

You usually remove restraints during the final phases of molecular dynamics simulations and geometry optimizations.

## Crossing Energy Barriers

Restraints can force molecules to cross energy barriers within reasonable calculation times. For example, you can flip a torsion angle using very high restraint values. This restraint still allows bond angles and distances for the atoms that define the dihedral angle to relax during a calculation. You can use this procedure with either geometry optimization or molecular dynamics.

## Limiting Conformational Changes during High Temperature Simulations

High temperature searches of conformational space (see “Quenched Dynamics” on page 80), can produce unwanted conformational changes, such as *cis-trans* peptide flips, ring inver-

sions, and other changes that you cannot reverse easily by geometry optimization. You can use restraints to prevent these changes.



## Docking Molecules

Restraints can facilitate docking a substrate molecule to a binding site. Restraints can also facilitate the interaction of two molecules in solution. In both cases, it is unlikely that two different neutral molecules would come into van der Waals contact with each other without the use of restraints.

*Example:* If a drug molecule interacts with a receptor molecule through hydrogen bonds, then you might restrain the distance between the donor and acceptor atoms involved in the hydrogen bonds. During a molecular dynamics simulation, these atoms would stay near an ideal value, while the rest of the molecular system fully relaxes.

*Example:* You could explore the possible geometries of two molecules interacting in solution and guess at initial transition structures. For example, if molecule A undergoes nucleophilic attack on molecule B, you could impose a distance restraint between the two atoms that would form a bond, allowing the rest of the system to relax. Simulations such as these can help to explain stereochemistry or reaction kinetics and can serve as starting points for quantum mechanics calculations and optimizations.

## Freezing Part of a System

You can completely freeze part of a molecule while allowing the remaining atoms to move in the field of the frozen atoms. This option is useful, for example, in a conformational search of part of a molecule.

To limit a molecular mechanics calculation to part of a molecule, select the atoms of interest. Only the selected atoms can move, but the other (frozen) atoms influence the calculation.

To limit a semi-empirical calculation to part of a molecule, select the atoms of interest. Then choose Extend to sp<sup>3</sup> on the Select menu to terminate correctly the selected group of atoms. Only the selected atoms move, but the frozen atoms influence the calculation. This is now possible with all the semi-empirical methods in HyperChem.

To limit an *ab initio* calculation to part of a molecular system, select the molecules of interest. Only the selected molecules move, but the frozen molecules influence the calculation. The *ab initio*

method implemented in HyperChem does not support any boundary atoms between the selected and unselected regions. In other words, the *ab initio* method does not accept partially selected molecule.

*Example:* You can use this technique to dock a flexible drug molecule (selected) to a rigid receptor site. You can then use restraints to search the conformations available to the drug (see “Using Geometric Restraints” on page 83) and to simulate the interactions of the drug with the macromolecule (see “Docking Molecules” on page 86).

*Example:* Molecular dynamics simulations of selected portions of proteins can demonstrate the motion of an amino acid sequence while fixing the terminal residues. These simulations can probe the motion of an alpha helix, keeping the ends restrained, as occurs naturally in transmembrane proteins. You can also investigate the conformations of loops with fixed endpoints.

## Solvent Simulations

Often you need to add solvent molecules to a solute before running a molecular dynamics simulation (see also “Solvation and Periodic Boundary Conditions” on page 65). In HyperChem, choose Periodic Box on the Setup menu to enclose a solute in a periodic box filled appropriately with TIP3P models of water molecules.

## Choice of Dielectric Constant

Before running a molecular dynamics simulation with solvent and a molecular mechanics method, choose the appropriate dielectric constant. You specify the type and value of the dielectric constant in the Force Field Options dialog box. The dielectric constant defines the screening effect of solvent molecules on nonbonded (electrostatic) interactions.

Use a constant dielectric of 1.0 with TIP3P water molecules in a periodic box. Because of the parameterization of TIP3P molecules, using a distance-dependent dielectric or a value other than 1.0 gives unnatural results.

A distance-dependent dielectric constant is commonly used to mimic the effect of solvent in molecular mechanics calculations, in the absence of explicit water molecules.

## Effects on Dynamic Motion

Including solvent in a molecular dynamics simulation creates a frictional force that damps some motion of the solute. This affects in particular the motions of exposed side chain in proteins.

*Caution:* Solvation in simulations can significantly increase computation time so that the simulation may be impractical.

## Collecting Averages from Simulations

Molecular dynamics calculations can automatically average and save these values:

- Kinetic energy (EKIN)
- Potential energy (EPOT)
- Total energy (ETOT)
- Temperature (TEMP)
- Any length, bond angle, or torsion defined in a named selection (see the *HyperChem Reference Manual*)
- The RMS deviation ( $D_x$ ) of any of the preceding values<sup>21</sup>

The calculation of average energies and their deviations from the mean are useful in several aspects of molecular dynamics simulations, such as these:

- Evaluating the stability of a simulation and the extent of equilibration.
- Identifying possibly interesting low energy conformations.

21.  $D_x = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$ , where  $x_i$  = value of EKIN, ETOT, EPOT, TEMP, or another value,

$$\langle x^2 \rangle = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right), \text{ and } \langle x \rangle^2 = \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2.$$

You choose the values to average in the Molecular Dynamics Averages dialog box. As you run a molecular dynamics simulation, HyperChem stores data in a CSV file. This file has the same name as the HIN file containing the molecular system, plus the extension `.csv`. If the molecular system is not yet stored in a HIN file, HyperChem uses the filename `chem.csv`.

For more information about the contents of CSV files, see appendix A, “HyperChem Files”, in the *HyperChem Reference Manual*.

## Evaluating Stability and Equilibration

Successful molecular dynamics simulations should have a fairly stable trajectory. Instability and lack of equilibration can result from a large time step, treatment of long-range cutoffs, or unrealistic coupling to a temperature bath.

### Constant Energy Simulations

You can check the success of a simulation by studying averages of the energy terms and their deviations. Since energy is conserved in a constant energy simulation, you can monitor the equilibration by dividing the deviation of the total energy (D ETOT) by the average total energy (ETOT). The data to calculate these averages is in the CSV file. This is a convenient way to obtain the averages for a molecular dynamics run:

4. Before starting a molecular dynamics simulation, L-click on Averages in the Molecular Dynamics Options dialog box.
5. Choose D ETOT and Add to average this value. Repeat this process for ETOT. L-click OK.
6. Set options for the simulation and L-click Proceed.
7. After a molecular dynamics simulation or the playback of a simulation, L-click on Averages in the Molecular Dynamics Options dialog box.
8. L-click on D ETOT. The average appears below, after Value. Write down this number. Repeat for ETOT. This average replaces the previous average.
9. Calculate the ratio.

For simulations of liquids, the generally accepted upper limit for the ratio is 0.0001. For proteins and macromolecules, this number

can be larger, about 0.001. However, constant energy simulations of proteins are uncommon.

You can also use deviation of the total energy (D ETOT) divided by the deviation of the kinetic energy (D EKIN) to assist in evaluating the quality of a simulation. Acceptable ratios are less than 0.05.

## Constant Temperature Simulations

To some extent you can monitor constant temperature simulations by the temperature (TEMP) and its deviation (D TEMP) or by kinetic energy (EKIN) and its deviation (D EKIN). Plot these values using the HyperChem Molecular Averages dialog box.

Large deviations in temperature and kinetic energy indicate there is a problem with the simulation or improper equilibration. Large fluctuations often accompany conformational or other geometry changes that induce rapid changes in the potential energy. It is useful to monitor both the kinetic and the potential energies as a function of time.

## Conformational Searches

A molecular dynamics simulation used for a conformational search can provide a quick assessment of low energy conformers suitable for further analysis. Plot the average potential energy of the molecule at each geometry. This plot may also suggest conformational changes in a molecule.

Evaluating average distances, angles, and torsion angles, plus their deviations, can facilitate understanding of detailed molecular properties and functional characteristics.

*Example:* The distance between two ends of a large, flexible molecule can provide information about its structural properties or its interaction with solvent. Analysis of an angle can reveal a hinged motion in a macromolecule.

*Example:* You can monitor improper torsion angles to determine which side of a substrate molecule faces the active site of a protein. Select three atoms on the substrate molecule and a fourth in the active site. These atoms define an improper torsion angle. Save this selection as a named selection. Then observe a plot of this improper torsion angle (in the Molecular Dynamics Results dialog

box) to follow the motion of the substrate as it interacts with the macromolecule.

## Setting Up a Molecular Dynamics Simulation

### Heating Time

The choice of heating time depends on the purpose of the molecular dynamics simulation. If the simulation is for conformational searches, the heating step is not critical for a successful calculation. The heating step may be rapid to induce large structural changes that provide access to more of the conformational space.

Heating and equilibration are critical (see page 75 and page 76) for investigating equilibrium properties of a molecular system. Several strategies are available to heat and equilibrate molecules.

For simulations of biological molecules, many researchers advocate slow heating. This allows the molecule to heat while keeping the structure close to the starting conformation. These simulations often investigate the potential energy surface near an important local minimum, such as a crystal structure. The heat time might be in the range of 5–10 ps, with a temperature step of 6 K, until the system reaches a simulation temperature of 300 K. The heating rate (30 K/ps) should allow a molecule to equilibrate during each temperature step.

For constant energy simulations without temperature regulation, use heating steps of about 0.5 ps and a heating time of 20–30 ps. In general, short heating times and large temperature steps perturb the initial system more than longer heating times and small temperature steps.

### Simulation or Run Time

The simulation or run time includes time for the system to equilibrate at the simulation temperature plus the time for data collection, while the trajectory evolves. Simulation times depend on the time scale of the property you are investigating.

*Example:* For a medium-sized molecule in the gas phase, with vibrations larger than  $50\text{ cm}^{-1}$ , a trajectory of only 10 ps should be adequate to explore motions due to vibrations.

*Example:* For protein molecules with motions occurring at a frequency of about  $1 \text{ cm}^{-1}$ , the vibrational period is about 30 ps. Clearly, simulations of hundreds of picoseconds are necessary to probe adequately the motions of this system.

## Cooling (Annealing) Time

Use the cooling time for simulated annealing (see “Cooling” on page 78). For simulated annealing following high temperature molecular dynamics (for example, 600 K), cool the system slowly to 300 K or lower. The result is a single molecular structure. To save computing time and obtain more annealed structures, start each new high temperature simulation from the coordinates and velocities of the previous high temperature simulation.

## Step Size

Step size is critical in all simulations. This is the increment for integrating the equations of motion. It ultimately determines the accuracy of the numerical integration. For molecules with high frequency motion, such as bond vibrations that involve hydrogens, use a small step size.

Temperature also determines step size. An acceptable time step for room temperature simulations is about 0.5–1 fs for All Atom systems or for simulations that do not constrain hydrogen atoms. For United Atom systems or systems containing only heavy atoms, you can use steps of 1–2 fs.

In general, smaller time steps generally lead to more accurate trajectories, but they can increase computing time significantly. Since each integration step requires about the same computing time, decreasing the time step from 1.0 fs to 0.1 fs may require an order of magnitude more time. You must balance accuracy with available computing resources.

Higher temperatures correspond to increased velocities and greater distances traveled by atoms between time steps. Generally, the length of a time step should allow only a small energy change. However, high temperature simulations can cause large energy changes between time steps, resulting in inaccurate integrations and unstable molecular dynamics trajectories. If a high temperature conformational search uses a large time step ( $> 1 \text{ fs}$ ), a molec-

ular structure can explode. These simulations sometimes require time steps as small as 0.1 fs.

## Temperature

The temperature of a simulation depends on your objectives. You might use high temperatures to search for additional conformations of a molecule (see “Quenched Dynamics” on page 80). Room temperature simulations generally provide dynamic properties of molecules such as proteins, peptides, and small drug molecules. Low temperatures (<250 K) often promote a molecule to a lower energy conformation than you could obtain by geometry optimization alone.

## Scripts

Molecular dynamics trajectories can be generated initially (real-time) or played back later at faster speed (playback-time). At playback time, it is possible to run any HyperChem script at each playback time step. This can be a very generic operation. For example, a trajectory might be generated and at playback time a quantum mechanical single point calculation might be requested, along with display of the electron density. This would result in a “movie” showing how the 3D electron density changes in time along a trajectory of the molecular system. This capability could be use, for example, to show the orbital changes during a chemical reaction.

The procedure for running a script during molecular dynamics play back is to place a script with a *specific name* in the HyperChem Program folder. This name is **chem-dynamics.scr**. If such a script exists, then it will be run at every playback time step. If it doesn't exist with this specific name, then play back will occur as normal. This functionality is similar to that for chem.scr which is run at instantiation time, if the script exists in the program folder.

## Bond Breaking

Quantum mechanical calculation of molecular dynamics trajectories can simulate bond breaking and formation. Although you do not see the appearance or disappearance of bonds, you can plot the distance between two bonded atoms. A distance exceeding a theoretical bond length suggests bond breaking.





# Chapter 7

## Langevin Dynamics and Monte Carlo

### Langevin Dynamics

The molecular dynamics method is useful for calculating the time-dependent properties of an isolated molecule. However, more often, one is interested in the properties of a molecule that is interacting with other molecules. With HyperChem, you can add solvent molecules to the simulation explicitly, but the addition of many solvent molecules will make the simulation much slower. A faster solution is to simulate the motion of the molecule of interest using Langevin dynamics.

Langevin dynamics simulates the effect of molecular collisions and the resulting dissipation of energy that occur in real solvents, without explicitly including solvent molecules. This is accomplished by adding a random force (to model the effect of collisions) and a frictional force (to model dissipative losses) to each atom at each time step. Mathematically, this is expressed by the Langevin equation of motion (compare to Equation (22) in the previous chapter):

$$\mathbf{a}_i = \frac{\mathbf{F}_i}{m_i} - \gamma \mathbf{v}_i + \frac{\mathbf{R}_i}{m_i}.$$

Here,  $\gamma$  is the friction coefficient of the solvent, in units of  $\text{ps}^{-1}$ , and  $\mathbf{R}_i$  is the random force imparted to the solute atoms by the solvent. The friction coefficient is related to the diffusion constant  $D$  of the solvent by Einstein's relation:  $\gamma = k_B T / mD$ . The random force is calculated as a random number, taken from a Gaussian distribu-

tion, with an average value of zero and no correlation with the atom's velocity.

## Integration Algorithm

To integrate the Langevin equation, HyperChem uses the method of: M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987; Ch.9, page 261:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + c_1\Delta t\mathbf{v}(t) + c_2\Delta t^2\mathbf{F}(t)/m + \Delta\mathbf{r}^G$$

$$\mathbf{v}(t+\Delta t) = c_0\mathbf{v}(t) + c_1\Delta t\mathbf{F}(t)/m + \Delta\mathbf{v}^G$$

where:

$$c_0 = e^{-t}$$

$$c_1 = (1 - c_0)/(\gamma \Delta t)$$

$$c_2 = (1 - c_1)/(\gamma \Delta t)$$

and  $\Delta\mathbf{r}^G$  and  $\Delta\mathbf{v}^G$  are random numbers chosen from a Gaussian distribution, with:

$$\langle \Delta\mathbf{r}^G \rangle = 0$$

$$\langle \Delta\mathbf{v}^G \rangle = 0$$

and standard deviations:

$$\sigma_r^2 = \Delta t^2(k_B T/m)(\gamma \Delta t)^{-1}(2 - (\gamma \Delta t)^{-1}(3 - 4 e^{-\gamma \Delta t} + e^{-2\gamma \Delta t}))$$

$$\sigma_v^2 = (k_B T/m)(1 - e^{-2\gamma \Delta t})$$

$\sigma_r$  and  $\sigma_v$  are correlated, with correlation coefficient  $c_{rv}$ , defined through the relation:

$$c_{rv}\sigma_r\sigma_v = \Delta t(k_B T/m)(\gamma \Delta t)^{-1}(1 - e^{-2\gamma \Delta t})$$

Assuming that the interatomic force term  $\mathbf{F}(t)$  varies linearly with time, the equations above can be rewritten in a form which produces more accurate simulations:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + c_1\Delta t\mathbf{v}(t) + c_2\Delta t^2\mathbf{F}(t)/m + \Delta\mathbf{r}^G$$

$$\mathbf{v}(t+\Delta t) = c_0\mathbf{v}(t) + (c_1 - c_2)\Delta t\mathbf{F}(t)/m + c_2\Delta t\mathbf{F}(t+\Delta t)/m + \Delta\mathbf{v}^G$$

As  $\gamma$  approaches zero, this method reduces to the velocity Verlet algorithm:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \Delta t\mathbf{v}(t) + \frac{1}{2}\Delta t^2\mathbf{F}(t)/m$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \frac{1}{2}\Delta t[\mathbf{F}(t) + \mathbf{F}(t+\Delta t)]/m$$

## Setting Up a Langevin Dynamics Simulation

In general, Langevin dynamics simulations run much the same as molecular dynamics simulations. There are differences due to the presence of additional forces. Most of the earlier discussions (see pages 71–94 and p. 342–360 of this manual) on simulation parameters and strategies for molecular dynamics also apply to Langevin dynamics; exceptions and additional considerations are noted below.

### Heating Time, Run Time, and Cooling Time

Most of the considerations described previously regarding the heat, run, and cool phases of a molecular dynamics simulation apply also to simulations that employ Langevin dynamics (see “Setting Up a Molecular Dynamics Simulation” on page 91). As with molecular dynamics, the time needed for these phases depends upon the purpose of the simulation and on the intrinsic dynamics of the system being studied as well. Gradual heating is needed for investigations of equilibrium properties. Conformational searches, on the other hand, are often successful with larger temperature steps and may be carried out with a very short heat period.

### Time Step

When the friction coefficient is set to zero, HyperChem performs regular molecular dynamics, and one should use a time step that is appropriate for a molecular dynamics run. With larger values of the friction coefficient, larger time steps can be used. This is because the solution to the Langevin equation in effect separates the motions of the atoms into two time scales: the short-time (fast) motions, like bond stretches, which are approximated, and long-time (slow) motions, such as torsional motions, which are accurately evaluated. As one increases the friction coefficient, the short-time motions become more approximate, and thus it is less important to have a small timestep.

## Temperature

Temperature is handled the same way in Langevin dynamics as it is in molecular dynamics. High temperature runs may be used to overcome potential energy barriers. Cooling a system to a low temperature in steps may result in a different stable conformation than would be found by direct geometry optimization.

## Constant Temperature vs. Constant Energy

As with molecular dynamics, Langevin dynamics simulations may be run at constant temperature, using the heat bath described under “Temperature Control” on page 74 to regulate the temperature. In the absence of coupling to a heat bath, the simulations are conducted at constant energy. For a constant temperature simulation, the bath relaxation constant determines the strength of the coupling to the bath. A small bath relaxation constant results in tight coupling and holds the temperature closer to the chosen temperature. A larger number corresponds to weaker coupling, allowing more fluctuations in temperature (and possibly an unstable trajectory).

## Friction coefficient

The friction coefficient determines the strength of the viscous drag felt by atoms as they move through the medium; its magnitude is related to the diffusion coefficient,  $D$ , through the relation  $\gamma = k_B T / mD$ . Because the value of  $\gamma$  is related to the rate of decay of velocity correlations in the medium, its numerical value determines the relative importance of the systematic dynamic and stochastic elements of the Langevin equation. At low values of the friction coefficient, the dynamical aspects dominate and Newtonian mechanics is recovered as  $\gamma \rightarrow 0$ . At high values of  $\gamma$ , the random collisions dominate and the motion is diffusion-like.

*Example:* Conformations of molecules like n-decane can be globally characterized by the end-to-end distance,  $R$ . In a comparison of single-molecule Brownian (Langevin) dynamics to molecular dynamics, the average end-to-end distance for n-decane from a 600 ps single-molecule Langevin dynamics run was almost identical to results from 19 ps of a 27-molecule molecular dynamics run. Both simulations were at 481K; the time step and friction coefficient for the Langevin simulation were 0.001 ps and 3.759 ps<sup>-1</sup>, respectively.<sup>16</sup>

*Example:* The dynamical effects of solvent on supercoiled DNA have been explored extensively by Langevin dynamics simulations. In one study, the effects of solvent damping and the random forces due to molecular collisions on the dynamics of DNA supercoiling were examined by systematically varying the viscosity (friction coefficient,  $\gamma$ ) over ten orders of magnitude. Three distinct physical regimes of DNA behavior were identified, from the low- $\gamma$  region where DNA motions are mostly harmonic, to the high friction limit where the dynamics are dominated by random forces and the global modes are effectively frozen by overdamping. Analogous studies of the dynamics of the damped harmonic oscillator exhibit similar behavior.<sup>17</sup>

## Additional Procedures

Refer to the following sections of this manual for related material: “Strategies” on page 79; “Using Geometric Restraints” on page 83; “Collecting Averages from Simulations” on page 88; “Molecular Dynamics on a Potential Energy Surface” on page 342.

# Monte Carlo Simulations

## Introduction

Monte Carlo simulations are commonly used to compute the average thermodynamic properties of a molecule or a system of molecules, and have been employed extensively in the study of the structure and equilibrium properties of liquids and solutions.<sup>18</sup> Monte Carlo methods have also been used to conduct conformational searches under non-equilibrium conditions.

Monte Carlo calculations are somewhat similar to the molecular (or Langevin) dynamics calculations discussed earlier. All function by repeated application of a computational algorithm that generates a new configuration from the current configuration. The

16. W.f. van Gunsteren, H.J.C. Berendsen and J.A.C.Rullmann. “Stochastic dynamics for molecules with constraints. Brownian dynamics of n-alkanes.” *Molecular Physics*, 1981, Vol. 44, No.1, 69-95.

17. G. Ramachandran and T. Schlick. “Solvent effects on supercoiled DNA dynamics explored by Langevin dynamics simulations.”, *Phys.Rev. E*, Vol. 51, No. 6, p.6188 (1995).

18. Allen, M. P., and Tildesley, D.J., *Computer Simulation of Liquids*, Oxford University Press, New York (1987).

sequence of configurations of the molecular system constitutes a trajectory, and after an initial equilibration phase, system properties may be sampled and averaged for the duration of the run to obtain ensemble averages. Dynamics simulations use an equation of motion as the basis for generating new configurations, which constitute the dynamical trajectory of the system. Monte Carlo simulations employ a statistical sampling technique to generate configurations that represent a trajectory in phase space.

Thus, unlike molecular dynamics or Langevin dynamics, which calculate ensemble averages by calculating averages over time, Monte Carlo calculations evaluate ensemble averages directly by sampling configurations from the statistical ensemble.

If run long enough, Monte Carlo and molecular dynamics should give the same average results for the same system (provided that the system is ergodic). However, only molecular dynamics can calculate time-dependent properties of a system, such as rotational frequencies or transition rates. On the other hand, Monte Carlo is generally better at sampling the allowed states of a system, and thus often can calculate the average properties more quickly and accurately.

## Background and Method

In order to compute average properties from a microscopic description of a real system, one must evaluate integrals over phase space. For an  $N$ -particle system in an ensemble with distribution function  $\mathbf{P}(r^N)$ , the experimental value of a property  $A(r^N)$  may be calculated from

$$\langle A(r^N) \rangle = \int A(r^N) P(r^N) d r^N .$$

The problem with direct evaluation of this multi-dimensional integral (apart from the huge number of phase space points to be sampled) is that most of the configurations sampled contribute nothing to the integral, having energy that is so high that the probability of their occurrence is vanishingly small. The trick, then, is to generate trajectories that sample commonly occurring configurations more than rare ones.

Thermodynamically, the probability of finding a system in a state whose energy is  $\Delta E$  above the ground state is proportional to  $\exp(-\Delta E/kT)$ . HyperChem uses the Metropolis method,<sup>19</sup> which chooses random configurations with this probability, to concen-

trate sampling of configurations in regions of space that make important contributions to the calculation of thermodynamic averages.

In the Metropolis method, a randomly selected atom attempts to move by a random distance in a random direction. The energy change associated with this move is calculated. If the energy change is negative, then the move is accepted, and the energy of the system is added to the currently running average. If the energy change is positive, then the move is accepted with probability  $\exp(-\Delta E/kT)$ . So, the larger the positive change in energy, the less likely it is that the move will be accepted. If a move is rejected, then the energy of the old configuration is again added to the average.

In HyperChem, the atoms are not chosen randomly, but instead, each atom is moved once per Monte Carlo step.

## Monte Carlo Trajectories and Simulation Parameters

### Initial Configurations

A good starting configuration for a Monte Carlo simulation is one that can relax quickly to a structure that is representative of the system to be studied. It is not necessary to begin with an energy-minimized structure (as with molecular dynamics); in fact, a random configuration could be used, at least in principle. However, in order to minimize computational time it is best to begin with a physically reasonable choice. Many simulations of the liquid state have used a crystal lattice as a reproducible and well-defined starting point. With adequate sampling, results of a simulation will be independent of the starting configuration; in fact, dependence of the results on the starting configuration can be monitored as one index of the reliability of the simulation.

### Temperature

The effect of temperature in Monte Carlo simulations is primarily to modulate the strength of intermolecular interactions, since temperature enters the simulation only through the Boltzmann factor  $\exp(-\Delta E/kT)$ , where  $\Delta E$  represents a difference in potential

19. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087-92 (1953).



energy. Thus, at higher temperatures, configurations of higher energy become more probable and contribute more to computed averages. The simulation temperature also determines the average kinetic energy, which contributes a fixed amount to the configurationally averaged potential energy to give the average total energy.

### **Step Size and Acceptance Ratio**

The step size,  $\Delta r$ , is the maximum allowed atomic displacement used in the generation of trial configurations. The default value of  $r$  in HyperChem is 0.05 Ångstroms. For most organic molecules, this will result in an acceptance ratio of about 0.5, which means that about 50% of all moves are accepted.

Increasing the size of the trial displacements may lead to more complete searching of configuration space, but the acceptance ratio will, in general, decrease. Smaller displacements generally lead to higher acceptance ratios but result in more limited sampling. There has been little research to date on what the optimum value of the acceptance ratio should be. Most researchers tend to try for an average value around 0.5; smaller values may be appropriate when longer runs are acceptable and more extensive sampling is necessary.

### **Initial Phase and Equilibration**

The first phase of a run will, in general, be used to allow the system to move away from high-energy and thus statistically insignificant configurations. During this period, the potential energy may either increase or decrease, depending on the starting point; its instantaneous value should be monitored. The number of steps required before equilibration has been achieved and equilibrium averages can be accumulated is strongly dependent on the system, the starting point, the step size, and the temperature of the simulation. Typical values for Lennard-Jones liquids are 500-1000 steps.

### **Equilibration and Statistical Averaging**

A thermodynamically stable system conserves energy. Thus, by monitoring the potential energy one can confirm that a stable (and productive) phase of the simulation has begun. Absence of systematic drift in computed averages is often used as a check on the stability of a Monte Carlo trajectory. Fluctuations in the energy should be computed and used to quantitatively assess the reliabil-

ity of the run; from statistical mechanics, one expects the mean-squared fluctuations in the energy for the canonical ensemble to be simply related to the constant-volume heat capacity:  $\langle(\delta E)^2\rangle = k_B T^2 C_V$ . For a simulation of finite duration this equality will not be strictly obeyed, but it is useful as a limiting value.

Averaging and storage of data for a Monte Carlo run are carried out in much the same way as for molecular dynamics (see p. 350-357).

## Heating and Cooling

The use of heating and cooling phases are optional and not often used in Monte Carlo simulations. However, they may be useful for some applications.



# Chapter 8

## Using Molecular Mechanics Methods

HyperChem offers four molecular mechanics force fields: MM+, AMBER, BIO+ (CHARMM), and OPLS (see “References” on page 110). To run a molecular mechanics calculation, you must first choose a force field. The following sections discuss considerations in choosing a force field.

### Availability of Parameters

Force fields give the best results for molecules similar to those used to develop its parameters. Choose a force field developed for a range of molecules similar to your molecular system.

### Force Field Features

The force field equations for MM+, AMBER, BIO+, and OPLS are similar in the types of terms they contain: bond, angle, dihedral, van der Waals, and electrostatic. There are some differences in the forms of the equations that can affect your choice of force field.

*Note:* The BIO+ force field is an implementation of the CHARMM (Chemistry at HARvard Macromolecular Mechanics) force field developed in the group of Martin Karplus at Harvard University. Like AMBER and OPLS, it is primarily designed to explore macromolecules.

#### MM+

MM+ is unique among the force fields in the way it treats bonds and angles. Both the bond and angle terms can contain higher

order terms than the standard quadratic. These bond and angle potentials express a harmonic motion better than a harmonic potential.

MM+ also contains a stretch-bend cross term, called a Urey-Bradley term. The other force fields in HyperChem normally evaluate nonbonded interactions for atoms separated by three or more bonds (1–4 and larger interactions). A Urey-Bradley term includes 1–3 interactions, which are critical for accurately simulating molecules. For example, the bond angles for heavy atoms in cyclobutane are compressed, compared to the natural bond angle for C sp<sup>3</sup> orbitals, and the carbon-carbon bonds have higher *p* character. This results in a weaker bond and an increased carbon-carbon bond length. A Urey-Bradley term allows for these structural changes and recognizes the unique 1–3 interactions which occur in strained molecules.

HyperChem supplements the standard MM2 force field (see “References” on page 110) by providing additional parameters (force constants) using two alternative schemes (see the second part of this book, *Theory and Methods*). This extends the range of chemical compounds that MM+ can accommodate. MM+ also provides cut-offs for calculating nonbonded interactions and periodic boundary conditions.

*Note:* MM+ is derived from the public domain code developed by Dr. Norman Allinger, referred to as MM2(1977), and distributed by the Quantum Chemistry Program Exchange (QCPE). The code for MM+ is not derived from Dr. Allinger’s present version of code, which is trademarked MM2™. Specifically, QCMP010 was used as a starting point for HyperChem MM+ code. The code was extensively modified and extended over several years to include molecular dynamics, switching functions for cubic stretch terms, periodic boundary conditions, superimposed restraints, a default (additional) parameter scheme, and so on.

The HyperChem MM+ code and program also differ from MM2(1977) by having parameters in text files separate from the code. These parameter files are available for your modification and additions. The parameters distributed with HyperChem include the public domain values, generally referred to as the MM2(1991) parameter set, that Dr. Allinger contributed to HyperCube, Inc. Parameters not obtained from Dr. Allinger are appropriately labeled in the distributed parameter files.

## Electrostatic Interactions

Another difference between the force fields is the calculation of electrostatic interactions. AMBER, BIO+, and OPLS use point charges to model electrostatic interactions. MM+ calculates electrostatic interactions using bond dipoles. The bond dipole method may not adequately simulate very polar or charged systems.

## Accuracy of Force Fields

For biological polymers, molecular mechanics force fields are not well substantiated by experimental data. You should be cautious about relying on predictions from these calculations.

Use the force fields that have demonstrated accuracy for particular molecules or simulations. For example, OPLS reproduces physical properties in liquid simulations extremely well. MM+ reproduces the structure and thermodynamic properties of small, nonpolar molecules better than AMBER, BIO+, and OPLS.

## Previous Experiences

Previous investigations might influence the choice of a molecular mechanics method. If molecular mechanics calculations of a particular compound or molecule type already exist, choose the same force field so you can make comparisons easily.

## Choosing Force Field Options

The *HyperChem Reference* manual and *Getting Started* discuss the sequence of steps to perform a molecular mechanics calculation. These steps include choosing a force field, force field options, and possible restraints.

## Dielectric Function

You can use two types of dielectric functions: a constant and a distance-dependent dielectric. Use constant dielectric for *in vacuo* systems and for molecular systems with explicit solvent molecules.

Also use constant dielectric for MM+ and OPLS calculations. Use the distance-dependent dielectric for AMBER and BIO+ to mimic the screening effects of solvation when no explicit solvent molecules are present. The scale factor for the dielectric permittivity,  $\epsilon$ , can vary from 1 to 80. HyperChem sets  $\epsilon$  to 1.5 for MM+. Use 1.0 for AMBER and OPLS, and 1.0–2.5 for BIO+.

## 1–4 Nonbonded Scale Factors

AMBER, BIO+, and OPLS scale 1–4 van der Waals and 1–4 electrostatic interactions. Although the value of the 1–4 nonbonded scale factors is an option in HyperChem, you should generally use recommended values. This is because during parameterization, the force field developers used particular values for the 1–4 nonbonded scale factors, and their parameters may not be correct for other scale factors.

The van der Waals scale factors used during force field parameterization are 0.5 for AMBER, 1.0 for BIO+, and 0.125 for OPLS. For 1–4 electrostatic interactions, use 0.5 for AMBER, BIO+, and OPLS.

## Nonbonded Cutoffs

You can choose to calculate all nonbonded interactions or to truncate (cut off) the nonbonded interaction calculations using a switched or shifted function. Computing time for molecular mechanics calculations is largely a function of the number of nonbonded interactions, so truncating nonbonded interactions reduces computing time. You must also truncate nonbonded interactions for periodic boundary conditions to prevent interaction problems between nearest neighbor images.

Use these guidelines for nonbonded interactions:

- For small and medium-sized molecules, calculate all nonbonded interactions.
- For larger molecules, such as proteins, use a switching function to dramatically decrease computing time.
- Use a switching function for periodic boundary conditions.
- Use a shifted function only to reproduce reported results. Since a shifted dielectric potential affects the entire potential energy surface, it is not recommended.

A switched function extends over the range of inner ( $R_{\text{on}}$ ) to outer ( $R_{\text{off}}$ ) radius and a shifted function from zero to outer ( $R_{\text{off}}$ ) radius. Beyond the outer radius, HyperChem does not calculate non-bonded interactions. The suggested outer radius is approximately 14 Ångstroms or, in the case of periodic boundary conditions, less than half the smallest box dimension. The inner radius should be approximately 4 Ångstroms less than the outer radius. An inner radius less than 2 Ångstroms may introduce artifacts to the structure.

## Restrains

You can restrain atoms during molecular mechanics or quantum mechanics calculations. Choosing restraints, via the Restraint Forces dialog box, applies additional harmonic forces—which you specify, to interatomic distances, angles, or dihedrals that you have set up as named selections.

*Note:* Restraints apply to distances, angles and dihedrals between bonded or nonbonded atoms. You can also restrain atoms to points in space.

You need to specify two parameters: the equilibrium value of the internal coordinate and the force constant for the harmonic potential. The equilibrium restraint value depends on the reason you choose a restraint. If, for example, you would like a particular bond length to remain constant during a simulation, then the equilibrium restraint value would probably be the initial length of the bond. If you want to force an internal coordinate to a new value, the equilibrium internal coordinate is the new value.

Recommended values for the force constant are 7.0 kcal/mol Å<sup>2</sup> for an interatomic distance, 12.5 kcal/mol degree<sup>2</sup> for an angle, and 16.0 kcal/mol degree<sup>2</sup> for a dihedral angle. Use the recommended values and then, if the internal coordinate is not sufficiently restrained, try a larger force constant.

See “Using Geometric Restraints” on page 83 for examples of using restraints.



## References

- MM2 Allinger, N.L. *J. Am. Chem. Soc.* 99:8127, 1977.
- AMBER Weiner, S.J.; Kollman, P.A.; Case, D.A.; Singh, U.C., Ghio, C.; Alagona, G.; Profeta Jr., S.; Weiner, P. A new force field for molecular mechanical simulation of nucleic acids and proteins *J. Am. Chem. Soc.* 106:765–784, 1984.
- Weiner, S.J.; Kollman, P.A.; Nguyen, D.T.; Case, D.A. An all atom force field for simulations of proteins and nucleic acids *J. Comput. Chem.* 7:230–252, 1986.
- Cornell, W.D.; Cieplak P.; Bayly, C.I.; Gould, I.R.; Merz, K.M., Jr.; Ferguson, D.M.; Spellmeyer, D.C.; Fox, T.; Caldwell, J.W.; Kollman, P.A., A second generation force field for the simulation of proteins and nucleic acids, *Journal of the American Chemical Society* 117, 5179-5197 (1995).
- CHARMM Brooks, B.R.; Brucoleri, R.E.; Olafson, B.D.; States, D.J.; Swaminathan, S.; Karplus, M. CHARM: A program for macromolecular energy, minimization, and dynamics calculations *J. Comput. Chem.* 4:187–217, 1983.
- MacKerell, A.D., Jr.; Bashford, D.; Bellott, M.; Dunbrack, R.L., Jr.; Evanseck, J.; Field, M.J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph, D.; Kuchnir, L.; Kuczera, K.; Lau, F.T.K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D.T.; Prodhom, B.; Reiher, W.E., III; Roux, B.; Schlenkrich, M.; Smith, J.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M, *J. Phys. Chem B*, 102, 3586.
- OPLS Jorgensen, W.L.; Tirado-Rives, J. The OPLS potential functions for proteins. Energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* 110:1657–1666, 1988
- Pranate, J.; Wierschke, S.; Jorgensen, W.L. *J. Am. Chem. Soc.* 113:2810, 1991.

# Chapter 9

## Using Quantum Mechanics Methods

There are three steps in carrying out any quantum mechanical calculation in HyperChem. First, prepare a molecule with an appropriate starting geometry. Second, choose a calculation method and its associated (Setup menu) options. Third, choose the type of calculation (single point, geometry optimization, molecular dynamics, Langevin dynamics, Monte Carlo, or vibrational analysis) with the relevant (Compute menu) options.

### Obtaining a Starting Structure

Use the HyperChem Model Builder to construct an approximate geometry. If you want to look at the relative energies of a set of molecules, you may want to carry out the calculations using fully optimized molecular geometries. These geometry optimization calculations can use either molecular mechanics or quantum mechanics to further refine the molecular geometry beyond that given by the Model Builder.

Geometry optimization using an *ab initio* or DFT quantum mechanical method may take much longer than using a semi-empirical quantum mechanical method. In turn, semi-empirical quantum mechanical calculations take longer than molecular mechanics (force field) calculations. You might first run a molecular mechanics optimization to get close to the optimized geometry and then refine with a semi-empirical optimization. Similarly, you might run a semi-empirical optimization before you run an *ab initio* or DFT geometry optimization. The closer the starting geometry is to the final geometry, the faster HyperChem can reach an optimized structure. For some molecules, such as certain inorganic molecules, molecular mechanics methods may not have appropriate parameters and will yield poor geometries. In those cases, use

the Model Builder to get a reasonable starting geometry, with the possibility of refining the geometry by semi-empirical calculations before submitting it to *ab initio* or DFT computation.

## Calculating Part of a Molecular System

You can perform quantum mechanical calculations on a part of a molecular system, such as a solute, while using molecular mechanics for the rest of the system, such as the solvent surrounding the solute. This boundary technique is available in HyperChem for all quantum mechanical methods. It is somewhat less complete with *ab initio* and DFT calculations than with semi-empirical calculations, however. With *ab initio* and DFT calculations the boundary must occur between molecules rather than inside a molecule.

Choose the region (single or multiple molecules) of interest for an *ab initio* or DFT calculation from the total molecular system. HyperChem performs the *ab initio* or DFT calculation for the selected region using the perturbation of an electrostatic potential arising from the net charges on the atoms of the unselected part. (For further details of this electrostatic potential perturbation implemented in HyperChem, please see the second part of this book, *Theory and Methods*).

Choose the atoms of interest for the semi-empirical calculation, then use the Extend to sp<sup>3</sup> option on the Select menu to establish the appropriate atomic boundaries for the quantum mechanics calculation. HyperChem substitutes parameterized pseudo-fluorine atoms for the portions of the molecule not included directly in the calculation (see the second part of this book, *Theory and Methods*).

The algorithms of the mixed classical-quantum model used in HyperChem are different for semi-empirical and *ab initio* or DFT methods. The semi-empirical methods in HyperChem treat boundary atoms (atoms that are used to terminate a subset quantum mechanical region inside a single molecule) as specially parameterized pseudo-fluorine atoms. However, HyperChem will not carry on mixed model calculations, using *ab initio* or DFT quantum mechanical methods, if there are any boundary atoms in the molecular system. Thus, if you would like to compute a wavefunction for only a portion of a molecular system using *ab initio* or DFT methods, you must select single or multiple isolated mole-

cules as your selected quantum mechanical region, without any boundary atoms.

Semi-empirical methods could thus treat the receptor portion of a single protein molecule as a quantum mechanical region but *ab initio* and DFT methods cannot. However, both semi-empirical and *ab initio* or DFT methods could treat solvents as a perturbation on a quantum mechanical solute. In the future, HyperChem may have an algorithm for correctly treating the boundary between a classical region and an *ab initio* or DFT quantum mechanical region in the same molecule. For the time being it does not.

## Setting Up Options

Certain options are needed to setup for running quantum mechanics calculations in HyperChem via the corresponding option dialog boxes on the Setup menu.

### Selecting Options for an *Ab Initio* or DFT Method

The *ab initio* or DFT options are available via the Ab Initio menu item or Density Functional menu item on the Setup menu.

### Choosing a Basis Set

Any set of one-electron functions can be a basis set in the LCAO approximation. However, a well-defined basis set will predict electronic properties using fewer terms than a poorly-defined basis set. So, choosing a proper basis set in *ab initio* and DFT calculations is critical to the reliability and accuracy of the calculated results.

There may be as many basis sets defined for polyatomic calculations as there are quantum chemists! One would like to define, in advance, the standard basis sets that will be suitable to most users. However, one also wants to allow sophisticated users the capability to modify existing basis sets or define their own basis sets. We have thus defined a HyperChem basis set file format and included with HyperChem a number of these \*.BAS files that define standard basis sets. Users, however, can define as many of their own basis sets as they like using this file format. The details of the HyperChem basis set file format are described in the *HyperChem Reference manual*.

Many conventional and commonly-used *ab initio* and DFT basis sets are supported in HyperChem. These basis sets include but are not limited to:

- STO-1G and STO-1G\* (H and He)<sup>22,23,24</sup>;
- STO-2G and STO-2G\* (H to Xe);
- STO-3G and STO-3G\* (H to Xe);
- STO-4G and STO-4G\* (H to Xe);
- STO-5G and STO-5G\* (H to Xe);
- STO-6G and STO-6G\* (H to Xe);
- 3-21G, 3-21G\*, and 3-21G\*\* (H to Ar)<sup>25,26,27</sup>;
- 4-21G, 4-21G\*, and 4-21G\*\* (H to Ne)<sup>28,29,30</sup>;
- 6-21G, 6-21G\*, and 6-21G\*\* (H to Ar);
- 4-31G, 4-31G\*, and 4-31G\*\* (H to Ne);
- 5-31G, 5-31G\*, and 5-31G\*\* (H to F);
- 6-31G, 6-31G\*, and 6-31G\*\* (H to Ar);
- 6-311G, 6-311G\*, and 6-311G\*\* (H to Ar)<sup>31,32</sup>;

22. Hehre, W.J.; Stewart, R.F.; Pople, J.A. Self-consistent molecular-orbital methods. I. Use of Gaussian expressions of Slater-Type Atomic Orbitals *J. Chem. Phys.* 51:2657-2664, 1969.

23. Collins, J.B.; Schleyer, P.V.; Binkley, J.S.; Pople, J.A. Self-consistent molecular-orbital methods. XVII. Geometries and binding energies of second-row molecules. A comparison of three basis sets *J. Chem. Phys.* 64:5142-5151, 1976.

24. Stewart, R.F. Small Gaussian expansions of Slater-Type Orbitals *J. Chem. Phys.* 52:431-438, 1970.

25. Binkley, J.S.; Pople, J.A.; Hehre, W.J. Self-consistent molecular orbital methods. 21. Small split-valence basis sets for first-row elements *J. Am. Chem. Soc.* 102:939-947, 1980.

26. Gordon, M.S.; Binkley, J.S.; Pople, J.A.; Pietro, W.J.; Hehre, W.J. Self-consistent molecular-orbital methods. 22. Small split-valence basis sets for second-row elements *J. Am. Chem. Soc.* 104:2797-2803, 1982.

27. Pietro, W.J.; Francl, M.M.; Hehre, W.J.; Defrees, D.J.; Pople, J.A.; Binkley, J.S. Self-consistent molecular orbital methods. 24. Supplemented small split-valence basis sets for second-row elements *J. Am. Chem. Soc.* 104:5039-5048, 1982.

28. Hehre, W.J.; Ditchfield, R.; Pople, J.A. Self-consistent molecular orbital methods. XII. Further extensions of Gaussian-Type basis sets for use in molecular orbital studies of organic molecules *J. Chem. Phys.* 56:2257-2261, 1972.

29. Hariharan, P.C.; Pople, J.A. The influence of polarization functions on molecular orbital hydrogenation energies *Theor. Chim. Acta.* 28:213-222, 1973.

30. Gordon, M.S. The isomers of silacyclopropane *Chem. Phys. Lett.* 76:163-168, 1980.

- D95, D95\* and D95\*\* (H to Cl)<sup>33</sup>.

## Add Extra Basis Function

HyperChem offers an easy way to interactively add certain basis functions to a molecular system. The Extra Basis Function dialog box can be used to add an S, P, D, SP, or SPD shell to the selected atom(s). These extra basis functions are primitives with no contractions. Thus, the extra basis functions are uniquely defined by the shell type and the value of the exponent.

One and only one extra shell can be applied to any atom in the current version of HyperChem. Different extra shells can be applied to different atoms but any atom can receive only one extra shell. If you would like to add more than one extra shell to any atom, you will need to modify a basis set file and create a new basis set. To use this new basis set you will need to include it in the [basisset] section of the Registry.

A basis set for an atom thus consists of a standard basis set (3-21G, for example) and an optional extra shell.

## Applying a Basis Set

You can use multiple basis sets in a single molecular system. The Apply Basis Set in HyperChem applies the currently selected basis set to the selected atoms or to all the atoms in HyperChem if there is no current selection. For example, some heavy atoms might have a 6-31G basis set (s and p only) while other heavy atoms might use a 6-31G\* basis set (with d-orbitals). This is an unusual but flexible option for *ab initio* or DFT calculations.

## Charge and Spin Multiplicity

Specify the extra (net) charge and the spin multiplicity in the Ab Initio Options dialog box.

The extra charge defines whether the current molecular system is a neutral system, positively charged system (cation), or negatively charged system (anion).

31. Krishnan, R.; Kinkley, J.S.; Seeger, R.; Pople, J.A. Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions *J. Chem. Phys.* 72:650-654, 1980
32. McLean, A.D.; Chandler, G.S. Contracted Gaussian basis sets for molecular calculations. I. Second row atoms, Z=11-18 *J. Chem. Phys.* 72:5639-5648, 1980.
33. Dunning, T.H.; Hay, P.J. in *Modern Theoretical Chemistry* Plenum, New York, 1976.

Closed-shell molecules have a multiplicity of one (a singlet). A radical, with one unpaired electron, has a spin multiplicity of two (a doublet). A molecular system with two unpaired electrons (usually a triplet) has a multiplicity of three. In some cases, however, such as a biradical, two unpaired electrons may also be a singlet.

## Convergence Criteria

Convergence limit and Iteration limit specify the precision of the SCF calculation. Convergence limit refers to the difference in total electronic energy (in kcal/mol) between two successive SCF iterations yielding a converged result. Iteration limit specifies the maximum number of iterations allowed to reach that goal.

You can always use the default values. If the calculation exceeds the iteration limit before it reaches the convergence limit, then most likely there is a convergence failure. Simply increasing this limit is unlikely to help. The DIIS convergence accelerator may help in some cases.

## UHF versus RHF

Choose UHF (spin Unrestricted Hartree-Fock) or RHF (spin Restricted Hartree-Fock) calculations according to your molecular system. HyperChem supports UHF for both open-shell and closed-shell calculations and RHF for closed-shell calculations only. The closed-shell UHF calculation may be useful for studying dissociation of molecular systems. ROHF (spin Restricted Open-shell Hartree-Fock) is not supported in the current version of HyperChem (for *ab initio* or DFT calculations).

One would normally choose RHF for closed-shell singlets and UHF for open-shell doublets and triplets.

## Convergence Acceleration

Choose the DIIS SCF convergence accelerator to potentially speed up SCF convergence. DIIS often reduces the number of iterations required to reach a convergence limit. However, it takes memory to store the Fock matrices from the previous iterations and this option may increase the computational time for individual iterations because the Fock matrix has to be calculated as a linear combination of the current Fock matrix and Fock matrices from previous iterations.

## Calculate Gradient

Specifies the calculation of gradients (the first derivatives of the total energy with respect to the nuclear positions). The RMS gradient gives an indication of the deviation from an optimized structure. The computations of two-electron integrals and their derivatives are time-consuming, because of the huge number of the two-electron integrals even for a medium size of molecule. You may not be interested in gradients for single point calculations, so you can turn off (not check) gradient calculation to speed up your task. This option applies to single point calculations only. HyperGauss always computes the gradients in doing geometry optimization, molecular dynamics, and vibration calculations.

## Calculate MP2 Correlation Energy

Specifies the calculation of electron correlation energy using the Møller-Plesset second order perturbation theory (MP2). This option can only be applied to Single Point calculations.

MP2 correlation energy calculations may increase the computational time because a two-electron integral transformation from atomic orbitals (AO's) to molecular orbitals (MO's) is required. HyperChem may also need additional main memory and/or extra disk space to store the two-electron integrals of the MO's.

## Two-Electron Repulsion Integrals

The calculation of the two-electron repulsion integrals in *ab initio* or DFT methods is inevitable and time-consuming. The computational time is mainly dominated by the performance of the two-electron integral calculation. The following items can control the performance of the two-electron integrals.

### Two-electron Integral Cutoff

Set this threshold to a small positive constant (the default value is  $10^{-10}$  Hartree). This threshold is used by HyperChem to ignore all two-electron repulsion integrals with an absolute value less than this value. This option controls the performance of the SCF iterations and the accuracy of the wave function and energies since it can decrease the number of calculated two-electron integrals.



## Two-electron Integral Buffer Size

HyperChem computes two-electron integrals and saves them in main memory (two-electron integral buffer). The units of this buffer are in double-precision words (8 bytes per double-precision word in Windows). Once this buffer is full, these two-electron integrals are written to a temporary file on a disk (the selected disk can be set in the Registry, before you start HyperChem, or with a script command, after you start HyperChem). A large buffer may reduce the processing time through fewer disk accesses. If the two-electron integral buffer size is big enough to hold all the integrals, HyperChem does not use the disk.

## Regular Two-Electron Integral Format

Specifies the use of a “regular” format for saving the two-electron integrals. HyperChem uses 16 bytes to store every integral. The first 8 bytes stores the four indices of an integral and the second 8 bytes stores its value. HyperChem only stores an integral and its associated indices when the integral's absolute value is greater than or equal to the two-electron integral cutoff. The two-electron integral and its indices are stored without any modification when you choose this regular two-electron integral format. These two-electron integrals and their indices can be printed out to a log file by choosing a proper setting for QuantumPrintLevel in the Registry.

## Raffenetti Two-Electron Integral Format

Specifies the use of the Raffenetti two-electron integral format<sup>34</sup>. HyperChem can calculate all two-electron integrals and store them in a special form that makes it easier to generate a Fock matrix during the SCF iterations. The Raffenetti two-electron integral format may take more main memory or disk space for the two-electron integrals, particularly in a UHF calculation, but may still be faster than the regular two-electron integral format for the same calculation.

The Raffenetti format is not available when performing MP2 and CI calculations or DFT calculations.

34. Raffenetti, R.C. Pre-processing two-electron integrals for efficient utilization in many-electron self-consistent field calculations. *Chem. Phys. Letters* 20:335-338, 1973.

## Direct SCF Calculation

When the Direct SCF calculation<sup>35</sup> option is chosen, the two-electron integrals are recomputed in every iteration. A Direct SCF calculation will be considerably slower than a regular SCF calculation which computes all the two-electron integrals and then saves them, retrieving the saved integrals every iteration to form the Fock matrices. The Direct SCF calculation avoids using disk space or a large main memory. This option may be practical for large molecular systems run on a desk-top PC where there is little available space on the disk.

## Initial Guess of MO Coefficients

An initial guess at the molecular orbital coefficients is necessary for an SCF calculation. Usually, the initial guess is obtained by solving the Hartree-Fock-Roothaan equations with the replacement of the Fock matrix by the core Hamiltonian. This initial guess of the MO coefficients is usually acceptable. However, in some cases, the core Hamiltonian leads to incorrect occupied orbitals or the initial guess may be far from the final converged SCF MO coefficients. Hence, HyperChem supports a few alternative methods for generating an initial guess at the MO coefficients. These alternative methods are:

- Projected Hückel: the initial guess at the MO coefficients is obtained from an extended Hückel calculation;
- Projected CNDO: the initial guess at the MO coefficients is obtained from a CNDO calculation;
- Projected INDO: the initial guess at the MO coefficients is obtained from an INDO calculation.

However, these alternative methods can be only applied to certain elements. For example, the projected CNDO/INDO may be used only for molecular systems with atomic numbers less than or equal to 18 (Ar). Elements beyond 18 are not available in the projected CNDO/INDO initial guess.

## Number of d Orbitals

There are two different sets of d-type functions (d orbitals) used in *ab initio* and DFT calculations. One 3d set consists of five 3d func-

35. Almlof, J.; Faegri Jr., K.; Korsell, K. *J. Comp. Chem.* 3:385, 1982.

tions —  $3d_0$ ,  $3d_{+1}$ ,  $3d_{-1}$ ,  $3d_{+2}$ , and  $3d_{-2}$  and is normally used in STO-NG basis sets. The other is a set of six 3d functions —  $3d_{xx}$ ,  $3d_{yy}$ ,  $3d_{zz}$ ,  $3d_{xy}$ ,  $3d_{xz}$ , and  $3d_{yz}$ , and is used in the split-valence basis sets, such as, 3-21G, 4-31G, 6-31G, etc.

The contraction exponents and coefficients of the d-type functions were optimized using five d-primitives (the first set of d-type functions) for the STO-NG basis sets and six d-primitives (the second set of d-type functions) for the split-valence basis sets. Thus, five d orbitals are recommended for the STO-NG basis sets and six d orbitals for the split-valence basis sets.

## Ghost-Atoms

With HyperChem you can always name a selection of atoms using the Name Selection item of the Select menu. A named selection called “**ghost-atoms**” has a special meaning, however. In most situations any atom that is a “ghost” atom, i.e. is a member of the ghost-atom set is treated just like any other atom and the named selection “ghost-atoms” is treated like any other named selection. [An alternative to naming “ghost-atoms” is to just make the selection, GHOST ATOMS]. However HyperChem may treat the ghost atoms differently from the regular atoms when performing certain *ab initio* or DFT calculations. If you have specified the use of “ghost-atoms” in the Ab Initio and DFT Advanced Options dialog box, then a ghost atom is represented in *ab initio* or DFT calculations by its basis set only and does not have any nucleus or electrons associated with it. Since there are atomic orbitals (basis functions) belonging to a ghost atom, there will be Mulliken charges attached to the ghost atoms after performing an *ab initio* or DFT calculation.

The concept of “ghost-atoms” only applies to Single Point calculations in the current version of HyperChem.

A major use of ghost atoms is to be able to offset the effect of basis set superposition effects (BSSE). For example, as two monomers come together the basis set of one monomer extends the basis set of the other, lowering the dimer energy more than just from the physical monomer-monomer interaction. A way to offset this in computing the dimerization energy is to use the same basis set in both monomer and dimer calculations. That is, the monomer energy is the energy of the dimer with ghost-atoms for the atoms of one of the monomers and “real” atoms for the other monomer.

## Configuration Interaction

Use Configuration Interaction to predict the electronic spectra of molecules. The Configuration Interaction wave function computes a ground state plus low lying excited states. You can obtain electronic absorption frequencies from the differences between the energies of the ground state and the excited states.

A configuration interaction calculation is available only for single points when the reference ground state is obtained from an RHF calculation.

The calculation mixes all single determinant wavefunctions that can be obtained from the ground state by exciting electrons from a subset of the occupied orbitals (of the ground state) to a subset of the unoccupied orbitals. The subsets are specified as a fixed number (highest occupied or lowest unoccupied) or by an energy criterion associated with the energy difference between the occupied orbital and the unoccupied orbital.

If you use the Conjugate Direction optimizer, you can optimize the geometries of excited states from configuration interaction calculations.

## Selecting Options for the Extended Hückel Method

### Charge and Spin Multiplicity

Specify the extra (net) charge and spin multiplicity in the Semi-empirical Options dialog box.

The extra charge defines whether the current molecular system is a neutral system, positively charged system (cation), or negatively charged system (anion).

Closed-shell molecules have a multiplicity of one (a singlet). A radical, with one unpaired electron, has a multiplicity of two (a doublet). A molecular system with two unpaired electrons (usually a triplet) has a multiplicity of three. In some cases, however, such as a biradical, two unpaired electrons may also be a singlet.

### Hückel Constant

The Hückel constant ( $k$ ) scales the interaction energy between two atomic orbitals (see “Extended Hückel Method” on page 130). HyperChem uses the default value of 1.75 (see the second part of

this book, *Theory and Methods*). You should use this value for most cases. A suggested range for experimental adjustment of this constant is 1.6–2.0.<sup>36</sup>

You can select Unweighted constant or Weight diffuseness for the Hückel constant. The unweighted option uses only the scale factor  $k$ . The Weight diffuseness option weighs the contributions from the two binding energies differently, but absorbs this weighting into the Hückel constant. The option you choose makes little difference in cases of orbitals with similar diffuseness, as for most organic molecules. It may be useful in calculations involving third row elements.

## d Orbitals

The Extended Hückel method also allows the inclusion of d orbitals for third row elements (specifically, Si, P, S and Cl) in the basis set. Since there are more atomic orbitals, choosing this option results in a longer calculation. The major reason to include d orbitals is to improve the description of the molecular system.

In some situations, using this option may be important. For example, if p orbitals on electronegative atoms interact with d orbitals, (as for a silicon atom bonded to an amine group), you may want to include d orbitals.

## Selecting Options for NDO Methods

After you select a method for a semi-empirical calculation (using the Semi-empirical item on the Setup menu), choose Options in the dialog box to set conditions for the calculation. You see the Semi-empirical Options dialog box. The following sections explain these options.

### Convergence Criteria

Convergence limit and Iteration limit specify the precision of the calculation. Convergence limit refers to the difference in energy (in kcal/mol) between two successive SCF cycles. Iteration limit specifies the maximum number of cycles allowed to reach that goal.

You can usually use the default values. If the calculation exceeds the iteration limit before it reaches the convergence limit, then

36. Hoffmann, R. An Extended Hückel Theory. I. Hydrocarbons *J. Chem. Phys.* 39(6):1397-1412, 1963.

there is most likely convergence failure. Simply increasing the limits is unlikely to help. The DIIS convergence accelerator, (see “SCF Convergence” on page 49), which is available for all the SCF semiempirical methods, might help in some cases. Most often, however, the reason for convergence failure lies in the molecular geometry. For example, two nuclei very close together often lead to convergence failure. If convergence problems arise, check the starting geometry for unrealistic features.

## Charge, Spin, and Excited State

Specify the charge and spin multiplicity in the Semi-empirical Options dialog box.

Closed-shell molecules have a multiplicity of one (a singlet). A radical, with one unpaired electron, has a multiplicity of two (a doublet). A molecular system with two unpaired electrons (usually a triplet) has a multiplicity of three. In some cases, however, such as a biradical, two unpaired electrons may also be a singlet.

This dialog box also contains the option for specifying that the molecular system is in the first excited singlet state (Next lowest) or the Lowest electronic state.

## UHF versus RHF

Next, you choose a UHF or RHF calculation. HyperChem can compute open-shell (non-singlet) systems with either the half-electron RHF or the UHF method (see “Hartree-Fock Wave Functions” on page 38).

## Convergence Acceleration

The DIIS convergence accelerator is available for all the SCF semiempirical methods. This accelerator may be helpful in curing convergence problems. It often reduces the number of iteration cycles required to reach convergence. However, it may be slower because it requires time to form a linear combination of the Fock matrices during the SCF calculation. The performance of the DIIS accelerator depends, in part, on the power of your computer.

## Configuration Interaction

Configuration Interaction (or electron correlation) improves energy calculations using CNDO, INDO, MINDO/3, MNDO,

MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO for these electron configurations

- closed-shell singlet ground states
- half-electron, excited singlet states
- half-electron, doublet, triplet, and quartet open-shell ground states.

See also “Extending the Wave Function Calculation” on page 39.

Use Configuration Interaction to predict electronic spectra of molecules. The Configuration Interaction wave function provides a ground state plus some excited state energies. You can obtain electronic absorption frequencies from the differences between the energies of the ground state and the excited states.

You can also use a RHF wave function with Configuration Interaction for calculations involving bond breaking, instead of using a UHF wave function. Using RHF plus Configuration Interaction conserves spin symmetry.

*Note:* Configuration Interaction significantly increases computing time. For calculations of ground-state energies, MINDO/3, MNDO, AM1, and PM3 methods may already include in their parameters some effects of Configuration Interaction.

## Log File for Results

### Log File for Results

After you choose the computation method and options, you can use Start Log on the File menu to record results, such as total energies, orbital energies, dipole moments, atomic charges, enthalpies of formation (for the CNDO, INDO, MINDO/3, MNDO, MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO methods), etc.

Use the QuantumPrintLevel setting in the Start Log dialog box to set the amount of information recorded in a log file.

For more information about log files, see Appendix A in the *HyperChem Reference manual*.

## Types of Calculations

Single-point, geometry optimization, molecular dynamics and vibration calculations are all available with either *ab initio*, DFT, or semi-empirical SCF methods. After obtaining a wavefunction via any of these calculations, you can plot two-dimensional contour diagrams of the electrostatic potential surrounding a molecule, the total electronic density, the spin density, one or more molecular orbitals, and the electron densities of individual orbitals. After a vibration calculation, you can display a predicted infrared spectrum and display normal modes as vector attached to atoms or by animation of each mode. If you have performed a single-point calculation with a singly-excited configuration interaction, you can display a simulated ultraviolet-visible spectrum.

The setup of these calculations is very similar for both quantum and molecular mechanics. In practice, molecular dynamics calculations using the *ab initio*, DFT, and semi-empirical quantum mechanical SCF methods are limited to relatively small systems. Each time step requires a complete calculation of the wave function and the forces.

*Note:* You can superimpose harmonic restraining forces to interatomic distances, angles, or dihedrals that you have set up as named selections. You can also restrain atoms to points in space. See “Using Geometric Restraints” on page 83 and “Restraints” on page 109.

## Single Point Calculations and CI

HyperChem always computes the electronic properties for the molecule as the last step of a geometry optimization or molecular dynamics calculation. However, if you would like to perform a configuration interaction calculation at the optimized geometry, an additional single point calculation is required with the CI option being turned on.

For quantum mechanical methods, many more options are available for a single point calculation than for molecular mechanics. This is because the quantum mechanical methods yield various electronic properties. Among the options available following a Single Point calculation is the plotting of contour diagrams. These plots include the electrostatic potential surrounding a molecule, the total electronic density, the spin density, and the contour dia-



grams describing individual molecular orbitals. HyperChem can produce different plots in succession without repeating the single point calculation, as long as the coordinates of the molecule are unaltered. To plot these pictures after a calculation, choose either Plot Molecular Graphs or Orbitals from the Compute menu of HyperChem.

## Optimization Methods

Four optimization techniques are available for geometry optimizations with quantum mechanical methods: a steepest descent method, two conjugate gradient methods, and eigenvector following. A generally less desirable conjugate directions optimizer is also available for certain situations where gradients are not available. To select these methods, choose Geometry Optimization on the Compute menu. The Semi-empirical Optimization or *Ab Initio* or Density Functional Optimization dialog box appears. You have the choice of the same methods as for molecular mechanics calculations (see “Geometry Optimizations” on page 59).

In general, the steepest descent method is useful when a molecule is far from a minimum. Conjugate gradient methods are better when the minimum is nearby. You choose the precision of the optimization in the Semi-empirical Optimization or *Ab Initio* or Density Functional Optimization dialog box. Suitable default values for ending an optimization calculation are either an RMS gradient of 0.1 kcal/mol/Å or a maximum number of cycles that is 15 times the number of atoms involved in the calculation. In general, you must use a gradient limit. For improved precision, use a lower gradient limit.

The conjugate directions optimizer will take longer than the other optimizers to find a minimum since it does not have information on gradients. It may require a relatively large number of cycles to find a minimum. Each time it is started over again (somewhat) since without gradients it doesn't know *where it is starting from or how far away the minimum is*.

**Note:** You cannot use the Extended Hückel method or any one of the SCF methods with the CI option being turned on for geometry optimizations, molecular dynamics simulations or vibrational calculations, in the current version of HyperChem [Geometry optimization by Conjugate Directions is available with the CI option].

## Transition State Search

You can use the *ab initio* or Density Functional method (with any basis set), and all semi-empirical methods except Extended Hückel, to perform a transition state search. Two different methods are available: eigenvector following and synchronous transit. In the synchronous transit method, both linear synchronous transit (LST) and quadratic synchronous transit (QST) are available. To select any of these methods, choose Transition State on the Compute menu. If you choose eigenvector following, the calculation will start with the calculation of the Hessian matrix. Then the dialog box will appear, in which you can choose the mode to be followed, and specify the calculation termination conditions (RMS gradient and number of SCF cycles). If the synchronous transit method is chosen, the dialog box appears, in which you can specify a linear or quadratic search, and can specify convergence criteria.

In general, *ab initio* or DFT calculations are preferable for transition state searching. The time required for the calculation to finish may sometimes be the limiting factor in the choice of the calculation method. Semi-empirical methods may be less successful in transition state searching, since they were originally parametrized for energy minimization searching. (The ZINDO/S method is especially likely to give poor results for transition state searching since it was parameterized for spectral calculations.) Many times, a tighter SCF convergence criteria may be needed, as well as a larger number of SCF cycles, in order to locate the transition state successfully with a semi-empirical method.

*Note:* You can not use the Extended Hückel method, nor any of the other SCF methods with the CI option turned on, for geometry optimization or molecular dynamics simulations.

## Molecular Dynamics

You can use any *ab initio* or DFT SCF calculation and all the semi-empirical methods, except Extended Hückel, for molecular dynamics simulations. The procedures and considerations are similar for simulations using molecular mechanics methods (see “Molecular Dynamics” on page 71).

These simulations take much longer than molecular mechanics simulations. They may, however, model bond breaking and forma-

tion more accurately. While HyperChem doesn't rearrange its molecular graph dynamically, you can detect the changes by monitoring interatomic distances (see “Bond Breaking” on page 93).

## Energy Conservation in Molecular Dynamics Calculations

In order to conserve the total energy in molecular dynamics calculations using semi-empirical methods, the gradient needs to be very accurate. Although the gradient is calculated analytically, it is a function of wavefunction, so its accuracy depends on that of the wavefunction. Tests for CH<sub>4</sub> show that the convergence limit needs to be at most 1e-6 for CNDO, INDO, and TNDO and 1e-7 for MINDO/3, MNDO, MNDO/d, AM1, and PM3 for accurate energy conservation. ZINDO/S is not suitable for molecular dynamics calculations.

## Vibrations

A vibrations calculation is the first step of a vibrational analysis. It involves the time consuming step of evaluating the Hessian matrix (the second derivatives of the energy with respect to atomic Cartesian coordinates) and diagonalizing it to determine normal modes and harmonic frequencies. For the SCF methods the Hessian matrix is evaluated by finite difference of analytic gradients, so the time required quickly grows with system size.

Once the vibration calculation completes, you can analyze and display the results by using Vibrational Spectrum menu item.

## Contour Plots and Orbitals

Once you have calculated an *ab initio*, DFT, or a semi-empirical wave function via a single point calculation, geometry optimization, molecular dynamics or vibrations, you can plot the electrostatic potential surrounding the molecule, the total electronic density, the spin density, one or more molecular orbitals  $\psi_i$ , and the electron densities of individual orbitals  $\psi_i^2$ . You can examine orbital energies and select orbitals for plotting from an orbital energy level diagram.

## Vibrational Spectrum

To perform a vibrational analysis, choose Vibrations on the Compute menu to invoke a vibrational analysis calculation, and then choose Vibrational Spectrum to visualize the results. The Vibrational Spectrum dialog box displays all vibrational frequencies and a simulated infrared spectrum. You can zoom and pan in the spectrum and pick normal modes for display, using vectors (using the Rendering dialog box from Display/Rendering menu item) and/or animation.

Generally, a vibration analysis calculation should be performed after a geometry optimization with the same method. This ensures that the calculation of second derivatives is performed at a configuration for which all first derivatives are zero.

## Electronic Spectrum

Use the Electronic Spectrum dialog box to display and analyze the UV-visible spectrum produced by a singly excited CI calculation. This dialog box is available from the Compute menu only after you do a single point CI semi-empirical, DFT, or *ab initio* calculation. Electronic Spectrum is then activated on the Compute menu. The DFT values are to be considered suspect because the orbitals are Kohn-Sham orbitals not Hartree-Fock orbitals and their use in a CI requires re-examination. For example, the matrix elements between the ground state and singly-excited states from DFT are no longer zero as the CI code assumes.

## Saving Information in a Log File

You can obtain detailed results of calculations by using Start Log on the File menu. The amount of calculational results is controlled by the QuantumPrintLevel. For more details of the log file and information saved in the log file, see the HyperChem *Reference* manual.

## Extended Hückel Method

Extended Hückel is the simplest and fastest semi-empirical method included in HyperChem, but it is also the least accurate. It is particularly simple in its treatment of electron-electron interactions: it has no explicit treatment of these interactions, although it may include some of their effects by parameterization.

Extended Hückel provides the approximate shape and energy ordering of molecular orbitals. It also yields the approximate form of an electron density map. This is the only requirement for many qualitative applications of quantum mechanics calculations, such as Frontier Orbital estimates of chemical reactivity (see “Frontier Molecular Orbitals” on page 146).

HyperChem determines the energy and form of the molecular orbitals using a single set of parameters that represents the binding energies of electrons in each atomic orbital. Binding energy refers to the attraction between an electron in an atomic orbital and the nucleus of the same atom. These energies are the diagonal elements of the Hamiltonian matrix. The off-diagonal elements are the interaction energies between two atomic orbitals. Interaction energies are the average of the binding energies multiplied by the overlap integral over the two atomic orbitals. HyperChem scales this result by the Hückel constant,  $k$ , which has a default value of 1.75 (see the second part of this book, *Theory and Methods*).

### Limitations of Extended Hückel

The neglect of electron-electron interactions in the Extended Hückel model has several consequences. For example, the atomic orbital binding energies are fixed and do not depend on charge density. With the more accurate NDO semi-empirical treatments, these energies are appropriately sensitive to the surrounding molecular environment.

HyperChem cannot perform a geometry optimization or molecular dynamics simulation using Extended Hückel. Stable molecules can collapse, with nuclei piled on top of one another, or they can dissociate into atoms. With the commonly used parameters, the water molecule is predicted to be linear.

## NDO Methods

The Extended Hückel method neglects all electron-electron interactions. More accurate calculations are possible with HyperChem by using methods that neglect some, but not all, of the electron-electron interactions. These methods are called Neglect of Differential Overlap or NDO methods. In some parts of the calculation they neglect the effects of any overlap density between atomic orbitals. This reduces the number of electron-electron interaction integrals to calculate, which would otherwise be too time-consuming for all but the smallest molecules.

### Defining Electron-Electron Interactions

NDO calculations use the Hartree-Fock (HF) approximation to solve the Schrödinger equation. HF methods deal with several kinds of electron-electron interactions. By understanding these interactions, you can appreciate differences between the NDO methods and gain insight into why the NDO approximation works well or fails.

Electrons repel each other electrostatically, and the repulsion between an electron in one atomic orbital and an electron in another (or the same, if doubly occupied) is called a coulomb interaction. The identical nature of electrons requires a correction for electrons of the same spin, and this is often described as if it were a real interaction. The contributions to matrix elements describing this correction are called atomic exchange integrals. This is because they arise from terms in the many-electron wave function involving the interchange of coordinates of an electron pair (see “Exclusion Principle” on page 35). In a typical calculation, there are many more exchange-type integrals than coulomb integrals. Think of an exchange integral as an electron in the overlap region between two atomic orbitals interacting with an electron in an overlap region between two other atomic orbitals.

### Treatment of Electron-Electron Interactions

The NDO methods in HyperChem are built on three different approximations to the treatment of electron-electron interactions. The NDO methods are discussed in the following sections.

## CNDO, INDO, MINDO/3, ZINDO/1, and ZINDO/S Methods

The simplest method is CNDO (Complete Neglect of Differential Overlap). The electron repulsion between electrons in different orbitals depends only on the nature of the atoms involved, and not on the particular orbital. This creates a very simple picture. One of its disadvantages is that because it neglects almost all exchange integrals, it cannot calculate differences between states of multiplicity arising from the same electronic configuration. CNDO treats singlet-triplet energy gaps poorly.

The INDO method (Intermediate NDO) corrects some of the worst problems with CNDO. For example, INDO exchange integrals between electrons on the same atom need not be equal, but can depend on the orbitals involved. Though this introduces more parameters, additional computation time is negligible. INDO and MINDO/3 (Modified INDO, version 3) methods are different implementations of the same approximation.

ZINDO/1 and ZINDO/S are Dr. Michael Zerner's INDO versions and used for molecular systems with transition metals. ZINDO/1 is expected to give geometries of molecules, and ZINDO/S is parametrized to give UV spectra.

## MNDO, MNDO/d, AM1, and PM3 Methods

The NDDO (Neglect of Diatomic Differential Overlap) approximation is the basis for the MNDO, MNDO/d, AM1, and PM3 methods. In addition to the integrals used in the INDO methods, they have an additional class of electron repulsion integrals. This class includes the overlap density between two orbitals centered on the same atom interacting with the overlap density between two orbitals also centered on a single (but possibly different) atom. This is a significant step toward calculating the effects of electron-electron interactions on different atoms.

Calculating the extra integrals takes time. MNDO, MNDO/d, AM1, and PM3 calculations typically take about one and one-half times as long as INDO or MINDO/3 calculations.

AM1 is generally the most accurate computational method included in HyperChem and is often the best method for collecting quantitative information. PM3 is functionally similar to AM1, but uses an alternative parameter set (see "PM3" on page 156).

## TNDO Methods

The TNDO (Typed Neglect of Differential Overlap) approximation is a new semi-empirical method with considerable research needed before its parameters are fully known for the variety of situations in which it might be used. It is essentially identical to CNDO (TNDO/1) or INDO (TNDO/2). The only difference is that the semi-empirical parameters are a function not of the atomic number but of the *atom type*. It uses Amber atom types and is in some sense a combination of molecular mechanics and semi-empirical quantum mechanics.

To be fully effective TNDO needs lots of research effort to develop parameters for the different atom types (chemical environments). For example, unlike INDO, which has one  $\beta$  bonding parameter for all C-C bonds, TNDO/2 has different  $\beta$  values for C-C single bonds, double bonds, triple bonds, aromatic bonds, etc. The atom parameters depend upon specific chemical environments as well as the property one is trying to predict. That is, in the general TNDO scheme one does not have one generic set of parameters for all situations but different parameter sets for different properties. Thus, in HyperNMR, which uses TNDO, there is one parameter set for predicting chemical shifts and a different parameter set for predicting spin-spin coupling.

For only the TNDO semi-empirical method a scheme, associated with the menu item Select Parameter Set, is available for quickly choosing and copying parameter sets.

## Practical Uses of NDO Methods

NDO methods work best for molecules with electrons that are spread evenly throughout, with no significant charge polarization. Hydrocarbons are classic examples: all NDO methods work well on nonpolar hydrocarbons. Molecular systems with heteroatoms provide a better test of these methods. Groups with several electronegative atoms close together, such as  $\text{NO}_2$ , are the most difficult to treat. Among inorganic, main group compounds, even the best semi-empirical methods can fail dramatically with, for example, interhalogen molecules.



## Parameterization

The various semi-empirical methods in HyperChem differ in many technical details. Treatment of electron-electron interactions is one major distinguishing feature. Another important distinguishing feature is the approach used to parameterize the methods. Based on the methods used for obtaining parameters, the NDO methods fall into two classes:

- CNDO and INDO were developed by the Pople group at Carnegie Mellon University. This group chose parameters based primarily on theory. Their intent was to give them values as close as possible to those possibly obtained by an exact Hartree-Fock calculation, using the same basis set.
- MINDO/3, MNDO, and AM1 were developed by the Dewar group at the University of Texas at Austin. This group chose many parameters, such as heats of formation and geometries of sample molecules, to reproduce experimental quantities. The Dewar methods yield results that are closer to experiment than the CNDO and INDO methods.
- MNDO/d was developed by Walter Thiel a former member of the Dewar group principally to deal with main group elements where d orbitals are important.
- PM3, developed by James J.P. Stewart<sup>37</sup>, is a reparameterization of AM1, which is based on the neglect of diatomic differential overlap (NDDO) approximation. NDDO retains all one-center differential overlap terms when Coulomb and exchange integrals are computed. PM3 differs from AM1 only in the values of the parameters. The parameters for PM3 were derived by comparing a much larger number and wider variety of experimental versus computed molecular properties. Typically, non-bonded interactions are less repulsive in PM3 than in AM1. PM3 is primarily used for organic molecules, but is also parameterized for many main group elements.
- ZINDO/1 is based on a modified version of the intermediate neglect of differential overlap (INDO), which was developed by Michael Zerner of the Quantum Theory Project at the University of Florida. Zerner's original INDO/1 used the Slater

37. Stewart, J. J. P. Optimization of Parameters for Semiempirical Methods. I. Method *J. Comput. Chem.* 10:209, 1989 and Stewart, J. J. P. Optimization of Parameters for Semiempirical Methods. II. Applications *J. Comput. Chem.* 10:221, 1989.

orbital exponents with a distance dependence for the first row transition metals only. However, in HyperChem constant orbital exponents are used for all the available elements, as recommended by Anderson, Edwards, and Zerner, *Inorg. Chem.* 28, 2728-2732, 1986.

- ZINDO/S is an INDO method parameterized to reproduce UV visible spectroscopic transitions when used with the singly excited CI method. It was developed in the research group of Michael Zerner of the Quantum Theory Project at the University of Florida.
- TNDO is based on CNDO and INDO as modified by Hypercube, Inc. starting with Release 1.0 of HyperNMR.

## Results of Semi-Empirical Calculations

You can use the information obtained from semi-empirical calculations to investigate many thermodynamic and kinetic aspects of chemical processes. Energies and geometries of molecules have clear relationships to chemical phenomena. Other quantities, like atomic charges and Frontier Orbitals, are less defined but provide useful qualitative results.

### Energies of Molecules

The total energy in an Molecular Orbital calculation is the net result of electronic kinetic energies and the interactions between all electrons and atomic cores in the system. This is the potential energy for nuclear motion in the Born-Oppenheimer approximation (see page 33).

The stable geometry of a molecule has a minimal total energy. Geometries at different energy minima (local minima plus the global minimum) describe different stable or metastable conformations and isomers of a molecule. Geometry optimizations with NDO methods generally yield geometries corresponding to energy minima.

While the HyperChem status line reports the binding energy of the electrons, all the SCF methods also supply heat of formation. MINDO/3, MNDO, AM1, and PM3 are parameterized by fitting to experimentally determined heats (enthalpies) of formation for a set of molecules at 298 K. The heat of formation is calculated for

these methods by subtracting atomic heats of formation from the binding energy. Results are reported in the log file for a calculation (see page 124). Heat of formation is the value usually reported when describing results and is more useful than the directly calculated binding energy.

The following data (Table 1) for molecules, including hydrocarbons, strained ring systems, molecules with heteroatoms, radicals, and ions comes from a review by Stewart.<sup>38</sup> For most organic molecules, AM1 reports heats of formation accurate to within a few kilocalories per mol. For some molecules (particularly inorganic compounds with several halogens, such as perchloryl fluoride, even the best semi-empirical method fails completely.

Table 8-1. Heats of formation of selected molecules from AM1 calculations

Molecular Formula	Name	Heat of Formation (experiment, kcal/mol)	Heat of Formation (AM1, kcal/mol)
C <sub>7</sub> H <sub>6</sub> O <sub>2</sub>	Benzoic Acid	-70.1	-68.0
C <sub>7</sub> H <sub>8</sub>	Norbornadiene	+59.7	+67.7
C <sub>5</sub> H <sub>8</sub> O	Cyclopentanone	-46.0	-36.1
C <sub>2</sub> H <sub>4</sub> O	Ethylene oxide	-12.6	-9.0
C <sub>5</sub> H <sub>8</sub>	Spiropentane	+44.3	+50.5
C <sub>4</sub> H <sub>10</sub>	n-butane	-30.4	-31.2
CF <sub>4</sub>	Carbon tetrafluoride	-223.3	-225.7
C <sub>3</sub> H <sub>5</sub>	Allyl radical	+40.0	+38.6
CH <sub>2</sub> (triplet)	Methylene	+92.3	+80.8
CH <sub>3</sub> O	Methoxy radical	-0.5	-3.7
C <sub>7</sub> H <sub>7</sub> <sup>+</sup>	Benzyl cation	+216.0	+222.1
CH <sub>2</sub> F <sub>2</sub> <sup>+</sup>	Difluoromethane cation	+185.2	+151.6

38. Stewart, J. J. P. MOPAC: A Semiempirical Molecular Orbital Program *J. Computer-Aided Mol. Design* 4, 1-105:1990.

Molecular Formula	Name	Heat of Formation (experiment, kcal/mol)	Heat of Formation (AM1, kcal/mol)
O <sub>3</sub> FCl	Perchloryl fluoride	-5.1	+246.5

You can investigate the energetics of chemical equilibrium by comparing the heats of formation of reactants and products. This produces one of the most useful results of a chemical calculation. The accuracy and reliability of the heats of formation depend on the method used (see “Choosing a Semi-Empirical Method” on page 153).

A common application of the direct calculation of molecular energy is the study of organic reaction mechanisms. You can investigate the energies of different potential intermediates, species not easily studied by experiment. A review by Thiel<sup>39</sup> lists many such applications for MNDO, including studies of rearrangements, questions of stereoselectivity, regioselectivity, and photochemical reactions. The accuracy of calculations in predicting relative energies of related species is generally greater than the absolute heats of formation (see Tables 1 and 2). This shows how much interest there is in the study of reaction mechanisms.

Data on proton affinities (gas phase) of many different compounds (see Table 2) demonstrate the high level of accuracy possible in determining energies of related species. In this report by Dewar and Dieter<sup>40</sup>, the enthalpy of formation of H<sup>+</sup> is the experimental value (367.2 kcal/mol). The calculated value for H<sup>+</sup> is unreliable.

Table 8-2. Proton affinities of selected compounds, from AM1 calculations

Base	Conjugate Acid	Proton Affinity (experimental, kcal/mol)	Proton Affinity (AM1, kcal/mol)
CH <sub>4</sub>	CH <sub>5</sub> <sup>+</sup>	132.0	134.0
C <sub>6</sub> H <sub>6</sub> (benzene)	C <sub>6</sub> H <sub>7</sub> <sup>+</sup>	181.3	183.2

39. Thiel, W. Semiempirical Methods: Current Status and Perspectives *Tetrahedron*, 44:7393, 1988.

40. Dewar, J. S.; Dieter, K. M. *J. Am. Chem. Soc.* 108: 8075, 1986.

Base	Conjugate Acid	Proton Affinity (experimental, kcal/mol)	Proton Affinity (AM1, kcal/mol)
CH <sub>3</sub> NH <sub>2</sub>	CH <sub>3</sub> NH <sub>3</sub> <sup>+</sup>	214.1	211.1
Ph-NH <sub>2</sub>	Ph-NH <sub>3</sub> <sup>+</sup>	209.5	211.4
CH <sub>3</sub> CN	CH <sub>3</sub> CNH <sup>+</sup>	188.4	190.4
H <sub>2</sub> O	H <sub>3</sub> O <sup>+</sup>	166.5	164.5
CH <sub>3</sub> CHO	CH <sub>3</sub> CHOH <sup>+</sup>	186.6	184.9

## Geometries of Molecules

The geometries obtained from optimizations with semi-empirical calculations describe the shapes of molecules. The calculations have varying degrees of accuracy and take more time than molecular mechanics methods. The accuracy of the results depends on the molecule.

Molecular mechanics force fields have much information built into them and can be accurate for the molecules used in their parameterization. For molecules outside the limited scope for which they were designed, semi-empirical methods are more reliable. Stewart<sup>38</sup> reports a statistical study of geometries obtained using MNDO and AM1. For second row elements, the calculations of bond lengths are accurate to an average of approximately 0.01 Ångstrom for AM1, and almost twice this error for MNDO. For third row elements, the error is about twice that of the second row elements. Bond angles are predicted correctly to within a few degrees for almost all organic molecules.

## Energies of Transition States

Many transition states of chemical reactions contain symmetry elements not present in the reactants and products. For example, in the umbrella inversion of ammonia, a plane of symmetry exists only in the transition state.

Symmetrical transition states are the lowest energy configuration within that symmetry. If a geometry optimization starts off within that symmetry, then the calculation can find the transition state.

*Example:* If you start an optimization of a planar ammonia molecule and constrain it to that geometry, the calculation finds the transition state.

*Example:* Ethane is stable in the staggered (*gauche*) conformation. The transition state for rotating a methyl group in ethane has the eclipsed conformation. A geometry optimization starting from an eclipsed conformation yields the transition state.

Semi-empirical methods take their parameters from data on stable molecules, so the parameters are less suited to describing transition states than to describing minimum energy geometries. Activation energies derived from these methods are generally less reliable than are reaction energies. Calculations of rotational barriers using MNDO and AM1 provide examples of the accuracy of activation energies (Table 3). The following data is from Stewart's review.<sup>38</sup>

Table 8-3. Barriers to rotation and inversions, from AM1 calculations

Molecule	Barrier to	Experiment (kcal/mol)	AM1 (kcal/mol)	MNDO (kcal/mol)
Ethane	Methyl rotation	2.9	1.25	1.01
Ammonia	Inversion	6	4.24	11.58
Methanol	Methyl rotation	1.1	1.04	0.74

## Molecular Orbital Energies and Ionization Potentials

Koopmans' Theorem states that the first ionization energy of a closed-shell molecule is approximated to second order by the energy of the highest occupied orbital (HOMO). During ionization, the remaining electrons are reorganized, contributing an additional energy term not considered in this theorem. However, Koopmans' theorem does hold for many situations and enables interpretation of photoelectron spectra from Molecular Orbital calculations. It is also true that orbital energies can reasonably approximate higher ionization potentials.

Errors in ionization potentials are typically a few tenths of an electron volt, a small percentage of the total. Stewart gives tables with sample values.<sup>24, 41</sup>

## Dipole Moments

The molecular dipole moment is perhaps the simplest experimental measure of charge density in a molecule. The accuracy of the overall distribution of electrons in a molecule is hard to quantify, since it involves all of the multipole moments. Experimental measures of accuracy are necessary to evaluate results. The values for the magnitudes of dipole moments from AM1 calculations for a small sample of molecules (Table 4) indicate the accuracy you may expect. This data is a sampling from Stewart's review.<sup>38</sup>

Table 8-4. Dipole moments of selected molecules, from AM1 calculations

Formula	Name	Experiment (Debyes)	AM1 (Debyes)
C <sub>3</sub> H <sub>4</sub> O	Acrolein	3.12	2.53
NH <sub>3</sub>	Ammonia	1.47	1.85
H <sub>2</sub> O	Water	1.85	1.86
C <sub>6</sub> H <sub>6</sub> O	Phenol	1.45	1.23
HSiF <sub>3</sub>	Trifluorosilane	1.27	1.55
C <sub>6</sub> H <sub>4</sub> F <sub>2</sub>	o-difluorobenzene	2.59	2.68

The value for water in Table 4 is particularly interesting. AM1 reproduces the water molecule's electron distribution very well and can give accurate results for hydrogen bonds.

The HyperChem log file includes calculated dipole moments of molecules. To set the amount of information collected in the log file, change the value of the QuantumPrintLevel setting in the chem.ini file. Note that the sign convention used in the quantum mechanical calculation of dipoles is opposite to that used in molecular mechanics dipole calculations; this reflects the differing sign conventions of physics and chemistry.

41. Stewart, J.J.P. in *Reviews of Computational Chemistry*, Lipkowitz, K.; Boyd, D.B., Eds. VCM Publishers, New York, 1990.

## Electrostatic Potential

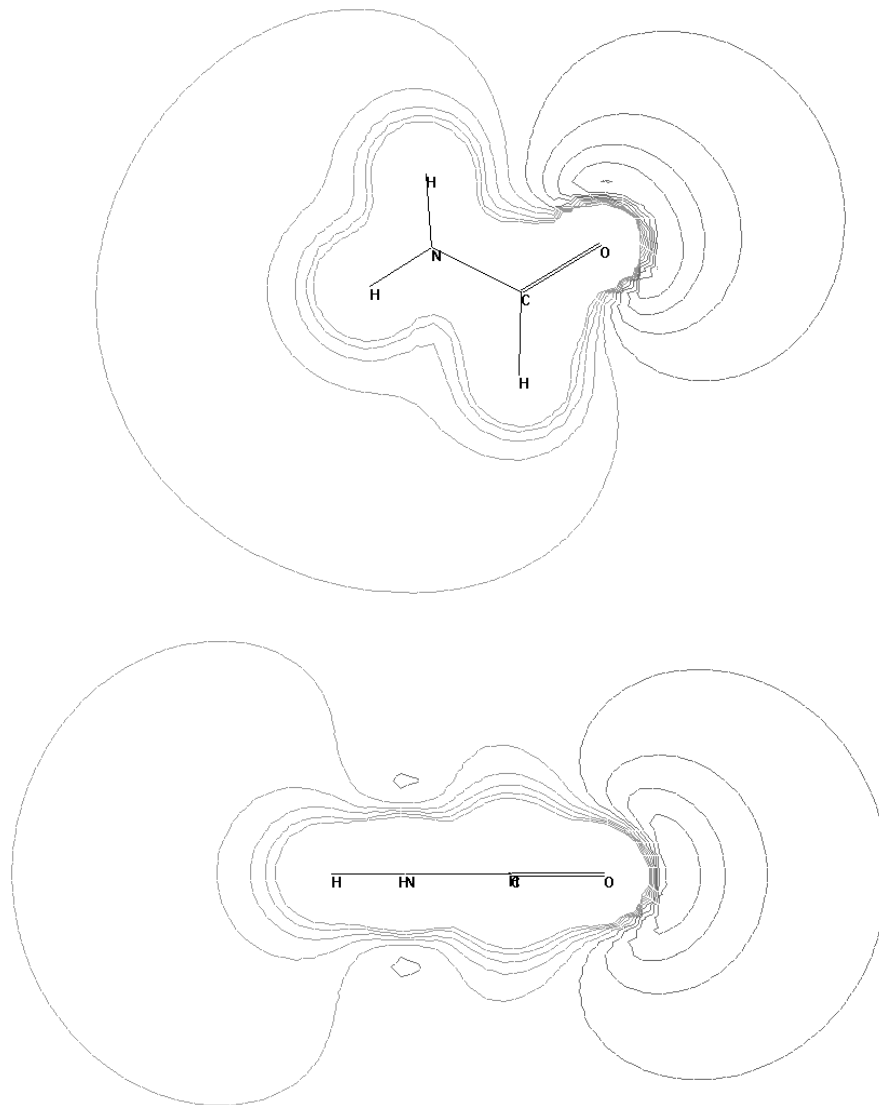
Electron distribution governs the electrostatic potential of molecules. The electrostatic potential describes the interaction of energy of the molecular system with a positive point charge. Electrostatic potential is useful for finding sites of reaction in a molecule: positively charged species tend to attack where the electrostatic potential is strongly negative (electrophilic attack).

HyperChem displays the electrostatic potential as a contour plot when you select the appropriate option in the Contour Plot dialog box. Choose the values for the starting contour and the contour increment so that you can observe the minimum (typically about  $-0.5$  for polar organic molecules) and so that the zero potential line appears.

View the contour map in several planes to see the general form of the distribution. As long as you don't alter the molecular coordinates, you don't need to repeat the wave function calculation. Use the left mouse button and the HyperChem Rotation or Translation tools (or Tool icons) to change the view of a molecule without changing its atomic coordinates.

Atomic charges indicate where large negative values (sites for electrophilic attack) are likely to occur. However, the largest negative value of the electrostatic potential is not necessarily adjacent to the atom with the largest negative charge. An example is formamide ( $\text{NH}_2\text{CHO}$ ). The largest negative atomic charge occurs on the nitrogen, but the most negative values of the electrostatic potential occur at the oxygen lone pair sites. Protonation most favorably occurs at these sites. This illustrates the value of electrostatic potential compared to simple atomic charges in predicting reactivity.





In this example, the deepest minimum occurs at oxygen lone pair sites. The contour starting value is  $-0.4$  and the increment is  $0.04$ . Fifteen contour levels were calculated on a  $60$  by  $60$  grid. The most negative (Mulliken) atomic charge in this system is that of the nitrogen ( $-0.448$ ), with oxygen ( $-0.371$ ) being the only other negative atom in the system. Electrostatic potential predicts protonation on the oxygen atom.

## Atomic Charges

Energy, geometry, dipole moment, and the electrostatic potential all have a clear relation to experimental values. Calculated atomic charges are a different matter. There are various ways to define atomic charges. HyperChem uses Mulliken atomic charges, which are commonly used in Molecular Orbital theory. These quantities have only an approximate relation to experiment; their values are sensitive to the basis set and to the method of calculation.

It is interesting to calculate the atomic charges of the water molecule using the various methods in HyperChem, and compare the results to the optimized point charges for the TIP3P model of water. The TIP3P model, which HyperChem uses in its periodic box, gives atomic charges that reproduce accurately the electrostatic interactions between water molecules. The charge on the oxygen atom is  $-0.834$ , while the charges on the oxygens from the semi-empirical calculations at optimized geometries are  $-0.383$  (AM1),  $-0.326$  (MNDO),  $-0.505$  (MINDO/3),  $-0.312$  (INDO), and  $-0.266$  (CNDO). At the CNDO geometry, the Extended Hückel charge on oxygen is  $-0.884$ . Extended Hückel, which gives the least accurate results of any method, yields a charge closest to the electrostatic potential values of TIP3P. Note that the TIP3P values are appropriate for a particular use; it would be incorrect to call them the proper atomic charges. However, it is clear that reliable energies and geometries do not translate into a meaningful set of atomic charges.

Despite these reservations, Mulliken population-derived atomic charges are easy to compute. Empirical investigation shows that they have various uses; they provide approximate representation of the 3D charge distribution within a molecule.

AMBER, BIO+, and OPLS calculations use information on atomic charges. Atomic charges can come from these sources:

- HyperChem residue templates. HyperChem uses these templates when it reads a Protein Data Bank file or when you construct a molecule from residues.
- Set Charge on the Build menu. You can use this command to assign charges to individual atoms.
- Semi-empirical calculations.
- *Ab initio* calculations

For example, semi-empirical calculations on a substrate molecule provide a set of charges that you can use in a molecular mechanics calculation of the interaction of that substrate with another molecule. To include the effects of polarization, repeated semi-empirical calculations can provide a set of charges that respond to the environment.

If the MM+ option to use bond dipoles for nonbonded electrostatic calculations is set, then MM+ ignores atomic charges but uses dipole moments supplied in its parameter set (in the mmpstr.\* files).

One of the common uses of Mulliken atomic charges is to indicate chemical reactivity.

## Chemical Reactivity

Because of thermodynamic forces, many elementary reactions (those that take place in a single step) favor the most stable product. In these cases, calculating energies of the reaction products yields straightforward, well-defined information about the likely products. Electrophilic, aromatic substitutions are an example of this type of reaction.

Other reactions are controlled kinetically, and the most stable product is not the major one observed. In these cases, you must look at the reactant side of the reaction coordinate to discover factors determining the outcome. Klopman<sup>42</sup> and Salem<sup>43</sup> developed an analysis of reactivity in terms of two factors: an electrostatic interaction approximated by atomic charges and a Frontier orbital interaction. Fleming's book provides an excellent introduction to these ideas.<sup>44</sup>

The Klopman-Salem equation partitions contributions to the interaction energy of two molecules into two teams, as they approach each other (equation 29).

42. Klopman, G. *J. Am. Chem. Soc.* 90:223, 1968.

43. Salem, L. *J. Am. Chem. Soc.* 90:543,90:553, 1968.

44. Fleming, I. *Frontier Orbitals and Organic Chemical Reactions* Wiley and Sons, London, 1976.

$$\Delta E = \sum_{A < B} \frac{q_A q_B}{R_{AB}} + \left( \sum_r^{\text{occ}} \sum_s^{\text{unocc}} - \sum_s^{\text{occ}} \sum_r^{\text{unocc}} \frac{2 \left( \sum_{\mu\nu} C_{\mu r} \cdot C_{\nu s} \cdot H_{\mu\nu} \right)^2}{\epsilon_r - \epsilon_s} \right) \quad (29)$$

$\Delta E$  is the interaction energy;  $q_A$  and  $q_B$  are charges on atoms A and B, separated by  $R_{AB}$ , on different molecules.  $r$  and  $s$  are molecular orbitals on the two different molecules.  $\mu$  and  $\nu$  label the atomic orbitals that contribute to these molecular orbitals, with coefficients  $C_{\mu r}$  and  $C_{\nu s}$ .  $H_{\mu\nu}$  is the matrix element between atomic orbitals  $\mu$  and  $\nu$ , which is a measure of the energy of their interaction, roughly proportional to their overlap. The energies of the molecular orbitals are  $\epsilon_r$  and  $\epsilon_s$ .

The first term includes the electrostatic attractions and repulsions between the net charges on pairs of atoms, one from each molecule. The second involves interactions between occupied and vacant molecular orbitals on the two molecules. The hypothesis is that the reaction proceeds in a way to produce the most favorable interaction energy. In one extreme, this can be accomplished by a favorable electrostatic interaction between charges. At the other extreme, it is accomplished by a favorable overlap of Frontier orbitals. Note that the Klopman-Salem equation is approximate, and that even within the framework of the equation, both factors are always at work.

Frontier Orbital theory supplies an additional assumption to this calculation. It considers only the interactions between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). These orbitals have the smallest energy separation, leading to a small denominator in the Klopman-Salem equation. The Frontier orbitals are generally diffuse, so the numerator in the equation has large terms.

The interaction between charges favors reaction between sites (on the two species) that have extreme charge values: the most positive charge interacts with the most negative. This ionic reaction generally involves strongly polar reactants. The second term favors interactions where the HOMO and the LUMO can overlap most favorably. This generally occurs for less polar reactants and is

important when the two have similar energies, so that the denominator is small.

## Atomic Charges and Reactivity

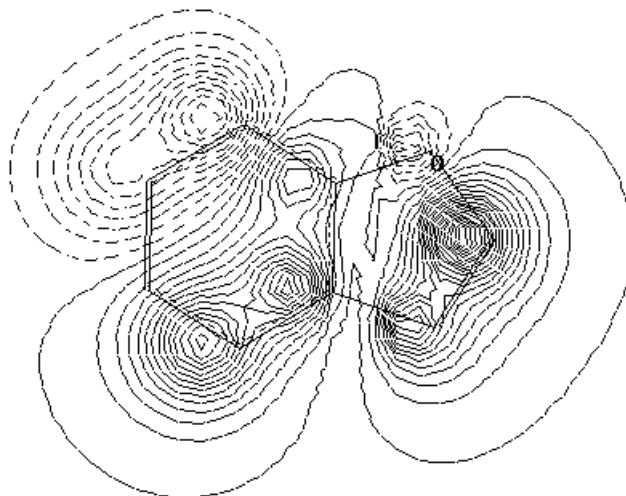
It is generally true that nucleophiles attack molecules at sites of positive charge, and that electrophiles attack sites of negative charge. You can estimate sites of ionic reactivity from the atomic charges in a molecule, particularly for reactions involving “hard” nucleophiles and electrophiles. While the electrostatic potential is a rigorous expression of this idea, the charges provide a quick summary of the main features of the electron distribution. Predictions of ionic reactivity based on atomic charges are generally more reliable when comparing different atoms of the same element.

*Caution:* For ionic reactions in solution, solvent effects can play a significant role. These, of course, are neglected in calculations on a single molecule. You can obtain an indication of solvent effects from semi-empirical calculations by carefully adding water molecules to the solute molecule.

## Frontier Molecular Orbitals

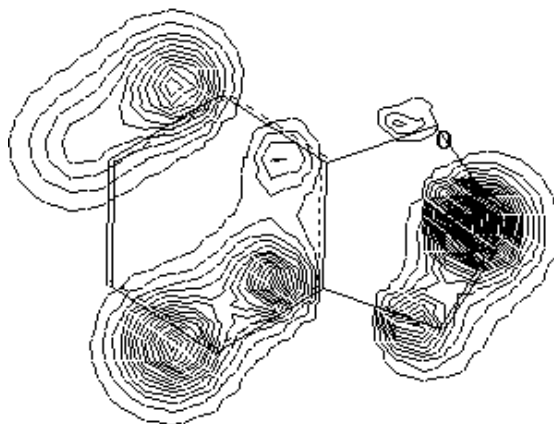
You can interpret the stereochemistry and rates of many reactions involving “soft” electrophiles and nucleophiles—in particular pericyclic reactions—in terms of the properties of Frontier orbitals. This applies in particular to pericyclic reactions. Overlap between the HOMO and the LUMO is a governing factor in many reactions. HyperChem can show the forms of orbitals such as HOMO and LUMO in two ways: a plot at a slice through the molecule and as values in a log file of the orbital coefficients for each atom.

*Example:* An example of Frontier Orbital theory is in predicting sites of nitration (electrophilic attack) on aromatic compounds. If you plot the HOMO as a contour map, the region of highest density (regardless of sign) is generally the site of electrophilic attack. Alternatively, you can look in the log file for the atom with the largest molecular orbital coefficient. This is generally the site of reaction.



In this example, the HOMO is plotted one Ångström above the plane of the molecule. Since it is of  $\pi$  symmetry, it has a node in the plane of the molecule. It shows the site of electrophilic attack at the carbon adjacent to the oxygen atom. This is also the experimentally observed site. The orbital comes from an Extended Hückel calculation of an MM+ optimized geometry.

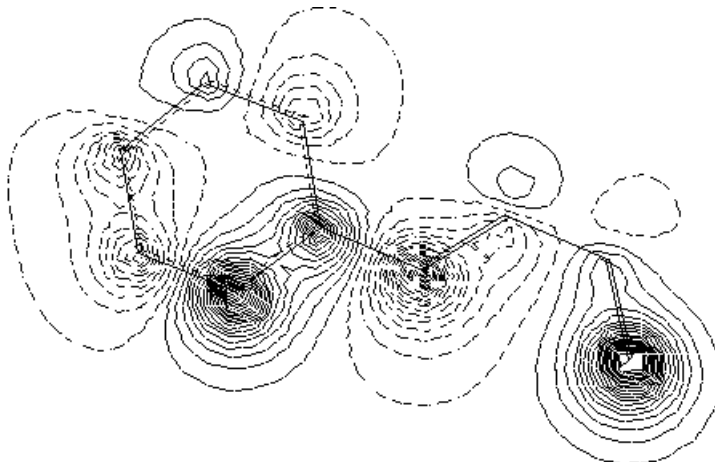
Plotting the same orbital as its density, which is the square of the wave function, emphasizes the difference in magnitude at the different sites.



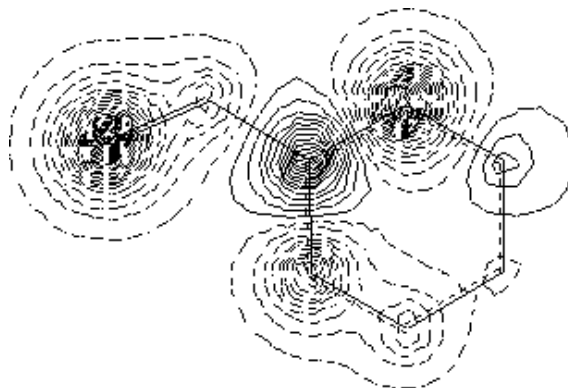
Frontier Orbital theory is qualitative, so there is no need for great accuracy in the calculation as long as it produces the approximate form of the orbitals.

*Example:* Another example of frontier orbital theory uses the reaction of phenyl-butadiene with phenylethylene. This reaction is a [4 + 2] pericyclic addition to form a six-membered ring. It could proceed with the two phenyl rings close to each other (head to head) or further away from each other (head to tail).

An initial assumption is that the reaction takes place with maximum overlap between the HOMO on one molecule and the LUMO on the other. The HOMO of phenylbutadiene has most density on the terminal carbon atom.



The LUMO of phenylethylene also has most density at the terminal (beta) carbon, so that these two atoms attach to each other, yielding the sterically unfavorable product.



This example uses AM1 for optimization and orbital calculations. For more examples of Frontier Orbital theory, see Fleming's book.<sup>29</sup>

## Vibrational Analysis and Infrared Spectroscopy

The quality of the vibrational frequencies varies widely with the semi-empirical method that is used. Generally, AM1, and PM3 are in closer agreement with experiment than methods based on CNDO or INDO.

The vibrational frequencies are derived from the harmonic approximation, which assumes that the potential surface has a quadratic form.

Large amplitude (floppy) vibrational modes often exhibit significant anharmonicity that may increase errors in computed frequencies. In addition to anharmonicity, usually there is coupling between vibrational modes.

If there are negative frequencies in an IR spectrum, it is a sign that you are not at a minimum energy structure. A valid minimum energy structure possesses only positive frequencies.

Results of a recent literature study indicate that frequencies computed using semi-empirical PM3, AM1, and MNDO methods compare well to values obtained at the *ab initio* level using medium size basis sets. Of these three methods, PM3 showed the closest correspondence to experimental values, which is generally about 10 percent too high in value from stretches (Seeger, D.M.; Korzeniewski, C.; Kowalchuk, W., *J. Phys.Chem.* 95:68-71, 1991).



The following table shows the accuracy of computed fundamental frequencies for CO<sub>2</sub> (cm<sup>-1</sup>):

Normal Mode	CNDO	INDO	AM1	PM3	Experiment
bend	571	553	525	521	667
asymmetric stretch	1888	2080	1480	1406	1340
symmetric stretch	6427	5737	2565	2386	2349

### Experimental Characteristic IR Fundamental Frequencies

After you compute an IR spectrum with HyperChem, you can use the table below to assign computed IR lines and qualitatively assess the accuracy of the computation.

Functional Group	Frequency, cm <sup>-1</sup>	Intensity <sup>a</sup>	Assignment
1. Alkanes	2850-3000	s	C - H stretch
	1450-1470	s	CH <sub>2</sub> and CH <sub>3</sub> bend
	1370-1380	s	
	720-725	m	
2. Alkenes			
a) (RCH = CH <sub>2</sub> )	3080-3140	m	= C - H stretch
	1645	m	C = C stretch
	990	s	C - H out-of-plane bend
	910	s	
b) R <sub>2</sub> C = CH <sub>2</sub>	3080-3140	m	= C - H stretch
	1650	m	C = C stretch

Functional Group	Frequency, $\text{cm}^{-1}$	Intensity <sup>a</sup>	Assignment
	890	s	C – H out-of-plane bend
c) <i>cis</i> -RCH = CHR	3020	w	= C - H stretch
	1660	m	C = C stretch
	675-725	m	C – H out-of-plane bend
d) <i>trans</i> -RCH = CHR	3020	w	= C - H stretch
	1675	vw	C = C stretch
	970	s	C – H out-of-plane bend
e) $\text{R}_2\text{C} = \text{CHR}$	3020	w	= C - H stretch
	1670	w	C = C stretch
	790-840	s	C – H out-of-plane bend
f) $\text{R}_2\text{C} = \text{CR}_2$	1670	vw	C = C stretch
3.Alkynes			
a) $\text{RC} \equiv \text{CH}$	3300	s	$\equiv$ C – H stretch
	2100-2140	m	$\text{C} \equiv \text{C}$ stretch
	600-700	s	$\equiv$ C – H bend
b) $\text{RC} \equiv \text{CR}$	2190-2260	vw	$\text{C} \equiv \text{C}$ stretch
4.Alkyl Halides			
a) R – F	1000-1350	vs	C – F stretch
b) R – Cl	750-850	s	C – Cl stretch
c) R – Br	500-680	s	C – Br stretch
d) R – I	200-500	s	C – I stretch
5.Alcohols			
a) $\text{RCH}_2\text{OH}$	3600	var	free O – H stretch

Functional Group	Frequency, $\text{cm}^{-1}$	Intensity <sup>a</sup>	Assignment
	3400	s	bonded O – H stretch
	1050		C – O stretch
b) $\text{R}_2\text{CHOH}$	3600	var	free O – H stretch
	3400	s	bonded O – H stretch
	1150	s	C – O stretch
c) $\text{R}_3\text{COH}$	3600	var	free O – H stretch
	3400	s	bonded O – H stretch
	1200	s	C – O stretch
6. Ethers	1070-1150	s	C – O stretch
7. Aldehydes	1725	s	C = O stretch
	2720, 2820	m	C – H stretch
8. Ketones			
a) acyclic	1715	s	C = O stretch
b) three-membered	1850	s	C = O stretch
c) four-membered	1780	s	C = O stretch
d) five-membered	1745	s	C = O stretch
e) six-membered	1715	s	C = O stretch
f) seven-membered	1705	s	C = O stretch

a. vs=very strong, s=strong, m=medium, w=weak, vw=very weak, v=variable

## UV-visible Spectra

The longest wavelength absorption transition for ethene calculated by HyperChem using PM3 is 207 nm, which compares favorably with the experimental value of 190-200 nm.

After you compute an electronic spectrum with HyperChem, you can use the table below to assign computed transitions and qualitatively assess the accuracy of the computation.:

Substrates	Absorption maxima (nm)
Simple alkenes	190-200
Acyclic dienes	220-250
Cyclic dienes	250-270
Styrenes	270-300
Saturated ketones	270-280
$\alpha$ , $\beta$ - Unsaturated ketones	310-330
Aromatic ketones and aldehydes	280-300
Aromatic compounds	250-280

## Choosing a Semi-Empirical Method

The choice of the NDO method depends on several factors including your previous experience and preferences. If you want to compare the results to other studies, you must use the same semi-empirical method. Since some methods can converge much more quickly than others, you might want to use a fast method to obtain an approximation of the final answer, and then a more accurate method for the final result.

### Extended Hückel

Extended Hückel (EHT) is the computational equivalent of the orbital interaction pictures familiar to chemists (see “Generating and Viewing Orbitals and Electronic Plots” on page 9). The energies of atomic orbitals are well defined, and the total energy is the sum of occupied orbital energies. Neither factor is true for other methods. Extended Hückel is useful for examining the general form and energy ordering of molecular orbitals. The method is simple and it provides qualitative answers to questions about electronic structure.

For transition metals the splitting of the d orbitals in a ligand field is most readily done using EHT. In all other semi-empirical methods, the orbital energies depend on the electron occupation. HyperChem's molecular orbital calculations give orbital energy spacings that differ from simple crystal field theory predictions. The total molecular wavefunction is an antisymmetrized product of the occupied molecular orbitals. The virtual set of orbitals are the residue of SCF calculations, in that they are deemed least suitable to describe the molecular wavefunction.

Normally, you would expect all 2p orbitals in a given first row atom to be identical, regardless of their occupancy. This is only true when you perform calculations using Extended Hückel. The orbitals derived from SCF calculations depend sensitively on their occupation. For example, the 2px, 2py, and 2pz orbitals are not degenerate for a CNDO calculation of atomic oxygen. This is especially important when you look at d orbital splittings in transition metals. To see a clear delineation between  $t_{2u}$  and  $e_g$  levels you must use EHT, rather than other semiempirical methods.

See also "Limitations of Extended Hückel" on page 130.

## CNDO

Although CNDO is less accurate than other NDO methods, there are two common reasons for using it:

- CNDO and INDO are generally faster than the MINDO/3, MNDO, AM1, and PM3 and require much less memory. You can apply them to molecules that are too large for MINDO/3, MNDO, AM1, and PM3.
- Apart from speed, an appealing aspect of CNDO is its simplicity. It uses fewer parameters than any other method except for Extended Hückel and, consequently, it is easier to understand the results of modifying a calculation.

*Note:* Do not use CNDO on any problem where electron-spin is critically important. Its complete neglect of atomic exchange integrals makes it incapable of dealing with these problems.

## INDO

INDO is faster than MINDO/3, MNDO, AM1, and PM3 and, unlike CNDO, can deal with spin effects. It is a particularly appealing choice for UHF calculations on open-shell molecules. It is also available for mixed mode calculations (see the previous section). INDO shares the speed and storage advantages of CNDO and is also more accurate. Although it is preferred for numerical results, it loses some of the simplicity and interpretability of CNDO.

## MINDO/3

MINDO/3 is the earliest of the Dewar methods. It provides more accurate geometries and heats of formation than CNDO or INDO, and has been used widely. The limitations of the INDO approximation, on which MINDO/3 is based, frequently lead to problems of accuracy when dealing with molecules containing heteroatoms.

MINDO/3 is particularly good for describing carbocations, including nonclassical carbocations, and polynitro organic compounds. For these problems it gives better results than MNDO and AM1, even though those methods are generally more accurate. Stewart<sup>23</sup> reports that for a set of 11 nitro and polynitro organic compounds, the average error in heats of formation is 3.5 kcal/mol for MINDO/3, compared to 34.5 kcal/mol for MNDO or 13.7 kcal/mol for AM1.

## MNDO

MNDO was introduced to correct some of the problems with MINDO/3. It has been used widely to calculate heats of formation, molecular geometries, dipole moments, ionization energies, electron affinities, and other properties. It has problems dealing with sterically crowded molecules (too unstable), four-membered rings (too stable), hydrogen bonding (almost nonexistent), and hypervalent compounds (too unstable). Also, nitrobenzene incorrectly yields an out-of-plane nitro group, and the peroxide bond is too short by about 0.17 Ångstrom.

Although AM1 is generally a significant improvement over MNDO, MNDO gives better results for some classes of molecule, such as some phosphorus compounds.

## MNDO/d

MNDO/d is a version of MNDO with d-orbital capability. In addition to the previous MNDO parameter sets, parameters for several new elements are included in the implementation of the MNDO/d method, such as Na, Mg, Cd and Sn. The MNDO/d method and its parameter set referred to can be found in the following references: W. Theil and A. Voityuk, *Theo. Chim. Acta* 81, pp391-404 (1992); W. Thiel and A. Voityuk, *Int.J. Quant. Chem.* 44, pp807-829 (1992);

## AM1

Many problems with MNDO involve cases where the NDO approximation electron-electron repulsion is most important. AM1 is an improvement over MNDO, even though it uses the same basic approximation. It is generally the most accurate semi-empirical method in HyperChem and is the method of choice for most problems. Altering part of the theoretical framework (the function describing repulsion between atomic cores) and assigning new parameters improves the performance of AM1. It deals with hydrogen bonds properly, produces accurate predictions of activation barriers for many reactions, and predicts heats of formation of molecules with an error that is about 40 percent smaller than with MNDO.

Problems still exist with AM1. Treatment of phosphorus-oxygen bonds is inaccurate, nitro compounds are still too positive in energy, and the peroxide bond, for example, is still too short. In many cases, PM3 is an improvement over AM1.

## PM3

PM3 is a reparameterization of AM1, which is based on the neglect of diatomic differential overlap (NDDO) approximation. NDDO retains all one-center differential overlap terms when Coulomb and exchange integrals are computed. PM3 differs from AM1 only in the values of the parameters. The parameters for PM3 were derived by comparing a much larger number and wider variety of experimental versus computed molecular properties. Typically, nonbonded interactions are less repulsive in PM3 than in AM1. PM3 is primarily used for organic molecules, but is also parameterized for many main group elements.

PM3 can also be used to study transition metal (tm) compounds; new parameters include the following elements Ti, Mn, Fe, Co, Ni, Cu, Zr, Mo, Rh, Pd, Hf, Ta and W.

## ZINDO/1

The ZINDO/1 method is the most suitable semi-empirical method in HyperChem for determining structures and energies of molecules with first or second transition row metals.

The ability to perform molecular orbital (MO) calculations on metals is extremely useful because molecular mechanics methods are generally unable to treat metals. This is because metals have a wide range of valences, oxidation states, spin multiplicities, and have unusual bonding situations (e.g.,  $d\pi$ - $p\pi$  back bonding). In addition, the nondirectional nature of metallic bonding is less amenable to a ball and spring interpretation.

Conversely, these factors dictate that molecular orbital calculations on metals yield less reliable results than with organic compounds.

## ZINDO/S

ZINDO/S is parameterized to reproduce spectroscopic transitions, therefore we do not recommend using this method for geometry optimization. You can obtain better results by performing a single-point calculation with ZINDO/S on a geometry obtained from the Model Builder, an optimization using one of HyperChem's other methods, or an external source.

For transition metal complexes with several possible spin arrangements, a separate calculation within each spin multiplicity may be required to find the ground state of the complex.

When computing UV visible spectra, you should do a CI singles calculation. RHF or UHF calculations are sufficient to reproduce the proper order of molecular orbitals in most complexes.

## TNDO

TNDO is essentially an experimental method. While simple, it can be the most accurate of any of the semi-empirical methods. To be accurate, however, requires a good set of parameters for the prop-



erty of interest and the set of atom types included in set of molecules for which predictions are required. If only the CNDO or INDO parameters are available then it reduces to a method with the relatively modest predictability that CNDO or INDO provide.

## Further Reading

Documentation on the reliability of the different NDO methods for various applications is scattered throughout the chemistry literature. Original papers describing the methods present relevant material. The CNDO and INDO methods are discussed in books by Pople and Beveridge<sup>45</sup> and by Murrell and Harget<sup>46</sup>. Compilations exist for MINDO/3 and MNDO in a book by Clark.<sup>47</sup> For MNDO, AM1, and PM3<sup>23</sup>, see the MOPAC documentation by Stewart<sup>24</sup> and a review article by Stewart<sup>27</sup>.

45. Pople, J. A.; Beveridge, D. L. *Approximate Molecular Orbital Theory* McGraw-Hill, New York, 1970.

46. Murrell, J. N.; Harget, A. J. *Semi-empirical Self-consistent-field Molecular Orbital Theory of Molecules* Wiley Interscience, New York, 1971.

47. Clark, T. A. *Handbook of Computational Chemistry*, John Wiley and Sons, New York, 1985.

# **HyperChem<sup>®</sup> Computational Chemistry**

**Part 2  
Theory and Methods**



# Chapter 10

## Theory and Methods Introduction

This part describes the essentials of HyperChem's theoretical and computational chemistry or how HyperChem performs chemical calculations that you request from the Setup and Compute menus. While it has pedagogical value, it is not a textbook of computational chemistry; the discussions are restricted to topics of immediate relevance to HyperChem only. Nevertheless, you can learn much about computational chemistry by reading this manual while using HyperChem.

The theory and methods discussed in this book are HyperChem's two fundamental force-energy-generator modules: one for molecular mechanics and one for quantum mechanics. Molecular mechanics and quantum mechanics are described in subsequent chapters as modules capable of delivering an energy, or derivatives of the energy. Other chapters describe the uses for these energies and their derivatives in more generic parts of HyperChem.

### HyperChem Architecture

While you may not necessarily perceive the difference, HyperChem is designed to consist of two basic components: a *front end* and a *back end*.

The front end is what you see and what you interact with. It provides a user interface to molecular modeling and provides the visualization of molecules and the results of computations. The front end can be thought of as the molecular modeling component of HyperChem.

The back end is the component of HyperChem that performs the more time-consuming scientific calculations. This is where molecular mechanical and quantum mechanical calculations are performed. The back end can be thought of as the computational chemistry component of HyperChem.

## The Back Ends

As an aid in understanding how to use HyperChem effectively, this section describes the essentials of the front end and back end architecture. While this may not be necessary, an intuition as to how the program operates can be useful in optimizing its efficient use. A network version of HyperChem will allow the front end to run on a local workstation or personal computer with the back end running somewhere else in the network, possibly on a large, parallel-processing machine at a different geographical location. The non-network version simply merges the front end and back end into the same machine.

The user only interacts with the front end. The front end collects input from the user, initiates back end calculations, collects results from the back end, and then, if requested, displays these results to the user. The front end launches a back end program, sends it input data, and then receives output results from it. The back end programs included with HyperChem are HyperMM+, HyperNewton (performs AMBER, BIO+, and OPLS calculations), HyperEHT (performs Extended Hückel calculations), HyperNDO (performs CNDO, INDO, MINDO/3, MNDO, AM1, PM3, ZINDO/1, and ZINDO/S calculations), HyperTNDO (performs TNDO calculations), HyperGauss (performs *ab initio* quantum mechanical calculations), and HyperDFT (performs density functional calculations). When the front end initiates a back end program, an icon will appear (for the Microsoft Windows version only) representing the back end program. The icons are a falling red apple, indicating a molecular mechanics calculation and an orbiting electron rendering of an atom, indicating a semi-empirical quantum mechanical calculation or a blue psi wavefunction symbol indicating an *ab initio* or DFT calculation. While the icon is visible, a back end program is active. The user can explicitly stop a back end program but only the front end can start a program.

A back end program essentially acts as a computational server for the front end. It receives input, computes something, sends it back to the front end and then looks for further input or commands from the front end. It can be used over and over for different calculations. If it has been explicitly stopped, the front end will start the back end executing again as necessary. If the front end determines that it needs to stop a back end program or start another copy it will do so. In particular, switching between calculation methods may stop a back end program and start another one.

For example, if the user requests a molecular dynamics calculation using the AMBER force field, HyperChem starts a copy of HyperNewton running and sends it a copy of the chosen AMBER parameter set. It then sends it a copy of the current molecular system in the work space and the appropriate parameters for the molecular dynamics run. The back end runs the dynamics trajectory periodically sending back results to the front end to update the display of the molecule, plot structural or energetic values, etc. The front end and back end communicate via messages that are as appropriate to a distributed computing environment as they are to the single machine configuration.

The internal architecture of HyperChem back ends is different from that expected to be used by third-party packages. To a third-party agent wishing to interface with HyperChem, HyperChem always acts as a server. Thus a third-party molecular dynamics package would ask HyperChem to send the coordinates of a molecule rather than HyperChem determining on its own that it should send coordinates at the appropriate time.

## **HyperChem Philosophy**

The HyperChem philosophy associated with back end computations is one which is intended to instill confidence, as far as is possible, in the scientific results emanating from HyperChem. This philosophy is one of openness — openness about the product, the calculations being performed, the science embodied in the product, etc. Apart from protecting the proprietary code associated with a commercial product, Hypercube wishes to document and describe as fully as is possible the calculations that HyperChem performs. There should be no mystery about the scientific results obtained with HyperChem.

HyperChem should not be viewed as a black box that computes only what its designers thought correct. It has an open architecture that makes it possible to customize it many ways. As far as is possible, the parameters of molecular mechanics and semi-empirical calculations are in the user's hands. As the techniques of software engineering advance and our expertise in building new releases of HyperChem progresses, we intend to make as many facets of HyperChem computations available to the user as possible. In the past, it was thought necessary to provide source code to users to allow them to customize a product. The ability to customize a product to one's own use is now attainable much more simply by means of an open architecture and well-defined, documented ways to use and customize the product.

In order to balance public domain science with a high quality commercial software product it has been necessary for us to re-implement almost every aspect of computational chemistry embodied in HyperChem. All HyperChem source code is written in C or C++, specified, designed, and implemented by HyperChem's developers. We have stood on the scientific shoulders of giants, but we have not used their FORTRAN code! Thus, although we have had access to MOPAC and other public domain codes for testing and other purposes, HyperChem computes MINDO, MNDO, and AM1 wave functions, for example, with HyperChem code, not MOPAC code. We have made the effort to implement modern chemical science in a modern software-engineered product.

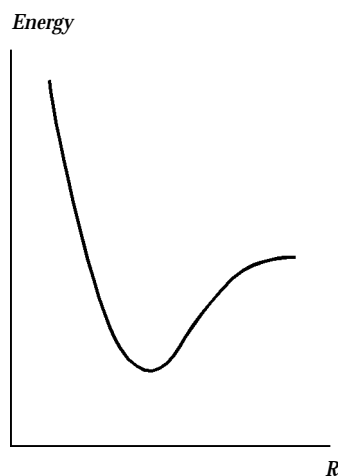
## Background on Computational Chemistry

The principal theory behind HyperChem is the concept of a potential energy surface and the distinction between classical and quantum energies, kinetic and potential energies, energies of electrons versus energies of nuclei, etc. This section provides a concise approach to the problem of defining a potential energy surface for the motion of nuclei and begins with the usual quantum mechanical definitions prior to a more rigorous quantum mechanical definition.

### Potential Energy Surfaces

A potential energy surface is simply a specification of the classical potential energy,  $V$ , as a function of molecular structure. For exam-

ple, the potential surface (in this case a potential curve) for a diatomic molecule is sketched qualitatively in the following illustration. The potential energy curve shows the potential energy of the molecule as a function of the internuclear distance  $R$ . Now, in fact, there are six degrees of freedom on which the energy might depend—for instance the X, Y, and Z coordinates of each of the two atoms. However, five of these degrees of freedom correspond to translations (3) and rotations (2) of the rigid molecule and do not affect the energy of the system. This leaves only the internuclear distance as the variable upon which the potential energy depends.



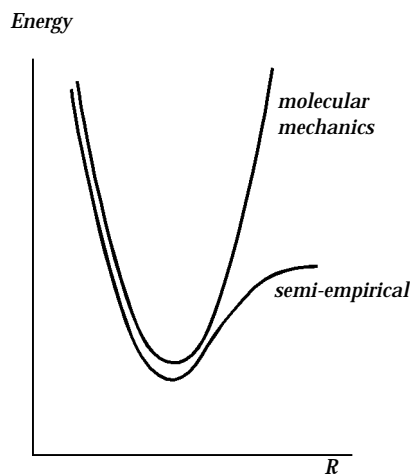
The curve above shows that as  $R \rightarrow \infty$ , the potential energy approaches a constant, which is the energy of the two individual atoms. Further, there is a global minimum for this potential surface at intermediate distances. At very short distances, the energy rises to  $+\infty$  as the two atoms repel each other.

The semi-empirical methods of HyperChem are quantum mechanical methods that can describe the breaking and formation of chemical bonds, as well as provide information about the distribution of electrons in the system. HyperChem's molecular mechanics techniques, on the other hand, do not explicitly treat the electrons, but instead describe the energetics only as interactions among the nuclei. Since these approximations result in substantial computational savings, the molecular mechanics methods can be applied to much larger systems than the quantum mechanical methods. There are many molecular properties, however, which



are not accurately described by these methods. For instance, molecular bonds are neither formed nor broken during HyperChem's molecular mechanics computations; the set of fixed bonds is provided as input to the computation.

This difference is shown in the next illustration which presents the qualitative form of a potential curve for a diatomic molecule for both a molecular mechanics method (like AMBER) or a semi-empirical method (like AM1). At large internuclear distances, the differences between the two methods are obvious. With AM1, the molecule properly dissociates into atoms, while the AMBER potential continues to rise. However, in explorations of the potential curve only around the minimum, results from the two methods might be rather similar. Indeed, it is quite possible that AMBER will give more accurate structural results than AM1. This is due to the closer link between experimental data and computed results of molecular mechanics calculations.



HyperChem provides three types of potential energy surface sampling algorithms. These are found in the HyperChem Compute menu: Single Point, Geometry Optimization, and Molecular Dynamics.

### Single Point

A single point calculation, as its name suggests, performs a calculation at only a single point on the potential surface. For a diatomic molecule, this might be a calculation at  $R=2.0 \text{ \AA}$ , for example. The results of a single point calculation give the potential energy of the system at that geometry, as well as the gradient at that point. For single parameter potential curves like that shown above, the gradient describes the steepness of the potential curve at that point along the direction in which the energy decreases. For a poly-

atomic system, the situation is more complicated, but essentially the same — the gradient gives the direction in which the energy goes most steeply downhill, along with the steepness of the downhill slope. The RMS gradient that is reported is just the root-mean-square average of the Cartesian components of the gradient vector.

## Geometry Optimization

A geometry optimization samples single points on the potential surface, searching for a minimum. At a potential minimum, the gradient is zero, and any small change in geometry causes the potential energy to rise. The technique used to search for the minimum is called the optimization algorithm, and several of these are available for selection in the optimization dialog box. The optimization algorithms are recipes for using the past history of points sampled from the potential surface to determine the next point to examine.

## Molecular Dynamics

Molecular dynamics involves the addition of kinetic energy to the above potential energy surface description and the subsequent motion of the molecular system over the potential surface. This motion follows the laws of classical mechanics according to Newton. Normally (except for the constant temperature algorithm of HyperChem), the classical total energy (sum of kinetic energy and potential energy) is conserved and the motion is faster (larger kinetic energy) near minima in the potential surface (smaller potential energy). If a set of initial conditions is defined (initial velocities and a particular point on the potential surface), then Newton's laws cause the molecular system to evolve along a path that is referred to as the molecular dynamics trajectory. This trajectory traverses the potential surface in ways that are of considerable interest to explore. Both the end point of a trajectory and the path taken to get there are of interest in molecular modeling.

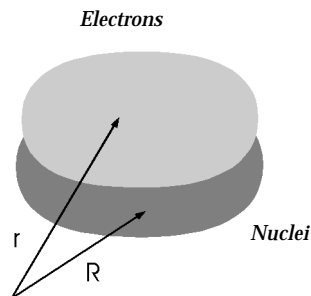
## The Born-Oppenheimer Approximation

The first basic approximation of quantum chemistry is the Born-Oppenheimer Approximation (also referred to as the clamped-nuclei approximation). The Born-Oppenheimer Approximation is used to define and calculate potential energy surfaces. It uses the heavier mass of nuclei compared with electrons to separate the

problem of computing the energy of a collection of nuclei and electrons (a molecular system) into two problems: an electronic problem and a nuclear motion problem. The electronic problem solves for the wave function of the electrons in the field of fixed (clamped) nuclei. The nuclear motion problem subsequently solves for the motion of nuclei (which can be referred to as atoms here) on a potential energy surface. The potential energy surface is the energy of the associated electronic problem, as a function of the position of the clamped nuclei (clamped only during each single energy calculation).

## The Hamiltonian of a Collection of Nuclei and Electrons

A molecular system in HyperChem consists of  $N$  nuclei  $A$  (having positive charges,  $+Z_A$ ) and  $M$  electrons  $i$  (each with negative charge,  $q_i = -1$ ). The nuclei are described by a vector,  $\mathbf{R}$ , with  $3N$  Cartesian  $X$ ,  $Y$ , and  $Z$  components. The electrons are described by a vector,  $\mathbf{r}$ , with  $3M$  Cartesian  $x$ ,  $y$ , and  $z$  components. The electrons are explicitly considered only in semi-empirical calculations.



A Hamiltonian is the quantum mechanical description of an energy contribution. The *exact* Hamiltonian for a molecular system is:

$$\mathcal{H}(\mathbf{R}, \mathbf{r}) = \mathcal{T}_n(\mathbf{R}) + \mathcal{T}_e(\mathbf{r}) + \mathcal{V}_{ee}(\mathbf{r}, \mathbf{r}) + \mathcal{V}_{ne}(\mathbf{R}, \mathbf{r}) + \mathcal{V}_{nn}(\mathbf{R}, \mathbf{R}) \quad (1)$$

where  $\mathcal{T}_n(\mathbf{R})$  is the total kinetic energy of the nuclei,  $\mathcal{T}_e(\mathbf{r})$  is the total kinetic energy of the electrons,  $\mathcal{V}_{ee}(\mathbf{r}, \mathbf{r})$  is the potential energy of interaction of electrons with electrons,  $\mathcal{V}_{ne}(\mathbf{R}, \mathbf{r})$  is the potential energy of interaction of the nuclei with electrons, and  $\mathcal{V}_{nn}(\mathbf{R}, \mathbf{R})$  is the potential energy of interaction of nuclei with nuclei.

## The Electronic Hamiltonian

Since nuclei are much heavier than electrons and move slower, the Born-Oppenheimer Approximation suggests that nuclei are stationary and thus that we can solve for the motion of electrons only. This leads to the concept of an *electronic Hamiltonian*, describing the motion of electrons in the potential of fixed nuclei.

$$\mathcal{H}_{\text{elec}}(\mathbf{r}) = \mathcal{J}_{\text{e}}(\mathbf{r}) + \mathcal{V}_{\text{ee}}(\mathbf{r}, \mathbf{r}) + \mathcal{V}_{\text{ne}}(\mathbf{R}, \mathbf{r}) \quad (2)$$

HyperChem's semi-empirical calculations solve (approximately) the Schrödinger equation for this electronic Hamiltonian leading to an electronic wave function  $\Psi_{\text{elec}}(\mathbf{r})$  for the electrons:

$$\mathcal{H}_{\text{elec}}(\mathbf{r})\Psi_{\text{elec}}(\mathbf{r}) = E_{\text{elec}}(\mathbf{R})\Psi_{\text{elec}}(\mathbf{r}) \quad (3)$$

The eigenvalue of this Schrödinger equation, the electronic energy  $E_{\text{elec}}$ , depends parametrically, as shown, on the coordinates of the nuclei (assumed to be fixed for the purposes of calculating each  $E_{\text{elec}}(\mathbf{R})$ , but variable in general). The electronic energy,  $E_{\text{elec}}(\mathbf{R})$  combined with  $\mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R})$  is the total energy of Single Point semi-empirical calculations.

## The Nuclear Hamiltonian

In the same way that we assumed nuclei move much slower than electrons, we can assume electrons move much faster than nuclei and the detailed motion of electrons might be replaced by their average position. Thus Equation (1) above becomes:

$$\mathcal{H}(\mathbf{R}) = \mathcal{H}(\mathbf{R}, \langle \mathbf{r} \rangle) = \mathcal{J}_{\text{n}}(\mathbf{R}) + \langle \mathcal{H}_{\text{elec}} \rangle + \mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R}) \quad (4)$$

or

$$\mathcal{H}(\mathbf{R}) = \mathcal{J}_{\text{n}}(\mathbf{R}) + E_{\text{elec}}(\mathbf{R}) + \mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R}) \quad (5)$$

This last equation is the nuclear Schrödinger equation describing the motion of nuclei. The electronic energy computed from solving the electronic Schrödinger equation (3) on page 170 plus the nuclear-nuclear interactions  $\mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R})$  provide a potential for nuclear motion, a Potential Energy Surface (PES).

$$\mathcal{V}_{\text{PES}}(\mathbf{R}) = \mathcal{V}(\mathbf{R}) = E_{\text{elec}}(\mathbf{R}) + \mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R}) \quad (6)$$

This potential energy surface, which is just values of the Single Point energy of HyperChem, can be thought of as describing the

interactions between the atoms (described by coordinates  $\mathbf{R}$ ) of a molecular system. It is commonly referred to in quantum mechanical calculations as the total energy, although it is only the potential energy as far as the motion of a collection of atoms is concerned. It provides a potential for a nuclear Hamiltonian:

$$\mathcal{H}(\mathbf{R}) = \mathcal{T}_n(\mathbf{R}) + \mathcal{V}(\mathbf{R}) \quad (7)$$

Solutions to a Schrödinger equation for this last Hamiltonian (7) describe the vibrational, rotational, and translational states of a molecular system. This release of HyperChem does not specifically explore solutions to the nuclear Schrödinger equation, although future releases may. Instead, as is often the case, a classical approximation is made replacing the Hamiltonian by the classical energy:

$$E = T + \mathcal{V}(\mathbf{R}) \quad (8)$$

where  $T$  is the classical kinetic energy and  $\mathcal{V}$  is the potential energy. The potential energy  $\mathcal{V}$  is either the total energy,  $E_{\text{elec}}(\mathbf{R}) + \mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R})$ , reported from a semi-empirical quantum mechanical calculation or the (potential) energy of a molecular mechanics calculation. In either case,  $\mathcal{V}$  is the energy computed in Single Point or Optimization calculations. The kinetic energy,  $T$ , is only computed during Molecular Dynamics calculations, where a distinction must be made between kinetic and potential energy. Elsewhere, the total energy usually means the total potential energy of the nuclei (atoms) as a function of the coordinates of the nuclei (atoms).

The quantity,  $\mathcal{V}(\mathbf{R})$ , the sum of the electronic energy  $E_{\text{elec}}$  computed in a wave function calculation and the nuclear-nuclear coulomb interaction  $\mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R})$ , constitutes a potential energy surface having  $3N$  independent variables (the coordinates  $\mathbf{R}$ ). The independent variables are the coordinates of the nuclei; but having made the Born-Oppenheimer approximation, we can think of them as the coordinates of the atoms in a molecule.

## Molecular Mechanics versus Quantum Mechanics

Molecular quantum mechanics finds the solution to a Schrödinger equation for an electronic Hamiltonian,  $H_{\text{elec}}$ , that gives a total energy,  $E_{\text{elec}}(\mathbf{R}) + \mathcal{V}_{\text{nn}}(\mathbf{R}, \mathbf{R})$ . Repeated solutions at different nuclear configurations,  $\mathbf{R}$ , lead to some approximate potential energy sur-

face  $\mathcal{V}(\mathbf{R})$ . This procedure requires the solution, albeit a very approximate solution, of some Schrödinger equation for every value of  $\mathbf{R}$ . This can be very time consuming and impractical for large molecular systems.

Molecular mechanics approximates  $\mathcal{V}(\mathbf{R})$  by analytical functions that are deduced and subsequently parameterized via accurate calculations or experimental results. In many situations, the potential energy surface obtained is more accurate than that calculated from first principles because of resorting to experiment for defining the parameters of the analytical functions. The analytical function (potential energy surface)  $\mathcal{V}(\mathbf{R})$  is sometimes referred to as a force field. Most force fields contain terms associated with bond stretching, angle bending, torsional twisting, and other terms that have chemical significance. The common difficulty with molecular mechanics is that a  $\mathcal{V}(\mathbf{R})$  may not have been defined or parameterized for the molecular system of interest.

## Classical Mechanics on a Potential Energy Surface

Rather than solve a Schrödinger equation with the Nuclear Hamiltonian (above), a common approximation is to assume that atoms are heavy enough so that classical mechanics is a good enough approximation. Motion of the particles on the potential surface, according to the laws of classical mechanics, is then the subject of classical trajectory analysis or molecular dynamics. These come about by replacing Equation (7) on page 171 with its classical equivalent:

$$E(\mathbf{R}) = \sum_i \frac{1}{2} m_i (d\mathbf{R}_i/dt)^2 + \mathcal{V}(\mathbf{R}) \quad (9)$$

In this context,  $E$  is the total classical energy including kinetic energy. You can then investigate the potential energy surface in a purely classical way using the positions ( $\mathbf{R}_i$ ) and velocities ( $\mathbf{V}_i = d\mathbf{R}_i/dt$ ) of the constituent atoms.

## Force-Energy Generators

HyperChem performs a number of different types of calculations such as geometry optimization, molecular dynamics, etc. Each of these types of calculations is designed to be, as far as is possible, independent of the technology that produces the potential energy surface. The technology that produces the potential energy surface is referred to here as a force-energy generator. Thus, geometry optimization of a molecule is fed by routines that compute the potential energy  $\psi(\mathbf{R})$  and its derivatives  $d\psi(\mathbf{R})/d\mathbf{R}$ ,  $d^2\psi(\mathbf{R})/d\mathbf{R}^2$ , etc. without concern whether these force-energy generators are molecular mechanics or semi-empirical quantum mechanics. New force-energy generators could, in principle, be added easily. An example would be the addition of an *ab initio* force-energy generator for which all the other methodology of HyperChem, such as molecular dynamics, would become available.





# Chapter 11

## Molecular Mechanics

Molecular Mechanics uses an analytical, differentiable, and relatively simple potential energy function,  $\mathcal{V}(\mathbf{R})$ , for describing the interactions between a set of atoms specified by their Cartesian coordinates  $\mathbf{R}$ .

Unlike first principles of quantum mechanical calculations, molecular mechanics might be thought of as simply a fitting procedure, attempting to obtain as accurately as possible a representation of  $\mathcal{V}(\mathbf{R})$  with no particular regard for theoretical foundations. However, it is found that the most successful fitting procedures, having generic utility, lead to terms in the potential that can be ascribed to chemically meaningful interactions. For example, molecular mechanics potentials typically have simple analytic terms that provide an energetic penalty for deviations from standard bond lengths, bond angles, and bond dihedral angles, together with simple analytic terms for long-range coulombic and van der Waals interactions.

To be useful, molecular mechanics potentials have to balance the simplicity of the analytical form of the potential (so that the energy and its derivatives can be rapidly evaluated, even for large molecules) with the accuracy of describing important characteristics of the exact  $\mathcal{V}(\mathbf{R})$ . Developing these molecular mechanics force fields has been an active and important activity for many years, resulting in a number of standard methodologies as described below. None of the standard force fields used in HyperChem, however, have complete general utility and they must be used with care. Force fields are usually developed for specific types of molecules and their use in other situations will either fail (for lack of parameters for the analytic expressions) or give poor results (because it is employed beyond the original range of molecular situations for which the force field was developed).

## Background

Molecular mechanics has three important concepts:

- Functional form
- Atom types
- Parameter sets

Each molecular mechanics method has its own functional form: MM+, AMBER, OPLS, and BIO+. The functional form describes the analytic form of each of the terms in the potential. For example, MM+ has both a quadratic and a cubic stretch term in the potential whereas AMBER, OPLS, and BIO+ have only quadratic stretch terms. The functional form is referred to here as the force field. For example, the functional form of a quadratic stretch with force constant  $K_r$  and equilibrium distance  $r_0$  is:

$$E_{\text{quadratic stretch}} = K_r(r - r_0)^2 \quad (10)$$

Each force field includes a set of atom types. Consider the quadratic stretch term shown above. In principle, every different bond in every molecule would have its own parameters  $r_0$  and  $K_r$ . This would be unmanageable for a generic method that attempts to compare molecules. Instead, atoms are classified as having a given type. Thus, you can distinguish an sp-hybridized carbon from an sp<sup>2</sup>-hybridized carbon from an sp<sup>3</sup>-hybridized carbon by saying they are different types (C2, C3, and C4 in MM+, for example). You can then describe quadratic stretching constants  $r_0$  and  $K_r$  for each pair of atom types (C2-C2, C2-C3, C3-C3, etc. as in MM+). The atom types in HyperChem are a characteristic of the force field with one set of atom types for each force field.

Finally, each force field may have multiple parameter sets (the values of  $r_0$  and  $K_r$ , for example). The AMBER force field and AMBER set of types may be used with, for example, the AMBER/2 or AMBER/3 set of parameters.

These ideas plus others related to the molecular mechanics options of HyperChem are discussed in this chapter.

# Atom Types

The concept of atom types might be considered the most fundamental topic in molecular mechanics and lies behind all aspects of the approach. Atom types, not atoms, are the fundamental basis for calculating interactions in molecular mechanics.

## Definition of Atom Type

The atom type defines the chemical environment of an atom. The basic idea is that not all carbon atoms in molecules are the same and can be distinguished by the following:

- Hybridization
- Formal charge on the atom
- Immediate bonded neighbors

The chemical environment for an atom in a molecule is probably unique to that molecule. Chemistry tries to find unifying concepts and the atom type is one of those unifying concepts. For example, the AMBER force field defines five atom types for oxygens:

- O** carbonyl oxygen
- OH** hydroxyl (alcohol) oxygen
- O2** carboxylic acid oxygen or phosphate oxygen
- OS** ester or ether oxygen
- OW** oxygen in water

The molecular mechanics interactions are computed between types not elements. Thus, you would compute different values for the long range nonbonded interaction of two water oxygens than you would for the interaction of water oxygen and an ester oxygen. These values would differ from the values for the interaction of two ester oxygens. Of course, if you want to neglect the chemical environment in calculating interactions, you could define some default type for all oxygens to make all oxygens identical.

In principle, atom types could be associated with a particular parameter set rather than the functional form or force field. In HyperChem, however, atoms types are rigorously tied to a force field: MM+, AMBER, OPLS, and BIO+. Each of the force fields has

a default set of atom types and all parameter sets for a given force field must use the same set of atom types. It is possible, however, to redefine or modify the atom types associated with a force field as described in the next section.

Atom types are defined in the file `chem.rul`. The atom types for each of the force fields are listed in the files pointed to by the `AtomTypeMass` entries associated with the specific force fields in the Registry. They are usually named `*typ.txt` or `*typ.dbf`, depending on whether text or dBASE format is used. For the AMBER force field, all variations use the same type file, defined in the [Amber] section of the Registry. It is usually called `ambertyp.txt`, if a text format file is used, indicated by `FileFormat=Text` or `ambertyp.dbf`, if a dBASE III format file is used, indicated by `FileFormat=dbf` [ dBase formats for parameters are being discontinued in favor of text files].

The concept of atom types is particularly relevant to the united atom approach. In AMBER, for example, an all atom treatment of a methyl group uses the type `CT` to describe the carbon and the type `HC` to describe the hydrogens. A united atom approach collapses the hydrogens into the carbon nucleus to represent the whole methyl group as one type, `C3`, located at the position of the carbon but with quite different interaction characteristics (parameters). For each force field, the list of types and their atomic masses is stored in the file pointed to by the `AtomTypeMass` entry in the Registry or `chem.ini`. The masses of `CT` and `C3` are respectively 12.01 and 15.03, with the larger number including the mass of the hydrogen atoms.

While atom types are tied to a specific force fields, it is easy to modify each force field's atom types; the functional form cannot be modified but atom types can. The next section describes how atom types are defined.

## The Typing Rules

Atom types represent the chemical environment of an atom. The atom types associated with a given force field could be hard-wired to have specific values and meaning. HyperChem also allows flexible definitions of the atom types and the associated chemical environments. The chemical environment of an atom (a set of rules for defining a type) and the default rules are in a standard ASCII text file, `chem.rul`, included with HyperChem. You can

modify this file and compile it in a binary form that HyperChem uses to calculate the atom types from the chemical environment of an atom.

The syntax and semantics of the typing rules in a chem.rul file are included in the *HyperChem Reference Manual*. The following example illustrates their use. The five AMBER types for oxygen atoms shown above are defined in chem.rul by the following rules:

```
O:
    ; carbonyl
    connected to =C?
    =O.

    ; neutral carboxylic acid
    connected to (H)C(=O)?
    =OH.

    ; negative carboxylic acid
    connected to ~C~O?
    =O2.

    ; ether
    connected to (-C)-C?
    =OS.

    ; ester
    connected to (C=O)C?
    =OS.

    ; water
    connected to H2?
    =OW.

    ; hydroxyl
    connected to H?
    =OH.

    ; phospho ester
    connected to (C)P?
    =OS.
```

```

; phosphate
connected to P?
=O2.

; default
=OS.

```

These rules are executed in order until a set of rules is satisfied. If all the rules fail then the last default type OS is assigned. The tests on neighboring lines are AND'ed together and sets of tests (clauses) separated by blank lines are OR'ed.

The above rules state that if an oxygen is connected to a carbon by a double bond it is a carbonyl oxygen with type O. Otherwise, if it is connected to both a hydrogen and a carbon that is itself connected by a double bond to another oxygen then it is a neutral carboxylic oxygen with type OH. Otherwise, if it is connected by an aromatic bond to a carbon which in turn is connected by an aromatic bond to another oxygen then it is a negative carboxylic oxygen with type O2. Otherwise, if it is connected by single bonds to two carbon atoms it is an ether oxygen of type OS. Otherwise, if it is connected to a carbonyl group and to a carbon it is an ester oxygen with type OS. Otherwise, if it is connected to two hydrogens it is a water oxygen with type OW. Otherwise, if it is connected to a hydrogen it is an hydroxyl oxygen with type OH. Otherwise, if it is connected to a carbon and a phosphorus it is a phospho ester oxygen with type OS. Otherwise, if it is connected to a phosphorus it is a phosphate oxygen with type O2. Otherwise, for lack of other information it is defined to be of type OS.

Note that two different environments, although they might be distinguished by tests (such as for ether and ester) can share an atom type (such as OS). A refinement of the AMBER force field would use separate types for these two along with different parameters for the different types.

## Redefining or Adding Types

To redefine an atom type associated with a force field, adjust the rules in the chem.rul file to represent the new chemical environment for a particular type and then compile the new types. It is always desirable to save the original chem.rul under another name prior to modifying chem.rul. Having modified chem.rul, you can

start a copy of HyperChem and click Build/Compile Type Rules to create a new typerule.bin, the binary file which HyperChem always uses to calculate atom types.

To add new types to a force field, follow this procedure:

1. Add the new type to the chem.rul file.
2. Add the new type and its mass to the file pointed to by the AtomTypeMass entry of the Registry or the chem.ini file for the appropriate force field.
3. Compile the chem.rul file to a new typerule.bin file as described in the previous paragraph.
4. Add nonbonded and other parameters to the appropriate files.

The new parameter files must be compiled for each parameter set used with the particular force field that has the new type, as follows:

1. Start HyperChem.
2. Choose the appropriate force field (MM+, AMBER, OPLS, or BIO+).
3. Select Setup/Select Parameter Set and specify a particular parameter set.
4. Select Setup/Compile Parameter File to create a new binary \*.par parameter file that includes the new type.
5. Repeat steps 3 and 4 for each of the parameter sets available with the particular force field being modified.

## Force Fields

This section describes HyperChem's four force fields, MM+, AMBER, OPLS, and BIO+, providing auxiliary information for all force field calculations.

*Note:* All of the force fields provided in HyperChem are built on new implementations of force fields developed by various computational chemistry research groups. However, HyperChem improves on the original force fields and uses new code.



## Background

Many of the ideas and issues surrounding the use of molecular mechanics (or force field technology) in computational chemistry are common to all force fields and in this section we describe many of these basic ideas.

### Energetic Terms in the Potential

The interaction potential  $\mathcal{V}(\mathbf{R})$  describes both bonding and non-bonding interactions. The bonding interactions are usually formulated as a strain energy that is zero at some ideal configuration of the atoms and describe how the energy increases as the ideal configuration is deformed. Bonding interactions usually refer to atoms in the following relationships:

- Directly bonded (a 1-2 bond stretch relationship)
- Geminal to each other (a 1-3 angle bending relationship)
- Vicinal to each other (a 1-4 dihedral angle rotation relationship)

The nonbonded interactions usually include the following:

- An exchange repulsion when atoms get too close
- A long range attraction arising from dispersion forces
- Electrostatic interactions coming from the interaction of charges, dipoles, quadrupoles, etc.

The exchange repulsion and dispersive attraction combine in what is referred to as a van der Waals term. Sometimes a potential is added to account for hydrogen bonding explicitly; while in other situations this is expected to fall out of ordinary electrostatic interactions.

In addition to these basic terms, force fields often have cross terms that combine the above interactions. For example there may be a term which causes an angle bend to interact with a bond stretch term (opening a bond angle may tend to lengthen the bonds involved).

The interaction potential  $\mathcal{V}(\mathbf{R})$  in used by the force fields in HyperChem share the following types of terms:

## Bond Stretching

This term is associated with deformation of a bond from its standard equilibrium length. For small displacements from equilibrium, a harmonic function is often used:

$$E_{\text{bond}} = \sum_{\text{bonds}} K_r (r - r_0)^2 \quad (11)$$

A larger value for the stretch force constant  $K_r$  leads to a greater tendency for the bond to remain at its equilibrium distance  $r_0$ . Higher powers of  $r - r_0$ , giving cubic, quartic, or higher terms are also common. A Morse function might also be employed.

## Bond Angle Bending

This term is associated with the deformation of an angle from its normal value. For small displacements from equilibrium, a harmonic function is often used:

$$E_{\text{bond angle}} = \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2 \quad (12)$$

A larger value for the bending force constant  $K_\theta$  leads to a greater tendency for the angle to remain at its equilibrium value  $\theta_0$ . There may be cubic, quartic, etc. terms as with the corresponding bond stretch term in addition to the quadratic term shown here.

## Dihedrals

This term is associated with the tendency of dihedral angles to have a certain  $n$ -fold symmetry and to have minimum energy for the *cis*-, *gauche*-, or *trans*-conformation, etc.

$$E_{\text{dihedral}} = \sum_{\text{dihedral}} \frac{V_n}{2} [1 + \cos(n\phi - \phi_0)] \quad (13)$$

The period of the interaction is  $360/n$ . The phase angle  $\phi_0$  shifts the curve to the left or right. For  $n=1$  and  $\phi_0=0$ , the curve represents the situation where the energy is a minimum for the *trans*-conforma-

tion with a barrier of  $V_n$  to the highest energy *cis*-conformation. A phase angle of  $\phi_0 = 180$  represents the opposite situation with a minimum at the *cis*-conformation and a maximum at the *trans*-conformation. By including sums of terms of the above kind, dihedral angle interactions of arbitrary complexity can be described. In HyperChem,  $n$  is restricted to values from 1 to 6, so up to six Fourier terms may be used for a particular dihedral angle.

## Improper Dihedrals

This term is associated with maintaining the planarity of planar atoms, or preventing inversion of tetrahedral carbons with one hydrogen in a united atom representation. This type of term usually employs an energetic penalty for deformation of a dihedral angle defined by the central atom and its three bonded neighbors. The functional form employed may be that of equation (12) or equation (13) on page 183, or a different scheme altogether (as in MM+).

## van der Waals

This term describes the repulsive forces keeping two nonbonded atoms apart at close range and the attractive force drawing them together at long range.

$$E_{\text{vanderWaals}} = \sum_{ij \in \text{vdW}} \left[ \frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} \right] \quad (14)$$

The above potential is referred to as a Lennard-Jones or 6-12 potential and is summed over all nonbonded pairs of atoms  $ij$ . The first positive term is the short range repulsion and the second negative term is the long range attraction. The parameters of the interaction are  $A_{ij}$  and  $B_{ij}$ . The convenient analytical form of the 6-12 potential means that it is often used, although an exponential repulsion term is usually considered to be a more accurate representation of the repulsive forces (as used in MM+).

The force fields in HyperChem that use the above 6-12 potential allow six ways of specifying the constants  $A_{ij}$  and  $B_{ij}$ ; three by single atom type and three by pairs of atom types. Single atom type means that there are constants for individual atom types,  $i$ , that are combined by a combining rule that results in a parameters for

a specific pair,  $ij$ , of atom types. Pairs of atom types means that parameter files contain explicit parameters for a pair of atom types  $ij$  and that no combining of single atom type parameters is necessary. If in the first case, there were  $N$  parameters, the second case would require  $N(N-1)/2$  parameters for an equivalent set. If present, the constants specified by pairs of atoms types are used to override the values generated from single atom types.

The form that single atom type constants take is selected by setting the Registry/chem.ini parameter set entry 6-12AtomVDWFormat to one of RStarEpsilon, SigmaEpsilon or SlaterKirkwood. This specifies the combination rules that are used for the file pointed to by the 6-12AtomVDW entry in the Registry or the chem.ini file for the same parameter set.

For the 6-12AtomVDWFormat entry set to RStarEpsilon, the combination rules used are:

$$A_{ij} = \left( \frac{r_i^*}{2} + \frac{r_j^*}{2} \right)^{12} \sqrt{\varepsilon_i \varepsilon_j} \quad (15)$$

$$B_{ij} = 2 \left( \frac{r_i^*}{2} + \frac{r_j^*}{2} \right)^6 \sqrt{\varepsilon_i \varepsilon_j} \quad (16)$$

where  $r_i^*/2$  is half the minimum energy separation for two atoms of type  $i$ ,  $\varepsilon_i$  is the well depth for two atoms of type  $i$ . These parameters are found in the parameter file.

For the 6-12AtomVDWFormat entry set to SigmaEpsilon, the combination rules used are:

$$A_{ij} = 4(\sigma_i \sigma_j)^6 \sqrt{\varepsilon_i \varepsilon_j} \quad (17)$$

$$B_{ij} = 4(\sigma_i \sigma_j)^3 \sqrt{\varepsilon_i \varepsilon_j} \quad (18)$$

where  $\sigma_i$ , the zero energy separation for two atoms of type  $i$

( $r = \sqrt[6]{2} \sigma_i$ ), and  $\varepsilon_i$ , the well depth for two atoms of type  $i$ , are the parameters from the parameter file.

For the 6-12AtomVDWFormat entry set to SlaterKirkwood, the combination rules used are:

$$B_{ij} = \frac{C\alpha_i\alpha_j}{\sqrt{\frac{\alpha_i}{N_i} + \frac{\alpha_j}{N_j}}} \quad (19)$$

$$A_{ij} = \frac{1}{2}B_{ij}(R_i + R_j)^6 \quad (20)$$

where  $\alpha_i$ , the polarizability of an atom of type  $i$ ,  $N_i$ , the effective number of valence electrons for an atom of type  $i$ , and  $R_i$ , the van der Waals radius of an atom of type  $i$ , are the parameters from the parameter file. The constant  $C$  combines the permittivity of free space ( $\epsilon_0$ ), the charge of an electron ( $e$ ), the mass of an electron ( $m_e$ ) and Planck's constant ( $h$ ):

$$C = \frac{3eh}{4\sqrt{\pi\epsilon_0 m_e}} \quad (21)$$

The form of Lennard-Jones constants specified by pairs of atom types is selected by setting the parameter set entry 6-12PairVDWFormat in the Registry or the chem.ini file to one of RStarEpsilon, SigmaEpsilon or AijBij. This specifies the interpretation of the file, if any, pointed to by the 6-12PairVDW entry in the Registry or the chem.ini file for the same parameter set. Values specified in this fashion override the values generated by single atom type constraints, so they are typically used just to specify parameters for certain cases where the atom combination rules don't give acceptable results.

For the 6-12PairVDWFormat entry set to RStarEpsilon, the values are converted by:

$$A_{ij} = r_{ij}^{*12} \epsilon_{ij} \quad (22)$$

$$B_{ij} = 2r_{ij}^{*6} \epsilon_{ij} \quad (23)$$

where  $r_{ij}^*$ , the minimum energy separation for two atoms of type  $i$  and  $j$ , and  $\epsilon_{ij}$ , the well depth for two atoms of type  $i$  and  $j$ , are the parameters from the parameter file.

For the 6-12PairVDWFormat entry set to SigmaEpsilon, the values are converted by:

$$A_{ij} = 4\sigma_{ij}^{12} \epsilon_{ij} \quad (24)$$

$$B_{ij} = 4\sigma_{ij}^6 \epsilon_{ij} \quad (25)$$

where  $\sigma_{ij}$ , the zero energy separation for two atoms of type  $i$  and  $j$  ( $r = \sqrt[6]{2}\sigma_{ij}$ ), and  $\epsilon_{ij}$ , the well depth for two atoms of type  $i$  and  $j$ , are the parameters from the parameter file.

For the 6-12PairVDWFormat entry set to AijBij, the values in the parameter file are not converted.

Usually, atoms with a 1-4 vicinal relationship or more are considered to be nonbonded. Sometimes, however, only atoms with a 1-5 relationship are considered to be fully nonbonded and the atoms with a 1-4 relationship have scaled down nonbonded interactions or are deleted completely from the nonbonded computations, or different parameters are used.

## Electrostatic

This term describes the classical nonbonded electrostatic interactions of charge distributions.

$$E_{\text{electrostatic}} = \sum_{ij \in \text{electrostatic}} \left[ \frac{q_i q_j}{\epsilon R_{ij}} \right] \quad (26)$$

The above potential describes the monopole-monopole interactions of atomic charges  $q_i$  and  $q_j$  a distance  $R_{ij}$  apart. Normally these charge interactions are computed only for nonbonded atoms and once again the 1-4 interactions might be treated differently from the more normal nonbonded interactions (1-5 relationship or more). The dielectric constant  $\epsilon$  used in the calculation is sometimes scaled or made distance-dependent, as described in the next section.

Electrostatic terms other than the simple charge interactions above are commonly included in molecular mechanics calculations, particularly dipole-dipole interactions. More recently, second-order electrostatic interactions like those describing polarizability have been added to some force fields.

## Hydrogen Bonding

This term is an explicit recognition of the importance of hydrogen bonding to molecular interactions.

$$E_{\text{H-bonded}} = \sum_{ij \in \text{H-bond}} \left[ \frac{C_{ij}}{R_{ij}^{12}} - \frac{D_{ij}}{R_{ij}^{10}} \right] \quad (27)$$

Not all force fields add a specific hydrogen bonding term, as there is evidence from accurate quantum mechanical calculations that hydrogen bonding is accounted for by normal electrostatic interactions. In HyperChem, only the AMBER force field employs an explicit hydrogen bonding term. The 12–10 form was suggested by Pauling but other functional forms for explicit hydrogen bonding have also been used.

## Effective Dielectric Constants

Many molecular mechanics potentials were developed at a time when it was computationally impractical to add large numbers of discrete water molecules to the calculation to simulate the effect of aqueous media. As such, techniques came into place that were intended to take into account the effect of solvent in some fashion. These techniques were difficult to justify physically but they were used nevertheless.

The first modification is to simply scale the dielectric permittivity of free space ( $\epsilon_0$ ) by a scale factor  $D$  to mediate or dampen the long range electrostatic interactions. Its value was often set to be between 1.0 and 78.0, the macroscopic value for water. A value of  $D=2.5$ , so that  $\epsilon = 2.5\epsilon_0$ , was often used in early CHARMM calculations.

Alternatively, a distance dependent dielectric constant is often used in the absence of explicit water molecules. This sets  $\epsilon$  in equation (26) to  $D \epsilon_0 \frac{R_{ij}}{R_0}$ , proportional to the distance between charges.

The electrostatic interaction then uses  $R_{ij}^{-2}$  and does not require the evaluation of a square root. To quote Weiner *et al.*, the developers of AMBER, “A rationale for using a distance-dependent dielectric constant is that it mimics the polarization effect in attractive interactions, with closer interactions weighted more

heavily. Second, it helps compensate for the lack of explicit solvation by implicitly damping longer range charge interactions more than shorter range ones. There is empirical and computational support for such a model, given that solvent (water) is not explicitly included in the calculation.”

## Cutoffs

When dealing with large systems where the number of nonbonded interactions dominate the calculation time, a time-saving approximation introduces a cutoff distance. This neglects nonbonded interactions beyond this distance. While this approximation is particularly difficult to justify for long range coulombic interactions (unit charges separated by 20 Å still contribute 17 kcal/mol to the energy), a cutoff in the range of 8–15 Å is still commonly used.

For the periodic boundary conditions described below, the cutoff distance is fixed by the nearest image approximation to be less than half the smallest box length. With a cutoff any larger, more than nearest images would be included.

When the cutoff is sharp, discontinuities in the forces and resultant loss of conservation of energy in molecular dynamics calculations can result. To minimize edge effects of a cutoff, often the cutoff is implemented with a switching or shifting function to allow the interactions to go smoothly to zero.

The switching function used by HyperChem, called *switch* below, alters the nonbonded energy (van der Waals, hydrogen bond, and electrostatic) in the following way:

$$E(\text{switched}) = E(\text{non-switched}) \times \text{switch}(R_{ij}^2, R_{\text{inner}}^2, R_{\text{outer}}^2) \quad (28)$$

where:

$$\text{switch}(x, a, b) = 1 \quad x \leq a \quad (29)$$

$$\text{switch}(x, a, b) = 0 \quad x \geq b$$

$$\text{switch}(x, a, b) = \frac{(b-x)^2(b+2x-3a)}{(b-a)^3} \quad a < x < b$$

Between the inner radius  $R_{\text{inner}}$  of the switch and the outer radius  $R_{\text{outer}}$  of the switch the interaction goes smoothly to zero.



HyperChem uses as its default an inner radius of 10 Å and an outer radius of 14 Å.

HyperChem can also use a *shifted* nonbonded potential, where the nonbonded energy is modified by:

$$E(\text{shifted}) = E(\text{unshifted})(1 - 2R_{ij}^2/R_{\text{outer}}^2 + R_{ij}^4/R_{\text{outer}}^4) \quad (30)$$

for  $R_{ij}$  less than or equal to  $R_{\text{outer}}$  and  $E(\text{shifted}) = 0$  for  $R_{ij}$  greater than  $R_{\text{outer}}$ . Shifted cutoffs apply to van der Waals, hydrogen bonding and electrostatic interactions, but not to 1–4 interactions with the BIO+ force field.

### Terms Involving Vicinal 1–4 Interactions

Although interactions between vicinal 1–4 atoms are nominally treated as nonbonded interactions, most of the force fields treat these somewhat differently from normal 1–5 and greater nonbonded interactions. HyperChem allows each of these 1–4 nonbonded interactions to be scaled down by a scale factor < 1.0 with AMBER or OPLS. For BIO+ the electrostatic may be scaled and different parameters may be used for 1–4 van der Waals interactions. The AMBER force field, for example, normally uses a scaling factor of 0.5 for both van der Waals and electrostatic interactions.

## MM<sup>+</sup>

The MM+ force field is an extension of MM2 which was developed by Allinger and co-workers [N. L. Allinger, *J. Am. Chem. Soc.*, 99, 8127 (1977), N. L. Allinger and Y. H. Yuh, Quantum Chemistry Program Exchange, Bloomington, Indiana, Program #395, U. Burkert and N. L. Allinger, *Molecular Mechanics*, ACS Monograph 177 (1982).] and is designed primarily for small organic molecules although it is being expanded to peptides [J. Lii et. al., *J. Comp. Chem.* 10, 503 (1989)] and other systems as well [K. B. Lipkowitz, QCPE Bulletin, Indiana University, 12, 1, (Feb., 1992)]. The efforts of the Allinger group have been focused more on very accurate results for certain classes of molecules than on developing a generic (but less accurate) method that can be applied to almost all situations in organic chemistry. Thus, the MM2 force field will fail in many situations where no parameters exist. With the MM+ force field specified, HyperChem first tries to perform a calculation with MM2 (1991) parameters but has a default scheme when there

are no MM2 parameters. A message in the status line and log file indicates that default parameters are being used. These default calculations are described later in this chapter.

HyperChem's MM+ force field uses the latest MM2 (1991) parameters and atom types (provided directly by Dr. Allinger) with the 1977 functional form [N. L. Allinger, *J. Am. Chem. Soc.*, 99, 8127 (1977), N. L. Allinger and Y. H. Yuh, Quantum Chemistry Program Exchange, Bloomington, Indiana, Program #395, U. Burkert and N. L. Allinger, *Molecular Mechanics*, ACS Monograph 177 (1982).] modified to incorporate nonbonded cutoffs (using switched or shifted smoothing), periodic boundary conditions, and the bond stretch term switched from cubic form to quadratic form at long range (to avoid the long range repulsive region of standard MM2).

## Units

Like MM2, MM+ internally uses ergs for energy and reports its force constants (a stretching force constant, for example) in units of millidynes per Ångstrom. Other force fields in HyperChem report their force constants in kcal/mol per Å<sup>2</sup>. In addition, there is a factor of 1/2 in the definition of a stretching force constant when comparing MM+ and other force fields. Thus, it is necessary to be quite careful in comparing force constants between different force fields. In the following discussion (and in HyperChem), every equation uses kcal/mol as the basic unit of energy and appropriate conversion constants are described. For example, a factor of 71.94 must be multiplied to MM+ stretching force constants for comparison with stretching force constants for the other force fields. These details affect the internal operation and the interpretation of the input values in parameter files but HyperChem reports all energetic results in energy units of kcal/mol.

## Bond Stretching

MM2 uses a cubic stretch term:

$$E_{\text{bond}} = 143.88 \sum_{\text{bonds}} \frac{1}{2} K_r (r - r_0)^2 [1 + CS(r - r_0)] \quad (31)$$

The cubic stretch term is a factor CS times the quadratic stretch term. This constant CS can be set to an arbitrary value by an entry

in the Registry or the chem.ini file. The default value for MM2 and MM+ is  $CS = -2.0$ .

However, this form becomes repulsive for  $r - r_0 > 2/3 CS$ , so it is unsuitable for bond lengths significantly larger than equilibrium. In the MM2 program, this is avoided by temporarily setting  $CS$  to zero, reverting to a quadratic form, which leads to discontinuities in the potential surface that are unacceptable for geometry optimization and molecular dynamics. Thus the MM+ force field applies the switching function from equation (29) on page 189 to give

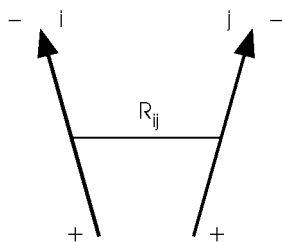
$$E_{\text{bond}} = 143.88 \sum_{\text{bonds}} \frac{1}{2} K_r (r - r_0)^2 \times \left[ 1 + \text{switch} \left( r - r_0, -\frac{1}{3}CS, -\frac{4}{3}CS \right) CS (r - r_0) \right] \quad (32)$$

Inside the point of inflection of equation (31) equation (32) is identical to MM2 with the cubic stretch term turned on. At very long bond distances, it is identical to MM2 with the cubic stretch term turned off.

Two  $r_0$  values are given for each MM+ bond,  $r_0^A$  and  $r_0^B$ . If  $r_0^B$  is available (has a non-zero value in the parameter file) then it is used in preference to the normal  $r_0^A$  for bonds where atom  $i$  and atom  $j$  have at least two hydrogen atoms directly attached to them. For example,  $\text{CH}_2\text{-CH}_2$ ,  $\text{CH}_2\text{-CH}_3$ , or  $\text{CH}_3\text{-CH}_3$  bonds may have their own equilibrium distance differentiated from the average single bond between carbon atoms.

## Bond Dipoles

MM+ calculations do not usually have an electrostatic charge-charge interaction nor define a set of atomic charges for atoms. Instead, the electrostatic contribution comes from defining a set of bond dipole moments associated with polar bonds. These bond moments are defined in the mmpstr.txt(dbf) file along with the bond stretching parameters and are given in units of Debyes. The center of the dipole is defined to be the midpoint of the bond and two dipoles  $\mu_i$  and  $\mu_j$ , separated by  $R_{ij}$ , as shown below:



The MM+ dipole interaction energy is:

$$E_{\text{dipole}} = 14.39418 \epsilon \sum_{ij \in \text{polar bonds}} \mu_i \mu_j \left[ \frac{\cos \chi - 3 \cos \alpha_i \cos \alpha_j}{R_{ij}^3} \right] \quad (33)$$

where  $\epsilon$ , the dielectric constant, is assumed to be larger than 1 (even in the gas phase there is assumed to be some screening of dipole interactions by the rest of the molecule). The MM2 value of 1.5 is used by HyperChem. The angle  $\chi$  is the angle between the two dipole vectors, and  $\alpha_i$  and  $\alpha_j$  are the angles that the two dipole vectors make with the  $\mathbf{R}_{ij}$  vector. The constant 14.39418 converts ergs/molecule to kcal/mol.

## Angle Bending

The quadratic angle bending term in MM+ is identical to that of equation (12) on page 183, apart from a factor 1/2. Three  $\theta_0$  values are given for each MM+ bond,  $\theta^A$ ,  $\theta^B$  and  $\theta^C$ . If  $\theta^B$  is available (has a non-zero value in the parameter file) then it is used in preference to the normal  $\theta^A$ , for angles where the central atom has *one* hydrogen atom directly attached to it. If  $\theta^C$  is available (has a non-zero value in the parameter file) then it is used in preference to the normal  $\theta^A$  for angles where the central atom has *two* hydrogen atoms directly attached to it. If no hydrogen atoms are attached to the central atom or if  $\theta^A$  or  $\theta^B$  values are not available (when they are relevant), the normal  $\theta^A$  is used.

MM+ also includes a sextic angle bending term. The final form for the angle bending energy is:

$$E_{\text{bond angle}} = 0.043828 \sum_{\text{angles}} \frac{1}{2} K_{\theta} (\theta - \theta_0)^2 [1 + \text{SF}(\theta - \theta_0)^4] \quad (34)$$

The sextic bending term is a scale factor SF times the quadratic bending term. This constant SF can be set to an arbitrary value by an entry in the Registry or the chem.ini file. The default value for MM+ is  $\text{SF} = 7.0 \times 10^{-8}$ . The constant 0.043828 converts the MM+ bending constants expressed in millidyne-Å per radian<sup>2</sup> to kcal/mol per degree<sup>2</sup> used by the other force fields. Note that the MM+ expression also includes a factor of 1/2 in the definition that is missing from other force fields.

The MM+ force field uses special values for the bending force constants when the atoms are in a three- or four-membered ring.

### Bond Stretch and Angle Bending Cross Term

Like MM2, MM+ includes coupling between bond stretching and angle bending. If the angle is defined to include atoms i, j, and k, where k is the central atom, then MM+ couples stretching of the ik and jk bonds with the angle:

$$E_{\text{stretch-bend}} = 2.51118 \sum_{\text{angles}} K_{\text{sb}} (\theta - \theta_0)_{\text{ikj}} [(r - r_0)_{\text{ik}} + (r - r_0)_{\text{jk}}] \quad (35)$$

If atom i or atom j is a hydrogen, the deformation  $(r - r_0)$  is considered to be zero. Thus, no stretch-bend interaction is defined for  $\text{XH}_2$  groups. The stretch-bend force constants are incorporated into the program and you cannot modify them. If R is an atom other than hydrogen, the values of the stretch-bend force constants are:

$$K_{\text{sb}} = 0.120 \text{ for } \text{XR}_2$$

X = atom in 1st long row

$$K_{\text{sb}} = 0.090 \text{ for } \text{XRH}$$

or

$$K_{\text{sb}} = 0.250 \text{ for } \text{XR}_2$$

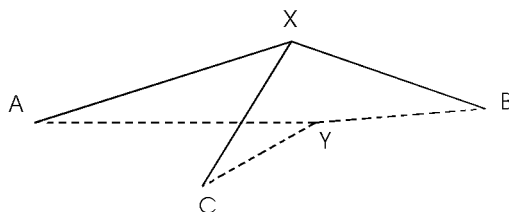
X = atom in 2nd long row

$$K_{sb} = -0.400 \text{ for XRH}$$

The constant 2.51118 converts between MM+ stretch-bend force constants expressed in millidynes per radian and HyperChem's default, kcal per degree.

## Out-of-Plane Bending

An atom that has  $sp^2$  hybridization tends to be coplanar with its attached atoms. This effect is accounted for by improper torsions in other force fields and by out-of-plane-bending interactions in MM+. Consider the following situation involving an atom X that is  $sp^2$  hybridized



Y is the projection of X onto the ABC plane. When the central atom (X) of these angles is  $sp^2$  hybridized, the angle bending calculations are modified to use the in-plane angles AYB, AYC, and BYC in equation (34) on page 194 with the standard force constants defined in `mmpben.txt(dbf)` rather than the standard angles AXB, AXC, and BXC. In addition, out-of-plane components are computed as well, for the out-of-plane angles XAY, XBY, and XCY. These last three calculations use equation (34) as well, but with a  $\theta_0$  equal to 0 and special out-of-plane bending constants,  $K_\theta$ , defined in `mmpoop.txt(dbf)`.

## Dihedrals

The dihedral angle or torsional energy interaction in MM+ is of the general form of equation (12) on page 183 but explicitly includes  $n=1, 2,$  and  $3$  with a phase angle  $\phi = 0$ :

$$E_{\text{dihedral}} = \quad (36)$$

$$\sum_{\text{dihedrals}} \frac{V_1}{2}(1 + \cos\phi) + \frac{V_2}{2}(1 - \cos 2\phi) + \frac{V_3}{2}(1 + \cos 3\phi)$$

The values of  $V_1$ ,  $V_2$ , and  $V_3$ , in kcal/mol, are listed in `mmp-tor.txt(dbf)`. The MM+ force field uses special values for the torsional force constants when the atoms are in a four-member ring.

## van der Waals

The MM+ van der Waals interactions do not use a Lennard-Jones potential but combine an exponential repulsion with an attractive  $1/R^6$  dispersion interaction. The basic parameters are a van der Waals radius  $r_i^*$  for each atom type and a hardness parameter  $\epsilon_i$  that determines the depth of the attractive well and how easy (or difficult) it is to push atoms close together. There are interactions for each nonbonded  $ij$  pair, including all 1–4 pairs. The parameters for a pair are obtained from individual atom parameters as follows:

$$r_{ij}^* = r_i^* + r_j^* \quad (37)$$

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \quad (38)$$

The van der Waals interaction is then calculated as:

$$E_{\text{vanderWaals}} = \sum_{ij \in \text{vdW}} \epsilon_{ij} (2.9 \times 10^5 \exp(-12.5\rho_{ij}) - 2.25\rho_{ij}^{-6}) \quad (39)$$

where

$$\rho_{ij} = \frac{R_{ij}}{r_{ij}^*} \quad (40)$$

At short distances ( $r_{ij} \leq 3.311$ ) the above expression is replaced by:

$$E_{\text{vanderWaals}} = 336.176 \sum_{ij \in \text{vdW}} \epsilon_{ij} \rho_{ij}^{-2} \quad (41)$$

For CH interactions the normal  $\epsilon$  and  $r^*$  values are replaced by special CH values:

$$\epsilon_{\text{CH}} = 0.046 \quad (42)$$

$$r_{\text{CH}}^* = 3.340 \quad (43)$$

For XH bonds, where X is any heavy atom, the hydrogen electron density is not thought to be centered at the position of the hydrogen nucleus but displaced along the bond somewhat, towards X. The MM+ force field reduces the XH bond length by a factor of 0.915 strictly for the purposes of calculating van der Waals interactions with hydrogen atoms.

## AMBER

The AMBER (Assisted Model Building and Energy Refinement) is based on a force field developed for protein and nucleic acid computations by members of the Peter Kollman research group at the University of California, San Francisco. The original AMBER has become one of the more widely used academic force fields and extensive work has gone into developing it — resulting in a number of versions of the method and associated parameters.

HyperChem originally gave results equivalent to Versions 2.0 and 3.0a of the AMBER program distributed by the Kollman group and parameter sets for both these versions are distributed with HyperChem. Subsequently, the ambers parameter set was added for sugars molecules. More recently, the Amber94 and Amber96 parameter sets were added corresponding to a slightly revised form of the Amber force field that eliminated explicit hydrogen bonding terms and included a whole new set of atom types. The current program includes choices of Amber2, Amber3, Ambers, Amber94, and Amber96 parameter sets and associated force field terms.

AMBER was first developed as a united atom force field [S. J. Weiner et al., *J. Am. Chem. Soc.*, 106, 765 (1984)] and later extended to include an all atom version [S. J. Weiner et al., *J. Comp. Chem.*, 7, 230 (1986)]. The newest version of Amber is due to Cornell et. al., *J. Am. Chem. Soc.* 117, 5179-5197 (1995). HyperChem allows the user to switch back and forth between the united atom and all atom force fields as well as to mix the two force fields within the same molecule. Since the force field was developed for macromolecules, there are few atom types and parameters for small organic systems or inorganic systems, and most calculations on such systems with the AMBER force field will lack explicit parameters. HyperChem, however, will choose a default parameter for a missing parameters as per the scheme originally imple-



mented in MM+. That is, default parameters are chosen as described later in this chapter.

## Bond Stretching

The functional form for bond stretching in AMBER is quadratic only and is identical to that shown in equation (11) on page 183.

The bond stretching force constants are in units of kcal/mol per  $\text{\AA}^2$  and are in the file pointed to by the QuadraticStretch entry for the parameter set in the Registry or the chem.ini file, usually called \*str.txt(dbf).

## Angle Bending

The functional form for angle bending in AMBER is quadratic only and is identical with that shown in equation (12) on page 183. The angle bending force constants are in units of kcal/mol per radian<sup>2</sup> and are in the file pointed to by the QuadraticBend entry for the parameter set in the Registry or the chem.ini file, usually called \*ben.txt(dbf).

## Dihedrals

The functional form for dihedral angle (torsional) rotation is identical to that shown in equation (13) on page 183. The barrier heights are in kcal/mol and are in the file pointed to by the FourierTorsion entry for the parameter set in the Registry or the chem.ini file, usually called \*tor.txt(dbf). If more than one term is available for a given dihedral angle in the parameter file, the terms are summed to give the total potential, i.e., the potential can be a Fourier sum over  $n$  in equation (14) where  $n$  is from 1 to 6.

## Improper Dihedrals

The functional form for improper dihedral term is identical to that shown in equation (13) on page 183. The barrier heights are in kcal/mol and are in the file pointed to by the FourierImpTorsion entry for the parameter set in the Registry or the chem.ini file, usually called \*imp.txt(dbf). The central atom in the improper torsion is the second or third atom (i.e. a central atom) in the dihedral angle. For atoms that have the improper flag set (either from templates or editing a HIN file) HyperChem uses the improper dihedral angle formed by neighbor 1 - neighbor 2 - central atom -

neighbor 3, where the order of neighbors is how they appear in a HIN file. Not all planar atoms customarily have associated improper torsions, and the order of atoms is arbitrary but was consistently chosen by the authors of the AMBER force field. The templates contain the standard AMBER definitions of improper torsions for amino and nucleic acids.

## van der Waals

The functional form for van der Waals interactions in AMBER is identical with that shown in equation (13) on page 183. The coefficients  $A_{ij}$  and  $B_{ij}$  are computed from the parameters in the file pointed to by the 6-12AtomVDW entry for the parameter set in the Registry or the chem.ini file, usually called \*nbd.txt(dbf), and optionally with the file pointed to by the 6-12PairVDW entry for the parameter set, usually called \*npr.txt(dbf). The standard AMBER parameter sets use equations (15) and (16) for the combination rules by setting the 6-12AtomVDWFormat entry to RStarEpsilon. The 1-4 van der Waals interactions are usually scaled in AMBER to half their nominal value (a scale factor of 0.5 in the Force Field Options dialog box).

## Electrostatic

The functional form for electrostatic interactions in AMBER is identical with that shown in equation (26) on page 187. You normally use a dielectric scaling of  $D=1$  with AMBER combined with a constant functional form when solvent molecules are explicitly included or a distance dependent form when trying to emulate the effects of solvent without explicitly adding solvent molecules. The charges  $q_i$  are not part of any AMBER parameter file. Standard atomic charges are placed on an atom only when the molecular system is created from templates, by using the database menu or reading a PDB file. Otherwise charges must be set manually or by a quantum mechanical calculation, or they default to zero and no electrostatic interactions result. The 1-4 electrostatic interactions are usually scaled in AMBER to half their nominal value (a scale factor of 0.5 in the Force Field Options dialog box).

## Hydrogen Bonding

The original AMBER force field uses a hydrogen bonding term identical to equation (27) on page 188. The coefficients  $C_{ij}$  and  $D_{ij}$  for appropriate donor-acceptor pairs are given in the file pointed to by the 10-12PairHBond entry for the parameter set in the Registry or the chem.ini file, usually called \*hbd.txt(dbf). If a non-bonded atom pair (excluding 1-4 interactions) has hydrogen bond parameters, the interaction is calculated using the hydrogen bonding term instead of the van der Waals term.

The newer, Cornell et. al. Amber94 and Amber96 versions of Amber do not use a hydrogen bonding term.

## Lone Pairs

The AMBER force field expects lone pairs to be added to all sulfur atoms and computes the interactions as if these lone pairs were atoms with a specific type just like any other atom. The templates automatically add the expected lone pairs to sulfur atoms when using the AMBER force field.

## OPLS

OPLS (Optimized Potentials for Liquid Simulations) is based on a force field developed by the research group of Bill Jorgensen now at Yale University and previously at Purdue University. Like AMBER, the OPLS force field is designed for calculations on proteins and nucleic acids. It introduces nonbonded interaction parameters that have been carefully developed from extensive Monte Carlo liquid simulations of small molecules. These non-bonded interactions have been added to the bonding interactions of AMBER to produce a new force field that is expected to be better than AMBER at describing simulations where the solvent is explicitly included and nonbonded interactions are particularly important.

The HyperChem OPLS force field gives results equivalent to the original OPLS force field.

The OPLS force field is described in two papers, one discussing parameters for proteins [W. L. Jorgensen and J. Tirado-Rives, *J. Amer. Chem. Soc.*, 110, 1657 (1988)] and one discussing parameters for nucleotide bases [J. Pranata, S. Wierschke, and W. L. Jorgensen,

*J. Amer. Chem. Soc.*, 113, 2810 (1991)]. The force field uses the united atom concept for many, but not all, hydrogens attached to carbons to allow faster calculations on macromolecular systems. The amino and nucleic acid residue templates in HyperChem automatically switch to a united atom representation where appropriate when the OPLS option is selected.

The OPLS atom types are a superset of the AMBER united atom types and the bonding parameters are just those of AMBER, supplemented where needed by the OPLS developers. The bond stretch, angle bending, dihedral angle and improper dihedral angle terms are identical to those of AMBER. Unlike AMBER, different combination rules are used for the van der Waals parameters, no hydrogen bonding term is used and no lone pairs are used.

## van der Waals

The van der Waals term in the OPLS force field is that of equation (14) on page 184. The coefficients  $A_{ij}$  and  $B_{ij}$  are computed from the parameters in the file pointed to by the 6-12AtomVDW entry for the parameter set in the Registry or the chem.ini file, usually called \*nbd.txt(dbf), and optionally with the file pointed to by the 6-12PairVDW entry for the parameter set, usually called \*npr.txt(dbf). The standard OPLS parameter sets use equations (17) and (18) on page 185 for the combination rules by setting the 6-12AtomVDWFormat entry to SigmaEpsilon. The 1-4 van der Waals interactions are usually scaled in OPLS to one-eighth their nominal value (a scale factor of 0.125 in the Force Field Options dialog box).

## Electrostatic

The OPLS form of electrostatic interactions is that of equation (26) on page 187. That is, it uses a charge-charge interaction just like AMBER. However, since the nonbonded potentials were developed from liquid studies with explicit water molecules, the use of a distance-dependent dielectric is excluded. The appropriate dielectric scaling factor for OPLS calculations is  $D=1$ . A set of charges are defined for protein and nucleic acid residues in the template (.tpl) files.

## BIO+

The BIO+ force field is an implementation of the CHARMM (Chemistry at HARvard Macromolecular Mechanics) force field developed in the group of Martin Karplus at Harvard University. Like AMBER and OPLS, it is primarily designed to explore macromolecules. It does not use lone pairs.

CHARMM was first developed as a united atom force field and parameters for some amino acids have been published [B. R. Brooks *et al.*, *J. Comp. Chem.*, 4, 187 (1983)]. Subsequent changes to the functional form and parameters were published [W. Reiher, Ph.D., Harvard 1985], but most parameter development remained unpublished and unavailable without purchasing the CHARMM program, i.e. until relatively recently. Subsequent parameter development in the Karplus group at Harvard employed an all-atom representation, using all dihedral angles with no hydrogen bonding term and a Urey-Bradley term [J. C. Smith and M. Karplus, *J. Am. Chem. Soc.* 114, 805-812, 1992]. More recently, full details of the CHARMM force field and parameters have been made available [A.D. MacKerell *et. al.* *J. Phys. Chem B*, 102, 3586] and a web site (<http://www.pharmacy.umaryland.edu/~alex/research.html>) describes the current parameter sets. Thus parameter sets charmm19, charmm22, and charmm27, etc. are now available in HyperChem for an up-to-date implementation of CHARMM. Beginning with the choice of the charmm22 set and beyond, the newer force field without hydrogen bonding terms and with a Urey-Bradley term is used. The HyperChem BIO+ force field thus gives results equivalent to CHARMM using the same CHARMM parameter sets.

### Bond Stretching

The functional form for bond stretching in BIO+, as in CHARMM, is quadratic only and is identical to that shown in equation (11) on page 183. The bond stretching force constants are in units of kcal/mol per  $\text{\AA}^2$  and are in the file pointed to by the Quadratic-Stretch entry for the parameter set in the Registry or the chem.ini file, usually called \*str.txt(dbf).

## Angle Bending

The functional form for angle bending in BIO+ is quadratic only and is identical with that shown in equation (12) on page 183. The angle bending force constants are in units of kcal/mol per radian<sup>2</sup> and are in the file pointed to by the QuadraticBend entry for the parameter set in the Registry or the chem.ini file, usually called \*ben.txt(dbf).

## Dihedrals

The functional form for dihedral angle (torsional) rotation is identical to that shown in equation (13) on page 183. The barrier heights are in kcal/mol and are in the file pointed to by the FourierTorsion entry for the parameter set in the Registry or the chem.ini file, usually called \*tor.txt(dbf). If more than one term is available for a given dihedral angle in the parameter file, the terms are summed to give the total potential, i.e. the potential can be a Fourier sum over  $n$  in equation (13), where  $n$  is from 1 to 6. The term in BIO+ is a generalization of that given for CHARMM in B. R. Brooks *et al.*, *J. Comp. Chem.*, 4, 187 (1983), in that  $\phi_0$  values other than 0° or 180° are allowed. In HyperChem, all bond dihedral angles are included in the energy calculation, equivalent to autogenerating dihedral angles in CHARMM. Thus, parameters developed for use with arbitrary subsets of bond dihedral angles, as is common with CHARMM, need to be scaled or have some parameters set to have zero force constants for use in HyperChem BIO+. Note: An extra column exists in the dihedral parameter file for specifying a factor to divide into the force constant.

## Improper Dihedrals

The functional form for an improper dihedral term is that shown in equation (12) on page 183 except that the summation is over improper torsion angles and  $q$  represents an improper torsion angle. For angles smaller than 6°, with  $q_0$  equal to 0°, a series expansion is used to avoid a numerically indeterminate derivative. The barrier heights are in kcal/mol and are in the file pointed to by the QuadraticImpTorsion entry for the parameter set in the Registry or the chem.ini file, usually called \*imp.txt(dbf). The central atom in the improper torsion is the first or fourth atom (i.e. a terminal atom) in the dihedral angle. For atoms that have the

improper flag set (either from templates or editing a HIN file) HyperChem uses the improper dihedral angle formed by central atom - neighbor 1 - neighbor 2 - neighbor 3, where the order of neighbors is how they appear in a HIN file. Not all planar atoms customarily have associated improper torsions. The order of atoms is arbitrary but has been consistently chosen by the original authors of the CHARMM force field. The templates contain equivalent CHARMM definitions of improper torsions for amino acids. Improper dihedral angles cannot be defined that do not have a central atom, as is sometimes done in CHARMM calculations.

## van der Waals

The functional form for van der Waals interactions in BIO+ is identical with that shown in equation (14) on page 184. The coefficients  $A_{ij}$  and  $B_{ij}$  are computed from the parameters in the file pointed to by the 6-12AtomVDW entry for the parameter set in the Registry or the chem.ini file, usually called \*nbd.txt(dbf), and optionally with the file pointed to by the 6-12PairVDW entry for the parameter set, usually called \*npr.txt(dbf). When using the newer, published CHARMM parameter sets, BIO+ uses equations (15) and (16) for the combination rules by setting the 6-12AtomVDWFormat entry to RStarEpsilon. When using earlier, published CHARMM parameter sets, BIO+ uses equations (19), (20) and (21) by setting the 6-12AtomVDWFormat entry to SlaterKirkwood.

The 1-4 van der Waals interactions cannot be scaled in CHARMM but in newer CHARMM parameter sets some atom types (usually united atoms) use different parameters for 1-4 interactions. These are specified for BIO+ in the file pointed to by the 6-12Atom14VDW entry, usually called \*nbd.txt(dbf). If an atom type is absent in the 6-12Atom14VDW file, the normal parameters are used. The format of the 6-12Atom14VDW file is also specified by the 6-12AtomVDWFormat entry for the parameter set.

Newer, published CHARMM parameter sets override some of the combination rule generated parameters for some atom type pairs. These parameters are found in the file pointed to by the 6-12PairVDW entry for the parameter set, usually called \*npr.txt(dbf). The values of  $A_{ij}$  and  $B_{ij}$  for these are computed using equations (22) and (23) on page 186 by setting the 6-12PairVDWFormat entry to RStarEpsilon.

## Electrostatic

The functional form for electrostatic interactions in CHARMM is identical with that shown in equation (26) on page 187. A dielectric scaling of  $D=2.5$  was common with older, published CHARMM parameter sets but newer sets seem to use a scaling of  $D=1$ , combined with a constant functional form when solvent molecules are explicitly included or a distance-dependent form when trying to emulate the effects of solvent without explicitly adding solvent molecules. The charges  $q_i$  are not part of any published CHARMM parameter file. Standard atomic charges are placed on an atom only when the molecular system is created from templates, by using the database menu or reading a PDB file. Otherwise charges must be set manually or by a quantum mechanical calculation, or they default to zero and no electrostatic interactions result. The 1-4 electrostatic interactions are scaled with newer, published CHARMM parameter sets to 0.4 or 0.5 times their nominal value (a scale factor of 0.4 or 0.5 in the Force Field Options dialog box).

## Hydrogen Bonding

The BIO+ force field option in HyperChem has no hydrogen bonding term. This is consistent with evolution and common use of the CHARMM force field (even the 1983 paper did not use a hydrogen bonding term in its example calculations and mentioned that the functional form used then was unsatisfactory and under review).

## Urey-Bradley

If one selects a parameter set charmm22 or later, a Urey-Bradley term (harmonic repulsion between the terminal atoms of a bond angle) is included in the force field, provided parameters are available.

## Parameter Sets

The concept of a parameter set is an important (but often inconvenient) aspect of molecular mechanics calculations. Molecular mechanics tries to use experimental data to replace *a priori* computation, but in many situations the experimental data is not known and a parameter is missing. Collecting parameters, verification of



their validity, and the relationship of these molecular mechanics parameters to chemical and structural moieties are all important and difficult topics.

A molecular mechanics method in HyperChem is defined by a set of atom types and a functional form for the energy and its derivatives— for example AMBER. For the AMBER method, you may use many different default and user-defined parameter sets. HyperChem comes with default parameter sets, Amber2, Amber3, AmberS, Amber94, and Amber96 in the case of the AMBER method. It is possible, however, to define an arbitrary number of additional parameter sets for the AMBER method (or for any of the other three force field methods). These user-defined parameter sets can be specified in the Parameters dialog box and integrated into HyperChem, in the same way as the default sets. [New parameter sets must now be added to the end of the list of parameter sets and apply only to the latest force fields].

## Text or DBF Form for Parameters

HyperChem provides two forms of parameter sets — an ASCII text form and the \*.dbf database form consistent with dBASE IV and a large number of other database packages. Since the principal difficulty with using molecular mechanics is having or obtaining appropriate parameters, you may want to explore parameter sets as a database in the \*.dbf form. [ With Release 7 of HyperChem dBase parameter sets are being phased out in favor of text files although older versions of dBase parameter sets are still supported].

The text form for parameters uses white space or commas to separate the fields (columns) of the parameter files. They can be read by ordinary text editors, word processors, etc. In the text form, parameters are easy to modify but not easy to compare, study, etc. Many database programs are capable of reading columns of text as a database, however. While spreadsheets are not, per se, databases, they can be useful for examining parameter sets. Microsoft Excel, for example, can read the text form of a parameter file and put the data in a form easily manipulated as a matrix or a database. The text form of parameters are stored, by default only, in \*.txt files.

The dBASE IV (or dBASE III) form of the parameter sets come in \*.dbf files analogous to the \*.txt files. These dbf files are binary files, so you can't read them directly. They are not required for the product and are provided only as an option. In most situations where

there is no extensive use of database operations on parameter sets, the text form is probably more convenient to use. Many spreadsheet programs, including Microsoft Excel, are also capable of reading \*.dbf files.

## Modifying Existing Parameter Sets

The entries in molecular mechanics parameter files are described in the *HyperChem Reference Manual*. Use the following procedure to modify an existing parameter set:

1. Modify, add, or delete a parameter in the appropriate file. Use a text editor for a \*.txt file or a database program for a \*.dbf file.
2. Start HyperChem.
3. Specify the modified parameter set with the Setup/Select Parameter Set command.
4. Select the Setup/Compile Parameters command to recompile the set of \*.txt files or \*.dbf files into a par file, a binary description of all the parameters of a given parameter set.

The modified parameters are then available to the current or future version of HyperChem. For example, the Amber2 set of parameters is described by the amberstr.txt, amberben.txt, etc. set of files. Modifying one or any of these files, saving it, and then, in HyperChem, selecting the Amber2 parameter set and asking for a compilation of parameters produces a new amber2.par file that is used from then on for the Amber2 set of parameters.

## Creating New Parameter Sets

New parameter sets can be created by essentially copying an old parameter set (copying the \*.str.\*, \*.ben.\*, etc. files while changing their root name), describing the new parameter set to HyperChem with its new name, via the Registry, and then modifying, adding, and deleting entries in the appropriate files prior to a compilation to the binary form. The difference between creating a new parameter set and modifying a parameter set, is giving a new name to the parameter set and its associated files and describing that new name in the Registry.

For example, if an Amber02 parameter set was to be added to the AMBER molecular mechanics method, you could simply copy the

files amberstr.txt(dbf), amberben.txt(dbf), etc. to the new files newamstr.txt(dbf), newamben.txt(dbf), etc. and then copy the Amber2 entry (multiple text lines) in the Registry into a new Amber02 entry, changing all the names in the Amber02 entry into the appropriate Amber02 names (Amber2 replaced by Amber02, amberben.txt(dbf) replaced by newamben.txt(dbf), etc.). Finally, the new parameter set should be compiled to amber02.par prior to its use.

## Compiling Parameters

Compilation of parameters makes HyperChem more efficient. It replaces a set of text or dBASE files (for one parameter set) by a single binary \*.par file. The compilation can be performed at any time from within HyperChem by selecting the parameter set (Setup/Select Parameter Set) and requesting a compilation (Setup/Compile Parameters). *If the parameter files are not modified, you do not have to compile them.*

## Registry Setup of Force Field Options

The Registry entry for molecular mechanics parameter sets is best described as it occurs with the default registry. The Registry contains four sections—[mm+], [amber], [bio+], and [opls]—describing the parameter files for each of the four molecular mechanics methods. Each method section contains as many sections as there are parameter sets for that method. Thus, the [amber] section contains the [amber2], [amber3], [ambers], [amber94], and [amber96] sections that describe the amber2, amber3, ambers, amber94, and amber96 parameter sets associated with the amber method. These CustomNames, amber2, amber3, ambers, amber94, and amber96 should first be described to the [amber] section as the custom names you choose for the parameter sets.

The force field section (for example [amber]) contains four entries:

- CurrentFiles describes the index of the current parameter set. CurrentFiles=0 says to use the first parameter set, CurrentFiles=1 says to use the second parameter set, etc.
- AtomTypeMass is set to the name of the file that lists the atom types associated with the force field and their masses (masses are associated with a type here not an atomic number). The file

can have any name but by convention is named, for example, as `ambertyp.txt(dbf)`.

- `FileFormat` describes whether this file is in text or dBASE format. This is, by convention, available in the file names, `*.txt` or `*.dbf`, but the `AtomTypeMass` file name is strictly arbitrary and the file is interpreted on the basis of the `FileFormat` setting.
- `CustomNames` describes the names of the individual parameter sets.

In addition, there are similar entries associated with the newer force field that begins with the Amber94 parameter set.

A section for a particular parameter set, for example, the `[amber2]` section, contains entries in a parameter set section that are as follows: The `FileFormat` entry describes the file format, text or dBASE, for the parameter files. The `Dielectric`, `DielectricScale`, `1-4ElectrostaticScale`, and `1-4vanderWaalsScale` entries describe the default settings of the Molecular Mechanics Options dialog box. The `6-12AtomVDWFormat` and `6-12PairVDWFormat` entries describe the way the Lennard-Jones van der Waals parameters are obtained æ either from SlaterKirkwood, SigmaEpsilon, or RStarEpsilon values listed in the associated nonbonding parameter files. In some situations, Lennard-Jones parameters come from parameters specific to individual *atoms* or to specific *pairs* of atoms as described earlier in the section of the manual that covers generic force fields.

The entries `QuadraticStretch`, `QuadraticBend`, `FourierTorsion`, `FourierImpTorsion`, `QuadraticImpTorsion`, `6-12AtomVDW`, `6-12PairVDW`, `6-12Atom14VDW`, and `10-12PairHBond` describe the names of the files holding the associated parameters.

In MM+, the parameter files differ sufficiently that they are described by their own entries, `MMPStretch`, `MMPBend`, `MMPBend3`, `MMPBend4`, `MMPOOPBend`, `MMPTorsion`, `MMPTorsion4`, `MMPAtomVDW` and `MMPCubicStretch`. `MMPBend3` and `MMPBend4` refer to bending constants in 3- and 4-membered rings. The `OOPBend` constants describe out-of-plane bending.

The reference manual gives a more complete description of parameter section of the HyperChem registry.

## Periodic Boundary Conditions and Solvent

Isolated gas phase molecules are the simplest to treat computationally. Much, if not most, chemistry takes place in the liquid or solid state, however. To treat these condensed phases, you must simulate continuous, constant density, macroscopic conditions. The usual approach is to invoke periodic boundary conditions. These simulate a large system (order of  $10^{23}$  molecules) as a continuous replication in all directions of a small box. Only the molecules in the single small box are simulated and the other boxes are just copies of the single box.

HyperChem allows solvation of arbitrary solutes (including no solute) in water, to simulate aqueous systems. HyperChem uses only rectangular boxes and applies periodic boundary conditions to the central box to simulate a constant-density large system. The solvent water molecules come from a pre-equilibrated box of water. The solute is properly immersed and aligned in the box and then water molecules closer than some prescribed distance are omitted. You can also put a group of non-aqueous molecules into a periodic box.

It is also possible to simulate liquid droplets by surrounding a solute by a finite number of water molecules and performing the simulation without a periodic box. The water, of course, eventually evaporates and moves away from the solute when periodic boundary conditions are not imposed. If the water is initially added via periodic boundary conditions, you must edit the resulting HIN file to remove the periodic boundary conditions, if a droplet approach is desired.

Periodic boundary conditions can also be used to simulate solid state conditions although HyperChem has few specific tools to assist in setting up specific crystal symmetry space groups. The group operations Invert, Reflect, and Rotate can, however, be used to set up a unit cell manually, provided it is rectangular.

### The Periodic Boundary Conditions

When you apply periodic boundary conditions to molecular mechanics simulations, the atoms of the system are placed in a rectangular box and treated as if they are surrounded by identically translated images of the box on all sides. A box of dimensions

$R_x \times R_y \times R_z$  has 26 nearest images of itself, translated by combinations of  $\pm R_x$ ,  $\pm R_y$  and  $\pm R_z$ . For crystalline systems, further layers of periodic images might be included, to account for long-range periodicity, and an atom would interact with *all* its periodic images. For liquid-phase systems periodic boundary conditions minimize edge effects of a necessarily finite system, rather than include long-range periodicity. HyperChem employs the usual *nearest image* or *minimum image* restriction, where an atom in the box only interacts with the closest image of another atom in the box. Thus with periodic boundary conditions, an atom only interacts with the set of images that are within  $\pm 1/2 R_x$ ,  $\pm 1/2 R_y$  and  $\pm 1/2 R_z$  of itself.

The nearest image restriction implies a sharp, rectangular cutoff of  $\pm 1/2 R_x$ ,  $\pm 1/2 R_y$  and  $\pm 1/2 R_z$ . This leads to discontinuities in the potential energy surface; if an atom moves too far away on one side it suddenly disappears and reappears on the other side (i.e. a different image becomes closest). The discontinuity at the switch-over point can lead to undesirable artifacts in geometry optimization and molecular dynamics. Another undesirable feature of this implied rectangular cutoff is that it is anisotropic, so that a rotation of the coordinate system changes the energy.

HyperChem avoids the discontinuity and anisotropy problem of the implied cutoff by imposing a smoothed spherical cutoff within the implied cutoff. When a system is placed in a periodic box, a switched cutoff is automatically added. The default outer radius, where the interaction is completely turned off, is the smallest of  $1/2 R_x$ ,  $1/2 R_y$  and  $1/2 R_z$ , so that the cutoff avoids discontinuities and is isotropic. This cutoff may be turned off or modified in the Molecular Mechanics Options dialog box after solvation and before calculation.

## Equilibrated Water Box

The water molecules used to solvate a solute comes from Jorgensen's Monte Carlo equilibrated box of 216 water molecules, described by the TIP3P potential function [W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.* 79, 926 (1983)]. This box is cubic and 18.70 Å on a side. When the requested box fits into this, it is just carved out of

the basic Jorgensen box. When a bigger box (more than 216 water molecules or perhaps an elongated box) is required, then the basic 216 molecule box is duplicated to create 3 x 3 x 3 x 216 water molecules in a box 56.10 Å on a side and the required box is carved out of this.

## Details of Solvation Methodology

Prior to solvation, the solute is oriented according to its inertial axes such that the box size needed to accommodate it is minimized (minimizing the number of water molecules). The principal inertial axis is oriented along the viewer's Z axis, for example. Then water molecules are eliminated if any of the three atoms are closer to a solute atom than the contact distance you specify.

## Restraints

You can add restraints to any molecular mechanics calculation (single point, optimization or dynamics). These might be NMR restraints, for example, or any situation where a length, angle, or torsion is known or pre-defined. Restraints with large force constants result in high frequency components in a molecular dynamics calculation and can result in instability under some circumstances.

### Definition of a Restraint

A restraint (not to be confused with a Model Builder constraint) is a user-specified one-atom tether, two-atom stretch, three-atom bend, or four-atom torsional interaction to add to the list of molecular mechanics interactions computed for a molecule. These added interactions are treated no differently from any other stretch, bend, or torsion, except that they employ a quadratic functional form. They replace no interaction, only add to the computed interactions.

Restraints are commonly used to restrain certain long range conformations. For example, you might restrain a certain end of a large molecule to be near another end. Alternatively, if the added force constant associated with the restraint is large enough, a local geometry, such as a bond length, bond angle, or torsion angle, could be almost forced to remain fixed. These restraints do not

actually restrain any specific geometric feature to remain constant but only make the energy rise when the geometric feature deviates from its restrained value. On the other hand, the Model Builder constraints are true constraints, normally fixing a geometric feature of a Model Built structure.

## Adding Restraints

Use Named Selections to add restraints:

1. Name any one-atom, two-atom, three-atom, or four-atom selection with the Select/Select command.
2. Use Restraints in the Setup menu to bring up the Restraint Forces dialog box
3. Add any of the currently named 1-, 2-, 3-, or 4-atom selections to the list of restraints.

If it is a two atom restraint, the dialog box requests a restrained 2-atom distance (bond length or nonbonded distance) and a stretching force constant. For a one atom restraint, the dialog box request a point to tether the atom to and a force constant. If the force constant is small, there is only a tendency towards keeping the two atoms at the specified distance. If the force constant is large, the distance remains near the specified distance since it requires a large energy to displace it significantly from that distance. The two atoms need not be bonded and in many restraint situations the two atoms definitely will not be bonded.

The 3-atom and 4-atom restraints behave just like those for two atoms, except that the dialog box requests an angle rather than a distance and the accompanying request for a force constant requires either a bending force constant (3-atom case) or a torsional force constant (4-atom case).

These molecular dynamics restraints are stored with the HIN file and are retained as long as the Named Selections are still active (structural changes not made to molecular system) or the restraints are still requested via the Restraint Forces dialog box.

The default restraints are appropriate for molecular dynamics calculations where larger force constants would create undesirable high frequency motions but much larger force constants may be desired for restrained geometry optimization.



## The Default (MM+) Force Field

A default force field was originally developed only for MM+. With more recent versions of HyperChem the scheme used for MM+ has been extended to provide default parameters for all the force fields in essentially an identical way to that used for MM+. It is still referred to here as the MM+ default force field and described *sp* below although it is now also used with Amber, OPLS, and CHARMM.

One of the major difficulties with molecular mechanics procedures (MM+ or otherwise) is that they almost always fail. That is, you find that force constants are not available for the molecule of interest. This is both the strength and weakness of molecular mechanics; it uses atom types to introduce specific chemical environments for the atoms within a molecule (to obtain accuracy in the calculations) but then requires knowledge of force constants specific to that chemical environment (as specific as stating that an atom is in a five-member ring containing one oxygen and one carbon, for example). As the number,  $N$ , of atom types rises the number of force constants needed to describe all possible occurrences of these atom type becomes very large. For torsions, for example,  $N^4$  force constants are needed. This does not even include the very real possibility that an atom type has not been defined for the particular chemical situation at hand.

In computational chemistry it can be very useful to have a generic model that you can apply to any situation. Even if less accurate, such a computational tool is very useful for comparing results between molecules and certainly lowers the level of pain in using a model from one that almost always fails. The MM+ force field is meant to apply to general organic chemistry more than the other force fields of HyperChem, which really focus on proteins and nucleic acids. HyperChem includes a default scheme such that when MM+ fails to find a force constant (more generally, force field parameter), HyperChem substitutes a default value. This occurs universally with the periodic table so all conceivable molecules will allow computations. Whether or not the results of such a calculation are realistic can only be determined by close examination of the default parameters and the particular molecular situation.

## The Wild Card Approach

The approach used to develop default parameters is best described using torsional parameters as an example. A general torsional rotation involves four connected atoms, atom<sub>A</sub>-atom<sub>X</sub>-atom<sub>Y</sub>-atom<sub>B</sub>, where atom<sub>A</sub> and atom<sub>B</sub> are the atoms at the ends of the torsion and atom<sub>X</sub> and atom<sub>Y</sub> are the two central atoms. Atom<sub>A</sub> is atom type A, for example. The parameters for these torsions are described by the atom types, A-X-Y-B in the torsional parameter file with the default name `mmptor.txt(dbf)`. If you have a molecule with a torsion of the type A-X-Y-B, HyperChem searches the `mmp.par` binary equivalent of `mmptor.txt(dbf)` and if there is a match, extracts and uses the parameters for the torsion.

When no explicitly correct force constant is found, HyperChem proceeds to stage two and finally to stage three. In stage two, you can use wildcards to relax the explicitness of the match between the atom types of the torsion in question and the available MM+ parameters. In the torsional case, as many as three searches of the `mmp.par` file are performed. If the exact match between A-X-Y-B and entries in `mmp.par` fails, then a search in `mmp.par` looks for an entry labeled, `**_X-Y_**`, where `**` is the designation for wildcard, don't-care, any-atom-type, unknown, etc. This search looks for a torsional parameter associated only with the central atoms of the torsion independent of the terminal vicinal atoms. If this fails, a third search looks for an entry in the `mmp.par` file that is labeled `**_**_**_**`. If such an entry exists, it indicates that this is the force constant (and additional torsional parameters) to use when no better match is available. Only the above three searches of `mmp.par` are made with no intermediate searches such as `**_X-Y-B` or `**_X-**_**`, etc. If the `**_**_**_**` entry for the completely generic torsion exists in `mmp.par`, then the search for torsional parameters for the A-X-Y-B torsion will obviously succeed at this point, if not before. If no `**_**_**_**` entry exists, and the search of `mmp.par` fails to find appropriate parameters, then HyperChem proceeds to stage three where default parameters are assigned based on perceived hybridization states of X and Y (and possibly A or B), the type of bond between X and Y etc. This last stage generally yields sets of parameters that are more reasonable than ones based on a single unique generic value, `**_**_**_**`, to be used everywhere that more explicit parameters are not available from `mmp.par`.

## The Default Force Field

If suitable parameters cannot be found in the MMP.PAR file, with or without wild cards, i.e. stage one and stage two fail, then the final stage three is entered and HyperChem uses a unique default scheme to find a parameter. At this point the HyperChem status line notes that default parameters are being used. This default scheme is similar to the Dreiding force field proposed by Mayo, Olafson, and Goddard [*J. Phys. Chem.* 94, 8897 (1990)], but has enough differences that it needs to be described in its own right. Specifically, the default scheme retains the MM2 functional form exactly as proposed by the Allinger group, but uses a scheme similar to that proposed for the Dreiding force field to obtain parameters. The parameters are obtained using a generic approach that considers only the hybridization of relevant atoms, the type of a bond (single, double, triple, or aromatic), and standard covalent radii for all elements in the periodic table. This scheme may not yield an optimized parameter but it does not require so specific a criterion as the explicit listing of a parameter in a parameter file associated with specific atom types.

Covalent radii for all the elements are readily available and the bond orders of all bonds are available from the molecular graph. Prior to describing the explicit default parameter scheme, it is necessary to indicate how a hybridization state is assigned to any atom in a molecule.

### Approximate Hybridization

An estimate of the hybridization state of an atom in a molecule can be obtained from the group of the periodic table that the atom resides in (which describes the number of valence electrons) and the connectivity (coordination of the atom). The HyperChem default scheme uses this estimate to assign a hybridization state to all atoms from the set (null, s, sp,  $sp^2$ ,  $sp^{2.5}$ , and  $sp^3$ ). The special value  $sp^{2.5}$  is used to denote aromatic systems such as benzene. In this release of HyperChem, hybridization states involving d orbitals are not considered. For the transition metals (or lanthanides and actinides) the number of valence electrons is only considered to be those normally residing in (ns,np) orbitals. This is, of course, an approximation that may not be justified in all cases but is the basis of the current default scheme where a hybridization is assigned based on the number of s and p valence electrons (0–8)

and the number of neighbors (0–4). The number of valence electrons assigned to each element is shown below:

1																		8
1	2												3	4	5	6	7	8
1	2												3	4	5	6	7	8
1	2	3	4	5	3	2	3	2	2	2	2	2	3	4	5	6	7	8
1	2	3	4	5	2	7	3	3	2	1	2	3	4	5	6	7	8	
1	2	3	4	5	6	7	4	4	4	3	2	3	4	5	6	7	8	
1	2	3																
			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4
			4	5	6	5	4	3	3	3	3	3	3	3	3	3	3	3

The number of neighbors is given by the molecular graph and the following rules determine a hybridization state for each atom in a molecule.

1. If the number of valence electrons is 0 or 8, the hybridization is described as null. This is used for Group 0 (Inert gases).
2. If the number of valence electrons is 1 (H, Na, K, etc.), the hybridization is described as s.
3. If the number of valence electrons is 2 (Be, Mg, etc.), the hybridization is set to  $sp^3$  for 4 neighbors,  $sp^2$  for 3 neighbors, and sp for 1 or 2 neighbors.
4. If the number of valence electrons is 3 (B, Al, etc.), the hybridization is set to  $sp^3$  for 4 neighbors,  $sp^2$  for three neighbors, and sp for 1 or 2 neighbors.
5. If the number of valence electrons is 4 (C, Si, etc.), the hybridization is set to  $sp^3$  for 4 neighbors.
6. If there are 3 neighbors and one or more of them is connected by an aromatic bond, then the hybridization is set to  $sp^{2.5}$  or the hybridization is set to  $sp^2$ .
7. If there are 2 neighbors and both bonds are single bonds then the hybridization is set to  $sp^3$ , or if one of the bonds is a single

- bond the hybridization is set to  $sp^2$ , or if neither of the bonds is single the hybridization is set to  $sp$ .
8. If there is 1 neighbor, then if the one bond is a triple bond the hybridization is set to  $sp$ , or if the one bond is an aromatic bond the hybridization is set to  $sp^{2.5}$ , or if the one bond is double the hybridization is set to  $sp^2$ , or if the one bond is single the hybridization is set to  $sp^3$ .
  9. If the number of valence electrons is 5 (N, P, etc.), the hybridization is set to  $sp^3$  for 4 or 3 neighbors, unless it is attached to an atom with a triple, double, or aromatic bond, in which case it is set to  $sp^2$ .
  10. If there are 2 neighbors and one or more of them is connected by an aromatic bond then the hybridization is set to  $sp^{2.5}$  or the hybridization is set to  $sp^2$ .
  11. If there is 1 neighbor, then if the one bond is triple the hybridization is set to  $sp$ , or if the one bond is aromatic the hybridization is set to  $sp^{2.5}$ , or if the one bond is double the hybridization is set to  $sp^2$ , or if the one bond is single the hybridization is set to  $sp^3$ .
  12. If the number of electrons is 6 (O, S, etc.), the hybridization is set to  $sp^3$  for 4, 3, or 2 neighbors, unless it is attached to an atom with a triple, double, or aromatic bond, in which case it is set to  $sp^2$ . If these atoms have 1 neighbor, and if the one bond is triple the hybridization is set to  $sp$ , or if the one bond is aromatic the hybridization is set to  $sp^{2.5}$ , or if the one bond is double the hybridization is set to  $sp^2$ , or if the one bond is single the hybridization is set to  $sp^3$ .
  13. If the number of valence electrons is 7 (Halides, etc.), the hybridization is described as s.

These rules may not be optimal when applied to certain transition metals (Lanthanides, etc.), but they generate a result for *any* molecule.

## Bond Stretching

The default parameters for bond stretching are an equilibrium bond length and a stretching force constant. The functional form is just that of the MM+ force field including a correction for cubic stretches. The default force constant depends only on the bond



Atom	Covalent Radius
O (sp <sup>2.5</sup> )	0.660
O (sp <sup>2</sup> )	0.560
O (sp)	0.528

## Parameters for Butadiene and Related Molecules

The normal molecular mechanics approach, as used with the AMBER, MM2, Dreiding or similar force fields, assigns bond length parameters on the basis of the “atom types” of the two bonded atoms. This results in the central sp<sup>2</sup>-sp<sup>2</sup> single bond of a molecule like butadiene having the same bond length parameters as each of the two terminal sp<sup>2</sup>-sp<sup>2</sup> double bonds in the same molecule. A single bond should, of course, be considerably longer than a double bond. This effect can be corrected by assigning the bond length parameter on the basis of “bond order” in addition to the “atom types” of the bonded atoms. The default MM+ force field applies such a correction, but the correction is not described in the manuals.

The correction is only applied in the default MM+ parameter scheme. When MM+ parameters are available in the mmp-str.txt(dbf) file, the normal approach based on only atom types is used. Thus, the correction is applied to butadiene when all the atom types are \*\*, for example, but not when atom types are present and their corresponding parameters available in the parameter files.

MM+ increases the bond length parameter for single bonds that are not sp<sup>3</sup>-sp<sup>3</sup> hybridized. This correction is a sum of two terms, one for each of the bonded atoms. The correction is 0.03 Ångstroms for sp<sup>2.5</sup> (aromatic) hybridization, 0.06 Ångstroms for sp<sup>2</sup> hybridization, and 0.09 Ångstroms for sp hybridization. Without this correction the bond length parameter for the central sp<sup>2</sup>-sp<sup>2</sup> single bond of butadiene, for example, would be 1.34 Ångstroms, the same as for the terminal double bonds. The correction adds 0.12 Ångstroms (0.06 + 0.06) to this, leading to a bond-length parameter for the central single bond of 1.46 Ångstroms, a considerably more accurate value.

## Angle Bending

All default bending force constants are assigned the value 100 kcal/mol/rad<sup>2</sup>

## Stretch-Bends

No default stretch-bending interactions are included.

## Out-Of-Plane Bends

The MM+ force field assigns default values for out of plane bending terms around an sp<sup>2</sup> center. If a central atom has some out of plane parameters, then the first out of plane parameter involving that central atom is used if a specific parameter is not found.

## Torsions

The functional form for default torsions is the MM+ form with three torsional constants V1, V2, and V3 for 1-fold, 2-fold, and 3-fold contributions. The default values for these constants depend on the particular chemical situation associated with the bond order of the bond and the hybridization of the relevant atoms. The rules are:

### Single Bonds

1. An ordinary sp<sup>3</sup>-sp<sup>3</sup> single bond has a three-fold barrier described by V3=2.0 kcal/mol.
2. An sp<sup>3</sup>-sp<sup>3</sup> single bond where each of the central atoms is in Group VIA (for example, hydrogen peroxide) has a two-fold barrier with optimum torsional angle of 90 degrees, as described by V2=-2.0 kcal/mol.
3. An sp<sup>3</sup>-sp<sup>2</sup> or sp<sup>3</sup>-sp<sup>2.5</sup> single bond where the sp<sup>3</sup> atom is in Group VIA (for example, the C-O bond of vinyl alcohol) has a two fold barrier with an optimal planar configuration described by V2=+2.0 kcal/mol.
4. An sp<sup>3</sup>-sp<sup>2</sup> or sp<sup>3</sup>-sp<sup>2.5</sup> single bond where the atom connected to the central sp<sup>2</sup> (sp<sup>2.5</sup>) atom is another sp<sup>2</sup> (sp<sup>2.5</sup>) atom, as in the H-C-C-double bond O torsion of acetic acid, is described by the MM+ parameters of acetic acid, V1=-0.167 kcal/mol and V3=-0.1 kcal/mol.



5. An  $sp^3-sp^2$  or  $sp^3-sp^{2.5}$  single bond where the atom connected to the central  $sp^2$  ( $sp^{2.5}$ ) atom is *not* another  $sp^2$  ( $sp^{2.5}$ ) atom, as in the H-C-C-H torsion of propene, is described by a three-fold barrier  $V3=2.0$  kcal/mol.
6. A planar  $sp^2-sp^2$  single bond or planar  $sp^2-sp^{2.5}$  single bond (for example, the central bond of butadiene, with isolated double bonds, or phenyl amine, where the nitrogen is  $sp^2$  hybridized) is described by a two-fold barrier,  $V2=5$  kcal/mol.
7. A conjugated  $sp^{2.5}-sp^{2.5}$  single bond (for example, the bond joining the two phenyl rings of biphenyl, the central bond of butadiene, with delocalized aromatic bonds, or phenyl amine, where N-C bond is labeled aromatic and nitrogen is  $sp^{2.5}$  hybridized) is described by a two-fold barrier,  $V2=10$  kcal/mol.

### Double Bonds

1. Isolated double bonds (as in ethylene) have a two-fold barrier of  $V2=45$  kcal/mol.
2. Aromatic bonds (as in benzene) have a two-fold barrier of  $V2=25$  kcal/mol.

In each of the cases above the total barrier heights are divided by the total number of torsions counted. For example, ethane uses a parameter  $V3=2.0/6$  for each of its six torsions, leading to a total barrier of 2.0 kcal/mol.

### Electrostatic

No default electrostatic interactions are included.

### van der Waals

The default exp-6 van der Waals interaction requires, as shown in equation (37) and (38) on page 196, a van der Waals radii,  $r^*$ , and a hardness parameter  $\epsilon$ . The default values for these parameters are based strictly on the atomic number and are given below.

The values of  $2r^*$  are:



**These values are from the same references as 2r\*. If a value is not available, a default value of 0.1 kcal/mol is used.**

# Chapter 12

## Quantum Mechanics

In addition to molecular mechanics calculations, HyperChem can perform various quantum mechanical calculations. These calculations are more universal than molecular mechanics, but are also more time consuming and less empirical.

Molecular mechanics depends on the concept of atom types and parameters associated with these atom types. Since the number of atom types is very large for the universe of possible molecules, parameters will probably be missing for a random new molecule unless a force field has been developed for molecules similar to the new molecule. Molecular mechanics predicts how the new molecule will behave based upon the behavior of known, similar molecules.

Quantum mechanical calculations generally have only *one carbon atom type*, compared with the *many types of carbon atoms* associated with a molecular mechanics force field like AMBER. Therefore, the number of quantum mechanics parameters needed for all possible molecules is much smaller. In principle, very accurate quantum mechanical calculations need no parameters at all, except fundamental constants such as the speed of light, etc.

HyperChem quantum mechanical calculations are referred to as either *ab initio*, DFT(density functional theory), or semi-empirical. *Ab initio* calculations only have parameters associated with the *basis set* of atomic orbitals (contracted basis functions) used, in linear combination, to describe the molecular orbitals. These basis sets are usually associated with shells, such as an *s* shell, *sp* shell, etc. The basis set is the only parameter in *ab initio* calculations. Density Functional Theory (DFT) calculations are similar to *ab initio* calculations but have additional parameters associated the *density functional*. The density functional attempts to describes how the energy of a molecule depends on the electron density of the molecule. These DFT calculations can give better results than *ab*

*initio* calculations but depend on having a good density functional. The principal variable in DFT calculations is knowing which density functional to use. Semi-empirical calculations use many explicit parameters associated with the specific atomic number (element) of an atom. The concept of atom *types* is not used in the conventional quantum mechanics methods. Semi-empirical quantum mechanics methods use a rigorous quantum mechanical formulation combined with the use of empirical parameters obtained from comparison with experiment. If parameters are available for the atoms of a given molecule, then *ab initio*, DFT, and semi-empirical calculations have an *a priori* aspect when compared with a molecular mechanics calculation, letting you explore new molecular systems beyond those for which the parameters were generated originally.

HyperChem include the commonly used basis sets. These basis sets are STO-nG and STO-nG\* with  $n = 1, 2, 3, 4, 5,$  and  $6$  for all the atoms with atomic numbers less than or equal to  $54$  (H - Xe) [W. J. Hehre, R. F. Stewart, and J. A. Pople, *J. Chem. Phys.*, **51**, 2657 (1969); J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, *J. Chem. Phys.*, **64**, 5142 (1976); R. F. Stewart, *J. Chem. Phys.*, **52**, 431 (1970)], 3-21G, 3-21G\*, 3-21G\*\*, 6-21G, 6-21G\*, and 6-21G\*\* for the elements H to Ar [J. S. Binkley, J. A. Pople, and W. J. Hehre, *J. Am. Chem. Soc.*, **102**, 939 (1980); M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre, *J. Am. Chem. Soc.*, **104**, 2797 (1982); W. J. Pietro, M. M. Francl, W. J. Hehre, D. J. Defrees, J. A. Pople, and J. S. Binkley, *J. Am. Chem. Soc.*, **104**, 5039 (1982)], 4-21G, 4-21G\*, and 4-21G\*\*, 4-31G, 4-31G\*, 4-31G\*\*, 6-31G, 6-31G\*, and 6-31G\*\* [W. J. Hehre, R. Ditchfield, and J. A. Pople, *J. Chem. Phys.*, **56**, 2257 (1972); P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta.*, **28**, 213 (1973); M. S. Gordon, *Chem. Phys. Lett.*, **76**, 163 (1980)], 6-311G, 6-311G\*, 6-311G\*\*, MC-311G, MC-311G\*, and MC-311G\*\* for the elements with atomic numbers less than or equal to  $18$  (H to Ar) [R. Krishnan, J. S. Kinkley, R. Seeger, and J. A. Pople, *J. Chem. Phys.*, **72**, 650 (1980); A. D. McLean and G. S. Chandler, *J. Chem. Phys.*, **72**, 5639 (1980), and D95, D95\*, and D95\*\* for the elements H to Cl [T. H. Dunning and P. J. Hay, in *Modern Theoretical Chemistry*, Plenum, New York, 1976.

HyperChem includes semi-empirical parameters for all the atoms with atomic numbers less than or equal to  $54$  and can perform calculations on any molecule composed of only those atoms. The few restrictions or exceptions to this rule will be pointed out as appropriate. Nevertheless, the probability of having appropriate param-

eters for a random calculation are much higher for semi-empirical calculations than for molecular mechanics calculations.

*Note:* This chapter explicitly describes *ab initio* calculations in addition to semi-empirical calculations. Almost everything associated with *ab initio* calculations is also relevant to the calculations of density functional theory (DFT). However, DFT calculations are explicitly described in the next chapter by themselves. Where and when it is necessary to distinguish details between the two types of calculations it will be noted. Often DFT calculations are also referred to as *ab initio* although that terminology is not used by the HyperChem manuals which explicitly distinguish *ab initio* from DFT.

## Background

This section describes the basics of HyperChem's simple molecular orbital calculations. It interprets HyperChem's results rather than introducing quantum chemistry. For a complete discussion, you should consult textbooks in quantum chemistry, such as I. N. Levine, *Quantum Chemistry*, 3rd Edition, Allyn and Bacon, 1986; and A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry*, McGraw-Hill Publishing Co., 1989.

## Approximate Solutions of the Schrödinger equation

The Schrödinger equation contains the essence of all chemistry. To quote Dirac: "The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known." [P.A.M. Dirac, *Proc. Roy. Soc. (London)* 123, 714 (1929)]. The Schrödinger equation is

$$\mathcal{H} \Psi = E \Psi \quad (44)$$

The input, the Hamiltonian  $\mathcal{H}$ , describes the particles of the system; the output,  $E$ , is the total energy of the system; and the wave function,  $\Psi$ , constitutes all we can know and learn about the particular molecular system represented by  $\mathcal{H}$ .

For small molecules, the accuracy of solutions to the Schrödinger equation competes with the accuracy of experimental results. However, these accurate *ab initio* or DFT calculations require enormous computation and are only suitable for the molecular systems

with small or medium size. *Ab initio* or DFT calculations for very large molecules are beyond the realm of current computers, so HyperChem also supports semi-empirical quantum mechanics methods. Semi-empirical approximate solutions are appropriate and allow extensive chemical exploration. The inaccuracy of the approximations made in semi-empirical methods is offset to a degree by recourse to experimental data in defining the parameters of the method. Indeed, semi-empirical methods can sometimes be more accurate than some poorer *ab initio* methods, which require much longer computation times.

In making certain mathematical approximations to the Schrödinger equation, we can equate derived terms directly to experiment and replace difficult-to-calculate mathematical expressions with experimental values. In other situations, we introduce a parameter for a mathematical expression and derive values for that parameter by fitting the results of globally calculated results to experiment. Quantum chemistry has developed two groups of researchers:

- Those who fairly successfully obtain improved *a priori* solutions, but are restricted to relatively small molecules.
- Those who want to explore chemistry widely with tools that are not always reliable, but which suggest many chemically interesting possibilities.

Obviously, the *ab initio* or DFT method in HyperChem is suitable for the former and the semi-empirical methods are more appropriate for the latter.

## Charge and Multiplicity

Once you make a set of approximations and introduce a set of parameters, you have an *a priori* method (a model of chemistry) for exploring arbitrary molecular systems. The input to these calculations is relatively straightforward: it consists of describing the number of nuclei, atomic number and position or configuration of the nuclei, and indicating how many electrons are to be added to the system of nuclei. This last input essentially describes the total charge of the system. For example, in a singly charged anion, there is one more electron than there is total nuclear charge; thus the charge on the system is  $-1$ . The charge of the system is a principal input quantity.

Secondly, you must describe the electron spin state of the system to be calculated. Electrons with their individual spins of  $s_i=1/2$  can combine in various ways to lead to a state of given total spin. The second input quantity needed is a description of the total spin  $\mathbf{S}=\sum_i \mathbf{s}_i$ . Since spin is a vector, there are various ways of combining individual spins, but the net result is that a molecule can have spin  $S$  of 0, 1/2, 1, .... These states have a *multiplicity* of  $2S+1 = 1, 2, 3, \dots$ , that is, there is only one way of orienting a spin of 0, two ways of orienting a spin of 1/2, three ways of orienting a spin of 1, and so on.

To define the state you want to calculate, you must specify the multiplicity. A system with an even number of electrons usually has a closed-shell ground state with a multiplicity of 1 (a singlet). A system with an odd number of electrons (free radical) usually has a multiplicity of 2 (a doublet). The first excited state of a system with an even number of electrons usually has a multiplicity of 3 (a triplet). The states of a given multiplicity have a spectrum of states—the lowest state of the given multiplicity, the next lowest state of the given multiplicity, and so on.

## Independent Electron Methods

The simplest approximation to the Schrödinger equation is an independent-electron approximation, such as the Hückel method for  $\pi$ -electron systems, developed by E. Hückel. Later, others, principally Roald Hoffmann of Cornell University, extended the Hückel approximations to arbitrary systems having both  $\pi$  and  $\sigma$  electrons—the Extended Hückel Theory (EHT) approximation. This chapter describes some of the basics of molecular orbital theory with a view to later explaining the specifics of HyperChem EHT calculations.

### The Independent Electron Approximation

Within the Born-Oppenheimer approximation discussed earlier, you can solve an electronic Schrödinger equation

$$\mathcal{H}_{\text{elec}} \Psi_{\text{elec}} = E_{\text{elec}} \Psi_{\text{elec}} \quad (45)$$

for  $\Psi_{\text{elec}}$ , which describes the motion and position of electrons in a molecule. The independent electron approximation assumes



that  $\mathcal{H}_{\text{elec}}$ , collectively describing all electrons, can be approximated as a sum of effective one-electron Hamiltonians,  $\mathcal{H}_i^{\text{eff}}$ :

$$\mathcal{H}_{\text{elec}} = \sum_i \mathcal{H}_i^{\text{eff}} \quad (46)$$

## Molecular Orbitals

Electrons are identical, and each term in this sum is essentially the same operator. You can then solve an independent-electron Schrödinger equation for a wave function  $\psi$ , describing an individual electron:

$$\mathcal{H}_i^{\text{eff}}\psi = \epsilon_i\psi \quad (47)$$

Multiple solutions  $\psi_i$  and  $\epsilon_i$  are possible for this last equation. The wave functions for individual electrons,  $\psi_i$ , are called *molecular orbitals*, and the energy,  $\epsilon_i$ , of an electron in orbital  $\psi_i$  is called the *orbital energy*.

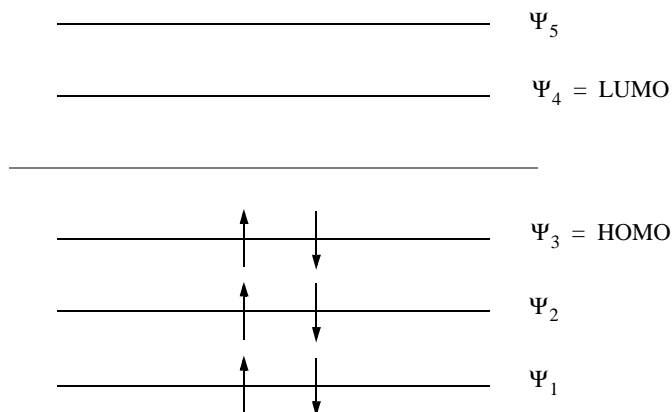
The molecular orbitals describe an electron such that the values of  $\psi_i^2(\mathbf{r}) d\mathbf{r}$  at a point  $\mathbf{r}$  describe the probability of the electron being in a small volume  $d\mathbf{r}$  around that point. The total probability of finding the electron somewhere is

$$\int \psi_i^2(\mathbf{r}) d\mathbf{r} = 1 \quad (48)$$

which says that the molecular orbitals are *normalized*.

## Orbital Energy Diagrams

You can order the molecular orbitals that are a solution to equation (47) according to their energy. Electrons populate the orbitals, with the lowest energy orbitals first. A normal, closed-shell, Restricted Hartree Fock (RHF) description has a maximum of two electrons in each molecular orbital, one with electron spin up and one with electron spin down, as shown:



If the number of electrons,  $N$ , is even, you can have a closed shell (as shown) where the occupied orbitals each contain two electrons. For an odd number of electrons, at least one orbital must be singly occupied. In the example, three orbitals are occupied by electrons and two orbitals are unoccupied. The highest occupied molecular orbital (HOMO) is  $\psi_3$ , and the lowest unoccupied molecular orbital (LUMO) is  $\psi_4$ . The example above is a singlet, a state of total spin  $S=0$ . Exciting one electron from the HOMO to the LUMO orbital would give one of the following excited states:

- Total spin  $S=0$  (a singlet) if the spins remained the same.
- Total spin  $S=1$  (a triplet) if a spin-down electron in the HOMO was promoted to be a spin-up electron in the LUMO. That is, the spin of one of the electrons was reversed when exciting it from the HOMO to the LUMO.
- If you add a single electron to the LUMO orbital above to create an anion, you obtain total spin  $S=1/2$  (a doublet).

## The MO-LCAO Approximation

To compute molecular orbitals, you must give them mathematical form. The usual approach is to expand them as a linear combination of known functions, such as the atomic orbitals of the constituent atoms of the molecular system. If the atomic orbitals, (1s, 2s, 2px, 2py, 2pz, etc.) are denoted as  $\phi_\mu$ , then this equation describes the molecular orbitals as linear combination of atomic orbitals (MO-LCAO):

$$\Psi_i = \sum_{\nu} C_{\nu i} \phi_{\nu} \quad (49)$$

where the elements of a matrix  $\mathbf{C}$  are the coefficients in the expansion.

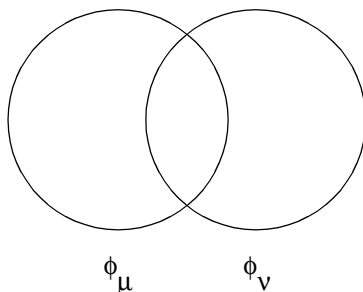
## The Matrix equations for the Molecular Orbitals

Now that you know the mathematical form, you can solve the independent-electron Schrödinger equation for the molecular orbitals. First substitute the LCAO form above into equation (47) on page 193, multiply on the left by  $\phi_{\mu}$  and integrate to represent the problem in terms of the matrix elements of a matrix  $\mathbf{H}$ ,

$$\mathbf{H}_{\mu\nu} = \int \phi_{\mu} H^{\text{eff}} \phi_{\nu} d\tau \quad (50)$$

The overlap between two atomic orbitals  $\phi_{\mu}$  and  $\phi_{\nu}$ ,

$0 < \text{Overlap} < 1$



defines the elements of a matrix  $\mathbf{S}$ ,

$$S_{\mu\nu} = \int \phi_{\mu} \phi_{\nu} d\tau \quad (51)$$

and the independent electron Schrödinger equation (47) reduces to the following matrix equation:

$$\mathbf{HC} = \mathbf{SC}\epsilon \quad (52)$$

where  $\epsilon$  is a diagonal matrix of the orbital energies,

$$\epsilon_{ij} = \epsilon_i \delta_{ij} \quad \delta_{ij} \text{ is } 1 \text{ (i equal j) or } 0 \text{ (i not equal j)} \quad (53)$$

## Solving for the Molecular Orbitals

Solving the previous matrix equation for the coefficients  $\mathbf{C}$  describing the LCAO expansion of the orbitals and orbital energies  $\epsilon$  requires a *matrix diagonalization*. If the overlap matrix were a unit matrix ( $\sum_{\mu\nu} = \delta_{\mu\nu}$ ), then you would simply diagonalize the matrix  $\mathbf{H}$  to yield the eigenvectors  $\mathbf{C}$  (the matrix of the diagonalizing unitary transformation), the eigenvalues  $\epsilon$ , and the resulting diagonal elements of  $\mathbf{H}$ :

$$\mathbf{C}^t \mathbf{H} \mathbf{C} = \epsilon \quad (54)$$

Because of the presence of the overlap matrix, however, you must first diagonalize the overlap matrix:

$$\mathbf{U}^t \mathbf{S} \mathbf{U} = \mathbf{s} \quad (55)$$

then form  $\mathbf{s}^{-1/2}$  as the diagonal matrix with elements  $(\mathbf{s}^{-1/2})_{ii} = [(\mathbf{s})_{ii}]^{-1/2}$

and the nondiagonal matrix  $\mathbf{S}^{-1/2}$  by the reverse transformation:

$$\mathbf{S}^{-1/2} = \mathbf{U}\mathbf{s}^{-1/2}\mathbf{U}^t \quad (56)$$

Finally, you must diagonalize  $\mathbf{S}^{-1/2}\mathbf{H}\mathbf{S}^{-1/2}$  to obtain the molecular orbitals and orbital energies:

$$(\mathbf{S}^{1/2}\mathbf{C})^t(\mathbf{S}^{-1/2}\mathbf{H}\mathbf{S}^{-1/2})(\mathbf{S}^{1/2}\mathbf{C}) = \varepsilon \quad (57)$$

## Self-Consistent Field Methods

In the true independent-electron methods you only have to obtain the matrix elements  $H_{\mu\nu}$  of some quite unspecified effective one-electron Hamiltonian  $\mathcal{H}^{\text{eff}}$ . These matrix elements, as in Extended Hückel Theory, can be quite empirical. The Hartree-Fock procedure or Self-Consistent Field (SCF) method, however, introduces very well defined mathematical approximations to end up with a one-electron eigenvalue problem, but one which must be solved iteratively to self-consistency.

### Hartree-Fock Method

While the equations of the Hartree-Fock approach can be rigorously derived, we present them *post hoc* and give a physical description of the approximations leading to them. The Hartree-Fock method introduces an effective one-electron Hamiltonian,  $\mathcal{F}$ , as in equation (47) on page 193:

$$\mathcal{F}\psi_i = \varepsilon_i\psi_i \quad (58)$$

but here the Fock operator,  $\mathcal{F}$ , depends implicitly upon the solutions  $\psi_i$ . The Fock operator is a sum of the kinetic energy operator for an electron, a potential that a single electron would feel coming from the fixed nuclei, and an average of the effects of all the other N-1 electrons.

The procedure is as follows:

1. Guess the position of each electron, that is, you guess each occupied orbital  $\psi_i$ .

2. Guess the average potential that a specific electron would feel coming from the other electrons; that is, you guess at the Fock operator.
3. Solve equation (58) for a new guess at the positions of the electrons.
4. Repeat the procedure until the wave function for an electron is consistent with the field that it and the other electrons produce.

## The Roothaan equations

The preceding discussion means that the Matrix equations already described are correct, except that the Fock matrix,  $\mathbf{F}$ , replaces the effective one-electron Hamiltonian matrix, and that  $\mathbf{F}$  depends on the solution  $\mathbf{C}$ :

$$\mathbf{F}(\mathbf{C})\mathbf{C} = \mathbf{S}\mathbf{C}\epsilon \quad (59)$$

These are the Roothaan SCF equations, which clearly can be solved iteratively—guess  $\mathbf{C}$ , form  $\mathbf{F}$ , diagonalize to a new  $\mathbf{C}$ , form a new  $\mathbf{F}$ , ..., and so on.

If you define a density matrix  $\mathbf{P}$  by summing over all occupied molecular orbitals:

$$P_{\mu\nu} = \sum_i^{\text{occ}} C_{\mu i} C_{\nu i} \quad (60)$$

then the rigorous definition of the Fock matrix is

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda\sigma} P_{\lambda\sigma} \left[ (\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma) \right] \quad (61)$$

where  $H_{\mu\nu}$  is a true one-electron Hamiltonian representing the kinetic energy and interaction of a single electron with the core of all the nuclei, and  $(\mu\nu|\lambda\sigma)$  is a so-called *two-electron integral*. The simpler two-electron integrals are easy to interpret:  $(\mu\mu|\nu\nu)$  is just the Coulomb repulsion between one electron described by the probability  $\phi_\mu^2$  and another electron described by the probability

$\phi_{\nu}^2$ , but others are less intuitive. The density matrix describes the total probability of finding an electron such that

$$\rho(\mathbf{r})d\mathbf{r} = \sum_{\mu\nu} P_{\mu\nu}\phi_{\mu}(\mathbf{r})\phi_{\nu}(\mathbf{r})d\mathbf{r} \quad (62)$$

is the probability of finding an electron in the small volume element  $d\mathbf{r}$  about the point  $\mathbf{r}$ .

## Spin Pairing

The Roothaan equations just described are strictly the equations for a closed-shell Restricted Hartree-Fock (RHF) description only, as illustrated by the orbital energy level diagram shown earlier. To be more specific:

- A closed-shell means that every occupied molecular orbital contains exactly two electrons.
- A restricted Hartree-Fock description means that spin-up and spin-down electrons occupy the same spatial orbitals  $\psi_i$ —there is no allowance for different spatial orbitals for different electron spins.

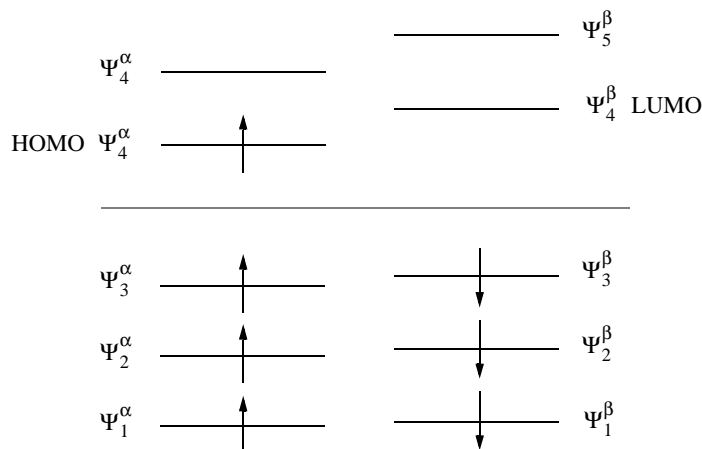
The occupancy can be described by

$$\psi_1(1)\bar{\psi}_1(2)\psi_2(3)\bar{\psi}_2(4)\psi_3(5)\bar{\psi}_3(6) \quad (63)$$

The notation here means that electron 1 occupies a spatial orbital  $\psi_1$  with spin up (no bar on top); electron 2 occupies spatial orbital  $\bar{\psi}_1$  with spin down (a bar on top), and so on. An RHF description of the doublet  $S=1/2$  state obtained by adding an electron to  $\psi_4$  would be

$$\psi_1(1)\bar{\psi}_1(2)\psi_2(3)\bar{\psi}_2(4)\psi_3(5)\bar{\psi}_3(6)\psi_4(7) \quad (64)$$

Conversely, an *unrestricted* Hartree-Fock description implies that there are two different sets of spatial molecular orbitals: those molecular orbitals,  $\psi_i^{\alpha}$ , occupied by electrons of spin up (alpha spin) and those molecular orbitals,  $\psi_i^{\beta}$ , occupied by electrons of spin down (beta spin) as shown next.



Notice that the orbitals are not paired;  $\psi_i^\alpha$  does not have the same energy as  $\psi_i^\beta$ . An unrestricted wave function like this is a natural way of representing systems with unpaired electrons, such as the doublet shown here or a triplet state:

$$\psi_1(1)\bar{\psi}_1(2)\psi_2(3)\bar{\psi}_2(4)\psi_3(5)\psi_4(6) \quad (65)$$

This last Restricted Hartree-Fock (RHF) state, if allowed to go unrestricted, would probably result in the following UHF state:

$$\psi_1^\alpha(1)\bar{\psi}_1^\beta(2)\psi_2^\alpha(3)\bar{\psi}_2^\beta(4)\psi_3^\alpha(5)\psi_4^\alpha(6) \quad (66)$$

with different orbitals for different spins. The equations to calculate these unrestricted molecular orbitals are generalizations of the Roothaan equations as first given by Pople and Nesbet.

## Pople-Nesbet Unrestricted equations

The Roothaan equations are the basic equations for closed-shell RHF molecular orbitals, and the Pople-Nesbet equations are the basic equations for open-shell UHF molecular orbitals. The Pople-Nesbet equations are essentially just the generalization of the Roothaan equations to the case where the spatial  $\psi_i^\alpha$  and  $\psi_i^\beta$ , as shown previously, are not defined to be identical but are solved independently.

These two cases are sufficient to cover all situations, but many theoretical scientists prefer an RHF description over a UHF descrip-



tion. A UHF wave function has certain undesirable characteristics as described below. The problem of defining and computing RHF descriptions for open-shell cases, however, is somewhat severe and it is usually necessary to indicate explicitly how open-shell RHF calculations are obtained (in HyperChem, they are obtained via the half-electron approximation). The UHF approximation can be applied universally without such considerations.

The unrestricted approach defines two different sets of spatial molecular orbitals—those that hold electrons with spin up:

$$\Psi_1^\alpha = \sum_{\nu} C_{\nu i}^\alpha \phi_{\nu} \quad (67)$$

and those that hold electrons with spin down:

$$\Psi_1^\beta = \sum_{\nu} C_{\nu i}^\beta \phi_{\nu} \quad (68)$$

The two sets of coefficients, one for spin-up alpha electrons and the other for spin-down beta electrons, are solutions of two coupled matrix eigenvalue problems:

$$\mathbf{F}^\alpha(\mathbf{C}^\alpha, \mathbf{C}^\beta)\mathbf{C}^\alpha = \mathbf{S}\mathbf{C}^\alpha \epsilon^\alpha \quad (69)$$

$$\mathbf{F}^\beta(\mathbf{C}^\alpha, \mathbf{C}^\beta)\mathbf{C}^\beta = \mathbf{S}\mathbf{C}^\beta \epsilon^\beta \quad (70)$$

The two equations couple because the alpha Fock matrix  $\mathbf{F}^\alpha$  depends on both the alpha and the beta solutions,  $\mathbf{C}^\alpha$  and  $\mathbf{C}^\beta$  (and similarly for the beta Fock matrix). The self-consistent dependence of the Fock matrix on molecular orbital coefficients is best represented, as before, via the density matrices  $\mathbf{P}^\alpha$  and  $\mathbf{P}^\beta$ , which essentially state the probability of describing an electron of alpha spin, and the probability of finding one of beta spin:

$$P_{\mu\nu}^\alpha = \sum_i^{\text{occ}} C_{\mu i}^\alpha C_{\nu i}^\alpha \quad (71)$$

$$P_{\mu\nu}^{\beta} = \sum_i^{\text{occ}} C_{\mu i}^{\beta} C_{\nu i}^{\beta} \quad (72)$$

$$\mathbf{P}^T = \mathbf{P}^{\alpha} + \mathbf{P}^{\beta} \quad (73)$$

$$F_{\mu\nu}^{\alpha} = H_{\mu\nu}^{\alpha} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T (\mu\nu|\lambda\sigma) - P_{\lambda\sigma}^{\alpha} (\mu\lambda|\nu\sigma) \quad (74)$$

$$F_{\mu\nu}^{\beta} = H_{\mu\nu}^{\beta} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T (\mu\nu|\lambda\sigma) - P_{\lambda\sigma}^{\beta} (\mu\lambda|\nu\sigma) \quad (75)$$

The equations and Roothaan equations are solved by the same techniques.

## Convergence

The solution of the RHF or UHF equations is an iterative procedure with two principal issues. First is the question of what to use for an initial guess and second, whether the solutions will converge quickly or at all. The initial guess affects the convergence also, as an exact guess would immediately converge.

An initial guess could be couched in terms of an initial guess at one of  $\mathbf{C}$ ,  $\mathbf{P}$ , or  $\mathbf{F}$ , which are equivalent starting points for the **CPFCPE**.. procedure. The simplest initial guess is to set  $\mathbf{P} = 0$  or equivalently,  $\mathbf{F} = \mathbf{H}$ . This is referred to as using the *core Hamiltonian* for an initial guess. HyperChem uses something very similar to this for an initial guess for the *ab initio* and semi-empirical methods. Alternatively, HyperChem offers some other methods, such as the projected Hückel, projected CNDO, and projected INDO, for an initial guess of MO for *ab initio* calculations.

The convergence of SCF interactions is not always successful. In the simplest iteration procedure, iterations proceed without the aid of either an external convergence accelerator or an extrapolator. This often leads to slow convergence.

Thus, HyperChem occasionally uses a three-point interpolation of the density matrix to accelerate the convergence of quantum mechanics calculations when the number of iterations is exactly divisible by three and certain criteria are met by the density matrices. The interpolated density matrix is then used to form the Fock matrix used by the next iteration. This method usually accelerates convergent calculations. However, interpolation with the MINDO/3, MNDO, AM1, and PM3 methods can fail on systems that have a significant charge buildup.

When three-point interpolation fails to yield a convergent calculation, you can request a second accelerator for any SCF calculation via the Semi-empirical Options dialog box and the Ab Initio Options dialog box. This alternative method, Direct Inversion in the Iterative Subspace (DIIS), was developed by Peter Pulay [P. Pulay, *Chem. Phys. Lett.*, 73, 393 (1980); *J. Comp. Chem.*, 3, 556(1982)]. DIIS relies on the fact that the eigenvectors of the density and Fock matrices are identical at self-consistency. At SCF convergence, the following condition exists

$$[\mathbf{P} \cdot \mathbf{F}] = 0 \quad (76)$$

The extent to which this condition does not occur is a measure of deviance from self-consistency. The DIIS method uses a linear combination of previous Fock matrices to predict a Fock matrix that minimizes  $[\mathbf{P} \cdot \mathbf{F}]$ . This new Fock matrix is then used by the SCF calculation.

The DIIS method is not without cost. A history of Fock matrices must be maintained and appropriate memory allocated. In addition, the computational cost of generating a new Fock matrix is significant. Also, in very rare cases, the solution found by this method is very different from that found in other ways.

## Spin Pairing—Restricted or Unrestricted?

You will need to decide whether or not to request Restricted (RHF) or Unrestricted (UHF) Hartree-Fock calculations. This question embodies a certain amount of controversy and there is no simple answer. The answer often depends simply on which you prefer or what set of scientific prejudices you have. Ask yourself whether you prefer orbital energy diagrams with one or two electrons per orbital.

To consider the question in more detail, you need to consider spin eigenfunctions. If you have a Hamiltonian  $\mathcal{H}$  and a many-electron spin operator  $\hat{S}$ , then the wave function  $\Psi$  for the system is ideally an eigenfunction of both operators:

$$\mathcal{H}\Psi = E\Psi \quad (77)$$

$$\hat{S}^2\Psi = S(S+1)\Psi \quad (78)$$

The functions that are exact eigenvalues of the spin operator are described as singlet ( $S=0$ ), doublet ( $S=1/2$ ), and so on. One school of thought says that since we can't obtain exact eigenvalues of  $\mathcal{H}$ , we should try to ensure that our wave functions are exact eigenvalues of  $\hat{S}^2$ , which we can easily do for RHF wave functions. UHF wave functions relax the restriction of being exact eigenvalues of  $\hat{S}^2$  for a lower energy  $E$ . A singlet UHF wave function, for example, might have a value of  $S=0.06$  and almost be a singlet. Because electron spins are not exactly paired (electrons of opposite spin occupy different spatial orbitals), an exact spin state is not obtained. Nevertheless, UHF wave functions have lower energy than their RHF counterparts. A second school of thought uses the variational principle to justify that UHF wave functions are better solutions (given their better energy).

Of the two, the UHF method provides a more continuous and universal model for chemistry. That is, you can obtain an unrestricted wave function more easily in various chemical situations than you can a restricted solution. An RHF solution for arbitrary bonding situations is sometimes hard to obtain. An example of this is the two-electron bond of, say  $\text{H}_2$ , as you stretch and break the bond. At normal bond lengths, the molecule is described as a singlet with two electrons occupying the same spatial molecular orbital. As you stretch and break the bond, however, the two electrons are certainly not best described as having the same spatial distribution—one electron should be on one hydrogen atom (a doublet) and the other electron on the other hydrogen atom (a second spin doublet). You cannot describe two hydrogen atoms by an RHF wave function with both electrons occupying the same spatial orbital; if you do, the hydrogen molecule does not dissociate correctly when you lengthen the bond. A UHF description of  $\text{H}_2$ , however, will dissociate correctly, and a UHF wave function with separate-orbitals-for-separate-spins can describe two isolated hydrogen atoms. At normal bond lengths, the UHF solution usually degenerates to the situation where the two spatial orbitals become identical. The UHF solution for  $\text{H}_2$ , for example, has a smooth potential energy

surface with no abnormalities. This makes a UHF computation desirable for studying chemical reactions involving the breaking of bonds. Such bond breaking can be poorly handled by RHF wave functions unless the reaction has no change in spin pairing as it proceeds.

A UHF wave function may also be a necessary description when the effects of spin polarization are required. As discussed in “Differences Between INDO and CNDO,” a Restricted Hartree-Fock description will not properly describe a situation such as the methyl radical. The unpaired electron in this molecule occupies a p-orbital with a node in the plane of the molecule. When an RHF description is used (all the s orbitals have paired electrons), then no spin density exists anywhere in the s system. With a UHF description, however, the spin-up electron in the p-orbital interacts differently with spin-up and spin-down electrons in the s system and the s-orbitals become spatially separate for spin-up and spin-down electrons with resultant spin density in the s system.

## Electronic States

HyperChem describes electronic states by their spin state (multiplicity,  $2S+1$ ) as either singlet ( $S=0$ ,  $2S+1=1$ ), doublet ( $S=1/2$ ,  $2S+1=2$ ), triplet ( $S=1$ ,  $2S+1=3$ ), or quartet ( $S=3/2$ ,  $2S+1=4$ ). Think of these states as having 0, 1, 2, or 3 unpaired electrons. An *even number of electrons* can lead to a singlet or a triplet, while an *odd number of electrons* leads to a doublet or a quartet. HyperChem does not explicitly consider states of higher multiplicity, such as quintets, and so on. The *ground state* (state of absolutely lowest energy) is not normally a quartet, but can commonly be a singlet, doublet, or a triplet. All other states are *excited states*. Systems with an even number of electrons commonly have a singlet ground state, but a triplet ground state is also possible depending on the arrangement of the molecular orbitals. For example, one of the first successes of molecular orbital theory was to explain why the  $O_2$  oxygen molecule has a triplet ground state (the HOMO orbitals are degenerate and occupied by two electrons). Molecules with an odd number of electrons (free radicals) generally have a doublet ground state.

HyperChem semi-empirical methods usually let you request a calculation on the *lowest energy state of a given multiplicity* or the *next lowest state of a given spin multiplicity*. Since most molecules with an even number of electrons are closed-shell singlets without

unpaired electrons, only the lowest triplet is made available and the next lowest triplet stated is grayed. For example, benzene has an even number of electrons and the ground state is a closed-shell singlet. You can request this ground state (the lowest singlet), the first excited singlet state (next lowest singlet), or the first excited triplet state (the lowest triplet). That is, either the highest occupied molecular orbital (HOMO) is occupied by two electrons or one electron is in the HOMO and one is in the lowest unoccupied molecular orbital (LUMO) leading to excited singlet or triplet states. For doublets and quartets, only the lowest state of the given multiplicity is available as an option.

For open-shell RHF semi-empirical calculations, the half-electron method [H. C. Longuet-Higgins and J. A. Pople, *Proc. Phys. Soc.*, 68, 591 (1955)] recommended by Dewar [M. J. S. Dewar and S. Olivella, *J. Chem. Soc. Fara. II*, 75, 829 (1979)]. It is used in the MOPAC and AMPAC series of programs [J. J. P. Stewart, *Quantum Chem. Prog. Exchange Bull.*, 10(4), 86 (1990); Dewar Research Group, *Quantum Chem. Prog. Exchange Bull.*, 7(2), 85 (1987)] in what amounts to a closed-shell calculation. For example, if the lowest doublet is requested, the orbital holding the unpaired electron is actually occupied by two 1/2-electrons with opposite spin. This is so that the full standard machinery of closed-shell Hartree-Fock calculations can be used to determine the orbitals. A correction is then made to the closed-shell SCF energy to properly represent the doublet state. For molecules with an even number of electrons and a closed-shell ground state this means that the first excited singlet and triplet states have the same orbitals—both being computed by pairs of 1/2-electrons in the HOMO and LUMO orbitals. The correction applied to obtain the energy of the 1/2-electron states, however, leads to the triplet being lower than the singlet, because of the exchange effect (described earlier in conjunction with the difference between CNDO and INDO).

The UHF option allows only the lowest state of a given multiplicity to be requested. Thus, for example, you could explore the lowest Triplet excited state of benzene with the UHF option, but could not ask for calculations on an excited singlet state. This is because the UHF option in HyperChem does not allow arbitrary orbital occupations (possibly leading to an excited single determinant of different spatial symmetry than the lowest determinant of the same multiplicity), nor does it perform a Configuration Interaction (CI) calculation that allows a multitude of states to be described.

For the RHF option, you can request a Configuration Interaction (CI) wave function. These calculations start with a set of computed RHF orbitals, either from the lowest singlet (doublet) or from the half electron singlet and triplet (doublet and quartet) orbitals. This initial calculation is referred to as the reference state. From this reference state, a set of microstates (different single determinants shown in an orbital energy diagram as different occupancies of the orbitals) is generated. These microstates are generated by promoting electrons from occupied orbitals closest to the HOMO orbital into unoccupied orbitals closest to the LUMO orbital until the lowest. Either an orbital criterion or energy criterion can be used to select microstates. A more accurate set of states is then obtained by taking appropriate linear combinations of these microstates. The coefficients of these microstates in the new, more accurate states are obtained by a diagonalization of the Hamiltonian matrix (the CI calculation). The states resulting from the CI calculation can all be inspected in the log file. The CI calculation takes a considerably longer time to obtain than the simple Hartree-Fock (SCF) calculations, and you should use the CI option only if you are prepared to wait for the (possibly) more accurate results.

Thus, in HyperChem a state is described for calculation in this sequence:

1. Define the multiplicity in the Semi-empirical Options dialog box or the Ab Initio Options dialog box.
2. Request one of two computational procedures, either RHF or UHF.
3. Request either the lowest energy state of the specified multiplicity or the next lowest energy state for semi-empirical calculations.
4. Request configuration interaction, if you want it.

Optimization and molecular dynamics cannot be performed when using configuration interaction. When a configuration interaction calculation is performed, gradients of the energy (and hence geometry optimization and molecular dynamics) are not available. For these cases, any geometry optimization should be done with another state or procedure, and then the configuration interaction calculation requested only to obtain spectral information, such as a ultraviolet-visible spectrum.

# Post Self-Consistent Field Calculations

## Configuration Interaction

For some systems a single determinant (SCF calculation) is insufficient to describe the electronic wave function. For example, square cyclobutadiene and twisted ethylene require at least two configurations to describe their ground states. To allow several configurations to be used, a multi-electron configuration interaction technique has been implemented in HyperChem.

Electronic spectroscopy requires the consideration of multiple electronic states for the same configuration of the nuclei. Configuration interaction is a procedure for generating and mixing many configurations (Slater determinants) together to arrive at a description of not only the ground state, but many excited states of a molecule. To a first approximation, only singly-excited configurations are needed to describe excited electronic states.

HyperChem uses single determinant rather than spin-adapted wave functions to form a basis set for the wave functions in a configuration interaction expansion. That is, HyperChem expands a CI wave function,  $\Phi_a$ , in a linear combination of single Slater determinants  $\Psi_i$

$$\Phi_a = \sum_i C_{ia} \Psi_i \quad (30)$$

The expansion coefficients  $C_{ia}$  are determined by solving the CI secular equation

$$\mathbf{HC} = \mathbf{CE} \quad (31)$$

where  $\mathbf{E}$  is the eigenvalue vector and represents the energies of the CI states.

The single Slater determinants are generated by taking an electron from one of the specified occupied orbitals of the ground reference state and placing that electron into one of the specified unoccupied orbitals (of the ground state). The Slater determinants that are mixed together in the CI calculation are completely specified by giving the occupied and unoccupied orbitals to use in generating the configuration, as specified in the Configuration Interaction dialog box.



The elements of the CI matrix are:

$$\langle \hat{\Phi}_a | \hat{H} | \Phi_b \rangle \quad (32)$$

Evaluation of the CI matrix elements is somewhat difficult. Fortunately, most matrix elements are zero because of the orthogonality of the MOs. There are only three types of non-zero elements which are needed to be computed.

1) Diagonal elements:

$$\langle \hat{\Phi}_a | \hat{H} | \Phi_a \rangle \quad (33)$$

2) Off-diagonal elements:

$$\langle \hat{\Phi}_a | \hat{H} | \Phi_b \rangle \quad (34)$$

where  $\Phi_a = \Phi_b$  except that  $\Psi_i$  in  $\Phi_a$  is replaced by  $\Psi_j$  in  $\Phi_b$ .

3) Off-diagonal elements:

$$\langle \hat{\Phi}_a | \hat{H} | \Phi_b \rangle \quad (35)$$

where  $\Phi_a = \Phi_b$  except that  $\Psi_i$  and  $\Psi_j$  in  $\Phi_a$  are replaced by  $\Psi_k$  and  $\Psi_l$  in  $\Phi_b$ .

Once the CI expansion coefficients are available, the expectation values of the  $\hat{S}^2$  operator can be obtained by

$$\langle \hat{\Phi}_a | \hat{S}^2 | \Phi_b \rangle \quad (36)$$

For the details of calculation for the matrix elements of operator  $\hat{H}$  and expectation values of  $\hat{S}^2$  in equations (33) to (36), please see J. P. Stewart, *J. Comp. Aided Mol. Design*, 4, 1 (1990).

## MP2 Correlation Energy

Configuration interaction (CI) is a systematic procedure for going beyond the Hartree-Fock approximation. A different systematic approach for finding the correlation energy is perturbation theory (Rayleigh-Schrödinger many-body perturbation theory — RSPT). In this approach, the total Hamiltonian of the system is divided or partitioned into two parts: a zeroth-order part,  $H_0$  (which has

known eigenfunctions and eigenvalues), and a perturbation,  $V$ . The exact energy is then expressed as an infinite sum of contributions of increasing complexity. The expressions for these contributions contain the eigenvalues of  $H_0$  and matrix elements of the perturbation between the eigenfunctions of  $H_0$ . By introducing the ordering parameter  $\lambda$ , the total Hamiltonian can be written as

$$H = H + \lambda V \quad (37)$$

The exact eigenfunctions and eigenvalues can now be expanded in a Taylor series in  $\lambda$ .

$$\Psi_\lambda = \Psi^{(0)} + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \dots \quad (38)$$

$$E_\lambda = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \dots \quad (39)$$

If  $H_0$  is chosen wisely, then the perturbation is small and the perturbation expansion (i.e., the sum of the 1st, 2nd, ..., nth-order energies) converges quickly. To obtain a perturbation expansion for the correlation energy, the best way is to choose the Hartree-Fock Hamiltonian as the zeroth-order Hamiltonian. The application to N-electron molecular systems is sometimes called Møller-Plesset (MP) perturbation theory. These methods, which can be terminated at second (MP2), third (MP3), or fourth order (MP4), with these three being the most frequently used in different *ab initio* programs, calculate the correlation energy and rely on a good description of the virtual orbitals in the original SCF function. The calculated total correlation energy is therefore quite dependent on the quality of the basis set.

Møller-Plesset calculations do not give the full correlation energy, and MP2 is estimated to give only about half of the total correlation energy. They, however, are very fast in comparison with configuration interaction calculations and appear to reproduce well the energetic effects of correlation. The MP2 method probably represents the simplest approximation for the correlation energy. The MP2 energy is given by

$$E^{(2)} = - \sum_{ar} \frac{V_{ar} V_{ra}}{\epsilon_a^{(0)} - \epsilon_r^{(0)}} \quad (40)$$

where  $\epsilon_a$  and  $\epsilon_r$  are the eigenvalues of the occupied and unoccupied MOs of the zero-order Hamiltonian and  $V_{ar}$  are the matrix elements of the perturbation Hamiltonian operator.

The amount of computation for MP2 is determined by the partial transformation of the two-electron integrals, what can be done in a time proportionally to  $m^4$  ( $m$  is the number of basis functions), which is comparable to computations involved in one step of CID (doubly-excited configuration interaction) calculation. To save some computer time and space, the core orbitals are frequently omitted from MP calculations. For more details on perturbation theory please see A. Szabo and N. Ostlund, *Modern Quantum Chemistry*, Macmillan, New York, 1985.

HyperChem supports MP2 (second order Møller-Plesset) correlation energy calculations using any available basis set. In order to save main memory and disk space, the HyperChem MP2 electron correlation calculation normally uses a so called “frozen-core” approximation, i.e. the inner shell (core) orbitals are omitted. A setting in CHEM.INI allows excitations from the core orbitals to be included if necessary (melted core). Only the single point calculation is available for this option.

## The Neglect of Differential Overlap Approximation

The principal semi-empirical schemes usually involve one of two approaches. The first uses an *effective* one-electron Hamiltonian, where the Hamiltonian matrix elements are given empirical or semi-empirical values to try to correlate the results of calculations with experiment, but no specified and clear mathematical derivation of the explicit form of this one-electron Hamiltonian is available beyond that given above. The extended Hückel calculations are of this type.

The second approach is based on an explicit mathematical approximation: the neglect of certain *differential overlap* terms. This approximation is not necessarily a valid one in any or all instances but can be justified to an extent. These approximations are offset by introducing empirical parameters and then using the new approximate form of the Schrödinger equation with its adjustable parameters as a fitting procedure for experimental results. A principal difficulty in any attempt to solve the Schrödinger equation for molecules is the large number ( $\sim N^4$ ) of two-electron integrals

(mn|ls) that have to be computed and used in some fashion in the computation. These integrals are defined as

$$(\mu\nu|\lambda\sigma) = \int \phi_{\mu}^A(\mathbf{r}_1)\phi_{\nu}^B(\mathbf{r}_1)r_{12}^{-1}\phi_{\lambda}^C(\mathbf{r}_2)\phi_{\sigma}^D(\mathbf{r}_2)d\mathbf{r}_1d\mathbf{r}_2 \quad (79)$$

where  $\phi_{\mu}^A$  is an atomic orbital centered at atom A. In general, these integrals involve atomic orbital basis functions  $\phi_{\mu}$  on four different centers A, B, C, and D. The principal function involved can be considered to be the two-center charge distribution,  $\phi_{\mu}^A\phi_{\nu}^B$ , which when integrated is the overlap integral

$$S_{\mu\nu} = \int \phi_{\mu}^A(\mathbf{r}_1)\phi_{\nu}^B(\mathbf{r}_1)d\mathbf{r}_1 \quad (80)$$

The differential overlap approximation says that  $S_{\mu\nu} \ll 1$  in many situations and that the overlap or its original argument under the integrand above can be ignored or approximated in various ways. The simplest approximation is the *complete neglect of differential overlap*

$$\phi_{\mu}\phi_{\nu} = \delta_{\mu\nu}\phi_{\mu}\phi_{\mu} \quad (81)$$

The second basic approximation is the neglect of diatomic differential overlap

$$\phi_{\mu}^A\phi_{\nu}^B = \phi_{\mu}^A\phi_{\nu}^A\delta_{AB} \quad (82)$$

Either of these approximations lead to a great simplification of the general equations, the greatest simplification being that associated with complete neglect of differential overlap (CNDO). An intermediate neglect of differential overlap (INDO, ZINDO/1, and ZINDO/S) scheme retains some of the one-center terms and the neglect of diatomic differential overlap (NDDO) schemes keep all of the one-center terms above. The TNDO method is a version of CNDO/INDO with parameters based on atom type rather than atomic number. The modified INDO (MINDO/3) method and those that are based on NDDO (MNDO, MNDO/d, AM1, and PM3) are all available in HyperChem and are discussed extensively below, in turn. First of all, however, we describe the results of SCF calculations that are independent of the details of whether the spe-

cific approximation involved is CNDO, MNDO, or some other approximate treatment of the general SCF equations.

## Characterizations of the Wave function

When you perform a single point semi-empirical or *ab initio* calculation, you obtain the energy and the first derivatives of the energy with respect to Cartesian displacement of the atoms. Since the wave function for the molecule is computed in the process, there are a number of other molecular properties that could be available to you. Molecular properties are basically an average over the wave function of certain operators describing the property. For example, the electronic dipole operator is basically just the operator for the position of an electron and the electronic contribution to the dipole moment is

$$\mu = \sum_i \int \psi_i(\mathbf{r}) \mathbf{r} \psi_i(\mathbf{r}) d\mathbf{r} \quad (83)$$

The dipole moment for a calculation is reported and is contained in the log file if logging is turned on. It is also reported on the status line if you Display Dipole Moment on the Display menu. Other expectation values besides the dipole moment (for example, quadrupole moment) could be reported with a wave function but the set reported with this release of HyperChem is limited to only a few. Below we discuss the properties or other characterizations of the calculated wave function that can be interactively visualized.

### Contour Plots

HyperChem allows the visualization of two-dimensional contour plots for a certain number of variables. These contour plots show the values of a spatial variable (a property  $f(x,y,z)$  in normal three-dimensional Cartesian space) on a plane that is parallel to the screen. To obtain these contour plots the user needs to specify:

- The plane to be contoured
- The contours to be shown
- The number and positioning of the grid of points on the plane to be used in computing values of the variable

The plane is specified as a plane parallel to the screen and offset from the center of mass of the current selection towards the user by a specified number of Ångstroms. For example, if you make no selection and specify an offset of 0.0, the plane is through the center of mass of the whole molecular system. If you select a single atom, again with an offset of 0.0, then the plane is that plane parallel to the screen that goes through the selected atom. If you specify an offset of 1.0 instead of 0.0, then the plane is 1.0 Å closer to the viewer than before. You can also specify a negative offset, which pushes the plane away from the user and into the screen.

You can specify the number and values of visible contour lines. You specify the total number of contour lines to be shown by simply stating the number,  $n > 0$ . You normally specify the values of the contour lines as *default values*. For this case, HyperChem computes the maximum and minimum values on the grid and then draws contours at these values plus  $n-2$  contour lines evenly spaced in between these maximum and minimum values. If you need non-default values, you can specify the starting value and then an increment to define the other  $n-1$  evenly spaced contour lines. If default values were computed previously, HyperChem suggests the starting value and increment of the previous default computation for the new non-default option.

The grid of computed values for the variable used in defining the contour is a grid of exactly the size of the current working area subdivided evenly such that the total number of grid points is as the user specifies.

When a molecule rotates, HyperChem abandons the current contour map because it is no longer valid and you must explicitly request a new contour map via the Graph check box.

## Total Electron Density

The density matrices are a representation, in the basis set  $\phi_\mu$  of the electron density at a point in space. The electron density  $\rho(\mathbf{r})$ , described above by equation (62) on page 198, represents a probability distribution function (the probability of an electron being at a particular point in space (rigorously, in a small volume  $d\mathbf{r}$  at the point  $\mathbf{r}$  in ordinary three-dimensional space). The total density can be subdivided into the probability of finding a spin-up (alpha) electron plus the probability of finding a spin-down (beta) electron:

$$\rho^T(\mathbf{r}) = \rho^\alpha(\mathbf{r}) + \rho^\beta(\mathbf{r}) \quad (84)$$

where the individual densities are defined to be identical in a closed-shell RHF calculation but can be different in a UHF calculation, according to:

$$\rho^\alpha(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_{\mu\nu} P_{\mu\nu}^\alpha \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\mathrm{d}\mathbf{r} \quad (85)$$

$$\rho^\beta(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_{\mu\nu} P_{\mu\nu}^\beta \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\mathrm{d}\mathbf{r} \quad (86)$$

The neglect of diatomic differential overlap approximation used in the semi-empirical methods reduces these formulae to much simpler formulae that can be computed quite rapidly,

$$\rho^\alpha(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_{A, \mu\nu \text{ on } A} P_{\mu\nu}^\alpha \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\mathrm{d}\mathbf{r} \quad (87)$$

$$\rho^\beta(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_{A, \mu\nu \text{ on } A} P_{\mu\nu}^\beta \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})\mathrm{d}\mathbf{r} \quad (88)$$

where the double sum is over each atom and then the orbitals on that atom. Without this last approximation the electron density calculation can be a long one, even longer than solving the approximate Schrödinger equation in the first place. The reason is the large number of grid points and the double sum over all pairs of atomic orbitals at each grid point. For *ab initio* and extended Hückel calculations where no neglect of overlap (differential or otherwise) is implied in the approximations, the complete calculation is performed and requires considerable patience!

The CNDO/INDO, TNDO, MINDO/3, ZINDO/1, and ZINDO/S methods might be expected to imply an even simpler equation for the electron density than the above. For example, a rigorous complete neglect of CNDO approximation, suggests that equations (87) and (88) should be replaced by expressions with a sum only

over diagonal elements of the density matrix. This would represent a molecular charge density that is the exact sum of atomic densities. Alternatively, expectation values like the dipole moment, computed from the this density matrix, would have only monopole charge on each atom contributions. The original authors of the CNDO method recognized that a hybridization term (coming from one-center integrals combining s and p type orbitals) was important to the dipole moment and included it as part of the method. In order to make the electron density consistent with expectation values and take account of these hybridization contributions to dipole moments, electrostatic fields, etc., all the semi-empirical SCF methods in HyperChem use equations (87) and (88) for the electron density and associated expectation values.

## Spin Density

For open-shell situations or for some closed-shell UHF situations (A UHF calculation for a normal closed-shell situation will degenerate into the closed-shell RHF solution in most cases. However, a true UHF solution is always obtained when a bond is stretched to such an extent that it is clearly broken and only an unrestricted open-shell descriptions of the two fragments is appropriate) non-zero spin densities exist and can be contoured. The definition of a closed-shell situation has subtleties, but a simplistic definition of it might be that the probability of finding a spin-up electron is identical to the probability of finding a spin-down electron, everywhere in space. Thus, in contrast to the total electron density defined as the sum of alpha and beta densities above, it is convenient to define the *spin density* as the difference:

$$\rho^S(\mathbf{r}) = \rho^\alpha(\mathbf{r}) - \rho^\beta(\mathbf{r})$$

The spin density defines the excess probability of finding spin-up over spin-down electrons at a point in space and is zero everywhere for closed-shell RHF situations. The spin density at the position of a nucleus is a prime determinant of electron spin resonance (ESR) spectra.

## Orbital Plots

For many reasons, including the Woodward-Hoffman rules that describe the likelihood of reaction based on arguments about the



shapes of orbitals, it is desirable to be able to visualize molecular orbitals.

You can use the semi-empirical and *ab initio* (or DFT) Orbitals dialog box in HyperChem to request a contour plot of any molecular orbital. When requested, the orbital is contoured for a plane that is parallel to the screen and which is specified by a subset selection and a plane offset, as described above. The index of the orbital and its orbital energy (in electron volts, eV) appears in the status line.

If a molecule is rotated by changing the position of the viewer (left mouse button rotation) then the molecule's position in the molecular coordinate system has not changed and another contour plot can be requested without recomputing the wave function. That is, many orbitals can be plotted after a single point *ab initio*, DFT, or semi-empirical calculation. Any contour map is available without recomputation of the wave function.

When you request an orbital, you also request a plot of either the orbital itself or of its square. The orbital  $\psi_i$  is a signed quantity with nodes where the sign changes and the value of the orbital goes to zero. The nodal structure of orbitals has chemical significance. For planar molecules you can distinguish s-orbitals which have non-zero value in the plane of the molecule from  $\pi$ -orbitals which have a node in the plane of the molecule. As energy increases, an orbital generally has more and more nodes. The orbital squared represents the probability distribution function for an electron in the orbital.

When you request an orbital, you can use the cardinal number of the orbital (ordered by energy and starting with number=1) or an offset from either the highest occupied molecular orbital (HOMO) or the lowest unoccupied molecular orbital (LUMO). Offset from the HOMO are negative and from the LUMO are positive. Often these frontier orbitals are the ones of most chemical interest.

## Electrostatic Potential

The results of electrostatic potential calculations can be used to predict initial attack positions of protons (or other ions) during a reaction. You can use the Contour Plot dialog box to request a plot of the contour map of the electrostatic potential of a molecular system after you done a semi-empirical, *ab initio*, or DFT calculation. By definition, the electrostatic potential is calculated using the following expression:

$$V(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} - \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} d\mathbf{r}'$$

The point  $\mathbf{r}$  is the position of a positive probe charge.  $Z_A$  is the nuclear charge on atom A located at position  $\mathbf{R}_A$ . The function  $\rho(\mathbf{r}')$  is the electronic density. In the above equation, the first term represents the contribution of the nuclei to the electrostatic potential and the second term is the electronic contribution. Substituting the electron density expression:

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu} \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \quad (89)$$

into the electrostatic potential expression above, the electrostatic potential can be rewritten as

$$V(\mathbf{r}) = \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} - \sum_{\mu\nu} P_{\mu\nu} \int \frac{\phi_\mu(\mathbf{r}') \phi_\nu(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} d\mathbf{r}' \quad (90)$$

The second summation is over all the orbitals of the system. This equation is used in HyperChem *ab initio* and DFT calculations to generate contour plots of electrostatic potential. If we choose the approximation whereby we neglect the effects of the diatomic differential overlap (NDDO), then the electrostatic potential can be rewritten as

$$V(\mathbf{r}) = \sum_A \left[ \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} - \sum_{\mu\nu}^{\text{on A}} P_{\mu\nu} \int \frac{\phi_\mu(\mathbf{r}') \phi_\nu(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} d\mathbf{r}' \right] \quad (91)$$

The second summation of the above is over the orbitals of atom A. HyperChem calculates the electrostatic potential using this last expression for the semi-empirical methods.

## Mixed Quantum/Classical Model

You can quickly select a portion of a molecule and then treat that portion quantum mechanically while treating the remaining portion as potential to be included as an external coulomb field in the quantum mechanical calculation. These calculations mix a quantum description (of the selected portion) with a classical (molecular mechanics) description of the point charges on the atoms of the unsettled portion. A complete merger of the two technologies (molecular mechanics and semi-empirical quantum mechanics) is not yet available (for performing molecular dynamics, for example, and having both the classical and quantum portions move); however, you can perform any quantum mechanical calculation on a selected portion of a molecule and have the remaining portion included (as a potential in the one-electron Hamiltonian matrix  $\mathbf{H}$ ).

Two basic issues must be solved in order to perform such calculations.

- How to decide on or form the boundary between the classical portion and the quantum portion. (Alternatively, you may ask which atoms will be quantum atoms and which will be classical atoms?)
- How to cap the quantum calculation so that there are no dangling electrons or bonds in the quantum mechanical calculation while still preserving the desired effect of the classical portion (which is now there, only in principle!).

### Choosing the Classical–Quantum Boundary

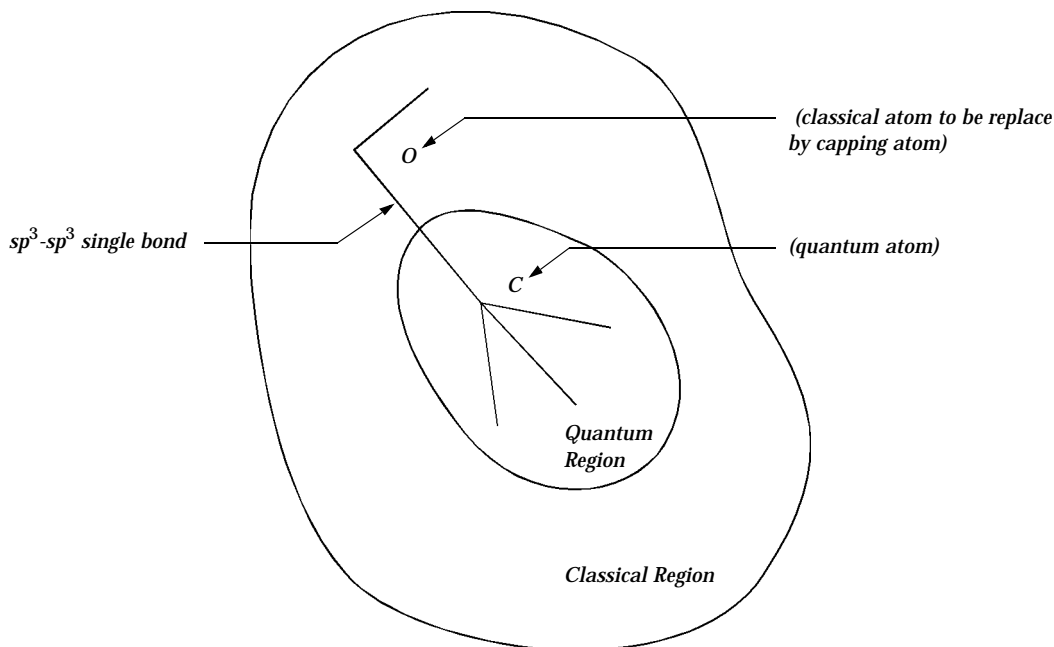
HyperChem assumes that it is easiest for you to just use subset selection to select that portion of the molecular system that is to be treated quantum mechanically. You can then extend the initial selection to form a convenient and universally acceptable boundary. Thus, you make a simple selection of atoms for the first pass at selecting the quantum mechanical portion. The selected atoms are quantum atoms and the unselected atoms are classical atoms.

If you request a semi-empirical quantum mechanical calculation now, HyperChem carries on as well as possible (as described below) in choosing how the atoms for quantum mechanical calculation is capped. If, however, an aromatic benzene ring (with delocalized

bonds, for example), has partially quantum atoms and partially classical atoms, it may be difficult for HyperChem to create a sensible result, although it will certainly try. A special menu item, Extend to  $sp^3$ , extends the selection outwards until it finds one of the following:

- The end of the molecule. Two atoms not part of the same connected molecule must be in separate molecules. Coming to the end of the molecule means reaching a terminal atom of the molecule such as a mono-coordinated hydrogen or fluorine, a carbonyl oxygen, etc.
- An  $sp^3$ - $sp^3$  single bond where the outside atom in the bond is not a mono-coordinated terminal atom such as hydrogen or fluorine.

The menu setting results in a boundary between the classical region and the quantum region that is either intermolecular or occurs in the middle of an  $sp^3$ - $sp^3$  single bond. As described below it will be easy for HyperChem to cap the quantum region if it ends this way.



Because it is desirable to break a peptide this way, some flexibility is required in the rigorous definition of  $sp^3$ - $sp^3$  single bond. In particular, the dative  $C_\alpha$ -N bond in the backbone of a peptide is considered to be such a bond since the definition is based on the number of neighbors — four for carbon, three for nitrogen, two for Oxygen, etc. If this were not the case, you couldn't break a protein into classical and quantum regions at all.

## Capping Atoms and their Parameters

Having found a place (the  $sp^3$ - $sp^3$  bond) to establish the boundary between classical atoms and quantum atoms, the next question is how to cap the quantum atoms. Let's first of all look at an illustrative example of the problem.

Consider an Alanine residue in a protein, [ $\dots C_\alpha$ -NH-CO- $C_\alpha$ H(CH<sub>3</sub>)-NH-CO- $C_\alpha$ ...]. If you first select, for example, the CH<sub>3</sub> methyl group and requested the extension to  $sp^3$ , the selected quantum atoms would be the fragment [-NH-CO-

$C_{\alpha}H(CH_3)-$ ]. This fragment has dangling bonds on both ends and prior to performing a quantum mechanical calculation, you wish to close them off by adding a capping (mono-coordinated) atom. The obvious choice is to add capping hydrogen atoms (H') and perform the quantum mechanical calculation on the closed-shell molecule  $[NHH'-CO-C_{\alpha}H(H')(CH_3)]$ .

The principal other consideration, however, is that these capping hydrogen atoms should simulate the effect of the rest of the protein that they replace, including any electronegative (electron withdrawing) or electropositive (electron donating) effects. In particular, the H' atoms should not be normal hydrogen atoms but rather (for a semi-empirical framework) be false capping hydrogen atoms that are parameterized (have their own parameters) to represent these electron withdrawing or donating effects. For example the H' on the right hand end of the fragment above (call it  $H^N$ ) replaces an electronegative nitrogen and should have its own special parameters while the H' on the left hand end of the fragment above (call it  $H^C$ ) replaces a relatively electroneutral carbon and should have a different set of parameters yet from the ordinary H or the capping atom  $H^N$ .

While these false hydrogen atoms would seem simple and convenient to use for the purposes described, experiments have shown that, having no  $\pi$ -electrons, capping hydrogen atoms leave out one principal effect of the classical atoms they replace — *hyperconjugation*. Hydrogen atoms can only pull or push around s-electrons but a Fluorine, with its p-orbitals, can also pull or push around p-electrons. The bottom line is that you cannot fully simulate the effect of the classical atoms unless you cap the quantum calculation with a false fluorine atom rather than a false hydrogen atom. If you use specially parameterized fluorine atoms,  $F^C$ ,  $F^N$ ,  $F^O$ , etc. to cap, then the charge distribution of the real system (a quantum calculation on the whole system) is correctly reproduced by a quantum calculation on the truncated system (selected quantum atoms plus capping atoms).

When a selection is used to distinguish a quantum region from a classical region and a subsequent semi-empirical calculation is performed on the quantum region, “phony” halogen atoms are used to cap the quantum region. These pseudo-halogen atoms are either fluorines, chlorines, bromines, or iodines depending on the row of the periodic table represented by the “connecting” atom of the

classical portion being replaced by the capping atom. Parameters for pseudo-halogen atoms depend on the connecting atom and are labeled by a negative atomic number in the respective \*.*abp* files.

The parameters for pseudo-halogen atoms have only been determined for a limited number of connecting atoms (C, N, O, Si, P and S) and only for CNDO/INDO methods. If other connecting atoms are tried, HyperChem will usually complain that the orbital exponent for that atom is zero since no parameters are available and the orbital exponent in the relevant \*.*abp* file is zero. The parameters for these capping atoms, where known, are added to the parameter files along with parameters for normal atoms. The parameters, for example for FN, a capping false fluorine atom meant as a replacement for a nitrogen (plus any other atoms attached to the nitrogen) are indicated in the parameter file associated with atomic number -7. Atomic number +7 is used for the normal nitrogen parameters.

Other considerations (such as mixed quantum and classical forces) arise when considering the complete merger of two or more methodologies in the same molecular system, but since they are not yet available in this release of HyperChem they need not be considered here.

*Note:* The capping atoms are only supported in the semi-empirical quantum mechanics methods in HyperChem. If you want to use the mixed model in the *ab initio* method in HyperChem, you must select an entire molecule or molecules without any boundary atom between the selected and unselected regions and then carry out the calculation.

## Supported Methods

HyperChem currently supports one first-principle method (*ab initio* theory), one independent-electron method (extended Hückel theory), and ten semi-empirical SCF methods (CNDO, INDO, MINDO/3, MNDO, MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO). This section gives sufficient details on each method to serve as an introduction to approximate molecular orbital calculations. For further details, the original papers on each method should be consulted, as well as other research literature. References appear in the following sections.

## Ab Initio Method

*Ab initio* methods are characterized by the introduction of an arbitrary basis set for expanding the molecular orbitals and then the explicit calculation of all required integrals involving this basis set. The same is true of Density Functional (DFT) calculations and much of the formalism below is also valid for DFT calculations. DFT calculations have a different effective Hamiltonian than Hartree-Fock calculations but the SCF procedure used to solve for the molecular orbitals (Kohn-Sham orbitals in one case and Hartree-Fock orbitals in the other case) are very similar. Density Functional calculations are explicitly described in the next chapter and we restrict this chapter to explicit *ab initio* calculations. You may at times need to mentally extend the discussion below to DFT calculations also. DFT calculations are sometimes also referred to as *ab initio* calculations but that terminology is not used here.

*Ab initio* calculations can be performed at the Hartree-Fock level of approximation, equivalent to a self-consistent-field (SCF) calculation, or at a post Hartree-Fock level which includes the effects of correlation — defined to be everything that the Hartree-Fock level of approximation leaves out of a non-relativistic solution to the Schrödinger equation (within the clamped-nuclei Born-Oppenheimer approximation).

HyperChem performs *ab initio* SCF calculations generally. It also can calculate the correlation energy (to be added to the total SCF energy) by a post Hartree-Fock procedure call MP2 that does a Møller-Plesset second-order perturbation calculation. The MP2 procedure is only available for single point calculations and only produces a single number, the MP2 correlation energy, to be added to the total SCF energy at that single point configuration of the nuclei.

HyperChem's *ab initio* calculations solve the Roothaan equations (59) on page 235 without any further approximation apart from the use of a specific finite basis set. Therefore, *ab initio* calculations are generally more accurate than semi-empirical calculations. They certainly involve a more fundamental approach to solving the Schrödinger equation than do semi-empirical methods.

In an *ab initio* method, all the integrals over atomic orbital basis functions are computed and the Fock matrix of the SCF computation is formed (equation (61) on page 235) from the integrals. The Fock matrix divides into two parts: the one-electron Hamiltonian



matrix,  $H$ , and the two-electron matrix,  $G$ , with the matrix elements

$$H_{\mu\nu} = \langle \mu | H^{\text{ef}} | \nu \rangle \quad (92)$$

and

$$G_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} \left[ (\mu\nu | \lambda\sigma) - \frac{1}{2} (\mu\lambda | \nu\sigma) \right] \quad (93)$$

respectively.

An *ab initio* calculation involves the calculation of the following types of integrals:

(1) Overlap integrals:

$$S_{\mu\nu} = \langle \mu | \nu \rangle \quad (94)$$

(2) Kinetic energy integrals:

$$T_{\mu\nu} = \langle \mu | \left( -\frac{1}{2} \nabla^2 \right) | \nu \rangle \quad (95)$$

(3) Nuclear-electron attraction energy integrals:

$$V_{\mu\nu} = \langle \mu | \frac{Z_A}{r_A} | \nu \rangle \quad (96)$$

(4) Electron-electron repulsion energy integrals:

$$(\mu\nu | \lambda\sigma) = \langle \mu\nu | \frac{1}{r_{12}} | \lambda\sigma \rangle \quad (97)$$

The first three types of integrals involve one or two centers. The fourth type of integral involves up to four centers.

To this point, the basic approximation is that the total wave function is a single Slater determinant and the resultant expression of the molecular orbitals is a linear combination of atomic orbital basis functions (MO-LCAO). In other words, an *ab initio* calculation can be initiated once a basis for the LCAO is chosen. Mathematically, any set of functions can be a basis for an *ab initio* calculation. However, there are two main things to be considered in the choice of the basis. First one desires to use the most efficient and accurate functions possible, so that the expansion (equation (49)

on page 232), will require the fewest possible terms for an accurate representation of a molecular orbital. The second one is the speed of two-electron integral calculation.

The Schrödinger equation for a hydrogen atom has an exact solution for the wave function of the form  $\exp(-\alpha r)$ . These Slater Type Orbitals (STO) are good basis functions. However, it is very difficult to calculate analytically the requisite multi-center two-electron integrals using STO basis functions. In 1950, S. F. Boys [S. F. Boys, *Proc. Roy. Soc.*, A200, 542 (1950)] proposed using Gaussian functions, which are of the form  $\exp(-\alpha r^2)$ . With a Gaussian basis set, an MO is expanded as a linear combination of Gaussian Type Orbitals (GTO). Each GTO is a fixed linear combination (contraction) of Gaussian functions (primitives). These contractions can be chosen, if desired, to have a GTO approximate an STO. The introduction of GTOs greatly simplifies the two-electron integral calculation, because the multi-center integrals involving GTOs can be much more easily computed than integrals involving STOs.

## Normalized Primitive Gaussian Functions

A Gaussian function is of the form

$$\exp(-\alpha r_A^2)$$

where  $\alpha$  is a constant. Boys introduced the generalized Cartesian Gaussian function,

$$x_A^l y_A^m z_A^n e^{-\alpha r_A^2}$$

for basis functions. Choosing different values of  $l$ ,  $m$ , and  $n$  gives different types of Gaussian orbitals. Only the lowest principal quantum number for any symmetry is used. That is, all  $s$  orbitals are expanded in  $1s$  primitive Gaussians. The normalized GTO primitives are

*1s*

$$\phi_{1s}^{\text{GF}}(\alpha, r_A) = \left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}} e^{-\alpha r_A^2} \quad (98)$$

*2p<sub>x</sub>*

$$\phi_{2p_x}^{\text{GF}}(\alpha, r_A) = \left( \frac{128\alpha^5}{\pi^3} \right)^{\frac{1}{4}} x_A e^{-\alpha r_A^2} \quad (99)$$

$2p_y$

$$\phi_{2p_y}^{\text{GF}}(\alpha, r_A) = \left( \frac{128\alpha^5}{\pi^3} \right)^{\frac{1}{4}} y_A e^{-\alpha r_A^2} \quad (100)$$

$2p_z$

$$\phi_{2p_z}^{\text{GF}}(\alpha, r_A) = \left( \frac{128\alpha^5}{\pi^3} \right)^{\frac{1}{4}} z_A e^{-\alpha r_A^2} \quad (101)$$

$3d_{xy}$

$$\phi_{3d_{xy}}^{\text{GF}}(\alpha, r_A) = \left( \frac{2048\alpha^7}{\pi^3} \right)^{\frac{1}{4}} x_A y_A e^{-\alpha r_A^2} \quad (102)$$

etc.

## Contracted Gaussian Functions

The contracted Gaussian functions are a linear combination of the primitive Gaussian functions. That is,

$$\phi_{\mu}^{\text{CGF}}(r_A) = \sum_{p=1}^L d_{p\mu} \phi_p^{\text{GF}}(\alpha_p, r_A) \quad (103)$$

where  $\alpha_p$  and  $d_{p\mu}$  are the exponents of the primitives and the contraction coefficients. The contracted basis functions may be chosen to approximate Slater functions, Hartree-Fock atomic orbitals, or any other set of functions desired. Integrals involving such basis functions reduce to the sum of integrals involving the primitive Gaussian functions. Even though many primitive integrals may need to be calculated for each basis function integral, the basis function integrals will be rapidly calculated providing that the method of computing primitive integrals is very fast.

Usually, contractions are determined from atomic SCF calculations. In these calculations one uses a relatively large basis of uncontracted Gaussians, optimizes all exponents, and determines

the SCF coefficients of each of the derived atomic orbitals. The optimized exponents and SCF coefficients can then be used to derive suitable contraction exponents and contraction coefficients for a smaller basis set to be used in subsequent molecular calculations.

### Minimal Basis Sets: STO-3G

A minimal basis set [W. J. Hehre, R. F. Stewart, and J. A. Pople, *J. Chem. Phys.*, 51, 2657 (1969); J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, *J. Chem. Phys.*, 64, 5142 (1976); R. F. Stewart, *J. Chem. Phys.*, 52, 431 (1970)] is a relatively inexpensive one and can be used for calculations on quite large molecules. It is minimal in the sense of having the smallest number of functions per atom required to describe the occupied atomic orbitals of that atom. This is not exactly true, since one usually considers 1s, 2s, and 2p, i.e., five functions, to construct a minimal basis set for Li and Be, for example, even though the 2p orbital is not occupied in these atoms. The 2sp (2s and 2p), 3sp, 4sp, 3d, ..., etc. orbitals are always lumped together as a “shell”, however. The minimal basis set thus consists of 1 function for H and He, 5 functions for Li to Ne, 9 functions for Na to Ar, 13 functions for K and Ca, 18 functions for Sc to Kr, ..., etc. Because the minimal basis set is so small, it generally can not lead to quantitatively accurate results. It does, however, contain the essentials of chemical bonding and many useful qualitative results can be obtained.

Because of the small number of functions in a minimal basis set, it is particularly important that these functions are of near optimum form. This immediately rules out a single Gaussian function. One would prefer to use Slater functions or functions that closely resemble the known shape of atomic orbitals. A significant advance in minimal basis calculations came with the development of computer programs like “Gaussian 70”, which could reproduce the results of minimal basis Slater orbital calculations using contracted Gaussian functions. The STO-NG method uses a contraction of N primitive Gaussians for each basis function, where the contraction coefficients and exponents are chosen so that the basis functions approximate Slater functions.

In the STO-NG method, the 1s, 2s, and 2p Slater functions are expanded in a set of primitive Gaussians

$$\phi_{1s}^{\text{CGF}}(\zeta = 1.0) = \sum_{i=1}^N d_{i,1s} \phi_{i,1s}^{\text{GF}}(\alpha_{i,1s}) \quad (104)$$

$$\phi_{2s}^{\text{CGF}}(\zeta = 1.0) = \sum_{i=1}^N d_{i,2s} \phi_{i,1s}^{\text{GF}}(\alpha_{i,2sp}) \quad (105)$$

$$\phi_{2p}^{\text{CGF}}(\zeta = 1.0) = \sum_{i=1}^N d_{i,2p} \phi_{i,2p}^{\text{GF}}(\alpha_{i,2sp}) \quad (106)$$

where the contraction coefficients (d's) and exponents ( $\alpha$ 's) are to be obtained by a least-squares fit and N is the number of primitive Gaussians, N = 1, 2, 3, ..., 6 corresponding to STO-1G, STO-2G, STO-3G, ..., STO-6G.

One of the unique aspects of the STO-NG method and the fitting procedure is the sharing of contraction exponents in 2sp, 3sp, ..., shells. Thus the exponents in the above three equations are constrained to be identical and the 2s and 2p fits are performed simultaneously. The reason for this constraint is that if 2s and 2p functions have the same exponents, then they have the same radial behavior, and during the radial part of the integral evaluation they can be treated as one function. That is, all integrals involving any sp shell are treated together and one radial integration is sufficient for up to 256 ( $4^4$ ) separate integrals. This grouping of basis functions by shells with shared exponents leads to considerable efficiency in integral evaluation. The general STO-NG procedure uses contraction lengths up to N = 6. However, time spent in integral calculations increases rapidly with the length of the contraction. It has been empirically determined that a contraction of length 3 is sufficient to lead to calculated properties that reproduce essentially all the valence features of a Slater calculation, and STO-3G has become a *de facto* standard for minimal basis calculations.

Once the least-squares fits to Slater functions with orbital exponents  $\zeta = 1.0$  are available, fits to Slater functions with other orbital exponents can be obtained by simply multiplying the  $\alpha$ 's in the above three equations by  $\zeta^2$ . It remains to be determined what Slater orbital exponents  $\zeta$  to use in electronic structure calculations. The two possibilities may be to use the "best atom" exponents ( $\zeta = 1.0$  for H, for example) or to optimize exponents in each

calculation. The “best atom” exponents might be a rather poor choice for molecular environments, and optimization of nonlinear exponents is not practical for large molecules, where the dimension of the space to be searched is very large. A compromise is to use a set of standard exponents where the average values of exponents are optimized for a set of small molecules. The recommended STO-3G exponents are

Element	$\zeta_{1s}$	$\zeta_{2sp}$	$\zeta_{3sp}$
H	1.24		
He	1.69		
Li	2.69	0.80	
Be	3.68	1.15	
B	4.68	1.50	
C	5.67	1.72	
N	6.67	1.95	
O	7.66	2.25	
F	8.65	2.55	
Ne	9.64	2.88	
Na	10.61	3.48	1.75
Mg	11.59	3.90	1.70
Al	12.56	4.36	1.70
Si	13.53	4.83	1.75
P	14.50	5.31	1.90
S	15.47	5.79	2.05
Cl	16.43	6.26	2.10
Ar	17.40	6.74	2.33
...			

The Slater exponents partially listed in the table above are used for all the STO-NG basis sets. The exponents for all the atoms with

atomic numbers less than and equal to 54 are available from HyperChem basis function \*.BAS files.

## Split-Valence Basis Sets

A minimal basis set has rather limited variational flexibility, particularly if exponents are not optimized. The first step in improving the minimal basis set is so-called split valence [J. S. Binkley, J. A. Pople, and W. J. Hehre, *J. Am. Chem. Soc.*, 102, 939 (1980); M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre, *J. Am. Chem. Soc.*, 104, 2797 (1982); W. J. Pietro, M. M. Francl, W. J. Hehre, D. J. Defrees, J. A. Pople, and J. S. Binkley, *J. Am. Chem. Soc.*, 104, 5039 (1982); W. J. Hehre, R. Ditchfield, and J. A. Pople, *J. Chem. Phys.*, 56, 2257 (1972); P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta.*, 28, 213 (1973); M. S. Gordon, *Chem. Phys. Lett.*, 76, 163 (1980); R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople, *J. Chem. Phys.*, 72, 650 (1980); A. D. McLean and G. S. Chandler, *J. Chem. Phys.*, 72, 5639 (1980); T. H. Dunning and P. J. Hay, in *Modern Theoretical Chemistry*, Plenum, New York, 1976] basis set in which two basis functions are used for each valence atomic orbital. Obviously, more computational time is needed by using the split-valence technique than using the minimal basis set for the same molecular system. However, a considerable increase in computational efficiency can be achieved if the exponents of the Gaussian primitives are shared between different basis functions.

At the split-valence level, primitive exponents are shared between s and p functions for the valence shells. In particular, a series of basis sets at the split-valence level has been defined and designated K-LMG where K, L, and M are integers. For the atoms Li to Ne, such a basis consists of an s-type inner-shell function with K primitive Gaussians, an inner set of valence s- and p-type functions with L primitive Gaussians, and another outer sp set with M Gaussians. For the atoms Na to Ar, the basis set consists of an s-type inner-shell with K primitives, an sp-type inner shell functions also with K primitives, an inner set of valence s- and p-type functions with L primitive Gaussians, and another outer sp set with M Gaussians. For hydrogen and helium, only two s-type valence functions (with L and M primitives) are used. The following table lists the KLM values for split-valence basis sets

Basis Set	K	L	M
3-21G	3	2	1

Basis Set	K	L	M
4-21G	4	2	1
6-21G	6	2	1
4-31G	4	3	1
5-31G	5	3	1
6-31G	6	3	1

Thus, the 4-31G basis set consists of 2 functions for H and He, 9 functions for Li to Ne, 13 functions for Na to Ar, ..., etc. For hydrogen the contractions are

$$\phi'_{1s}(r_A) = \sum_{i=1}^3 d'_{i,1s} \phi_{i,1s}^{\text{GF}}(\alpha'_{i,1s}, r_A) \quad (107)$$

and

$$\phi''_{1s}(r_A) = \phi_{i,1s}^{\text{GF}}(\alpha''_{i,1s}, r_A) \quad (108)$$

The outer hydrogen function  $\phi''_{1s}$  is uncontracted and the inner hydrogen function  $\phi'_{1s}$  is a contraction of three primitive Gaussians. The split-valence basis set is not a fit to any particular functional form, but is derived by choosing the form of the contraction and then minimizing the energy of an atomic calculation by varying the contraction coefficients and exponents. The 4-31G acronym implies that the valence basis functions are contractions of three primitive Gaussians (the inner function) and one primitive Gaussian (the outer function), whereas the inner shell functions are contractions of four primitive Gaussians. Hydrogen, of course, does not have inner shells.

For the atoms Li to F, the contractions are

$$\phi_{1s}(r_A) = \sum_{i=1}^4 d_{i,1s} \phi_{i,1s}^{\text{GF}}(\alpha_{i,1s}, r_A) \quad (109)$$

$$\phi'_{2s}(r_A) = \sum_{i=1}^3 d'_{i,2s} \phi_{i,1s}^{\text{GF}}(\alpha'_{i,2sp}, r_A) \quad (110)$$



$$\phi''_{2s}(r_A) = \phi_{i,2s}^{\text{GF}}(\alpha'_{i,2sp}, r_A) \quad (111)$$

$$\phi'_{2p}(r_A) = \sum_{i=1}^3 d'_{i,2p} \phi_{i,2p}^{\text{GF}}(\alpha'_{i,2sp}, r_A) \quad (112)$$

$$\phi''_{2p}(r_A) = \phi_{i,2p}^{\text{GF}}(\alpha''_{i,2sp}, r_A) \quad (113)$$

As in the STO-LG basis, the 2s and 2p functions share the exponents for computational efficiency. The contraction coefficients  $d_{1s}$ ,  $d_{2s}'$ ,  $d_{2s}''$ ,  $d_{2p}'$ , and  $d_{2p}''$  and the contraction exponents  $\alpha_{1s}$ ,  $\alpha_{2sp}'$ , and  $\alpha_{2sp}''$  were explicitly varied until the energy of an atomic SCF calculation reached a minimum. Unlike the STO-NG basis, which was obtained by a least-squares fit to known functions, or general contraction scheme based on contraction of previously determined uncontracted atomic calculations, the 4-31G basis sets were determined by choosing the specific form [equations (109) to (113)] for the contractions and then optimizing all contraction parameters. That is, the basis set was obtained by contraction first, then optimization, as opposed to optimization first, then contraction.

Since the basis set is obtained from atomic calculations, it is still desirable to scale exponents for the molecular environment. This is accomplished by defining an inner valence scale factor  $\zeta'$  and an outer valence scale factor  $\zeta''$  ("double zeta") and multiplying the corresponding inner and outer  $\alpha$ 's by the square of these factors. Only the valence shells are scaled.

The standard 4-31G scale factors are

Element	$\zeta'$	$\zeta''$
H	1.20	1.15
He	1.00	1.00
Li	1.03	1.12
Be	1.03	1.12
B	1.03	1.12
C	1.00	1.04
N	0.99	0.98

Element	$\zeta'$	$\zeta''$
O	0.99	0.98
F	1.00	1.00

## Polarized Basis Sets

The next step in improving a basis set could be to go to triple zeta, quadruple zeta, etc. If one goes in this direction rather than adding functions of higher angular quantum number, the basis set would not be well balanced. With a large number of s and p functions only, one finds, for example, that the equilibrium geometry of ammonia actually becomes planar. The next step beyond double zeta usually involves adding polarization functions, i.e., adding d-type functions to the first row atoms Li-F and p-type functions to H.

To see why these are called polarization functions, consider the hydrogen atom. The exact wave function for an isolated hydrogen atom is just the 1s orbital. If the hydrogen atom is placed in a uniform electric field, however, the electron cloud is attracted to the direction of the electric field, and the charge distribution about the nucleus becomes asymmetric. It is polarized. The lowest order solution to this problem is a mixture of the original 1s orbital and a p-type function, i.e., the solution can be considered to be a hybridized orbital. A hydrogen atom in a molecule experiences a similar, but nonuniform, electric field, arising from its nonspherical environment. Adding polarization functions, e.g., p-type functions, to a basis set for H directly accommodates this effect. In a similar way, d-type functions, which are not occupied in the first and second row atoms, play the role of polarization functions for these atoms.

Using the standard notation, a basis set with a single star (\*) stands for adding a d-type function to heavy atoms (atomic numbers greater than 2) and double stars (\*\*) stands for adding d-type functions to heavy atoms and p-type functions to light atoms (H and He). Hence, for example, 6-31G\* is the basis set 6-31G plus d-type functions added to all the heavy atoms and 6-31G\*\* is the basis set 6-31G plus d-type functions added to all the heavy atoms and p-type functions added to the light atoms (H and He). These rules apply to all the K-LMG and STO-NG basis sets except that the standard 3-21G\* basis set applies d-type functions only to the atoms beyond the first row atoms: that is, there are no d-type functions

added to the first row atoms. It has been empirically determined that adding polarization functions to the heavy atoms is more important than adding polarization functions to hydrogen. This is why the first star in a basis set stands for adding d-type functions to the heavy atoms.

The d-type functions used in the conventional STO-NG are Hermite Gaussians. That is, there are five 3d functions —  $3d_{3z^2-r^2}$ ,  $3d_{xz}$ ,  $3d_{yz}$ ,  $3d_{x^2-y^2}$ , and  $3d_{xy}$  (or  $3d_0$ ,  $3d_{+1}$ ,  $3d_{-1}$ ,  $3d_{+2}$ , and  $3d_{-2}$  respectively) for each d-type function. In contrast, the d-type functions used in the conventional split-valence basis sets (L-KMG) are Cartesian Gaussians. That is, there are six 3d functions —  $3d_{xx}$ ,  $3d_{yy}$ ,  $3d_{zz}$ ,  $3d_{xy}$ ,  $3d_{xz}$ , and  $3d_{yz}$  for each d-type function. These six Cartesian Gaussians are linear combinations of the usual five 3d Hermite Gaussians and a 3s function ( $x^2 + y^2 + z^2$ ). The d-type functions that are added to a basis to form a star basis are a simple set of uncontracted 3d primitive Gaussians (five Hermite Gaussians) in STO-NG basis sets but Cartesian Gaussians in the split-valence basis sets.

Calculations at the 6-31G\* and 6-31G\*\* level provide, in many cases, quantitative results considerably superior to those at the lower STO-3G and 3-21G levels. Even these basis sets, however, have deficiencies that can only be remedied by going to triple zeta (6-311G basis sets in HyperChem) or quadruple zeta, adding more than one set of polarization functions, adding f-type functions to heavy atoms and d-type functions to hydrogen, improving the basis function descriptions of inner shell electrons, etc. As technology improves, it will be possible to use more and more accurate basis sets.

## Two-Electron Integrals

In order to form the Fock matrix of an *ab initio* calculation, all the core-Hamiltonian matrix elements,  $H_{\mu\nu}$ , and two-electron integrals ( $\mu\nu|\lambda\sigma$ ) have to be computed. If the total number of basis functions is  $m$ , the total number of the core Hamiltonian matrix elements is

$$N_1 = \frac{1}{2}m(m+1) \quad (114)$$

after considering the symmetry of the core Hamiltonian matrix element,  $H_{\mu\nu} = H_{\nu\mu}$ ; and the total number of the two-electron integrals is

$$N_2 = \frac{1}{8}(m^4 + 2m^3 + 3m^2 + 2m) \quad (115)$$

After considering the symmetry of the two-electron integrals, we have

$$\begin{aligned} (\mu\nu|\lambda\sigma) &= (\nu\mu|\lambda\sigma) = (\nu\mu|\sigma\lambda) = (\mu\nu|\lambda\sigma) \\ &= (\lambda\sigma|\mu\nu) = (\sigma\lambda|\mu\nu) = (\sigma\lambda|\nu\mu) = (\lambda\sigma|\nu\mu) \end{aligned} \quad (116)$$

The total number of two-electron integrals is thus approximately equal to  $m^4/8$ . For example, for benzene,  $N_1 = 666$  and  $N_2 = 222,111$  in an STO-3G basis set ( $m=36$ ) and  $N_1 = 2,346$  and  $N_2 = 2,753,031$  when using a double zeta basis set such as 6-31G ( $m=68$ ).

Since the total number of two-electron integrals is so large even for a small molecule like benzene, optimal algorithms for computing the two-electron integrals are a major topic in *ab initio* calculations. There are two main concerns for computing the two-electron integrals: first of all, the speed of evaluating the two-electron integrals; second, the method of storing and retrieving the two-electron integrals. J. Pople and co-workers [i.e. see M. H. Gordon and J. A. Pople, *J. Chem. Phys.* 89, 5777 (1988); P. M. W. Gill, M. H. Gordon, and J. A. Pople, *J. Phys. Chem.* 94, 5564 (1990); P. M. W. Gill, B. G. Johnson, and J. A. Pople, *Int. J. Quan. Chem.* 40, 745 (1991)] have developed a series of procedures for computing two-electron integrals.

In an SCF calculation, many iterations may be needed to achieve SCF convergence. In each iteration all the two-electron integrals are retrieved to form a Fock matrix. Fast algorithms to retrieve the two-electrons integrals are important.

## Regular Integral Format

For fast access to the two-electron integrals, a four-dimensional array might be straightforward. The four indices of the four dimensional array correspond to the four basis function indices,  $\mu$ ,  $\nu$ ,  $\lambda$ , and  $\sigma$ , respectively. However, the four dimensional array may take a huge main memory or computer disk space even for a medium-size molecule. Therefore, this may not be practical.

An alternate method is to store the values and indices of two-electron integrals. The advantage is that the integrals with zero values or with a value less than a certain threshold can be ignored. However, the disadvantage is that the indices have to be stored as well as the integrals, and these indices take computer main memory or computer disk space. HyperChem uses this method when the user requests the Regular Integral Format.

HyperChem uses 16 bytes (two double-precision words) of storage for each electron repulsion integral. The first 8 bytes save the compressed four indices and the second 8 bytes store the value of the integral. Each index takes 16 bits. Thus the maximum number of basis functions is 65,535. This should satisfy all users of HyperChem for the foreseeable future.

Because of the use of two double-precision words for each integral, HyperChem needs, for example, about 44 MBytes of computer main memory and/or disk space to store the electron repulsion integrals for benzene with a double-zeta 6-31G basis set.

### Raffenetti Integral Format

Raffenetti [R. C. Raffenetti, *Chem. Phys. Lett.* 20, 335(1973)] proposed another way to store the two-electron integrals in *ab initio* calculations. Raffenetti rewrote (93) on page 262 to read

$$G_{\mu\nu} = \sum_{\lambda \geq \sigma} P'_{\lambda\sigma} D_{\mu\nu,\lambda\sigma} \quad (117)$$

with the modified density matrix P'

$$P'_{\lambda\sigma} = 4P_{\lambda\sigma} \left( 1 - \frac{1}{2} \delta_{\lambda\sigma} \right) \quad (118)$$

and supermatrix D

$$D_{\mu\nu,\lambda\sigma} = \left[ (\mu\nu|\lambda\sigma) - \frac{1}{4}(\mu\lambda|\nu\sigma) - \frac{1}{4}(\mu\sigma|\nu\lambda) \right] \left( 1 - \frac{1}{2} \delta_{[\mu\nu],[\lambda\sigma]} \right) \quad (119)$$

Obviously,

$$P'_{\lambda\sigma} = P_{\sigma\lambda} \quad (120)$$

and

$$D_{\mu\nu, \lambda\sigma} = D_{\nu\mu, \lambda\sigma} = D_{\mu\nu, \sigma\lambda} = D_{\lambda\sigma, \mu\nu} = \dots \quad (121)$$

So only the two-electron integrals with  $\mu \geq \nu$ , and  $\lambda \geq \sigma$  and  $[\mu\nu] \geq [\lambda\sigma]$  need to be computed and stored.  $D_{\mu\nu, \lambda\sigma}$  only appears in  $G_{\mu\nu}$ , and  $G_{\nu\mu}$ , whereas the original two-electron integrals contribute to other matrix elements as well. So it is much easier to form the Fock matrix by using the supermatrix  $D$  and modified density matrix  $P'$  than the regular format of the two-electron integrals and standard density matrix.

The Raffenetti integral format emphasizes the speed of computing the Fock matrix. the Raffenetti format is used only with *ab initio* calculations and is not available for the Density Functional calculation of the next chapter.

## Two-Electron Integral Cutoff

The total number of two-electron integrals is proportional to  $m^4$  for a molecular system with  $m$  basis functions. Some of these integrals may be zero because of the symmetry, and some may be very small in magnitude. Using the regular integral format or the Raffenetti integral format, each integral value and its indices take 16 bytes and all are saved to the computer main memory or disk. All saved two-electron integrals are then used in forming the Fock matrix in every iteration. Those integrals with zero value or with a very small magnitude do not make much contribution to the Fock matrix and to the total energy. Neglecting these integrals may not affect the accuracy of *ab initio* calculations. Thus, in order to save computer main memory or disk space and speed up the calculation of the SCF procedure, a two-electron integral cutoff is introduced. HyperChem uses the two-electron integral cutoff to determine which of the two-electron integrals need to be saved. The value of  $10^{-10}$  (Hartree) generally is good enough for most calculations. A small value is recommended for “tight” calculations and a large value for “loose” calculations.

## Direct SCF Calculation

Since the first formulation of the MO-LCAO finite basis approach to molecular Hartree-Fock calculations, computer applications of the method have conventionally been implemented as a two-step process. In the first of these steps a (large) number of integrals — mostly two-electron integrals — are calculated and stored on external storage. The second step then consists of the iterative solution

of the Roothaan equations, where the integrals from the first step are read once for every iteration.

In any scheme for performing *ab initio* calculations, the limitations of accuracy, as well as to the size of the system that can be studied are set by the two-electron integrals. Their number is of the order of  $m^4 / 8$ , where  $m$  is the number of basis functions; roughly proportional to the size of the system studied. If only integrals whose magnitude exceeds a certain threshold are considered, the size dependence is asymptotically reduced to  $m^2 \log m$  [J. Almlof, K. Faegri Jr., and K. Korsell, *J. Comp. Chem.* 3, 385 (1982)]. However, in practical applications that threshold must be tightened as large systems are considered, in order to reduce the accumulation of errors. Significant deviation from the  $m^4$  dependence is therefore only seen for extended systems, and a calculation on a molecule of chemically interesting size may easily require  $10^7$  to  $10^9$  integrals even for rather modest basis sets. Such large *ab initio* calculations may not be practical on a small- or medium-size computer.

Direct SCF calculations [J. Almlof, K. Faegri Jr., and K. Korsell, *J. Comp. Chem.* 3, 385 (1982)] offer a solution to this problem by eliminating the storage of two-electron integrals. This can, however, only be done at the expense of having to recompute integrals for every iteration.

With the current impressive CPU and main memory capacity of relatively inexpensive desktop PC's, non-direct SCF *ab initio* calculations involving 300-400 basis functions can be practical. However, to run these kinds of calculation, 20 GBytes of hard disk space might be needed. Such big disk space is unlikely to be available on desktop PCs. A direct SCF calculation can eliminate the need for large disk storage.

## Initial Guess of MO Coefficients

In any SCF method, an initial estimate of the MO coefficients is necessary before starting the SCF iterations. Normally, the initial guess to the MO coefficients is obtained by solving the Hartree-Fock Roothaan equations with replacement of the Fock matrix by the core Hamiltonian. The core Hamiltonian gives a reasonable good initial guess in most cases. In some cases, however, the core Hamiltonian gives a wrong order for the energy levels or the initial guess MO coefficients are far from the actual MO coefficients.

HyperChem provides alternative methods for generating an initial guess of the MO coefficients. These alternative methods are:

### Projected Hückel

HyperChem performs an empirical Hückel calculation to produce the MO coefficients for a minimal basis set and then projects these coefficients to the real basis set used in an *ab initio* calculation. The projected Hückel guess can be applied to molecular systems with an atomic number less than or equal to 54 (Xe).

### Projected CNDO/INDO

A projected CNDO/INDO guess uses the computed coefficients from a minimum basis set CNDO/INDO calculation and then projects these MO coefficients to the real basis set used in an *ab initio* calculation. The projected CNDO/INDO guess can be applied to molecular systems with atomic number less than or equal to 18 (Ar).

### Mixed Model

Selection is used for many purposes in HyperChem, normally in a select-operate mode. When a quantum mechanical calculation is requested, either semi-empirical or *ab initio*, selection serves to separate the system into a classical region and a quantum mechanical region (the selected region). HyperChem treats the unselected region of any molecular system as a classical perturbation for the quantum mechanical selected region. The charges in the unselected region form a background of electrostatic potential and perturb the properties of the selected quantum mechanical region. The interaction of the charges in the classical region with the quantum mechanical region are treated correctly by including them in the one-electron core Hamiltonian for the quantum region. The contribution of the charges in the classical region, to the core Hamiltonian, is treated in the following way:

$$\Delta H_{\mu\nu} = \sum_B \langle \mu | \frac{Q_B}{R_{1B}} | \nu \rangle \quad (122)$$

where the sum runs over all the atoms, B, in the unselected region.  $R_{1B}$  is the distance between the nucleus of B and the electron, and



$Q_B$  is the net charge (the nuclear charge minus the electron charge) on atom B.

## Ghost-Atoms

Atoms with an atomic number that is sometimes treated as if it were zero are known as ghost atoms. That is, a ghost atom is non-interacting and has effectively no nucleus or electrons. HyperChem defines a ghost atom in exactly the same way as a regular atom, and any basis set can be applied to a ghost atom. Under most situations ghost-atoms is just the name for a set of atoms (a named selection) and they act as any other atom. When performing an *ab initio* calculation, however, HyperChem may treat ghost atoms differently than regular atoms. A ghost atom is represented by a basis set, but does not have any nuclei attached to it, and therefore can be regarded as a point in space upon which orbitals can be centered. The concept of ghost atoms can be very useful in avoiding the basis set superposition error (BSSE) when describing a so called “basis set deficient” systems, i.e. systems with relatively large number of electrons represented by relatively poor basis sets. Some long range interactions (i.e. complexes, hydrogen bonding) can be poorly represented without the use of a ghost atom concept. Also ghost atoms provide a convenient way to specify some arbitrary point in molecular space at which one wishes to place basis functions.

In the current version of HyperChem, this option is only available for Single Point calculation.

## Extended Hückel Theory

Extended Hückel theory (EHT) is principally attributed to the work of Roald Hoffmann and his collaborators at Cornell University although other groups have certainly contributed to the development. EHT is normally used just as a rapid method for computing molecular orbitals, to investigate the orbital structure of molecules. The energies produced have to be somewhat suspect. For example, optimizing a molecule, i.e., minimizing the energy to predict the structure of the molecule, cannot give satisfactory results and is disallowed in HyperChem.

Since no iterations are involved, it is faster than the semi-empirical SCF methods and no convergence difficulties arise.

## Basic Method

Since EHT is an independent-electron method, it is defined by giving formulas for each of the  $H_{\mu\nu}$  matrix elements of the effective one-electron Hamiltonian  $\mathcal{H}^{\text{eff}}$ , described above. Once the matrix  $\mathbf{H}$  is available, the molecular orbitals  $\psi_i$  and their orbital energies  $\varepsilon_i$  are just solutions of a standard eigenvalue problem  $\mathbf{H}\mathbf{C} = \mathbf{S}\mathbf{C}\varepsilon$  where  $\mathbf{C}$  describes the MO-LCAO coefficients of the molecular orbitals in the atomic orbital basis set:

$$\psi_i = \sum_{\nu} C_{\nu i} \phi_{\nu} \quad (123)$$

and  $\varepsilon$  is a diagonal matrix of the orbital energies. The method thus requires:

- Specification of the atomic orbital basis set  $\{\phi_{\mu}\}$ .
- Computation of the overlap integrals  $S_{\mu\nu}$  in the atomic orbital basis set.
- Specification of the EHT Hamiltonian matrix elements  $H_{\mu\nu}$  in the atomic orbital basis set.

## Valence Atomic Orbitals

Most simple empirical or semi-empirical molecular orbital methods, including all of those used in HyperChem, neglect inner shell orbitals and electrons and use a *minimal basis set of valence Slater orbitals*.

Thus for H and He, the basis set consists of one orbital, a 1s atomic orbital. For atoms Li to Ne the 2 inner-shell electrons are combined with the nucleus and the basis set consists of 4 orbitals, the 2s, 2p<sub>x</sub>, 2p<sub>y</sub>, and 2p<sub>z</sub> atomic orbitals. This means, for example, that carbon has a nucleus with an effective positive charge of (6-2)=4, not the usual 6. For atoms Na to Al, the 10 inner shell electrons are combined with the nucleus and the basis set consists of the 4 orbitals, the 3s, 3p<sub>x</sub>, 3p<sub>y</sub>, and 3p<sub>z</sub> atomic orbitals. For atoms Si to Cl, there is the question whether to include the 3d orbitals into the basic set as they are close in energy to the 3s and 3p orbitals, but are unoccupied. The standard option of EHT in HyperChem includes the d-orbitals so that the basis set for Si to Cl consists of 9 orbitals, 3s,

$3p_x$ ,  $3p_y$ ,  $3p_z$ ,  $3d_z^2$ ,  $3d_{xz}$ ,  $3d_{yz}$ ,  $3d_{xy}$ , and  $3d_{x^2-y^2}$ . An option in the EHT Option dialog box neglects the 3d orbitals to speed up calculations. The standard parameter set, however, is derived assuming 3d orbitals are used.

So called Hydrogenic atomic orbitals (exact solutions for the hydrogen atom) have radial nodes (values of the distance  $r$  where the orbital's value goes to zero) that make them somewhat inconvenient for computation. Results are not sensitive to these nodes and most simple calculations use Slater atomic orbitals of the form

$$\phi_v(x, y, z) = \phi_v(r, \theta, \varphi) = N_{n,l,m} r^{n-1} e^{-\zeta r} Y_{l,m}(\theta, \varphi) \quad (124)$$

where  $N_{n,l,m}$  is a normalization constant,  $\zeta$  is the orbital exponent and  $Y$  is an angular function distinguishing  $s(l=0)$  from  $p(l=1)$  from  $d(l=2)$ , etc. A  $3p_x$  orbital, for example, has identical angular behavior to a  $2p_x$  orbital but differs in its radial function by an extra power of  $r$  and its orbital exponent  $\zeta$ . The orbital exponents are usually common between  $ns$  and  $np$  orbitals but different for  $nd$ . The default orbital exponents of the first three row elements used by HyperChem are shared by  $s$  and  $p$  orbitals and are

Element	Zeta (s and p)	Zeta (d)
H	1.3	
Li	0.650	
Be	0.975	
B	1.3	
C	1.625	
N	1.950	
O	2.275	
F	2.425	
Na	0.733	
Mg	0.950	
Al	1.167	
Si	1.383	1.383

Element	Zeta (s and p)	Zeta (d)
P	1.600	1.4
S	1.817	1.5
Cl	2.033	2.033

## Hamiltonian Matrix Elements

The diagonal matrix elements  $H_{\mu\mu}$ , for an atomic orbital  $\phi_{\mu}$ , can be shown to bear a very close association with the energy needed to ionize (remove) an electron from the orbital. Valence state ionization energies (VSIEs) can be determined from experiment and extended Hückel theory equates  $H_{\mu\mu}$  to the corresponding VSIE, with opposite sign:

$$H_{\mu\mu} = -\text{VSIE} \quad (125)$$

The VSIE values of the first three row elements used by HyperChem are shown in the following table.

ELEMENT	S-VSIE	P-VSIE	D-VSIE
H	13.6		
Li	5.4	3.5	
Be	10.0	6.0	
B	15.2	8.5	
C	21.4	11.4	
N	26.0	13.4	
O	32.3	14.8	
F	40.0	18.1	
Na	5.1	3.0	
Mg	9.0	4.5	
Al	12.3	6.5	
Si	17.3	9.2	6.0

ELEMENT	S-VSIE	P-VSIE	D-VSIE
P	18.6	14.0	7.0
S	20.0	13.3	8.0
Cl	30.0	15.0	9.0

The off-diagonal elements of Extended Hückel theory,  $H_{\mu\nu}$  ( $\mu \neq \nu$ ) represent the effects of bonding between the atoms and are assumed to be proportional to the overlap,  $H_{\mu\nu} \sim S_{\mu\nu}$ . An approximation for differential overlap referred to as the Mulliken approximation

$$\phi_{\mu}\phi_{\nu} = S_{\mu\nu} \frac{(\phi_{\mu}^2 + \phi_{\nu}^2)}{2} \quad (126)$$

can be used to estimate the off-diagonal elements

$$H_{\mu\nu} = kS_{\mu\nu} \frac{(H_{\mu\mu} + H_{\nu\nu})}{2} \quad (127)$$

The Hückel constant  $k$  has been inserted here as one more adjustable parameter. Note that the integrated form of equation (126) is exact.

## Choosing a Hückel Constant

The constant  $k$  can be adjusted to give best agreement with experiment. It is found that a good value to use is somewhat larger than would be indicated by the Mulliken approximation and a very standard value used by many groups is:

$$k = 1.75 \quad (128)$$

Alternatively, a slight modification to this formula makes  $k$  a function of the specific orbital pair,  $k_{\mu\nu}$ , rather than identical for each matrix element  $H_{\mu\nu}$ :

$$k_{\mu\nu} = k + \delta^2 + \delta^4(1 - k) \quad (129)$$

where

$$\delta = (H_{\mu\mu} - H_{\nu\nu}) / (H_{\mu\mu} + H_{\nu\nu}) \quad (130)$$

The derivation of this result introduces a generalization of the Mulliken approximation:

$$\phi_{\mu}\phi_{\nu} = 1/2S_{\mu\nu}((1 + \delta)\phi_{\mu}^2 + (1 - \delta)\phi_{\nu}^2) \quad (131)$$

where the weighting is based on a measure of the relative diffuseness,  $\delta$ , of  $\phi_{\mu}$  and  $\phi_{\nu}$ . HyperChem allows you to use a simple constant  $k$  or a diffuseness weighted constant  $k_{\mu\nu}$ .

## Mixed Model

In the Extended Hückel approximation, the charges in the unselected part are treated like classical point charges. The correction of these classical charges to the diagonal elements of the Hamiltonian matrix may be written as:

$$\Delta H_{\mu\mu} = -a \sum_j \frac{Q_j}{R_{ij}}$$

and the off-diagonal elements of the Hamiltonian matrix are computed based on the corrected diagonal elements. In this equation,  $a$  is a scale factor for the classical-quantum interaction,  $\mu$  is the  $\mu$ th atomic orbital belonging to atom  $i$ ,  $Q_j$  is a pure charge on atom  $j$  in the classical region,  $R_{ij}$  is the distance between atom  $i$  in the selected region and atom  $j$  in the unselected region, and the sum is over all atoms in the unselected region.

## CNDO

HyperChem implements the CNDO method based upon a complete neglect of differential overlap approximation. Specifically, HyperChem implements the CNDO/2 scheme developed by Pople, Segal, and Santry [J. A. Pople, D. P. Santry, and G. A. Segal, *J. Chem. Phys.*, 43, S129 (1965); J. A. Pople and G. A. Segal, *J. Chem. Phys.*, 43, S136 (1965); J. A. Pople and G. A. Segal, *J. Chem. Phys.*, 44, 3289 (1966)]. The method is the simplest of the zero differential overlap (ZDO) methods and generally thought to be the least accurate. The CNDO/1 method, as originally developed, was quickly modified and re-parameterized to CNDO/2, which is the method that has remained in common use for many years.

### The CNDO equations

The form of the CNDO/2 equations for the Fock matrix comes from the basic approximation  $\phi_\mu \phi_\nu = \delta_{\mu\nu} \phi_\mu \phi_\mu$ . This reduces the general two-electron integral  $(mn|ls)$  to  $(mm|ll)$ . However, in order that this approximation have rotational invariance (a calculation must give the same result if a molecule is rotated in the molecular coordinate system) it is necessary to make the further approximation that this last integral is independent of the specific orbital  $\phi_\mu$  or  $\phi_\lambda$  but dependent only on the atom A or atom B that the orbital resides on

$$(\mu\nu|\lambda\sigma) = \delta_{\mu\nu} \delta_{\lambda\sigma} \gamma^{AB} \quad (132)$$

where  $\gamma^{AB}$  is the average electrostatic repulsion between an electron on atom A (in any orbital) and an electron on atom B (in any orbital). These Coulomb integrals are fundamental to many semi-empirical methods. Sometimes they are computed (using spherical ns atomic orbitals for the mathematical functions) and sometimes they are parameterized to experimental results. In CNDO/2 they are computed.

The elements of the CNDO/2 Fock matrix (for the RHF case) thus become

$$F_{\mu\mu} = H_{\mu\mu} + \left( P_{AA} - \frac{1}{2} P_{\mu\mu} \right) \gamma^{AA} + \sum_{B \neq A} P_{BB} \gamma^{AB} \quad (133)$$

$$F_{\mu\nu} = H_{\mu\nu} - \frac{1}{2} P_{\mu\nu} \gamma^{AB} \quad (134)$$

In a ZDO computation where the overlap matrix is assumed to be the unit matrix, the diagonal elements of the density matrix become occupancy numbers for the atomic orbitals. Thus  $P_{\mu\mu}$  is the effective number of electrons occupying atomic orbital  $\phi_{\mu}^{\mu}$ . The sum of these numbers over all orbitals on a given atom is the total number of electrons,  $P_{AA}$  residing on that atom, in this case atom A. The net atomic charge is then the charge of the nucleus (with collapsed inner shell electrons)  $Z_A$  minus  $P_{AA}$

$$q_A = Z_A - P_{AA} = Z_A - \sum_{\mu}^A P_{\mu\mu} \quad (135)$$

It remains to specify the elements of the one-electron core Hamiltonian,  $H_{\mu\nu}$ , containing the kinetic energy and nuclear attraction integrals.

The diagonal elements of the core Hamiltonian simply represent the energy of an electron in an atomic orbital of the corresponding atom,  $U_{\mu\mu}$ , plus the attraction of an electron in that atomic orbital (on atom A) for the other nuclei (atoms B),

$$H_{\mu\mu} = U_{\mu\mu} - \sum_{B \neq A} V_{AB} \quad (136)$$



The off-diagonal elements represent the dominant effects of bonding and are set to be proportional to the overlap by a parameter dependent on the two atoms involved in the overlap,

$$H_{\mu\nu} = \beta_{AB}^O S_{\mu\nu} \quad (137)$$

One center overlaps are all zero so that the above describes only two center contributions.

It was found that computed bond lengths were much too short in CNDO/1 because of poor balance between the attractive  $V_{AB}$  terms and the repulsive  $\gamma^{AB}$  terms. Good theoretical justification can be made for a further simplification that approximates the nuclear attraction terms  $V_{AB}$  as follows,

$$V_{AB} = Z_B \gamma^{AB} \quad (138)$$

This approximation leads to the CNDO/2 scheme. It remains to explore the values of  $U_{\mu\mu}$  which are closely related to valence state ionization energies, as in Extended Hückel theory. In the CNDO/1 scheme they were set to just that. The CNDO/2 scheme considers the electron affinity (the process of adding an electron to orbital  $\phi_\mu$ ) as well as the ionization potential (subtracting an electron) and develops the following expression for  $U_{\mu\mu}$

$$U_{\mu\mu} = -\frac{1}{2}(I_\mu + A_\mu) - \left(Z_A - \frac{1}{2}\right) \gamma^{AA} \quad (139)$$

where  $I_\mu$  is the ionization potential of an electron in atomic orbital  $\phi_\mu$ ,  $A_\mu$  is the electron affinity of the same orbital (determined by adding an electron to the ionized atom) and the average of the two is a measure of the chemical concept, *electronegativity*.

The final expressions for the (UHF) Fock matrices are

$$F_{\mu\mu}^\alpha = -\frac{1}{2}(I_\mu + A_\mu) + \left[ (P_{AA} - Z_A) - \left( P_{\mu\mu}^\alpha - \frac{1}{2} \right) \right] \gamma^{AA} + \sum_{B \neq A} (P_{BB} - Z_B) \gamma^{AB} \quad (140)$$

$$F_{\mu\nu}^{\alpha} = \beta_{AB}^0 S_{\mu\nu} - P_{\mu\nu}^{\alpha} \gamma^{AB} \quad (141)$$

The RHF form of these equations simply substitutes  $\mathbf{P}^T = 1/2 \mathbf{P}^{\alpha} = 1/2 \mathbf{P}^{\beta}$

The CNDO parameters are discussed below.

## Expectation Values

You would normally expect to make the same zero differential overlap (ZDO) approximation in computing expectation values as in deriving the form for the Fock matrices. However, the original authors of the CNDO method found that when considering the dipole moment such a strict ZDO approximation reduces the dipole moment to just the classical value computed from the point charges given by equation (135) on page 285. It is expected, however, that one-center (s, p<sub>x</sub>), (s, p<sub>y</sub>), or (s, p<sub>z</sub>) hybridization terms are significant contributors to the dipole moment and these terms were included by the original CNDO authors. HyperChem continues to include these hybridization terms when computing dipole moments.

## Mixed Model (CNDO and INDO)

In the CNDO and INDO level of approximation, the interaction of the charges between the quantum mechanical region and the classical region is treated semi-empirically. The charges appearing in the classical region affects the charge distribution and the energy level of the quantum mechanical region. The contribution of the charges (including the electronic charges and the nuclear charges) in the classical region to the core Hamiltonian is treated in the following way:

$$\Delta H_{\mu\mu} = \sum_B (Z_B - Q_B) \gamma_{AB}$$

$\Delta H_{\mu\mu}$  corrects the core Hamiltonian element,  $H_{\mu\mu}$ , and

$$\Delta E_N = \sum_{(A,B)} \left( \frac{Z_A Z_B}{R_{AB}} - Z_A Q_B \gamma_{AB} \right)$$

corrects the nuclear-nuclear interaction energy. In these two equations, A is the index for an atom in the quantum region and B for

an atom in the classical region,  $Z_A$ , is the nuclear charge of atom A with a number of the valence electrons,  $Q_B$  is the electronic charge,  $R_{AB}$  is the distance between atom A and atom B, and  $\gamma_{AB}$  is the Coulomb interaction of two electrons located on atom A and atom B, respectively:

$$\gamma_{AB} = (nsns|n'sn's)$$

a Coulomb integral of the Slater orbitals. Therefore, in order to run a mixed model at the CNDO and INDO approximation level, HyperChem needs the Slater exponents of s-orbitals for the atoms in the classical region and limits the elements in the classical region to be in the range of H through Xe.

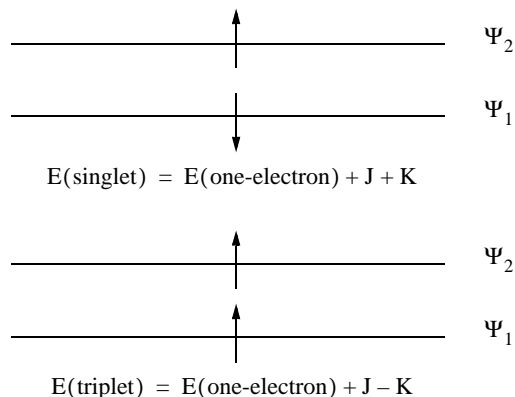
## INDO

### Exchange Phenomena

The difference between CNDO and INDO is best understood in relation to the phenomenon of exchange, which we briefly describe here.

Consider first of all the problem of two electrons, one occupying orbital  $\psi_1$  and the other occupying a different orbital  $\psi_2$ . The two electrons can have their electron spins parallel or anti-parallel, as shown below. The parallel arrangement leads to a triplet state with total spin  $S=1$  and the anti-parallel arrangement leads to a singlet state with total spin  $S=0$ . The total energy of the two states includes the kinetic energy of the electrons, the attraction of the electrons for any fixed nuclei (these two contributions sum to give the one-electron energy contributions) and electron-electron interactions. In each of the two cases there is a *Coulomb* interaction  $J = (\psi_1\psi_1|\psi_2\psi_2)$  between the two electrons. The difference in energy between the two states, with the triplet having the lower energy, comes from a special quantum mechanical effect called *electron exchange*. Since electrons are all identical, they can be exchanged without changing any observable. That electrons have individual spins,  $s=1/2$ , results in the quantum mechanical effect that the triplet (parallel spin) arrangement has a lower energy than the singlet (anti-parallel spin) associated with the *exchange* interaction  $K = (\psi_1\psi_2|\psi_1\psi_2)$ . The CNDO procedure basically neglects most of these exchange effects leading to a poor treatment of the differ-

ences between different spin states of molecules. The INDO procedure restores the major exchange interactions neglected in CNDO theory.



## Differences Between INDO and CNDO

The INDO (Intermediate Neglect of Differential Overlap) differs from CNDO in the treatment of one-center exchange integrals. The CNDO (Complete Neglect of Differential Overlap) treatment retains only the two-electron integrals  $(\mu\mu|\nu\nu) = \gamma_{\mu\nu}$ . The  $\gamma_{\mu\nu}$  are atomic Coulomb integrals approximated in CNDO by  $\gamma^{AB}$ . In order to take at least minimal account of exchange phenomena, the INDO procedure restores integrals based on *monatomic differential overlap*. That is, it retains, in addition to the CNDO terms, all integrals  $(\mu\nu|\lambda\sigma)$ , where the atomic orbitals  $\phi_\mu$ ,  $\phi_\nu$ ,  $\phi_\lambda$ , and  $\phi_\sigma$  are on the same atom A. In practice, this amounts to adding all one-center exchange integrals  $(\mu\nu|\mu\nu)$  as well as distinguishing between different one-center coulomb integrals  $\gamma^{AA}$ .

The one-center exchange integrals that INDO adds to the CNDO scheme can be related to the Slater-Condon parameters  $F^0$ ,  $G^1$ , and  $F^2$  used to describe atomic spectra. In particular, for a set of  $s$ ,  $p_x$ ,  $p_y$ ,  $p_z$  atomic orbitals, all the one-center integrals are given as:

$$\begin{aligned}
 (ss|ss) &= (ss|xx) = F^0 = \gamma^{AA} \\
 (sx|sx) &= \frac{1}{3}G^1 \\
 (xy|xy) &= \frac{3}{25}F^2
 \end{aligned} \tag{142}$$

$$(xx|xx) = F^0 + \frac{4}{25}F^2$$

$$(xx|yy) = F^0 - \frac{2}{25}F^2$$

There are similar expressions for symmetry related integrals (ss|yy), etc. For direct comparison with CNDO,  $F^0$  is computed as in CNDO. The other INDO parameters,  $G^1$ , and  $F^2$ , are generally obtained [J. I. Slater, *Quantum Theory of Atomic Structure*, McGraw-Hill Book Company, Vol. 1, New York, 1960.] from fits to experimental atomic energy levels, although other sources for these Slater-Condon parameters are available. The parameter file CINDO.ABP contains the values of  $G^1$  and  $F^2$  (columns 9 and 10) in addition to the CNDO parameters.

## Spin Interactions in INDO

The advantages of INDO over CNDO involve situations where the spin state and other aspects of electron spin are particularly important. For example, in the diatomic molecule NH, the last two electrons go into a degenerate p-orbital centered solely on the Nitrogen. Two well-defined spectroscopic states,  $^3S^-$  and  $^1D$ , result. Since the p-orbital is strictly one-center, CNDO results in these two states having exactly the same energy. The INDO method correctly makes the triplet state lower in energy in association with the exchange interaction included in INDO.

A second common example of the utility of INDO over CNDO is the methyl radical  $CH_3$ . This planar molecule is a free radical with an unpaired electron occupying a p-orbital centered on the carbon. The p-orbital is purely a p-orbital with a node in the plane of the molecule; CNDO results in zero spin density at the positions of the carbon and hydrogen nuclei. Electron Spin Resonance (ESR) spectra measure quantities proportional to the spin density at the nucleus and show that it is non-zero (negative) at the position of the hydrogen nucleus. An INDO calculation allows the spin of the unpaired electron to affect the spin distribution in the s-orbitals (because of one-center exchange interactions) and correctly results in negative spin density at the position of the hydrogen nucleus. Obtaining this result requires, in addition to INDO, the use of unrestricted Hartree-Fock (UHF) wave functions, which allow different spatial distributions for electrons of different spins.

While INDO calculations have more parameters and are somewhat more complex than CNDO calculations, they require essentially no extra computation time and in most situations they are superior to CNDO calculations.

## MINDO/3

MINDO/3 stands for Modified Intermediate Neglect of Differential Overlap, *version 3*; it is a modification of the INDO method. The method was proposed and developed by M. J. S. Dewar and co-workers at the University of Texas at Austin [N. C. Baird and M. J. S. Dewar, *J. Chem. Phys.*, 50, 1262(1969); M. J. S. Dewar and E. Haselbach, *J. Am. Chem. Soc.*, 92, 590(1970); M. J. S. Dewar and D. H. Lo, *J. Am. Chem. Soc.*, 94, 5296(1972); R. C. Bingham, M. J. S. Dewar, and D. H. Lo, *J. Am. Chem. Soc.*, 97, 1285(1975); 97, 1302(1975); 97, 1307(1975); 97, 1311(1975)].

Using the same nomenclature as for the INDO approximation, the elements of the MINDO/3 UHF Fock matrix are described below. When  $\phi_\mu$  and  $\phi_\nu$  are on different centers the off-diagonal elements are

$$F_{\mu\nu}^\alpha = H_{\mu\nu} - P_{\mu\nu}^\alpha(\mu\mu|v\nu) \quad (143)$$

( $\mu$  on A,  $\nu$  on B,  $A \neq B$ )

and when  $f_m$  and  $f_n$  are on the same center, the off-diagonal elements are

$$F_{\mu\nu}^\alpha = (2P_{\mu\nu}^{\alpha+\beta} - P_{\mu\nu}^\alpha)(\mu\nu|\mu\nu) - P_{\mu\nu}^\alpha(\mu\mu|v\nu) \quad (144)$$

( $\mu$  on A,  $\nu$  on A,  $\mu \neq \nu$ )

The diagonal elements of the Fock matrix are

$$F_{\mu\mu}^\alpha = H_{\mu\mu} + \sum_v^A (P_{v\nu}^{\alpha+\beta}(\mu\mu|v\nu) - P_{v\nu}^\alpha(\mu\nu|\mu\nu)) \quad (145)$$

$$+ \sum_B \gamma^{AB} \sum_A^B P_{\lambda\lambda}^{\alpha+\beta} \quad (146)$$

where  $\mathbf{P}^T \equiv \mathbf{P}^{\alpha+\beta} \equiv \mathbf{P}^\alpha + \mathbf{P}^\beta$ . By replacing the superscripts  $\alpha$  and  $\beta$  by  $\beta$  and  $\alpha$ , respectively, in the above three equations, you can easily get three similar equations for the Fock matrix elements for beta orbitals. Similar expressions to the above for Fock matrix elements of restricted Hartree-Fock (RHF) calculations can be generated by simply replacing  $\mathbf{P}^\alpha$  (or  $\mathbf{P}^\beta$ ) by  $1/2 \mathbf{P}^T$  in the above equations.

The two-center two-electron, one-center two-electron, two-center one-electron, one-center one-electron, and core-core repulsion integrals involved in the above equations are discussed below.

## Two-Center Two-Electron Integrals

All non-zero integrals over atomic orbitals on the two centers are set equal, as in CNDO/INDO, to an averaged  $\gamma^{AB}$ . Thus,  $(s_A s_A | s_B s_B) = (s_A s_A | p_B p_B) = (p_A p_A | p_B p_B) = \gamma^{AB}$ . The two-center Coulomb integrals, rather than being calculated from first principles using  $s$  orbitals as in CNDO/INDO, are approximated by an Ohno-Klopman [K. Ohno, *Theor. Chim. Acta*, 2, 219 (1964); G. Klopman, *J. Amer. Chem. Soc.*, 86, 4550 (1964); 87, 3300 (1965)] relation as follows:

$$\gamma^{AB} = \frac{1}{\left( R_{AB}^2 + \frac{1}{2} \left[ \frac{1}{A_A} + \frac{1}{A_B} \right]^2 \right)^{1/2}} \quad (147)$$

where  $A_A$  and  $A_B$  are the averages of the appropriate one center two-electron integrals.

## One-Center One-Electron Integral $H_{\mu\mu}$

The one-center one-electron integral  $H_{\mu\mu}$  represents the energy that an electron in an atomic orbital  $\phi_\mu$  would have if all other valence electrons were removed to infinity. This is calculated by adding to the one-electron energy of the atomic orbital in the fully ionized atom the potential due to all the other nuclei in the system. The one-electron energy is obtained parametrically, and is given by the symbol  $U_{\mu\mu}$ .  $H_{\mu\mu}$  is thus derived from the fundamental equation, identical to that of CNDO/INDO,



$$H_{\mu\mu} = U_{\mu\mu} - \sum_{B \neq A} V_{AB} \quad (148)$$

The core-electron integral,  $V_{AB}$ , as in CNDO/INDO, is then equated to the corresponding two-electron integral:

$$V_{AB} = Z_B \gamma^{AB}$$

The one-center integrals,  $U_{\mu\mu}$ , are computed according to techniques originally developed by Oleari et. al. [L. Oleari, L. DiSipio, and G. De Michelis, *Mol. Phys.*, 10, 97 (1966)].

### Two-Center One-Electron Integral $H_{\mu\nu}$ (Resonance Integral)

The two-center one-electron integral  $H_{\mu\nu}$ , sometimes called the resonance integral, is approximated in MINDO/3 by using the overlap integral,  $S_{\mu\nu}$ , in a related but slightly different manner to CNDO/INDO as

$$H_{\mu\nu} = S_{\mu\nu} \beta_{AB} (I_{\mu} + I_{\nu}) \quad (149)$$

where  $\beta_{AB}$  is a adjustable parameter dependent on the specific element pair (unlike CNDO/INDO where it ends up as an average over the two elements involved), and  $I_{\mu}$  and  $I_{\nu}$  are the ionization potentials of the atomic orbitals  $\mu$  and  $\nu$  on atoms A and B respectively.

### One-Center Two-Electron Integral

The one-center two-electron integrals used in MINDO/3 are derived from an analysis of atomic spectra [see, e.g., N. C. Baird and M. J. S. Dewar, *J. Chem. Phys.* 50, 1262(1969)]. Following Pople et al. [J. A. Pople, D. L. Beveridge, and P. A. Dobosh, *J. Chem. Phys.*, 47, 2026(1967)], the one-center two-electron integrals are written in terms of the Slater-Condon  $F^k$  and  $G^k$  parameters exactly as in equation (111) on page 230 for INDO. MINDO/3 uses the values quoted by Pople et al. for  $G^1$  and  $F^2$ , and calculates  $F^0$  for each atom.

## Core-Core Repulsion Integrals

From simple electrostatics the core-core repulsion (as used, for example, in CNDO/INDO) is:

$$E_N(A, B) = \frac{Z_A Z_B}{R_{AB}} \quad (150)$$

where  $Z_A$  and  $Z_B$  are the core charges (nuclear point charges minus inner shell electrons). Since, electron-electron and electron-core integrals do not immediately collapse to the form  $c/R_{AB}$  ( $R_{AB}$  is the distance of atom A and atom B,  $c$  is a constant) for distances beyond the van de Waals radii, there would be a net repulsion between two neutral atoms or molecules. To correct for this, core-core repulsion is approximated by:

$$E_N(A, B) = Z_A Z_B \gamma^{AB} \quad (151)$$

In addition to this term, account must be taken of the decreasing screening of the nucleus by the electrons as the interatomic distance becomes very small. At very small distances the core-core term should approach the classical form. To account for this, an additional term is added to the basic core-core repulsion integral in MINDO/3 to give:

$$E_N(A, B) = Z_A Z_B \left[ \gamma^{AB} + \left( \frac{1}{R_{AB}} - \gamma^{AB} \right) \exp(-\alpha_{AB} R_{AB}) \right] \quad (152)$$

Here  $\alpha_{AB}$  is a diatomic parameter. In the case of N-H and O-H interactions only, this expression is replaced by:

$$E_N(A, B) = Z_A Z_B \left[ \gamma^{AB} + \left( \frac{1}{R_{AB}} - \gamma^{AB} \right) \alpha_{AB} \exp(-R_{AB}) \right] \quad (153)$$

The MINDO/3 parameters appearing in the above equations are discussed later in the parameter section.

## Mixed Model

The basic idea of mixed model in MINDO/3 is the same as that used for CNDO and INDO and corrects  $\gamma_{AB}$ , which appears in the core Hamiltonian. Because the algorithm in calculating the Coulomb interaction in MINDO/3 is different from that used in CNDO and INDO, the procedure to correct  $\gamma_{AB}$  is also different from that in CNDO and INDO.

The core Hamiltonian,  $H_{\mu\mu}$  correction due to the interaction of charges in the quantum region and classical region is

$$\Delta H_{\mu\mu} = -\sum_B (Z_B - Q_B) \gamma_{AB}$$

and to the interaction energy of the nuclei in the quantum region and the charges in the classical region,  $E_N$ , is

$$\Delta E_N = \sum_{(A,B)} \left( (Z_A Z_B) \left[ \gamma_{AB} + \left( \frac{1}{R_{AB}} - \gamma_{AB} \right) \exp(-\alpha_{AB} R_{AB}) \right] - Z_A Q_B \gamma_{AB} \right)$$

The sum  $A$  is over all the atoms in the quantum region and  $B$  is over all the atoms in the classical region. The two-electron and two-center Coulomb integral,  $\gamma_{AB}$ , is computed in MINDO/3 by

$$\gamma_{AB} = \frac{1}{\left[ R_{AB}^2 + \frac{1}{4} \left( \frac{1}{A_A} + \frac{1}{A_B} \right)^2 \right]^{1/2}}$$

where  $R_{AB}$  is the distance between atom  $A$  and atom  $B$ .  $A_A$  and  $A_B$  are the averaged one-center, two-electron integral. Therefore, to run the mixed model at the MINDO/3 of approximation, HyperChem requires not only all the parameters for the atoms in the quantum mechanical region, but also the parameters for the pseudo-halogen atoms for the boundary atoms, if any, and some parameters for the atoms in the classical region. The parameters for the atoms in the classical region are  $A$ , the averaged one-center two-electron integral and  $\alpha_{AB}$ , the bond parameters.

## MNDO

MNDO is a Modified Neglect of Diatomic Overlap method based on the neglect of diatomic differential overlap (NDDO) approximation. The method was proposed and developed by M. J. S. Dewar and co-workers at the University of Texas at Austin [e.g. see M. J. S. Dewar and M. L. McKee, *J. Am. Chem. Soc.*, 99, 5231(1977); M. J. S. Dewar and W. Thiel, *J. Am. Chem. Soc.*, 99, 4899(1977); M. J. S. Dewar and H. S. Rzepa, *J. Am. Chem. Soc.*, 100, 58(1978); L. P. Davis, R. M. Guidry, J. R. Williams, M. J. S. Dewar, and H. S. Rzepa, *J. Comp. Chem.* 2, 433(1981)].

The elements of the MNDO Fock matrix based on the neglect of diatomic differential overlap approximation are described below. When  $\phi_\mu$  and  $\phi_\nu$  are on different centers the off-diagonal elements of the Fock matrix are:

$$F_{\mu\nu}^\alpha = H_{\mu\nu} - \sum_{\lambda}^A \sum_{\sigma}^B P_{\lambda\sigma}^\alpha (\mu\lambda|\nu\sigma) \quad (154)$$

( $\mu$  on A,  $\nu$  on B,  $A \neq B$ )

and when  $\phi_\mu$  and  $\phi_\nu$  are different atomic orbitals but on the same center then the off-diagonal elements of the Fock matrix are:

$$F_{\mu\nu}^\alpha = H_{\mu\nu} + P_{\mu\nu}^\alpha [3(\mu\nu|\mu\nu) - (\mu\mu|\nu\nu)] + \sum_B \sum_{\sigma}^B \sum_{\lambda}^B P_{\lambda\sigma}^{\alpha+\beta} (\mu\nu|\lambda\sigma) \quad (155)$$

( $\mu$  on A,  $\nu$  on A,  $\mu \neq \nu$ )

The diagonal elements of the Fock matrix are:

$$F_{\mu\mu}^\alpha = H_{\mu\mu} + \sum_{\nu}^A [P_{\nu\nu}^{\alpha+\beta} (\mu\mu|\nu\nu) - P_{\nu\nu}^\alpha (\mu\nu|\mu\nu)] + \sum_B \sum_{\lambda}^B \sum_{\sigma}^B P_{\lambda\sigma}^{\alpha+\beta} (\mu\mu|\lambda\sigma) \quad (156)$$

where  $\mathbf{P}^T \equiv \mathbf{P}^{\alpha+\beta} \equiv \mathbf{P}^\alpha + \mathbf{P}^\beta$ . By replacing the superscripts  $\alpha$  and  $\beta$  by  $\beta$  and  $\alpha$ , respectively, in the above three equations, you can easily get three similar equations for the Fock matrix elements for beta orbitals. Similar expressions to the above for Fock matrix elements of restricted Hartree-Fock (RHF) calculations can be generated by simply replacing  $\mathbf{P}^\alpha$  (or  $\mathbf{P}^\beta$ ) by  $-1/2 \mathbf{P}^T$  in the above equations.

The terms involved in the above equations are described below.

## Two-Center Two-Electron Integrals

The MNDO method has 22 unique two-center two-electron integrals for each pair of heavy (non-hydrogen) atoms in their local atomic frame. They are:

$$\begin{array}{lll}
 (ss|ss) & (ss|p_{\pi}p_{\pi}) & (ss|p_{\sigma}p_{\sigma}) \\
 (p_{\pi}p_{\pi}|ss) & (p_{\sigma}p_{\sigma}|ss) & (p_{\pi}p_{\pi}|p_{\pi}p_{\pi}) \\
 (p_{\pi}p_{\pi}|p'_{\pi}p'_{\pi}) & (p_{\pi}p_{\pi}|p_{\sigma}p_{\sigma}) & (p_{\sigma}p_{\sigma}|p_{\pi}p_{\pi}) \\
 (p_{\sigma}p_{\sigma}|p_{\sigma}p_{\sigma}) & (sp_{\sigma}|ss) & (sp_{\sigma}|p_{\pi}p_{\pi}) \\
 (sp_{\sigma}|p_{\sigma}p_{\sigma}) & (ss|sp_{\sigma}) & (p_{\pi}p_{\pi}|sp_{\sigma}) \\
 (p_{\sigma}p_{\sigma}|sp_{\sigma}) & (sp_{\pi}|sp_{\pi}) & (sp_{\sigma}|sp_{\sigma}) \\
 (sp_{\pi}|p_{\pi}p_{\sigma}) & (p_{\pi}p_{\sigma}|sp_{\pi}) & (p_{\pi}p_{\sigma}|p_{\pi}p_{\sigma}) \\
 (p_{\pi}p'_{\pi}|p_{\pi}p'_{\pi}) & & 
 \end{array}$$

With the exception of integral 22,  $(p_{\pi}p_{\pi}|p'_{\pi}p'_{\pi})$ , all the integrals can be computed *a priori* without loss of rotational invariance. That is, no integral depends on the value of another integral, except for this last one. It can, however, be shown that

$$(p_{\pi}p'_{\pi}|p_{\pi}p'_{\pi}) = \frac{1}{2}[(p_{\pi}p_{\pi}|p_{\pi}p_{\pi}) - (p_{\pi}p_{\pi}|p'_{\pi}p'_{\pi})] \quad (157)$$

The use of this formula for integral 22 gives rotational invariance.

The two-center two-electron repulsion integrals  $(\mu\nu|\lambda\sigma)$  represents the energy of interaction between the charge distributions  $\phi_{\mu}\phi_{\nu}$  at atom A and  $\phi_{\lambda}\phi_{\sigma}$  at atom B. Classically, they are equal to the sum over all interactions between the multipole moments  $M_{lm}$  of the two charge contributions, where the subscripts  $l$  and  $m$  specify the order and orientation of the multipole. MNDO uses the classical model in calculating these two-center two-electron interactions.

The electron density distributions are approximated by a series of point charges. There are four possible types of contributions, i.e.

- Monopole (1 charge): unit negative charge centered on the nucleus.
- Dipole (2 charges):  $+1/2$  charge located at position  $(x, y, z)$  and  $-1/2$  charge located at position  $(-x, -y, -z)$ .
- Linear quadrupole (3 charges):  $+1/2$  charge located at nucleus and  $-1/2$  charge located at positions  $(x, y, z)$  and  $(-x, -y, -z)$ .
- Square quadrupole (4 charges) — four charges of magnitude  $+1/4, -1/4, +1/4,$  and  $-1/4$  forming a square centered on the nucleus.

The following types of multipole distributions are used to represent the four types of atomic orbital products.

Atomic orbitals	Multipole distribution	Number of charges
$ ss\rangle$	Monopole	1
$ sp\rangle$	Dipole	2
$ pp\rangle$	Monopole plus linear quadrupole	4
$ pp'\rangle$	Square quadrupole	4

Each two-electron integral is the sum of all the terms arising from the charge distribution representative of the first pair of atomic orbitals interacting with the charge distribution representative of the second pair of atomic orbitals. Thus in the simplest case, the  $(ss|ss)$  interaction is represented by the repulsion of two monopoles, while a  $(p_{\pi} p_{\pi} | p'_{\pi} p'_{\pi})$ , a much more complicated interaction, is represented by 16 separate terms, arising from the four charges representing the monopole and linear quadrupole on one center interacting with the equivalent set on the second center.

Because the repulsion interaction energy of two point charges is inversely proportional to the distance separating the two charges, Dewar and co-workers, for example, represent the  $(ss|ss)$  two-center two-electron integral by:

$$(ss|ss) = \frac{1}{\left( (R + c_A + c_B)^2 + \frac{1}{4} \left( \frac{1}{A_A} + \frac{1}{A_B} \right)^2 \right)^{1/2}} \quad (158)$$

where  $A_A$  and  $A_B$  are parameters to be defined below and  $c_A$  and  $c_B$  are distances of the multipole charges from their respective nuclei A and B (a simple function of the atomic orbital type). In the case of an sp product, this is a vector of length  $D_1$  atomic units pointing along p axis,

$$D_1 = \frac{(2n+1)(4\zeta_s\zeta_p)^{n+1/2}}{3^{1/2}(\zeta_s + \zeta_p)^{2n+2}} \quad (159)$$

where  $\zeta_s$  and  $\zeta_p$  are the orbital exponents and  $n$  is the principal quantum number, equal to 2 for atoms in the first long row. The corresponding distances of the charges from the nucleus for the linear and square quadrupoles are  $2D_2$  and  $\sqrt{2}D_2$ , respectively, where

$$D_2 = \left(\frac{4n^2 + 6n + 2}{20}\right)^{1/2} \frac{1}{\zeta_p} \quad (160)$$

There are some boundary conditions which can be used to fix parameters  $A_A$  and  $A_B$ . For example, when the distance between nucleus A and nucleus B approaches zero, i.e.,  $R_{AB} = 0.0$ , the value of the two-electron two-center integral should approach that of the corresponding monocentric integral. The MNDO nomenclature for these monocentric integrals is,

$$\begin{aligned} G_{ss} &= (ss|ss) \\ G_{sp} &= (ss|pp) \\ H_{sp} &= (sp|sp) \\ G_{pp} &= (pp|pp) \\ G_{p_2} &= (pp|p'p') \\ H_{pp} &= (pp'|pp') \end{aligned} \quad (161)$$

Three cases can be identified:

- A monopole-monopole interaction, where the integral should converge to  $G_{ss}$
- A dipole-dipole interaction, where the integral should converge to  $H_{sp}$

- A quadrupole-quadrupole interaction, where the integral should converge to  $H_{pp} = 1/2 (G_{pp} - G_{p2})$

Using the above asymptotic forms of the two-center two-electron integrals, the parameters  $A_A$  and  $A_B$  can be derived. Certainly, parameter  $A_A$  is different for different orbitals even though they reside on the same atom. Dewar used AM to represent the parameter A obtained via  $G_{ss}$ , AD to represent the parameter A obtained via  $H_{sp}$ , and AQ to represent the parameter A obtained from  $H_{pp}$ .

Therefore, the two-center two-electron integral  $(ss | ss)$  can be written as:

$$(ss|ss) = \frac{1}{\left(R_{AB}^2 + \frac{1}{2}\left(\frac{1}{AM_A} + \frac{1}{AM_B}\right)^2\right)^{1/2}} \quad (162)$$

Equivalently,  $(ss | p_\pi p_\pi)$  can be represented as

$$\begin{aligned} (ss|p_\pi p_\pi) &= \frac{1}{\left(R_{AB}^2 + \frac{1}{2}\left(\frac{1}{AM_A} + \frac{1}{AM_B}\right)^2\right)^{1/2}} \\ &\quad + \frac{1/2}{\left(R_{AB}^2 + (2D_2^B)^2 + \frac{1}{2}\left(\frac{1}{AM_A} + \frac{1}{AQ_B}\right)^2\right)^{1/2}} \\ &\quad - \frac{1/4}{\left(R_{AB}^2 + \frac{1}{4}\left(\frac{1}{AM_A} + \frac{1}{AQ_B}\right)^2\right)^{1/2}} \end{aligned} \quad (163)$$

This last integral is the interaction of a monopole with the sum of a monopole and a linear quadrupole. This gives rise to a total of four terms. However, since the interaction of the monopole with each of two negative charges of the dipole are the same, only three terms remain. The remaining integrals are developed by similar manipulations.

### One-Center One-Electron Integral $H_{\mu\mu}$

In the MNDO method, the one-center one-electron integral,  $H_{\mu\mu}$ , is given by:



$$H_{\mu\mu} = U_{\mu\mu} - \sum_{B \neq A} V_{AB} \quad (164)$$

with  $V_{AB}$  approximated in a similar way to before:

$$V_{AB} = Z_B(\mu\mu|ss)$$

where  $(\mu\mu|ss)$  is the MNDO generalization of the CNDO/INDO and MINDO/3 two-electron integral  $\gamma^{AB}$ .

### Two-Center One-Electron Integral $H_{\mu\nu}$

In the MNDO method the resonance integral,  $H_{\mu\nu}$ , is proportional to the overlap integral,  $S_{\mu\nu}$ :

$$H_{\mu\nu} = S_{\mu\nu} \frac{(\beta_\mu + \beta_\nu)}{2} \quad (165)$$

where  $\beta_\mu$  and  $\beta_\nu$  are adjustable parameters characteristic of atomic orbitals  $\phi_\mu$  on atom A and  $\phi_\nu$  on atom B. For a given first-row or second-row element, there are at most two different  $\beta$  parameters, i.e.,  $\beta_s^A$  for the s-orbital and  $\beta_p^A$  for the p orbital of atom A. This

differs from MINDO/3 where  $N^2$  beta parameters are required compared to the  $O(N)$  beta parameters needed here.

### One-Center Two-Electron Integrals

The one-center two-electron integrals in the MNDO method are derived from experimental data on isolated atoms. Most were obtained from Oleari's work [L. Oleari, L. DiSipio, and G. DeMichells, *Mol. Phys.*, 10, 97(1977)], but a few were obtained by Dewar using fits to molecular properties.

For each atom there are a maximum of five one-center two-electron integrals, that is  $(ss|ss)$ ,  $(ss|pp)$ ,  $(sp|sp)$ ,  $(pp|pp)$ , and  $(pp|p'p')$ , where p and p' are two different p-type atomic orbitals. It has been shown that the extra one-center two-electron integral,  $(pp'|pp')$ , is related to two of other integrals by

$$(pp'|pp') = \frac{1}{2}[(pp|pp) - (pp|p'p')] \quad (166)$$

If the five independent one-center two-electron integrals are expressed by symbols such as  $G_{ss}$ ,  $G_{sp}$ , defined above, then the Fock matrix element contributions from the one-center two-electron integrals are:

$$F_{ss}^{\alpha} \leftarrow P_{ss}^{\beta} G_{ss} + (P_{px}^{\alpha+\beta} + P_{py}^{\alpha+\beta} + P_{pz}^{\alpha+\beta}) G_{sp} - (P_{px}^{\alpha} + P_{py}^{\alpha} + P_{pz}^{\alpha}) H_{sp} \quad (167)$$

$$F_{sp}^{\alpha} \leftarrow P_{sp}^{\alpha+\beta} H_{sp} - P_{sp}^{\alpha} (H_{sp} + G_{sp}) \quad (168)$$

$$F_{pp}^{\alpha} \leftarrow P_{ss}^{\alpha+\beta} G_{sp} - P_{ss}^{\alpha} H_{sp} + P_{pp}^{\beta} G_{pp} + (P_{p'p'}^{\alpha+\beta} + P_{p''p''}^{\alpha+\beta}) G_{p2} - \frac{1}{2} (P_{p'p'}^{\alpha} + P_{p''p''}^{\alpha}) (G_{pp} - G_{p2}) \quad (169)$$

$$F_{pp'}^{\alpha} \leftarrow P_{pp'}^{\alpha+\beta} (G_{pp} - G_{p2}) - \frac{1}{2} P_{pp'}^{\alpha} (G_{pp} + G_{p2}) \quad (170)$$

where  $\mathbf{P}^T \equiv \mathbf{P}^{\alpha+\beta} \equiv \mathbf{P}^{\alpha} + \mathbf{P}^{\beta}$ . By replacing the superscripts  $\alpha$  and  $\beta$  by  $\beta$  and  $\alpha$ , respectively, in the above three equations, you can easily get three similar equations for the Fock matrix elements for beta orbitals. Similar expressions to the above for Fock matrix elements of restricted Hartree-Fock (RHF) calculations can be generated by simply replacing  $\mathbf{P}^{\alpha}$  (or  $\mathbf{P}^{\beta}$ ) by  $1/2 \mathbf{P}^T$  in the above equations.

## Core-Core Repulsion Integrals

In practice the MNDO approximation to the screening effect is similar to that of MINDO/3 but has a different functional form:

$$E_N(A, B) = Z_A Z_B (s_A s_A | s_B s_B) \times [1 + \exp(-\alpha_A R_{AB}) + \exp(-\alpha_B R_{AB})] \quad (171)$$

The core-core repulsion integrals are different for O-H and N-H interactions. They are expressed as:

$$E_N(A, H) = Z_A Z_H (s_A s_A | s_H s_H) \times \left( 1 + \frac{\exp(-\alpha_A R_{AH})}{R_{AH}} + \exp(-\alpha_H R_{AH}) \right) \quad (172)$$

where  $a$  is an adjustable atomic parameter. The numerical values of the  $a$  parameter are the same for each element.

### Mixed Model (MNDO, AM1, and PM3)

The mixed models used in MNDO, AM1, and PM3, are identical, because all of these three methods are derived based on NDDO. The core Hamiltonian correction due to the interaction of the charges between the quantum mechanical region and the classical region is

$$\Delta H_{\mu\nu} = -\sum_B (Z_B - Q_B) \gamma_{\mu\nu}^{AB}$$

for both  $\mu$  and  $\nu$  are on atom  $A$ , and  $\gamma_{\mu\nu}^{AB}$  is

$$\gamma_{AB} = (\mu_A \nu_A | s_B s_B)$$

and calculated by the MNDO method. And the interaction energy between the nuclei in the quantum mechanical region and the charges (including the nuclear charges and the electronic charges) is

$$\Delta E_N = \sum_{(A,B)} \{ Z_A Z_B (s_A s_A | s_B s_B) \left[ 1 + \exp(-\alpha_A R_{AB}) + \frac{\exp(-\alpha_A R_{AB})}{R_{AB}} \right] - Z_A Q_B (s_A s_A | s_B s_B) \}$$

with

$$(s_A s_A | s_B s_B) = \frac{1}{\left[ R_{AB}^2 + \frac{1}{2} \left( \frac{1}{AM_A} + \frac{1}{AM_B} \right)^2 \right]^{1/2}}$$

where  $AM$  are the monopole-monopole interaction parameters.

### MNDO/d

This method is distinguishable from MNDO principally by its use of  $d$  orbitals. It is described in the following references: W. Theil and A. Voityuk, *Theo. Chim. Acta* 81, pp391-404 (1992); W. Thiel and A. Voityuk, *Int.J. Quant. Chem.* 44, pp807-829 (1992);

## AM1 and PM3

AM1 (Austin Model 1) is a modified MNDO method proposed and developed by M. J. S. Dewar and co-workers at the University of Texas at Austin [see, e.g., M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, and J. J. P. Stewart, *J. Am. Chem. Soc.*, 107, 3902(1985); M. J. S. Dewar and K. M. Dieter, *J. Am. Chem. Soc.*, 108, 8075(1986); J. J. P. Stewart, *J. Comp. Aided Mol. Design*, 4, 1(1990)].

PM3, developed by James J.P. Stewart, is a reparameterization of AM1, which is based on the neglect of diatomic differential overlap (NDDO) approximation. NDDO retains all one-center differential overlap terms when Coulomb and exchange integrals are computed. PM3 differs from AM1 only in the values of the parameters. The parameters for PM3 were derived by comparing a much larger number and wider variety of experimental versus computed molecular properties. Typically, nonbonded interactions are less repulsive in PM3 than in AM1. PM3 is primarily used for organic molecules, but is also parameterized for many main group elements.

While MINDO/3 has proved very effective in studies of a wide variety of hydrocarbons [M. J. S. Dewar and H. S. Rzepa, *J. Am. Chem. Soc.*, 99, 7432(1977)], problems arise in the case of molecules containing heteroatoms because of the neglect of one-center differential overlap in the INDO approximation on which MINDO/3 based. The problems are avoided in MNDO but at the expense of other weaknesses [M. J. S. Dewar and W. Thiel, *J. Am. Chem. Soc.*, 99, 4899(1977); 99, 4907(1977)]. In particular MNDO fails to reproduce hydrogen bonds correctly, gives energies that are too positive for crowded molecules (e.g. neopentane) and too negative for ones containing four-membered rings, and gives activation energies that tend to be too large. Dewar and co-workers found that the reason is because MNDO overestimates the repulsions between atoms at their Van der Waals distance. Hence they modified the core-core repulsion for N-H and O-H interactions to:

$$E_N(A, H) = Z_A Z_H (s_A s_A | s_H s_H) \\ \times \left( 1 + \frac{\exp(-\alpha_A R_{AH})}{R_{AH}} + \exp(-\alpha_H R_{AH}) \right) + \left( \frac{Z_A Z_H}{R_{AH}} \right) \\ \times \left( \sum_k a_{kA} \exp[-b_{kA} (R_{AH} - c_{kA})^2] + \sum_k a_{kH} \exp[-b_{kH} (R_{AH} - c_{kH})^2] \right) \quad (173)$$

and for all other interactions to:

$$E_N(A, B) = Z_A Z_B (s_A s_A | s_B s_B) \quad (174)$$

$$[1 + \exp(-\alpha_A R_{AB}) + \exp(-\alpha_B R_{AB})] + \left( \frac{Z_A Z_B}{R_{AB}} \right) \left( \sum_k a_{kA} \exp[-b_{kA} (R_{AB} - c_{kA})^2] + \sum_k a_{kB} \exp[-b_{kB} (R_{AB} - c_{kB})^2] \right)$$

Comparing the core-core repulsion of the above two equations with those in the MNDO method, it can be seen that the only difference is in the last term. The extra terms in the AM1 core-core repulsion define spherical Gaussian functions — the a, b, and c are adjustable parameters. AM1 has between two and four Gaussian functions per atom.

These are the only differences between the MNDO and AM1 functional form. Dewar's group regenerated AM1 parameters for the elements H, B, C, N, O, F, Al, Si, P, S, Cl, Zn, Ge, Br, and Sn and found that the main gains in AM1 over MNDO were the ability to reproduce hydrogen bonds and the promise of better activation energies for reactions. AM1 does not significantly change the computation time compared with MNDO.

## ZINDO/1

ZINDO/1 is based on a modified version of the intermediate neglect of differential overlap (INDO), which was developed by Michael Zerner of the Quantum Theory Project at the University of Florida. Zerner's original INDO/1 used the Slater orbital exponents with a distance dependence for the first row transition metals only. (See *Theoret. Chim. Acta (Berl.)* **53**, 21-54 (1979).) However, in HyperChem constant orbital exponents are used for all the available elements, as recommended by Anderson, Edwards, and Zerner, *Inorg. Chem.* **25**, 2728-2732, 1986.

As with the other semi-empirical methods, HyperChem's implementation of ZINDO/1 is restricted to spin multiplicities up to a quartet state. ZINDO/1 lets you calculate the energy states in molecules containing transition metals.

HyperChem's implementation of ZINDO/1 has been tested using parameters suggested by references to work done by Zerner on first row transition metals.

The basic equations of ZINDO/1 are the same as those in INDO, except for  $U_{\mu\nu}$ . Instead of using the electronegativity in INDO, ZINDO/1 uses the ionization potential for computing  $U_{\mu\nu}$ .

## Overlap Weighting Factors

The developers of ZINDO found that the parameters required to reproduce orbital energy orderings and UV spectra are different from those required to reproduce accurate structures by geometry optimization. They introduced a new pair of parameters, called the *overlap weighting factors*, to account for this. These parameters are provided in HyperChem in the Semi-empirical Options dialog box. Their effect is to modify the *resonance integrals* for the off-diagonal elements of the Fock matrix.

In semi-empirical formulations, resonance integrals account for sigma and pi bonding patterns in molecules. These one-electron two-center integrals represent the kinetic energy and electron-nuclear attraction energy associated with a charge distribution that lies between two atoms (overlap distribution). The actual magnitude of the resonance integral is proportional to the overlap integral  $S_{\mu\nu}$ . There are two general types of atomic orbital overlap: sigma-sigma and pi-pi. Sigma-pi overlap is generally absent due to orthogonality conditions. Adjusting the weights provides a mechanism to adjust the relative contributions of sigma versus pi bonding.

For geometry optimizations and comparison of total energies (which should be carried out with ZINDO/1, not ZINDO/S), both overlap weighting factors (Sigma-Sigma and Pi-Pi) should be set to 1 in the Semi-empirical Options dialog box.

If you use ZINDO/S for spectroscopic investigations and orbital energies, the default values are 1.267 for Sigma-Sigma (Ridley, J.E. and Zerner, M.C. *Theor. Chim. Acta* 42:223, 1976) and 0.585 for Pi-Pi. However, several different values for the Pi-Pi overlap weighting factor have been used in the literature (Bacon, A.D. and Zerner, M.C. *Theor. Chim. Acta* 53:21, 1979). For UV spectra and orbital eigenvalues of organic molecules, 0.585 is commonly used (Del Bene, J. and Jaffe, H.H. *J. Chem. Phys.* 48:1807, 1968; *ibid.*, 4050). For triplets, 0.64 and 0.68 were recommended. In recent studies,

0.585 was used with good success for transition-metal systems, see, for example, Stavrev, K.K., Zerner, M.C., Meyer, T.J., *J. Am. Chem. Soc.* 117:8684, 1995 (on Ru-complexes); Stavrev, K.K., and Zerner, M.C., *J. Chem. Phys.* 102:34, 1995 (Mn-centers); *Chem. Phys. Lett.* 233:179, 1995 (Cu-porphyrins); *ibid.* 263:667, 1996 (VC14); *Theor. Chim. Acta* 96(3):141 (Mo-nitrogenase); Cory, M.G., Stavrev, K.K. and Zerner, M.C., *Int. J. Quant. Chem.* 63:781, 1997 (Fe-proteins).

If you want to change the number that appears in the Semi-Empirical Options dialog box you could include a script command that sets the value explicitly in your chem.scr initialization script.

## Mixed Model

The mixed model used in ZINDO/1 is identical to that used in CNDO and INDO if there is no d-orbital involved in the quantum mechanical calculation. Once the system involves transition metals, the mixed model is slightly more complex. HyperChem treats the charges in the classical region as the following way:

$$\Delta H_{\mu\mu} = \sum_B (Z_B - Q_B) \gamma_{\mu\mu ss}^{AB}$$

corrects the core Hamiltonian element  $H_{\mu\mu}$ , and

$$\Delta E_N = \sum_{(A,B)} \left( \frac{Z_A Z_B}{R_{AB}} - Z_A Q_B \gamma_{ss ss}^{AB} \right)$$

corrects the interaction energy between the quantum mechanical nuclei and classical charges. Because the Slater exponent for *s* orbital may be different from the Slater exponent for *d* orbital,

$\gamma_{ss ss}^{AB}$  may not be equal to  $\gamma_{dd ss}^{AB}$ . To keep the rotational invariance,

four *s* type Slater orbitals are used in calculating  $\gamma_{dd ss}^{AB}$  with *d* orbital

Slater exponent for the *s* type Slater orbitals on atom *A* and *s* orbital Slater exponent for the *s* type Slater orbitals on atom *B*.

## ZINDO/S

ZINDO/S is an modified INDO method parameterized to reproduce UV visible spectroscopic transitions when used with the CI singles methods. It was developed in the research group of Michael Zerner of the Quantum Theory Project at the University of Florida.

As with the other semi-empirical methods, HyperChem's implementation of ZINDO/S is restricted to spin multiplicities of up to a quartet state. Higher spin systems may not be done using HyperChem.

The algorithms in ZINDO/S are almost the same as those in ZINDO/1, except of the one-center two-electron integral,  $F^0$ . ZINDO/S uses empirical value of  $F^0$  instead of using *ab initio* value in terms of the Slater orbitals.

## Mixed Model

ZINDO/S is different from ZINDO/1 because they use different algorithms in computing the Coulomb integrals. Hence the two equations used in the mixed model in ZINDO/1 are also employed in ZINDO/S. But the Coulomb integrals are computed by the following way:

$$\gamma_{ss\ ss}^{AB} = \frac{f_r}{\frac{2f_r}{\gamma_{ss}^A + \gamma_{ss}^B} + R_{AB}}$$

and

$$\gamma_{dd\ ss}^{AB} = \frac{f_r}{\frac{2f_r}{\gamma_{ss}^A + \gamma_{ss}^B} + R_{AB}}$$

where  $f_r$  is a constant and equal to 1.2 and  $\gamma_{ss}^A$  and  $\gamma_{dd}^A$  are two-electron one-center Coulomb integrals and they are semi-empirical parameters required by the ZINDO/S model.

## TNDO

The TNDO Hamiltonian is identical to the CNDO or INDO Hamiltonian. TNDO/1 is CNDO and TNDO/2 is INDO. The difference is that TNDO has different parameters from CNDO/INDO and parameters that depend upon the atom type of an atom rather than the atomic number of an atom. Thus, there are separate parameters for  $sp$ ,  $sp^2$ , and  $sp^3$  carbon atoms, not just single parameters for carbon. The method can be very accurate with the right parameters.



In general, however, TNDO is a very parameter dependent method with even different parameters sets for different computed molecular properties. For example, a parameter set might be designed specifically for the electronic spectra of dye molecules. Another parameter set might be designed for geometries of hydrocarbons and so on. Using this method effectively may require you to develop your own parameter sets.

# Chapter 13

## Density Functional Theory

In addition to the *ab initio* and semi-empirical quantum mechanical calculations described in the last chapter, HyperChem, beginning with Release 7, can perform quantum mechanical calculations according to *Density Functional Theory (DFT)*. These are a relatively new type of calculation that bears a very close resemblance to *ab initio* calculations; they are even referred to as *ab initio* in some contexts. We will, however, make a clear distinction between so-called *ab initio* and DFT calculations.

### Wavefunctions versus Electron Density

The traditional *ab initio* techniques attempt to solve the Schrödinger equation (using various approximations) for a wavefunction  $\Psi_{\text{elec}}$  describing a molecule or molecular system described by a Hamiltonian  $\mathbf{H}$ .

$$\mathbf{H}_{\text{elec}} \Psi_{\text{elec}} = E_{\text{elec}} \Psi_{\text{elec}} \quad (1)$$

Having obtained a wavefunction, which is a function of the  $3N$  coordinates of all  $N$  electrons, one can compute the electron density by integrating the square of the wavefunction over all particles but one,

$$\rho(\vec{r}_1) = \int \Psi^2(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) d\vec{r}_2 \dots d\vec{r}_N \quad (2)$$

That is, the electron density is much simpler than the wavefunction since it is a function of only 3 variables rather than  $3N$  variables. The electron density averages out details of the full  $N$ -body problem.

If one could compute the energy and properties of a molecule by computing only the electron density rather than the full wave-

function, one might expect a simplification of quantum chemistry. This is the hope of Density Functional Theory.

## The Hohenberg-Kohn Theorem

The formal rationale for Density Functional Theory begins with a remarkable theorem due to Hohenberg and Kohn. This theorem states that the energy  $E$  of a molecular system is a universal functional of the electron density,

$$E = E[\rho] \tag{3}$$

and that the exact energy is returned only when  $\rho$  is exact; approximate  $\rho$ 's lead to higher energies. By the term functional, we mean that  $E$  depends not on simple variables but on a complicated function, the density in three dimensional space. That is,  $E$  depends on the density only, but with an unknown functional dependence. By the term universal dependence, we mean that this dependence of the energy on the density is the same in methane as it is in fluorobenzene and if we only knew this dependence it would help provide solutions for all molecular systems.

In the above form, the Hohenberg-Kohn theorem is not of immediate practical help in solving for the energy and electron density of molecules. The theorem, however, allows a new formulation for a self-consistent-field (SCF) procedure that can be used to compute molecular charge densities. This new procedure is embodied in the Kohn-Sham equations.

## The Kohn-Sham Equations

### Overview

The Kohn-Sham equations can, in principal, lead to an exact solution for a molecular system. The equations resemble very closely the equations of Hartree-Fock theory of the last chapter. However, the interpretation of the solutions to the Kohn-Sham equations is quite different from the interpretation of the solutions to the Hartree-Fock equations. For example, the Kohn-Sham energy can be exact while the Hartree-Fock energy is well known to be missing what is referred to as the *correlation energy*. It is enormously useful

that the two sets of equations have so much in common as this makes it possible to use technology developed over many years for Hartree-Fock programs to be immediately useful in developing software to solve the Kohn-Sham equations.

The logic used to turn the Hohenberg-Kohn theorem into the Kohn-Sham equations is subtle and involves the use of an imaginary system of non-interacting electrons with an electron density identical to the true system. We will not offer a formal derivation of the Kohn-Sham equations here but simply state the results.

As in Hartree-Fock theory, there are two versions of the Kohn-Sham equations, one for closed-shell systems, analogous to the “restricted” Roothaan equations, and one for open-shell systems, analogous to the “unrestricted” Pople-Nesbet equations. We will state the equations in the generalized “unrestricted” form that is very close to that of UHF equations of the previous chapter.

## The Electron Density and Kohn-Sham Orbitals

The total electron density can be subdivided into a density of electrons with spin up (alpha) and a density of electrons with spin down (beta).

$$\rho^T = \rho^\alpha + \rho^\beta \quad (4)$$

As with the Pople-Nesbet equations, a basis set,  $\{\phi_\mu\}$ , is introduced and the electron densities can be represented by density matrices,

$$\mathbf{P}^T = \mathbf{P}^\alpha + \mathbf{P}^\beta \quad (5)$$

where,

$$P_{\mu\nu}^\alpha = \sum_i^{\text{occ}} C_{\mu i}^\alpha C_{\nu i}^\alpha \quad (6)$$

and

$$P_{\mu\nu}^{\beta} = \sum_i^{\text{occ}} C_{\mu i}^{\beta} C_{\nu i}^{\beta} \quad (7)$$

The C's are coefficients of basis functions in the expansion of the Kohn-Sham orbitals,

$$\Psi_i^{\alpha} = \sum_v C_{v i}^{\alpha} \phi_v \quad (8)$$

and

$$\Psi_i^{\beta} = \sum_v C_{v i}^{\beta} \phi_v \quad (9)$$

The Kohn-Sham orbitals are not unlike the molecular orbitals of Hartree-Fock theory but have energies that must be interpreted differently. As in Hartree-Fock theory, however, the total electron density is just the sum of the squares of the N occupied orbitals. The difference here is that this electron density can, in principal, be exact if the orbitals are solutions to the Kohn-Sham equations rather than the Hartree-Fock equations.

## The Kohn-Sham Equations

The Pople-Nesbet equations of Hartree-Fock Theory involve Fock matrices,  $F^{\alpha}$  and  $F^{\beta}$  that have the following form,

$$F_{\mu\nu}^{\alpha} = H_{\mu\nu}^{\alpha} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T (\mu\nu|\lambda\sigma) - P_{\lambda\sigma}^{\alpha} (\mu\lambda|\nu\sigma) \quad (10)$$

$$F_{\mu\nu}^{\beta} = H_{\mu\nu}^{\beta} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T(\mu\nu|\lambda\sigma) - P_{\lambda\sigma}^{\beta}(\mu\lambda|\nu\sigma) \quad (11)$$

The first term in the summations above is the *coulomb energy* and the second term is the *exchange energy*. Coulomb interactions occur between all the electrons, independent of spin, whereas exchange interactions only occur between electrons of parallel spin. In Hartree Fock theory, *the correlation energy*, the difference between the energy of the Hartree-Fock single determinant wavefunction approximation and the exact energy is, by definition, zero.

The Kohn-Sham equations or the fundamental equations of pragmatic Density Functional Theory involve replacing the exchange term above with a new exchange-correlation potential and equivalent matrix. That is,

$$F_{\mu\nu}^{\alpha} = H_{\mu\nu}^{\alpha} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T(\mu\nu|\lambda\sigma) + F_{\mu\nu}^{\text{XC}\alpha} \quad (12)$$

$$F_{\mu\nu}^{\beta} = H_{\mu\nu}^{\beta} + \sum_{\lambda\sigma} P_{\lambda\sigma}^T(\mu\nu|\lambda\sigma) + F_{\mu\nu}^{\text{XC}\beta} \quad (13)$$

where  $\mathbf{F}^{\text{XC}}$  is the exchange-correlation matrix (for either alpha or beta electrons) and these two equations constitute the general (unrestricted) matrix form of the Kohn-Sham equations.

These equations can be solved in a self-consistent fashion identical to the Pople-Nesbet equations of Hartree-Fock theory, the only difference is the replacement of the Hartree-Fock exchange term with an unknown, effective one-electron, exchange-correlation potential represented by the matrix  $\mathbf{F}^{\text{XC}}$ . The formulation can potentially give exact results. The problem, of course, is to know the exchange-correlation potential and be able to compute the matrix  $\mathbf{F}^{\text{XC}}$ .

For closed shells, where  $\rho^{\alpha} = \rho^{\beta}$ , the Kohn-Sham equations reduce to a set of equations analogous to the Roothaan equations.

## The Exchange-Correlation Functional

The complications of normal *ab initio* theory are in the complexity of the procedures that go beyond the Hartree-Fock approximation and attempt to calculate the correlation energy. Formulations such as full Configuration Interaction are easy to describe but almost impossible to implement for reasonable sized molecules. Density Functional Theory, as described by the Kohn-Sham equations, has transferred this complexity into an (unknown) effective one-electron exchange-correlation potential. The equations have the simplicity of the Hartree-Fock equations but with an unknown potential.

The energy that is a solution to the Kohn-Sham equations is given by

$$E = \sum_{\mu\nu} P_{\mu\nu} H_{\mu\nu} + \frac{1}{2} \sum_{\mu\nu\lambda\sigma} P_{\mu\nu} P_{\lambda\sigma} (\mu\nu|\lambda\sigma) + E_{XC}$$

This expression is identical to the Hartree-Fock energy except that the normal exchange term is missing and has been replaced by an effective *exchange-correlation energy*,  $E_{XC}$ , that hides all the complexity. In line with the Hohenberg-Kohn theorem, this energy is a functional of the Kohn-Sham density,

$$E_{XC} = E_{XC}[\rho^\alpha, \rho^\beta]$$

It is normal to describe an exchange-correlation energy density (per unit volume)  $f$ , such that

$$E_{XC} = \int f(\rho^\alpha, \rho^\beta) d\mathbf{r}$$

The relationship between the exchange-correlation energy,  $E_{XC}$ , and the exchange-correlation matrix of the Kohn-Sham equations has been formulated in a convenient form by Johnson, Gill, and Pople [J.Chem. Phys. **98**, 5612(1993)]. In its simplest form, the relationship is,

$$F_{\mu\nu}^{XC\alpha} = \int \frac{\partial f}{\partial \rho^\alpha} \phi_\mu \phi_\nu d\mathbf{r}$$

## A Simple Example: The Slater Functional

To illustrate the above discussion, we introduce a simple functional for the exchange energy. This functional, introduced by Slater in the context of so-called  $X\alpha$  theory, does not describe any correlation energy but is a functional that is a first approximation to the exchange energy. The functional is

$$f = -\frac{9}{4}\alpha\left(\frac{3}{4\pi}\right)^{1/3}((\rho^\alpha)^{4/3} + (\rho^\beta)^{4/3})$$

where  $\alpha$  is a parameter that Slater set to 1 but which has the value of  $2/3$  when this functional is derived as the exact exchange energy of a uniform electron gas (UEG). The corresponding elements of the exchange-correlation matrix of the Kohn-Sham equations are:

$$F_{\mu\nu}^{XC\alpha} = F_{\mu\nu}^{X\alpha} = \int -3\alpha\left(\frac{3}{4\pi}\right)(\rho^\alpha)^{1/3}\phi_\mu\phi_\nu d\mathbf{r}$$

The integrand in this case involves the electron density to the third power. In general integrals involving an exchange-correlation functional are too complicated to be performed analytically and such matrix elements must be obtained by numerical integration. Other exchange-correlation functionals are much more complicated than the simple Slater exchange functional and an important step in practical DFT calculations such as those performed in HyperChem is the requisite numerical integrations required. Practical schemes for these integrations will be discussed below.

## The Uniform Electron Gas and Local Density Approximations.

### Exchange versus Correlation Functional

Perhaps the principal issue in Density Functional Theory is developing accurate functions that describe either the exchange or the correlation energy. Some functionals describe only the exchange energy and some describe only the correlation energy. To describe the total exchange-correlation energy one can just add an exchange functional to a correlation functional. Other functionals



describe the total exchange-correlation energy without a recommended decomposition.

## Uniform Electron Gas (UEG)

A simple model system to try and understand the correct form that a functional could or should take is the uniform electron gas of constant density  $\rho$ . This system is simple enough that one can exactly calculate the exchange energy. The result in simple analytical form is just the Slater result discussed above. This result is also called the **Local Density Approximation (LDA)** or **Local Spin-Density Approximation (LSDA)** for exchange. It is called *local* because one takes the result for the uniform electron gas as describing the exchange energy at a local point  $\mathbf{r}$  but then integrates this result over all space for the non-uniform density of a molecular system. Rather than “local” it might just as well be called the UEG functional. An LSDA result specifically uses  $\rho^\alpha$  and  $\rho^\beta$  rather than just  $\rho$  but depending on the situation it may be identical to the LDA result. From now on we will simply refer to results for the uniform electron gas as LDA results (and functionals).

## LDA Correlation Functionals

While the Slater result describes the LDA result for the exchange energy in a simple analytic form, no such simple result is available for the correlation energy of a uniform electron gas. The LDA results for a correlation functional are more complicated (and slightly more variable in practice) than for the exchange functional. Most of the LDA functionals for correlation depend upon numerical results obtained by Monte Carlo simulation of the UEG. These “exact” results are then fit to an analytical functional form to obtain an LDA correlation functional. Examples are the VWN functional and the Perdew-Zunger 81 functional.

## Beyond the LDA - Gradient Corrected Functionals

The LDA results described above, either for exchange or correlation, or in combination constitute the simplest approach to Density Functional Theory. Many DFT calculations have been performed using the LDA and they provide a sort of benchmark for developing better functionals. These functionals directly use

results from the constant electron density situation (UEG). A next approximation would be to recognize that electron densities in molecules are not constant but are larger near a nucleus, go to zero at large distances, etc. That is, electron densities have a gradient. Equivalently, the next term in a Taylor expansion of a functional might involve a term  $d\rho/dr$ , the gradient of the electron density in three-dimensional space. Functionals that depend upon the gradient of the density as well as the density are called gradient-corrected functionals. Equivalently, one refers to the Generalized Gradient Approximation (GGA). The gradient of the electron density is usually described via three invariants,

$$\begin{aligned}\gamma_{\alpha\alpha} &= \nabla\rho^\alpha \cdot \nabla\rho^\alpha \\ \gamma_{\beta\beta} &= \nabla\rho^\beta \cdot \nabla\rho^\beta \\ \gamma_{\alpha\beta} &= \nabla\rho^\alpha \cdot \nabla\rho^\beta\end{aligned}$$

and the generalized gradient approximation has a function that is a generalization of the LDA functional.

$$f = f(\rho^\alpha, \rho^\beta, \gamma_{\alpha\alpha}, \gamma_{\beta\beta}, \gamma_{\alpha\beta})$$

The Kohn-Sham matrices for GGA functionals have a more complicated form than for LDA functionals. As before, a convenient expression is given by Pople et. al. as,

$$F_{\mu\nu}^{\text{XC}\alpha} = \int \left\{ \frac{\partial f}{\partial \rho^\alpha} \phi_\mu \phi_\nu + \left[ 2 \frac{\partial f}{\partial \gamma_{\alpha\alpha}} \nabla \rho^\alpha + \frac{\partial f}{\partial \gamma_{\alpha\beta}} \nabla \rho^\beta \right] \cdot \nabla (\phi_\mu \phi_\nu) \right\} d\mathbf{r}$$

## Energy Gradients

HyperChem uses gradients for many things - finding optimized structures, finding transition states, performing molecular dynamics, etc. The gradient of the energy in DFT calculations has been somewhat elusive and different formulations have different approximations to these gradients. HyperChem uses a rigorous approach outlined by Johnson, Gill and Pople that does not require second derivatives of the exchange-correlation energy as in

certain other formulations. The gradient of the exchange-correlation energy with respect to the coordinates of nucleus A are given by:

$$\begin{aligned} \nabla_A E_{XC} = & -2 \sum_{\mu} \sum_{\nu} P_{\mu\nu}^{\alpha} \int \left\{ \frac{\partial f}{\partial \rho^{\alpha}} \phi_{\nu} \nabla \phi_{\mu} + X_{\mu\nu} \left[ 2 \frac{\partial f}{\partial \gamma_{\alpha\alpha}} \nabla \rho^{\alpha} + \frac{\partial f}{\partial \gamma_{\alpha\beta}} \nabla \rho^{\beta} \right] \right\} d\mathbf{r} \\ & -2 \sum_{\mu} \sum_{\nu} P_{\mu\nu}^{\beta} \int \left\{ \frac{\partial f}{\partial \rho^{\beta}} \phi_{\nu} \nabla \phi_{\mu} + X_{\mu\nu} \left[ 2 \frac{\partial f}{\partial \gamma_{\alpha\beta}} \nabla \rho^{\alpha} + \frac{\partial f}{\partial \gamma_{\beta\beta}} \nabla \rho^{\beta} \right] \right\} d\mathbf{r} \quad (14) \end{aligned}$$

## Hybrid Functionals

On an empirical basis, it has been determined that rather than completely replacing the Hartree-Fock exchange term by an exchange functional of the kind being described by DFT, better results can often be obtained by retaining a percentage of so-called “exact” exchange, i.e. the Hartree-Fock exchange term. These have been referred to as *hybrid* functionals as opposed to *pure* functionals. Thus, for example, the B3-LYP functional discussed below contains a mixture of 20% exact exchange with 80% of a pure exchange functional.

## A Set of Functionals

HyperChem contains a number of functionals and here we list each with the appropriate literature reference.

### Exchange Functionals

#### Hartree-Fock

This is the normal exchange term of the Hartree-Fock equations. Using this functional only should give the same result as a normal ab initio SCF calculation.

## Slater

This is the LDA exchange result described above. Slater derived it in a different context. Its original derivation is probably due to Dirac in 1930. A good reference is one of Slater's books.

[J.C. Slater, *Quantum Theory of Molecules and Solids*, Vol. 4: *The Self-Consistent Field for Molecules and Solids* (McGraw-Hill, New York, 1974).]

## Becke-88

This functional constituted an important step in the development of density functional theory for molecules. Becke included a gradient correction to Slater's result that greatly improved the accuracy. Like many functionals it was of the general form,

$$E_X = \sum_{\sigma} \int e_{X\sigma}^{\text{LSDA}} g_{X\sigma} d\tau$$

where  $g=1$  is equivalent to the Slater LSDA result but, in general,  $g$  is some function of the electron density gradient

$$g_{X\sigma}(x) = g_{X\sigma}(|\nabla\rho^{\sigma}|/(\rho^{\sigma})^{4/3})$$

The Becke-88 functional is

$$g(x) = 1 + (bx^2)/a(1 + 6bx \operatorname{asinh} x)$$

where

$$a = \frac{3}{2} \left( \frac{3}{4\pi} \right)^{1/3}$$

and  $b$  is a parameter (.0042) used to fit experiment.

The Becke-88 functional is described in the reference:

[A.D. Becke, *Density-functional exchange energy approximation with correct asymptotic behaviour*, Phys. Rev. A, **38**, 3098(1988).]

## Perdew-Wang-91

A large number of the available functionals both for exchange and correlation have been developed by J.P. Perdew and his collaborators. The Perdew-Wang-91 exchange functional is similar to the Becke-88 functional but with a somewhat different  $g(x)$ . This functional is referred to as GGA-II by Perdew and collaborators. The functional is originally due to Perdew and Wang but the more transparent paper upon which the HyperChem implementation is based is the following:

[J.P. Perdew, J.A. Chevary, S.H. Vosko, K.A. Jackson, M.R. Pederson, D.J. Sing, and C. Fiolhais, *Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation*, Phys. Rev. B **46**, 6671 (1992).]

## Gill-96

The Gill exchange functional is a gradient-corrected functional but has the very simple form,

$$g(x) = 1 + bx^{3/2}$$

with  $b = 1/137$ . It is simple enough that there might eventually be a way to avoid the numerical integrations that plague density functional theory.

## PBE-96

More recently Perdew, Burke, and Ernzerhof have developed a GGA functional that appears to be as accurate as their GGA-II (PW91) functional but avoids the use of purely empirical parameters. Like some other functionals, both an exchange and a correlation functional have been defined. They are commonly recommended to be used in conjunction with each other but HyperChem allows each to be considered independently. The appropriate reference for the HyperChem implementation is:

[J.P. Perdew, K. Burke, M. Ernzerhof, *Generalized Gradient Approximations Made Simple*, Phys. Rev. Lett. **77**, 3865 (1996).]

## HCTH-98

Becke has defined a new hybrid exchange-correlation potential that we refer to in HyperChem as Becke-97. The pure form of the the corresponding exchange potential (as well as the correlation potential) has been adopted by the group of Handy et al. as convenient for defining new pure exchange and correlation potentials that they refer to as HCTH-98 potentials.

The Becke suggestion was to express  $g(x)$  as a sum of terms in a converging power series where the first term was essentially identical to an earlier Becke-86 (not Becke-88!) potential. Becke has suggested 3 terms in the series and Handy et. al. have used 5 terms. The five terms have been used to fit the potential to a set of accurate ab initio results. The exchange potential has 5 parameters and the correlation potential has 10 parameters. Handy et. al. have defined a number of parameter sets based on their set of molecules being fit. HyperChem reads a set of parameters from hcth.txt as the current set of parameters.

The HyperChem implimentation is from the following reference:

[F.A. Hamprecht, A.J. Cohen, D.J. Tozer, and N.C. Handy, *Development and Assessment of New Exchange-correlation Functionals*, J. Chem. Phys. **109**, 6264(1998)].

## Correlation Functionals

### VWN#5

The VWN correlatiooin potential developed by Vosko, Wilk, and Nusair is an LDA functional and one of the first relatively accurate LDA correlation functionals. The VWN authors defined a number of related potentials and recommended the #5 potential which is the one implimented in HyperChem. The program from Gaussian Inc. originally implimented a VWN functional, but not the #5. The VWN functional has a relatively complicate form with many parameters, the Random Phase Approximation, Pade Approximants, etc. and has little resonance with normal quantum chemistry. The authors used the Monte Carlo results of Ceperley and Alder on the correlation energy of the uniform electron gas to extract an analytical form for the correlation functional. The details are described in the paper:

[S.H. Vosko, L. Wilk, and M. Nusair, *Accurate Spin-Dependent Electron Liquid correlation Energies for Local Spin Density Calculations: A Critical Analysis*, Can. J. Phys. **58**, 1200 (1980)].

### Perdew-Zunger-81

Perdew and Zunger have also developed an LDA functional for the correlation energy by fitting to the same numerical Monte Carlo results of Ceperley and Alder. Their functional form is somewhat simpler than VWN but yields similar results. This Perdew-Zunger-81 LDA functional is sometimes used as the leading term in other GGA correlation potentials if it is not often used in isolation.

The fundamental reference for the Perdew-Zunger-81 functional is:

[J.P. Perdew and A. Zunger, *Self-Interaction Correction to Density-Functional Approximations for Many-Electron Systems*, Phys. Rev. B **23**, 5048 (1981)].

### Perdew-86

One of the first gradient-corrected correlation functionals was this one developed by John Perdew, also going under the name of GGA-I when combined with a gradient-corrected exchange functional developed by Perdew and Wang [J.P. Perdew and Y. Wang, *Accurate and Simple Density Functional for the Electronic Exchange Energy: Generalized Gradient Approximation*, Phys. Rev. B **33**, 8800 (1986)]. This exchange functional and GGA-I are not available in HyperChem although the correlation part of GGA-I, i.e. Perdew-86 is.

The basis reference for the HyperChem implementation of this gradient-corrected functional is:

[J.P. Perdew, *Density-Functional Approximation for the Correlation Energy of the Inhomogeneous Electron Gas*, Phys. Rev. B **33**, 8832 (1986)].

### LYP

The Lee-Yang-Parr correlation functional is described in the following two references:

[C. Lee, W. Yang, and R.G. Parr, Phys. Rev. **37**, 785 (1988) and B. Mihlich, A. Savin, H. Stoll, and H. Preuss, Chem. Phys. Lett. **157**, 200 (1989)].

However, the best description associated with the HyperChem implementation is given by:

[B.G. Johnson, P.M. Gill, and J.A. Pople, *Performance of a Family of Density-Functional Methods*, J. Chem. Phys. **98**, 5612 (1993)].

This last reference is also a valuable paper for the general details of how to implement a DFT program such as the one in HyperChem. Besides giving a straight-forward implementation of the LYP functional, they also give the appropriate details for the VWN and Becke-88 functionals.

### Perdew-Wang-91

This correlation functional is part of the GGA-II functional discussed above. The best source for the implementation details of this functional, although originally described by Perdew and Wang, are in the previously mentioned reference,

[J.P. Perdew, J.A. Chevary, S.H. Vosko, K.A. Jackson, M.R. Pederson, D.J. Sing, and C. Fiolhais, *Atoms, Molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation*, Phys. Rev. B **46**, 6671(1992).]

### PBE-96

As discussed above for exchange potentials, Perdew, Burke, and Ernzerhof have simplified the GGA-II functionals (PW-91) both for exchange (above) and for correlation (here). They eliminated a number of empirical parameters without sacrificing accuracy, according to the authors. One believes that the authors now recommend PBE-96 over PW-91 although HyperChem has both available for further clarification of the individual advantages.

The fundamental reference is:

[J.P. Perdew, K. Burke, M. Ernzerhof, *Generalized Gradient Approximations Made Simple*, Phys. Rev. Lett. **77**, 3865 (1996).]

### HCTH-98

As discussed for exchange potentials, Handy et. al. have developed a many-parameter exchange-correlation functional using the ideas of Becke-97 but without the exact Hartree-Fock exchange component. Becke has argued that accurate DFT results require a component of Hartree-Fock exchange. It is, however, more convenient



and easier to develop fast, new DFT methods if only a pure exchange functional is used. HCTH divides into a pure exchange functional and a pure correlation functional. The correlation functional has 10 parameters and different versions of these parameters exist for different situations. For example, the authors recommend a specific set of parameters for van der Waal's interactions [Arguments have been given by others that such interactions are not explicitly included in the standard DFT formulation!]

The basic paper describing the HCTH functional is:

[F.A. Hamprecht, A.J. Cohen, D.J. Tozer, and N.C. Handy, *Development and Assessment of New Exchange-correlation Functionals*, J. Chem. Phys. **109**, 6264(1998)].

## Custom and Hybrid Functionals

These functionals do not divide simply into a single exchange functional and a single correlation functional.

### B3-LYP

This functional, defined, by Becke, contains an exchange functional that consists of:

20% - Hartree-Fock Exchange

8% - Slater Exchange

72% - Becke-88 Exchange

plus a correlation functional that consists of:

19% VWN#5 Correlation

81% - LYP Correlation

This unusual combination [A.D. Becke, J. Chem. Phys. **98**, 5648 (1993)] was empirically determined by comparing with the results of very accurate calculations.

### B3-PW91

Like B3-LYP, this hybrid functional was determined empirically by Becke. The components are identical to B3-LYP except that the Perdew-Wang-91 correlation functional replaces the LYP correlation functional.

## EDF1

This functional is a slightly rearranged form of two versions of Becke-88 (with different  $b$  parameters) plus a custom version of the LYP correlation functional (with different  $a$ ,  $b$ ,  $c$ , and  $d$  parameters). It was optimized for a specific basis set (6-31+G\*). It performs better than Becke88-LYP or B3-LYP for this particular basis set and a representative set of molecules.

The basis reference is:

[R.D. Adamson, P.M.W. Gill, and J.A. Pople, *Empirical Density Functionals*, Chem. Phys. Lett. **284**, 6 (1998)].

## Becke-97

As discussed in the context of HCTH-98, the Becke-97 functional is a hybrid functional that looks like HCTH (but with 3 parameters rather than 5, i.e. 5 terms in the powers series for  $g(x)$ ). It has an admixture (19.43%) of Hartree-Fock exchange in it, as recommended by Becke. A principal difference between Becke-97 and HCTH-98 is that HCTH is pure while Becke-97 is a hybrid. The details of Becke-97 are given in the following reference:

[A.D. Becke, *Density-functional Thermochemistry. V. Systematic Optimization of Exchange-Correlation Functionals*, J. Chem. Phys. **107**, 8554 (1997)].

# Numerical Integration Schemes

The calculation of matrix elements of  $F^{XC}$ , the exchange-correlation matrix of the Kohn-Sham equations, requires computing integrals such as

$$\int \frac{\partial f}{\partial \rho^\alpha} \phi_\mu \phi_\nu d\mathbf{r}$$

Because even the simplest functional involves terms like  $(\rho^\alpha)^{4/3}$ , analytic formulas for these integrals are not available. They must be performed numerically. This numerical evaluation of matrix elements is one of the severe computational burdens of Density Functional Theory. The integral above involves a complicated integrand integrated over all space. Because of the lack of a unique

origin, the presence of multiple nuclei with their associated discontinuities, etc. such an integration can be very difficult.

Fortunately, Becke [A.D. Becke, *A Multicenter Numerical Integration Scheme for Polyatomic Molecules*, J. Chem. Phys. **88**, 2547 (1988)] has developed a scheme for decomposing these integrations into a sum of “normal” single-center integrations associated with each nucleus -- with integrands that fall away prior to reaching the discontinuity associated with a neighboring nucleus. This integration *trick* is the one used by many programs and is used by HyperChem.

It remains to specify the numerical grid used to perform the Becke one-center integrations. These are usually decomposed into a set of radial points for  $dr$  and a set of angular points for  $d\Omega$ .

## Angular Integrations

Many implimentors have used Lebedev grids for the angular integration and HyperChem does the same. These grids come with discrete numbers of points and HyperChem allows up to 194 Points in a Lebedev grid.

## Radial Integrations

Two versions of radial integrations are offered in HyperChem. The first is 32 Gauss-Chebyshev points with a substitution to modify the range from  $(-1, 1)$  to  $(0, \text{infinity})$ . The second is an Euler-MacLaurin set of points as suggested Murray, Handy, and Lamming [Mol.Phys. 78, 997 (1993)]. Unlike Gauss-Chebyshev points these points come in any specified quantity. HyperChem allows up to 50 Euler-MacLaurin radial points.

In addition to these generic integration grids, HyperChem offers the Standard Grid #1 (SG-1) specified by Gill, Johnson, and Pople [Chem. Phys. Lett. 209, 506 (1993)]. This is essentially 50 radial Euler-MacLaurin points with the angular Lebedev grid dependent on the particular value according to an algorithm they give for partitioning the radial points. For example, fewer angular points are needed at small  $r$  near the nucleus. The SG-1 grid is built into HyperChem as a simple standard option without having to specify the grid further. If one desires to explore other grids one must use a more regular  $(m,n)$  set of  $m$  radial points and  $n$  angular points.

HyperChem tries to reduce the magnitude of the numerical integration scheme by trying to determine in advance when the integrand is negligible. An option in the user interface can assist in reducing the effort. HyperChem allows a user to specify that any point with an electron density less than a criterion will be neglected. The criterion is set by the user depending on the accuracy needed in the integration. Neglecting values of electron density less than  $10^{-6}$  would appear to be reasonable while still retaining chemical accuracy.



# Chapter 14

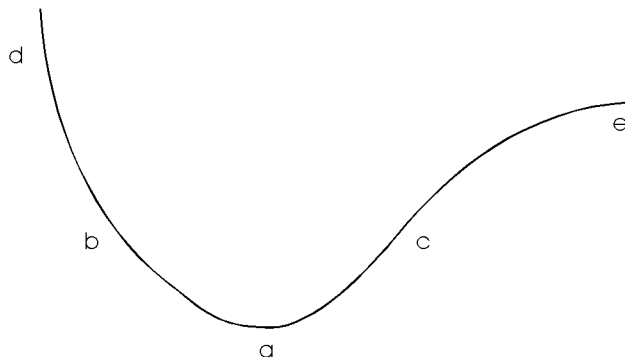
## Computational Options

HyperChem provides calculations that explore molecular potential energy surfaces. Indeed, most of computational chemistry relates in one way or another to molecular potential energy surfaces, the topography of the surface and motion on the surface.

### Single Points on a Potential Energy Surface

An  $N$ -atom molecular system may be described by  $3N$  Cartesian coordinates. Six independent coordinates (five for linear molecules, three for a single atom) describe translation and rotation of the system as a whole. The remaining coordinates describe the molecular configuration and the internal structure. Whether you use molecular mechanics, quantum mechanics, or a specific computational method (AMBER, CNDO, etc.), you can ask for the energy of the system at a specified configuration. This is called a single point calculation.

For a diatomic molecule, for example, there is only one internal coordinate and the energy as a function of configuration (internuclear distance) will look something like the following:



A single point calculation at the points b, c, d, or e will give a higher energy,  $E$ , than a single point calculation near the minimum at a.

In addition, a single point calculation yields information about the net forces on each atom. If  $(X_A, Y_A, Z_A)$  are the Cartesian coordinates of an atom A, then the gradient (force) on atom A has the components,  $(\partial E/\partial X_A, \partial E/\partial Y_A, \partial E/\partial Z_A)$ . The gradient at position b is positive for one atom and negative for the other atom in the direction of increasing the bond length. The gradient is large at d compared to the gradient at b, which is large compared to the gradient at points near the equilibrium distance a or near the dissociation limit e. Note that the gradient is not only zero at a minimum but at any extreme (maximum, transition state, etc.).

For multi-dimensional potential energy surfaces a convenient measure of the gradient vector is the root-mean-square (RMS) gradient described by

RMS Gradient =

$$(3N)^{-1} \left( \sum_A \left( \left( \frac{\partial E}{\partial X_A} \right)^2 + \left( \frac{\partial E}{\partial Y_A} \right)^2 + \left( \frac{\partial E}{\partial Z_A} \right)^2 \right) \right)^{1/2} \quad (175)$$

For a molecular mechanics calculation the energy and the gradient are essentially the only quantities available from a single point calculation. An analysis of the components of this molecular mechanics energy is placed in the log file for further detail. In the case of MM+, a much more complete description of the individual

contributions to the energy, including individual stretch energies, bending energies, etc., is placed in the log file along with the system dipole moment, if bond dipoles are available.

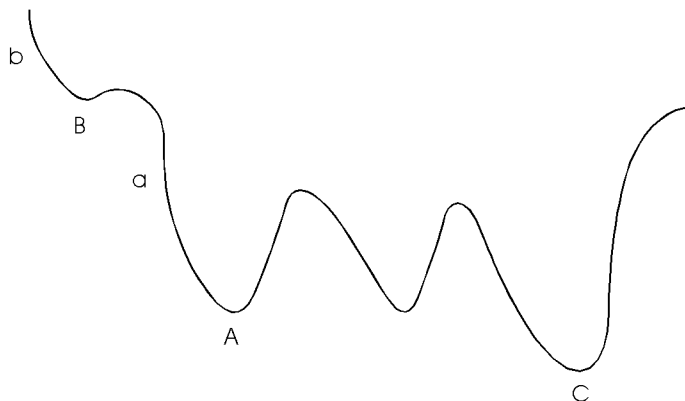
For a quantum mechanical calculation, the single point calculation leads to a wave function for the molecular system and considerably more information than just the energy and gradient are available. In principle, any expectation value might be computed. You can get plots of the individual orbitals, the total (or spin) electron density and the electrostatic field around the molecule. You can see the orbital energies in the status line when you plot an orbital. Finally, the log file contains additional information including the dipole moment of the molecule. The level of detail may be controlled by the `PrintLevel` entry in the `chem.ini` file.

## Local Minima on a Potential Energy Surface

While you might be interested in knowing the energy or gradient at any arbitrary point on a potential surface, you might be more interested in classifying the extrema of the potential surface. For a simple potential surface as illustrated above, there are two significant points on the surface, the global minimum and the dissociation limit (two isolated atoms). The first describes a stable configuration of this set of atoms and the second is an important configuration if one is studying chemical reactions. Both are desirable configurations to be obtained from an exploration of the potential surface. In this simple case, the dissociation limit can be known beforehand. In more complicated cases it is important to be able to discover configurations that represent dissociation into chemical products. The subject of chemical reactions (one of the two extrema shown above) is described below. Here we are more interested in the possibility of multiple minima on the same potential surface.

A HyperChem optimizer attempts to find extrema on a potential surface. That is, it finds points where all the components of the gradient vector go to zero. For the simple surface above, an optimizer would easily find the stable equilibrium configuration of the diatomic molecule, the only minimum on the surface. It would do this by starting at some initial guess, say point *c*, and iterating until it found point *a*. In more complicated cases, however, the potential surface is not so simple and might be qualitatively represented as something like the following.





This surface has multiple minima (four) with one global minimum. Starting at a is likely to lead to A, while starting at b could lead to B or to A depending on the shallowness of B, the step size, etc. Neither are likely to lead to the global minimum C. This general issue of optimizers leading to local minima rather than global minima is inherent in most procedures. The amount of effort to insure that a minimum is a global minimum is impractical in most problems with many degrees of freedom.

## Unconstrained Geometry Optimization

HyperChem has a set of optimizers available to explore potential surfaces. These differ in their generality, convergence properties and computational requirements. One must be somewhat pragmatic about optimization and switch optimizers or restart an optimizer when it encounters specific problems.

A particular problem with many optimizers relates to planar molecules. HyperChem directs optimization to proceed initially along directions associated with the force on an atom. If the starting point for an optimization is rigorously planar there may be no initial forces out of the plane, and optimization will lead only to the optimum planar form, which may or may not be a relevant structure. If this is the case, a slight nudge of any atom perpendicular to the plane will give a non-planar starting point and will lead to a lower energy non-planar structure.

HyperChem includes only unconstrained optimization. That is, given the coordinates  $\{X_A, Y_A, Z_A\}$  of a set of atoms A (the inde-

pendent variables), it attempts to find a minimum in the energy  $E$  (the objective function) as a function of the coordinates with no constraints on the value of these coordinates. In the process, there is no guarantee that a minimum is obtained rather than a transition state, etc. It is only guaranteed that the first derivatives have all gone to zero. The proper characterization of all extrema requires second derivatives which are not yet available in HyperChem.

HyperChem does not use constrained optimization but it is possible to restrain molecular mechanics and quantum mechanics calculations by adding extra restraining forces.

## Optimization Methods

Unconstrained optimization methods [W. H. Press, et. al., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 1986, Chapter 10] can use values of only the objective function, or of first derivatives of the objective function, second derivatives of the objective function, etc. HyperChem uses first derivative information and, in the Block Diagonal Newton-Raphson case, second derivatives for one atom at a time. HyperChem does not use optimizers that compute the full set of second derivatives (the Hessian) because it is impractical to store the Hessian for macromolecules with thousands of atoms. A future release may make explicit-Hessian methods available for smaller molecules but at this release only methods that store the first derivative information, or the second derivatives of a single atom, are used.

### Steepest Descent

Steepest Descent is the simplest method of optimization. The direction of steepest descent,  $\mathbf{g}$ , is just the negative of the gradient vector:

$$\mathbf{g} = -\nabla E = -\left(\frac{\partial E}{\partial X_1}, \frac{\partial E}{\partial Y_1}, \frac{\partial E}{\partial Z_1}, \frac{\partial E}{\partial X_2}, \frac{\partial E}{\partial Y_2}, \dots\right) \quad (176)$$

HyperChem uses the steepest descent by steps method. New points are found by:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \lambda_i \frac{\mathbf{g}_i}{|\mathbf{g}_i|} \quad (177)$$

where the step size  $\lambda_i$  is increased by a factor of 1.2 for the next step if the energy at  $\mathbf{x}_{i+1}$  is lower than at  $\mathbf{x}_i$ , or decreased by a half if the energy has risen.

The steepest descent by steps method may provide a reasonably good method to begin an optimization when the starting point is far from the minimum. However, it converges slowly near the minimum and it is principally recommended only to initiate optimization when the starting point is particularly bad.

The principal difficulty with steepest descent is that the successive directions of search,  $\mathbf{g}_i, \mathbf{g}_{i+1}, \dots$  are not conjugate directions.

Searching along the successive directions of steepest descent might, for example, keep skipping back and forth *across* a long narrow valley. A conjugate direction would be *along* the long narrow valley. Conjugate gradient methods are a considerable improvement over steepest descent methods and involve search down conjugate directions rather than steepest descent directions.

## Conjugate Gradient Methods

If you iterate in the steepest descent method, you are recomputing the first derivatives all the time but not essentially looking at how these first derivatives are changing. A better approach would be to use the information on a history of first derivatives to implicitly (or, explicitly, if the Hessian could be stored) gather second derivative information. The basis of conjugate gradient methods is that the history of first derivatives allows a more intelligent search direction than just the direction of steepest descent. These methods search in conjugate gradient directions  $\mathbf{h}_i$  rather than simple steepest descent directions  $\mathbf{g}_i$ .

Another difference from steepest descent is that a one-dimensional minimization is performed in each search direction. A line minimization is made along a direction  $\mathbf{h}_i$  until a minimum energy is found at a new point  $i+1$ ; then the search direction is updated and a search down the new direction  $\mathbf{h}_{i+1}$  is made. This is repeated until convergence is obtained. The line minimization method takes increasing steps along the search direction until a minimum in that direction is bracketed. It then interpolates to the

minimum of the cubic polynomial fit to the energy and its directional derivative at the bracketing points.

If a surface is quadratic it can be characterized by its Hessian matrix  $\mathbf{A}$  of second derivatives of the energy with respect to the Cartesian coordinates of the atoms A, B, ..., i.e.  $A_{ij} = \partial^2 E / \partial X_A \partial Y_B$ , etc. The conjugate directions are defined as a set of  $3N$  search directions having the property that for a quadratic surface a successive search down  $3N$  conjugate directions leads to the exact minimum. The conjugate directions are defined by:

$$\mathbf{h}_i \bullet \mathbf{A} \bullet \mathbf{h}_j = 0 \quad (178)$$

Conjugate Gradient methods compute the conjugate directions  $\mathbf{h}_i$  by iterative computation involving the gradient  $\mathbf{g}_i$  without recourse to computation of second derivatives. The steps in a conjugate gradient minimization are:

1. A starting point is defined and the initial conjugate direction is chosen to be the steepest descent direction,  $\mathbf{h}_0 = \mathbf{g}_0$ .
2. A line minimization is performed along the conjugate direction  $\mathbf{h}_i$  until a minimum is found that defines the next point, where the gradient is re-evaluated as  $\mathbf{g}_{i+1}$ . If the RMS gradient is below the convergence threshold the procedure is terminated, otherwise:
3. The conjugate direction is updated according to the formula,  $\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_{i+1} \mathbf{h}_i$ , where the coefficient  $\gamma_{i+1}$  is computed from the new and old gradients  $\mathbf{g}_{i+1}$  and  $\mathbf{g}_i$ . Two slightly different procedures for computing this coefficient are used in HyperChem and the two procedures are referred to as the Fletcher-Reeves method and the Polak-Ribiere method.
4. A return to step 2 is made with the new conjugate direction.

## Fletcher-Reeves

This procedure computes the coefficient  $\gamma_{i+1}$  as:

$$\gamma_{i+1} = \frac{\mathbf{g}_{i+1} \bullet \mathbf{g}_{i+1}}{\mathbf{g}_i \bullet \mathbf{g}_i} \quad (179)$$

The conjugate direction is reset to the steepest descent direction every 3N search directions or cycles, or if the energy rises between cycles.

### Polak-Ribiere

This procedure computes the coefficient  $\mathbf{g}_{i+1}$  as:

$$\gamma_{i+1} = \frac{(\mathbf{g}_{i+1} - \mathbf{g}_i) \bullet \mathbf{g}_{i+1}}{\mathbf{g}_i \bullet \mathbf{g}_i} \quad (180)$$

For quadratic functions this is identical to the Fletcher-Reeves formula but there is some evidence that the Polak-Ribiere may be somewhat superior to the Fletcher-Reeves procedure for non-quadratic functions. It is not reset to the steepest descent direction unless the energy has risen between cycles.

### Newton-Raphson Methods

The full Newton-Raphson method computes the full Hessian  $\mathbf{A}$  of second derivatives and then computes a new guess at the 3N coordinate vector  $\mathbf{x}$ , according to

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{A}^{-1} \bullet \mathbf{g}_i \quad (181)$$

Numerous optimization methods aim at approximating the Hessian (or its inverse) in various ways.

### Block Diagonal Newton-Raphson

The block diagonal Newton-Raphson method is an option for MM+ minimizations only. This method derives from Allinger's original MM2 program. It retains diagonal blocks of the Hessian associated with individual atoms and neglects off-diagonal blocks coupling the minimization of two different atoms.

Pragmatically, the procedure considers only one atom at a time, computing the 3x3 Hessian matrix associated with that atom and the 3 components of the gradient for that atom and then inverts the 3x3 matrix and obtains new coordinates for the atom according to the Newton-Raphson formula above. It then goes on to the next atom and moves it in the same way, using first and second derivatives for the second atom that include any previous motion of atoms.

The procedure uses second derivative information and can be quite efficient compared to conjugate gradient methods. However, the neglect of coupling in the Hessian matrix can lead to situations where oscillation is possible. Conjugate gradient methods, although perhaps slower in reaching a minimum, implicitly involve the full Hessian.

### Powell's Conjugate Directions

A conjugate direction method attempts to find search directions that are “non-interfering” or conjugate. This means that if you optimize along a direction  $\mathbf{u}$  to a minimum and then along a new direction  $\mathbf{v}$ , the optimization along  $\mathbf{v}$  will not “spoil” the optimization along  $\mathbf{u}$ .

Powell's method starts with a coordinate vector  $\mathbf{x}_i$  and searches along some direction  $\mathbf{u}_i$  to find a new coordinate vector  $\mathbf{x}_{i+1}$ . A new search direction  $\mathbf{u}_{i+1}$  then becomes  $\mathbf{x}_{i+1} - \mathbf{x}_i$ . This is continued until a set of search directions,  $\{\mathbf{u}_i, i=1..3N\}$  are conjugate and a minimum is found.

## Transition Structures on a Potential Energy Surface

As indicated earlier, potential energy surfaces are usually quite complex, with more than one minimum on the surface. In the illustration above, the potential surface had four minima that were discussed in the geometry optimization section. Between the energy minima there are three transition structures, located at the local maximum energy values along the reaction coordinate path. HyperChem has two methods to locate the transition energy structures: eigenvector following and synchronous transit.

Exploring reaction mechanisms is a major challenge for chemical theory. The geometries of reactants and products may generally be obtained using wide range of spectroscopic methods. The same techniques provide little if any information about connecting pathways. Activation energies, *i.e.*, energies of transition structures relative to reactants, can also be experimentally determined in most cases. Geometries of transition structures, local energy maxima along such reaction pathways, usually can only be hypothesized from theory. Computational methods can also be applied to characterize short-lived reactive intermediates, which correspond to shallow local minima on the potential energy surface. Such

reactive intermediates frequently can be determined experimentally, but detailed information about their geometries is usually very difficult to obtain.

Theory can be used to examine any arrangement of atoms in a molecule. A reaction potential surface can be explored to any desired degree, using the same computational methods as for geometry optimization of reactants and products. In theory, it is possible to determine whether or not a given structure corresponds to a local minimum (stable intermediate) or a saddle point (transition structure). Favored reaction pathways can then be obtained as those involving progression from reactants to products over the lowest-energy transition structures.

The success of simple theoretical models in determining the properties of stable molecules may not carry over into reaction pathways. Therefore, *ab initio* calculations with larger basis sets may be more successful in locating transition structures than semi-empirical methods, or even methods using minimal or small basis sets. Nevertheless, this area is one of the richest fields for chemical theory, with great potential to deepen our understanding of the way chemical reactions occur.

## Transition States Search Methods

In HyperChem, two different methods for the location of transition structures are available. Both are the combinations of separate algorithms for the maximum energy search and quasi-Newton methods. The first method is the eigenvector-following method, and the second is the synchronous transit method.

### Eigenvector Following Method

HyperChem uses the eigenvector following method described in Baker, J., *J. Comput. Chem.*, **7**, 385-395 (1986), where the details of the procedures can be found.

The eigenvector following algorithm comprises several consecutive steps. First is needed to make an initial guess  $X$  of the position of the transition state. The gradient vector  $\mathbf{g}$  and the Hessian matrix  $\mathbf{H}$  at the initial point are then calculated (or estimated). If analytical second derivatives are available in the calculational method, the exact Hessian can be calculated. For the other wave functions, the Hessian can be estimated by stepping through each

variable and doing a finite-difference calculation on the gradient. The second step involves the diagonalization of the Hessian and determination of the local surface characteristics, *i.e.*, the number of negative eigenvalues. Then  $\mathbf{g}$  is transformed into the local Hessian modes  $\mathbf{F}=\mathbf{U}\mathbf{g}$ .

The next step depends on the structure of the Hessian. If the Hessian has the wrong number of negative eigenvalues (more than one) then the next step would be P-RFO step, which involves separation of positive and negative hessian eigenvalues, that gives two matrix equations. If the mode following has been switched on, then eigenvector followed will be saved. The subsequent steps will determine which Hessian mode has the greatest overlap with the eigenvector followed on the previous cycle, and save and follow this mode.

The next step will determine optimization convergence. If the criteria are satisfied, HyperChem will stop at this point, having found the position of the transition state. If convergence criteria are not satisfied, it will calculate the energy and gradient vector at the new point, provided that maximum number of steps has not been exceeded.

## Synchronous Transit Method

HyperChem uses the synchronous transit method described in Peng, C., and Schlegel, H.B., *Israel Journal of Chemistry*, **33**, 449-454 (1993), and the details of the method can be found there.

The synchronous transit method is combined with quasi-Newton methods to find transition states. Quasi-Newton methods are very robust and efficient in finding energy minima. Based solely on local information, there is no unique way of moving uphill from either reactants or products to reach a specific reaction state, since all directions away from a minimum go uphill.

HyperChem has two synchronous transit methods implemented. The linear synchronous transit method (LST) searches for a maximum along a linear path between reactants and products. It may happen that this method will end up with a structure having two or more negative eigenvalues. The quadratic synchronous transit method (QST) is an improvement of LST approach and searches for a maximum along a parabola connecting reactants and products, and for a minimum in all directions perpendicular to the parabola.



Let  $\mathbf{X}$ ,  $\mathbf{R}$  and  $\mathbf{P}$  be the coordinates of the current point, the reactants, and products. (Note that it is necessary to evaluate the geometries of the reactants and products using the same method or the same basis set as the one used here to calculate the transition structure). In the linear synchronous transit approach,  $\mathbf{X}$  is on a path that is a linear interpolation between  $\mathbf{R}$  and  $\mathbf{P}$ . The quadratic synchronous transit method uses a curved path through  $\mathbf{X}$ ,  $\mathbf{R}$  and  $\mathbf{P}$ . In both cases, a maximum is found along the path. The tangent to the synchronous transit path (LST or QST) is used to guide the optimization to the quadratic region of the transition state. Then the tangent to the path is used to choose the best eigenvector for the ascent direction and the quasi-Newton method is used to complete the optimization. It is also possible to use an eigenvector-following step to complete optimization.

## **Molecular Dynamics on a Potential Energy Surface**

HyperChem uses molecular mechanics or quantum mechanics to compute a potential energy surface. This surface is the energy of a molecular system as a function of the 3N Cartesian coordinates of the N atoms comprising the molecular system. Having obtained such a surface or having a method to determine the energy and its derivatives at any point on the surface, you are in a position to investigate many chemical phenomena.

Reality suggests that a quantum dynamics rather than classical dynamics computation on the surface would be desirable, but much of chemistry is expected to be explainable with classical mechanics only, having derived a potential energy surface with quantum mechanics. This is because we are now only interested in the motion of atoms rather than electrons. Since atoms are much heavier than electrons it is possible to treat their motion classically. Quantum scattering approaches for small systems are available now, but most chemical phenomena is still treated by a classical approach. A chemical reaction or interaction is a classical trajectory on a potential surface. Such treatments leave out phenomena such as tunneling but are still the state of the art in much of computational chemistry.

## Temperature

Classical mechanics involves studying motion (trajectories) on the potential surface where the classical kinetic energy equates to the temperature. The relationship is that the average kinetic energy of the  $N$  atoms,  $1/2 N^{-1} \sum_A m_A (v_{xA}^2 + v_{yA}^2 + v_{zA}^2)$  is equated to  $3/2 kT$ , where  $m_A$  is the mass of the  $A^{\text{th}}$  atom,  $v_{xA}$  is the x-component of the velocity of the  $A^{\text{th}}$  atom,  $k$  is the Boltzmann constant and  $T$  is the temperature. When the number of degrees of freedom in the microscopic system is small, the relationship between the average velocity and the macroscopic temperature may be dubious unless many different configurations of the microscopic system are considered. For example, to study a reaction at a specific temperature requires using a Boltzmann distribution of initial conditions for the reacting (or non-reacting) trajectory.

## Statistical Averaging

Statistical mechanics states that the macroscopic values of certain quantities, like the energy, can be obtained by ensemble averaging over a very large number of possible states of the microscopic system. In many realms of chemistry, these statistical averages are what computational chemistry requires for a direct comparison with experiment. A fundamental principle of statistical mechanics, the Ergodic Hypothesis, states that it is possible to replace an ensemble average by a time average over the trajectory of the microscopic system. Molecular dynamics thus allows you to compute a time average over a trajectory that, in principle, represents a macroscopic average value. These time averages are fundamental to the use of molecular dynamics.

## Background

This section describes a minimal theoretical background so you can understand the process of creating a classical molecular dynamics trajectory and use it to obtain a visual or statistical result.

## Newton's Equations of Motion

A classical molecular dynamics trajectory is simply a set of atoms with initial conditions consisting of the 3N Cartesian coordinates of N atoms A ( $X_A, Y_A, Z_A$ ) and the 3N Cartesian velocities ( $\mathbf{v}_{xA}, \mathbf{v}_{yA}, \mathbf{v}_{zA}$ ) evolving according to Newton's equation of motion:

$$\mathbf{a}_{xA} = d\mathbf{v}_{xA}/dt = (m_A)^{-1}\mathbf{F}_{xA} = (m_A)^{-1}\partial E/\partial X_A \quad (182)$$

where  $\mathbf{a}_{xA}$  is the acceleration in the X direction of atom A, and  $\mathbf{F}_{xA}$  is the total force in the X direction on atom A arising from all the other atoms.

## Leap-frog Algorithm

The leap-frog algorithm uses the simplest central difference formula for a derivative

$$\frac{df}{dt} = \frac{f(t + \Delta t) - f(t - \Delta t)}{2\Delta t} \quad (183)$$

as the basis for a simple iterative scheme to integrate Newton's equations of motion. If  $\mathbf{v}_{-1/2}$  is a first guess at a velocity (for the time  $t_0 - 1/2\Delta t$ ) and  $\mathbf{x}_0$  is a position at time  $t_0$ , then by computing first a new velocity, then a new coordinate, then a new velocity, etc., we can integrate as follows:

$$\mathbf{v}_{\frac{1}{2}} = \mathbf{v}_{-\frac{1}{2}} + \mathbf{a}_0\Delta t \quad (184)$$

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v}_{1/2}\Delta t \quad (185)$$

$$\mathbf{v}_{3/2} = \mathbf{v}_{1/2} + \mathbf{a}_1\Delta t \quad (186)$$

$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{v}_{3/2}\Delta t \quad (187)$$

The new coordinates ( $\mathbf{x}$ ) and accelerations ( $\mathbf{a}$ ) are computed at integral times and the velocities ( $\mathbf{v}$ ) at half integral times. The time step  $\Delta t$  entered by the user is the time between evaluations of  $\mathbf{a}$ , i.e.,  $\Delta t = t_1 - t_0$ . The temperatures reported at integral times are the averages of the values on either side, determined from  $\mathbf{v}_{i+1/2}$  and  $\mathbf{v}_{i-1/2}$

## Statistical Mechanical Averages

Statistical mechanical averages in a molecular dynamics run are obtained by simply averaging an energetic or structural value over time steps. Thus if the values  $\{x_i, i\}$  are being computed in a trajectory, the statistical mechanical average is just

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i \quad (188)$$

For purposes of exploring fluctuations and determining the convergence of these statistical averages the root mean square (RMS) deviation in  $x$  is also computed:

$$D_x = [\langle x^2 \rangle - \langle x \rangle^2]^{1/2} \quad (189)$$

## Random Velocities or Restart Velocities

As described previously, the Leap-frog algorithm for molecular dynamics requires an initial configuration for the atoms and an initial set of velocity vectors  $\mathbf{v}_{-1/2}$ . These initial velocities can come from two sources; the status of the HyperChem molecular dynamics Restart check box determines the source of these velocities.

If Restart is not checked then the velocities are randomly assigned in a way that leads to a Maxwell-Boltzmann distribution of velocities. That is, a random number generator assigns velocities according to a Gaussian probability distribution. The velocities are then scaled so that the total kinetic energy is exactly  $3/2$  kT where T is the specified starting temperature. After a short period of simulation the velocities evolve into a Maxwell-Boltzmann distribution.

If Restart is checked then the velocities are the existing assigned velocities derived from a previous molecular dynamics simulation or included in the HIN file when it was first read in. You can thus restart a trajectory at exactly the point it was terminated by using the Restart check box.

The Restart check box can be used in conjunction with the explicit editing of a HIN file to assign completely user-specified initial velocities. This may be useful in classical trajectory analysis of chemical reactions where the initial velocities and directions of the reactants are varied to statistically determine the probability of

reaction occurring, or not, in the process of calculating a rate constant.

## The Basic Phases of a Trajectory

A given molecular dynamics trajectory can be divided into three sequentially-ordered phases:

- Heating
- Running
- Cooling

Any one of these is optional. The running phase is often thought of as first an equilibration phase and then an averaging phase, although HyperChem does not specifically distinguish these two.

The heating phase is used to take a molecular system smoothly from lower temperatures, indicative of a static initial (possibly optimized) structure, to the temperature  $T$  at which it is desired to perform the molecular dynamics simulation. The run phase then constitutes a simulation at temperature  $T$ . If the heating has been done carefully, it may be possible to skip the equilibration phase and start collecting statistics immediately. If the heating has been done rapidly, then it may be necessary to run the simulation for an extended period of time to equilibrate the sample fully prior to collecting statistics. The cooling phase is simply to enable simulated annealing as discussed below. Proper heating and equilibration are appropriate to the use of molecular dynamics in studying equilibrium statistical mechanical properties but may not be relevant for certain non-equilibrium studies.

## The Fundamental Time Step

Molecular dynamics involves the integration of Newton's equations of motion. To perform this integration, a discrete time step,  $\Delta t$ , must be used. A large value of this fundamental time step will result in inaccurate integration, and a small value will waste computer resources taking very small steps. The appropriate value of the time step is related to the frequency of the motions involved, which are in turn related to the value of force constants and atomic masses. Large force constants and small masses imply high frequency motion and need small time steps to preserve integration accuracy. If too large a time step is used, it is possible to have

a basic instability in the equations that results in a molecule blowing apart, i.e. in the molecular coordinates becoming too large, inappropriate, etc.

The time step should be at least an order of magnitude lower than the shortest period of internal motion. If hydrogens are present in a system near room temperature a value of 0.5 femtoseconds (0.0005 ps) is usually appropriate.

## Heating and Cooling

If the heating time  $t_h$  or cooling time  $t_c$  are non-zero, or if the run time  $t_r$  is non-zero and constant temperature is selected, velocities are adjusted (rescaled) during the molecular dynamics run to change the temperature of the system.

In the heating phase (assuming  $t_h$  is non-zero), the velocities are periodically rescaled to change the system temperature from the initial temperature  $T_1$  to the simulation temperature  $T_2$  in increments of the temperature step  $\Delta T$ . The heating period for rescaling the velocities,  $P_h$ , is defined by:

$$P_h = \frac{t_h}{\Delta t} \times \frac{\Delta T}{|T_2 - T_1|} \quad (190)$$

where  $\Delta t$  is the integration time step. Thus every  $P_h$  steps in the heating phase, the current velocities  $\mathbf{v}_{i+1/2}$  are rescaled to give velocities  $\mathbf{v}'_{i+1/2}$  with the desired temperature  $T_d$ :

$$\mathbf{v}'_{i+1/2} = \sqrt{\frac{T_d}{T_{i+1/2}}} \times \mathbf{v}_{i+1/2} \quad (191)$$

where  $i$  is the step number and  $T_{i+1/2}$  is the temperature corresponding to the velocities  $\mathbf{v}_{i+1/2}$  (rather than the temperature reported for step  $i$  that is the average of  $T_{i-1/2}$  and  $T_{i+1/2}$ ). The scaled velocities  $\mathbf{v}'_{i+1/2}$  are used to compute the next set of atomic coordinates. The desired temperature  $T_d$  is incremented from  $T_1$  to  $T_2$  in increments of the temperature step  $\Delta T$  (for the final heat phase rescaling at  $t_h$ ,  $T_d$  is set to  $T_2$ ). Note that the heating phase may actually be used to cool the system if  $T_1$  is greater than  $T_2$ .

In the cooling phase (assuming  $t_c$  is non-zero), the velocities are periodically rescaled to change the system temperature from the run temperature  $T_2$  to the final temperature  $T_3$  in increments of

the temperature step  $\Delta T$ . The cooling period for rescaling the velocities,  $P_c$ , is defined by:

$$P_c = \frac{t_c}{\Delta t} \times \frac{\Delta T}{|T_2 - T_3|} \quad (192)$$

where  $\Delta t$  is the integration time step. Thus every  $P_c$  steps in the cooling phase the current velocities  $\mathbf{v}_{i+1/2}$  are rescaled to give velocities  $\mathbf{v}'_{i+1/2}$  with the desired temperature  $T_d$ , as in the heating phase, except that the desired temperature  $T_d$  is changed from  $T_2$  to  $T_3$ , again in increments of the temperature step  $\Delta T$  (for the final cooling phase, rescaling at  $t_h + t_c$ ,  $T_d$  is set to  $T_3$ ). Note that the cooling phase may actually be used to heat the system if  $T_3$  is greater than  $T_2$ .

When using the heating and cooling features of HyperChem it should be remembered that it is accomplished through rescaling of the velocities, so if the velocities are zero no temperature can occur. This happens, for example, if you start with an exactly optimized structure and heat from a starting temperature  $T_1$  of zero, or use the restart option when velocities are all zero.

It is also desirable to let the system equilibrate somewhat at each value of  $T_d$ . This may be accomplished by increasing the rescaling periods  $P_h$  and  $P_c$  through a longer cooling time or larger temperature step, though the larger the temperature step, the greater the perturbation to the system when rescaling. A compromise value of  $\Delta T$  needs to be chosen that allows sufficient equilibration at each temperature but does not unduly perturb the system when rescaling.

## Equilibration at a Temperature T

When using molecular dynamics to study equilibrium properties like enthalpy (the average energy), etc., you want the average over a trajectory to be equivalent to an ensemble average. This means the system must be in equilibrium; the initial conditions have been forgotten and you are sampling from a set of phase space configurations representative of the macroscopic equilibrium state. You should not begin sampling for the purpose of collecting statistical averages until this equilibration is performed. The lack of any long term drift is one indication of possible equilibration. To achieve equilibration to a temperature  $T$ , it may be necessary to rescale velocities through the use of the constant temperature bath

algorithm, (the use of a heating phase with a small temperature step and overall temperature change such as one degree or less) or by re-initializing periodically. Equilibration requires the temperature to fluctuate about the requisite value  $T$ .

Naturally, the study of non-equilibrium properties involves different criteria although the equilibrium state and evolution towards the equilibrium state may be important.

## **Collecting Data**

Subsequent to equilibration, averages over the trajectory can be accumulated to describe statistical mechanical properties. For example, to calculate an average bond length, the bond should first be selected, prior to collecting molecular dynamics data or playing back snapshots, and made a named selection with the Select/Name Selection menu item. Then, the named selection should be placed in the Average only or Avg. & graph column of the Molecular Dynamics Averages dialog box invoked by the Averages button of the Molecular Dynamics Options dialog box. A molecular dynamics simulation will then average the bond length. The average may be viewed after the sampling by re-opening the Molecular Dynamics Averages dialog box and selecting the desired property, so that it is outlined, in the Average only or Avg. & graph column. If a graph was requested, the average appearing here is equivalent to the average over the simulation or playback period.

In deciding the convergence of these averages, the RMS deviation of a value from its average (i.e.,  $Dx$ ) may be a very useful indicator.

## **Free Dynamics or Constant Temperature Dynamics**

If the Constant temperature option has not been selected, HyperChem's molecular dynamics uses constant volume and constant total energy corresponding to a microcanonical ensemble. The temperature then becomes something to be calculated rather than specified (it is used, however, to specify the initial kinetic energy). This is free dynamics. The total energy remains constant but there is a free flow of energy back and forth between the potential energy and the kinetic energy (the temperature). If, initially, the system is far from equilibrium (high potential energy), potential energy is converted to kinetic energy as the trajectory evolves and the temperature rises significantly from the initial temperature. If a molecular dynamics simulation is stopped and started again without the Restart box being checked, a re-equilibration of



velocities will be performed and such excess energy purged. The temperature always fluctuates, however, rather than remaining constant, and may continue to drift as the trajectory evolves.

Berendsen *et al.* [H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. di Nola, and J. R. Haak, *J. Chem. Phys.* 81, 3684 (1984)] have described a simple scheme for constant temperature simulations that is implemented in HyperChem. You can use this constant temperature scheme by checking the constant temperature check box and specifying a bath relaxation constant  $\tau$ . This relaxation constant must be equal to or bigger than the dynamics step size  $\Delta t$ . If it is equal to the step size, the temperature will be kept as close to constant as possible. This occurs, essentially, by rescaling the velocities used to update positions to correspond exactly to the specified initial temperature. For larger values of the relaxation constant, the temperature is kept approximately constant by superimposing a first-order relaxation process on the simulation. That is:

$$\frac{dT}{dt} = \frac{(T_d - T)}{\tau} \quad (193)$$

where  $T_d$  is the specified desired temperature. This is accomplished by scaling the velocities every time step during the running phase by

$$\mathbf{v}'_{i+1/2} = \sqrt{\left(\frac{T_d}{T_{i+1/2}} - 1\right) \times \frac{\Delta t}{\tau} + 1} \times \mathbf{v}_{i+1/2} \quad (194)$$

If the relaxation constant is large there is little effect on a trajectory apart from the long term drift towards fluctuation about  $T_d$ , and the reduction of cumulative numerical roundoff error.

## Data, Averaging, Snapshot, and Screen Refresh Periods

HyperChem includes a number of time periods associated with a trajectory. These include the basic time step in the integration of Newton's equations plus various multiples of this associated with collecting data, the forming of statistical averages, etc. The fundamental time period is  $\Delta t_1 \cong \Delta t$ , the integration *time step* set in the Molecular Dynamics dialog box.

## The Data Collection Period

HyperChem runs the molecular dynamics trajectory, averaging and analyzing a trajectory and creating the Cartesian coordinates and velocities. The period for reporting these coordinates and velocities is the data collection period  $\Delta t_2$ . It is a multiple of the basic time step,  $\Delta t_2 = n_2 \Delta t_1$ , and is also referred to as a data step. The value  $n_2$  is set in the Molecular Dynamics options dialog box.

## The Statistical Averaging Period

You can average and plot various energetic and structural quantities and save the instantaneous values in a file (in comma separated value, or CSV format). The period for such activity is called the Average/graph period  $\Delta t_3$ . The quantities to be averaged are thus collected at times  $t_0$ ,  $t_0 + \Delta t_3$ ,  $t_0 + 2\Delta t_3$ , etc. The Average/graph period is specified in the Averages dialog box by  $n_3$  data steps, i.e. as a multiple of the data collection period,  $\Delta t_3 = n_3 \Delta t_2$ .

## The Snapshot Collection Period

HyperChem can store a file of snapshots of a trajectory for subsequent analysis. The period for doing so is referred to as the Snapshot period  $\Delta t_4$ . The snapshots are made at times  $t_0$ ,  $t_0 + \Delta t_4$ ,  $t_0 + 2\Delta t_4$ , etc. The Snapshot period is specified in the Snapshot dialog box by  $n_4$  data steps, i.e. as a multiple of the data collection period,  $\Delta t_4 = n_4 \Delta t_2$ .

## The Screen Refresh Period

HyperChem updates the screen during a trajectory at regular intervals so you can visualize the trajectory. Since this screen update may slow down a trajectory if it occurs too frequently, you can specify the duration of the Screen Refresh period  $\Delta t_5$ . The screen updates at times  $t_0$ ,  $t_0 + \Delta t_5$ ,  $t_0 + 2\Delta t_5$ , etc. The Screen Refresh period is specified in the Molecular Dynamics options dialog box by  $n_5$  data steps, i.e. as a multiple of the data collection period,  $\Delta t_5 = n_5 \Delta t_2$ .

## Averaging Energetic and Structural Data

Statistical mechanical averaging is a fundamental aspect of molecular dynamics. While there may be many reasons for wanting to generate a molecular dynamics trajectory, using it to obtain statis-

tical mechanics averages may be the most common. In the micro-canonical ensemble of constant volume and number of particles, the appropriate equilibrium macroscopic average of some variable  $x(\mathbf{r})$  depending on the configuration  $\mathbf{r}$  of the particles is given by,

$$\langle x \rangle = \left( \frac{1}{Z} \right) \int x(\mathbf{r}) \exp\left(-\frac{V(\mathbf{r})}{kT}\right) d\mathbf{r} \quad (195)$$

where  $Z$  is the partition function and  $\exp[-V(\mathbf{r})/kT]$  is the Boltzmann factor associated with the potential energy of the configuration  $\mathbf{r}$ . By the nature of the ergodic hypothesis, any equilibrium value computed this way can also be computed by a time average from a molecular dynamics trajectory. That is:

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i = \frac{1}{N} \sum_{i=1}^N x(\mathbf{r}_i, t_i) \quad (196)$$

where the  $x_i$  are the values of  $x$  at time step  $i$ . In addition to equilibrium values, it is possible to compute dynamic values, like correlation functions, from the molecular dynamics approach. Although HyperChem includes no specific facility for such dynamic properties, the values stored in the \*.csv file or the snapshots of the trajectory in a \*.csv file can be used off-line to compute correlation functions.

## Averaging Energetic Values

You can request the computation of average values by clicking the Averages button in the Molecular Dynamics Options dialog box to display the Molecular Dynamics Averages dialog box. The energetic quantities that can be averaged appear in the left Selection column. When you select one or more of these energetic quantities (EKIN, EPOT, etc.) and click Add, the quantity moves to the Average Only column on the right. You can move quantities back to the left column by selecting them and clicking Del. Quantities residing in the Average Only column are not plotted but are written to a CSV file and averaged over the molecular dynamics trajectory. If you return to this dialog box after generating the trajectory and select one of the quantities so that the outline appears around it in the Average Only column, the average value over the last trajectory is displayed at the bottom of the column next to the word "Value".

Quantities, which are selected from the Averages Only column and added to the Avg. & graph column, will be written out and averaged, as described above, but will also be plotted on the molecular dynamics graph. To inspect the computed average value, select the quantity so that the outline appears around it and the average is displayed beside Value.

HyperChem allows the computation and display of the potential energy (EPOT), the kinetic energy (EKIN), or the total energy ETOT = EPOT + EKIN. The temperature (TEMP) can also be chosen for averaging or plotting but is simply related to EKIN by  $EKIN = 3/2 Nk TEMP$ .

For free dynamics where the constant temperature check box in the Molecular Dynamics dialog box is not checked, the total energy ETOT should remain constant. A fluctuation of the total energy about its average value of about 1 part in  $10^4$  is normally considered acceptable. A larger fluctuation, or a drift in the total energy, is symptomatic of an unphysical phase space trajectory, and generally indicates that the time step is too large. In free dynamics, however, the individual kinetic and potential energy contributions to this total energy will fluctuate.

For constant temperature dynamics where the constant temperature check box in the Molecular Dynamics Options dialog box is checked, the energy will not remain constant but will fluctuate as energy is exchanged with the bath. The temperature, depending on the value set for the relaxation constant, will approach constancy.

In addition to the energy quantities EKIN, etc., it is possible to average and plot their standard deviations D EKIN, etc. as described below.

## Averaging Named Selections

In addition to averaging and plotting energetic values associated with a trajectory, it is possible to average or plot structural quantities. The structural quantities that can be averaged are those that represent 2, 3, or 4-atom structural variables. Monitoring such a value represents a study of diffusion. A 2-atom structural quantity is a distance between atoms; any bonded or nonbonded interatomic distance can be monitored. A 3-atom structural quantity is an angle, and a 4-atom structural quantity is a dihedral angle.

To average or plot a structural quantity, the structural quantity must first be selected and named by the normal process for creating named selections (select the atoms and then use the menu item Select/Name Selection to give the selected atoms a name). From then on the Molecular Dynamics Averages dialog box will show these named selections as possible candidates to be averaged or plotted in addition to energetic quantities described above.

To average a torsion, select the four atoms of the torsion, name the torsion "tor," for example, and then select "tor" as the quantity to be averaged from the Molecular Dynamics Averages dialog box.

## Deviations from the Average

In addition to being able to plot simple instantaneous values of a quantity  $x$  along a trajectory and reporting the average,  $\langle x \rangle$ , HyperChem can also report information about the deviation of  $x$  from its average value. These RMS deviations may have particular significance in statistical mechanics or just represent the process of convergence of the trajectory values.

If a request for a deviation in a quantity  $x$  is made, the instantaneous values that are plotted represent:

$$Dx = [\langle x^2 \rangle_i - \langle x \rangle_i^2]^{1/2} \quad (197)$$

The  $i$  subscripts emphasize that the averaging is only up to the current step in the trajectory, not necessarily over the whole trajectory:

$$\langle x \rangle_i = \frac{1}{i} \sum_{j=1}^i x_j \quad (198)$$

The single average value that is reported for this quantity in the Molecular Dynamics averages dialog box is the limit reached by the plotted values at  $i=N$ , i.e. the RMS value of  $x$ :

$$[\langle x^2 \rangle - \langle x \rangle^2]^{1/2} = \langle (x - \langle x \rangle)^2 \rangle = \lim_{i \rightarrow N} Dx \quad (199)$$

In summary, for  $x$ , the average value is the average of the plotted values; for  $Dx$  the average value is the final value of the plotted values.

## The CSV File

When a molecular dynamics average is requested over a trajectory, certain quantities can be plotted as previously described. It is desirable, however, to allow the user to collect a full set of appropriate data values for possible custom use. Since HyperChem is not a full plotting package, it cannot allow you every desired plot. Therefore, HyperChem collects the basic data used for averaging and plotting into a comma-separated-values file, with the extension `.csv`, for general use in other applications. For example, the values can be readily brought into Microsoft Excel and plotted and manipulated in innumerable ways using the convenient plotting and statistical analysis features of Excel.

For every quantity you request to average, a column of the CSV file contains the  $N$  values of  $x_i$ . These are exactly the values that are plotted, and can be used to create further custom averages. For example, if you were interested in NMR, you might wish to average  $\langle r^{-3} \rangle$  for certain internuclear distances  $r$ . The instantaneous values of  $r$  could be placed in the `chem.csv` file and a spreadsheet program such as Microsoft Excel be used to generate and plot values of  $r^{-3}$  and its average over various regions of the trajectory.

## Plotting Instantaneous Values Along the Trajectory

Molecular dynamics is essentially a study of the evolution in time of energetic and structural molecular data. The data is often best represented as a graph of a molecular quantity as a function of time. The values to be plotted can be any quantity  $x$  that is being averaged over the trajectory, or the standard deviation,  $Dx$ . You can create as many as four simultaneous graphs at once.

While you can follow the progress of a trajectory, you cannot get publication quality graphs from HyperChem. You can, however, use the information in the CSV file with other programs for creating graphs for publication.

## Obtaining and Understanding MD Graphs

The graph associated with choosing an  $x$  is just the sequence of values of  $x_i$  as a function of  $i$ . The values of  $i$  run from  $i=0$  to  $i=N$  where  $N$  is the total number of time steps requested, i.e. the total run time in picoseconds divided by the time increment  $\Delta t$ . The graph appears in a fixed-size window with the total graph (deter-

mined by the value of  $N$ ) scaled to fit the window, if necessary. The plot lines appear only after enough points (the number of points is set by the chem.ini setting DynamicsGraphOnset) have been created to establish an initial vertical scale for the graph. The graph then fills all of the vertical space apart from a single pixel. The vertical scale factor is kept constant unless you explicitly request a rescaling by clicking on the rescale button (or performing the equivalent by pressing the space bar). If no rescaling is performed the plot may go off scale and not be seen until a rescale is performed.

As many as four plots are possible with colors red, green, blue and black. Each plot is scaled independent of the others and the colored labels identify the minimum and maximum of the plot and the name of the quantity being plotted. In addition to the values of  $x$ , the RMS deviation in  $x$  ( $Dx$ ) can be plotted. The plot of  $Dx$  converges to the RMS deviation of  $x$  at the end of the run and represents the current value of the RMS deviation at any point in the plot.

You can request a plot by selecting a quantity from the Averages Only column of the Molecular Dynamics Averages dialog box and passing it to the right (to the Avg. & graph column) by clicking the Add button. Only four quantities can be plotted as indicated by their presence in this last column of the dialog box. A molecular dynamics run with any quantity in the Avg. & graph column automatically pops up the Molecular Dynamics Results window to show requested plots. This window is of fixed size and cannot be resized, but can be moved or iconized to place it out of the way. It can always be made visible again by double-clicking on the icon. In addition to the plots and their labels, the window contains two buttons, one (Rescale) for rescaling the plot and one (Done) for deciding that plots are to be discontinued. Discontinuing the plotting of values does not terminate the trajectory. If new values are to be plotted, the trajectory must be terminated (by selecting Cancel from the Menu bar or by pressing the [Esc] key) and restarted using the restart check box in the Molecular Dynamics Options dialog box. Restarting the trajectory results in no discontinuity in the trajectory but does initiate averaging from scratch. If complete continuity is desired it may be necessary to save the CSV file prior to restart and concatenate individual CSV files.

## Placing Graphs into Other Documents

Since the Molecular Dynamics Results window containing plots is a true window, an image of it alone can be captured into the clipboard or a file using Top-level in the File/Preferences/Setup Image dialog box. This captured image, in addition to showing the molecular dynamics plots, shows the Restart and Done buttons, etc. If you only want the plots, you can erase the details of the box with a paint program, such as Microsoft Windows Paintbrush which comes with Microsoft Windows.

## Collecting Trajectory for Subsequent Playback

Particularly in the past, molecular dynamics has been batch-oriented where the trajectory is calculated as a one-shot operation requiring large amounts of computer time over a period of months with the trajectory being stored on tape or disk. The trajectory was then analyzed, perhaps many times, in much shorter time frames by reading the trajectory tape. HyperChem, which utilizes faster computers, uses more real-time, interactive molecular dynamics trajectories and averages, allowing a degree of experimentation not possible with a batch approach. Nevertheless, computing a trajectory, particularly for large molecules, can be a very lengthy process; therefore, HyperChem can compute a trajectory for storage and later playback for analysis, visualization, etc.

The molecular dynamics analyzes times steps, also called snapshots (coordinates and velocities), for display, averaging, and plotting (possibly from other applications). In the present release of HyperChem, two particular sources are relevant (the DDE interface allows the possibility of other generators of snapshots as well). The first source are time steps that are computed, displayed, and averaged. This is the normal real-time use of HyperChem molecular dynamics.

The second source of time steps is individual snapshots stored in a snapshot file (\*.*snp*) on disk and analyzed later. The limiting rate is determined by disk access times. Thus HyperChem, in addition to the real-time operation of molecular dynamics, allows the playback of molecular dynamics from previously created trajectories (snapshot files). A given snapshot file can be played back over and over again with different aspects of the trajectory being investigated each time. For a large molecule the trajectory might be created overnight, for example, and then analyzed interactively, playing back particular parts of the trajectory, analyzing different struc-



tural changes, etc. If you request frequent display of the time steps of a trajectory, the playback speed is limited by the display update time rather than disk speed.

## Creating a Snapshot (SNP) file

To create a set of snapshots of any molecular dynamics run, press the Snapshot button of the Molecular Dynamics Options dialog box to bring up the Molecular Dynamics Snapshots dialog box for naming a snapshot file. A snapshot file contains snapshots of the coordinates and velocities of a molecular system along the trajectory. The dialog box allows you to name the file and decide at what frequency to take snapshots. For example, choosing the snapshot period to be two data steps implies that only every other time step is stored in the snapshot file.

These snapshots, created for playback purpose (or for third-party purposes), consist of two components. The first is simply a HIN file that contains the starting configuration for the molecular dynamics run. The name of the snapshot becomes the name of the HIN file. The second component is the actual snapshots of the coordinates (velocities) stored in a SNP file. A snapshot stored as XXX will have a file, xxx.hin, containing the molecular system's starting configuration and a file, xxx.snp, containing the energies, coordinates and velocities of subsequent configurations. The snapshot HIN file contains an entry (for example, dynamics c:\hyper\xxx.snp) indicating that there exists a corresponding SNP file. When you read the HIN file it warns you that editing the molecule will invalidate the attached SNP file.

The format of the binary snapshot file is described in the Reference Manual.

## Reading a (HIN, SNP) File for Playback

To playback a trajectory that has been stored in a snapshot file you must first read the appropriate HIN file unless you have already saved snapshots with the current system. When reading the HIN file a warning will be issued that the attached SNP file is only valid if the molecular system in the HIN file is not edited in any way. That is, the attached \*.snp trajectory applies only to the molecular system as it is currently constituted. Any editing of the molecular system such as deleting an atom, etc. will gray the playback option, making it inaccessible. It is possible, however, to change the view of the system, define new named selections, etc. as long

as its atomic information is not altered. You need not decide what structural features you are interested in at the time of creation of the trajectory; you could, for example, name a new torsion angle and average that torsion angle only at playback time, with no need for the torsion to be defined before the snapshot creation.

To playback a trajectory rather than create one, you simply check the playback check box in the Molecular Dynamics Options dialog box. The check box is not available if no SNP file exists (all options of the dialog box that are not relevant to playback are also grayed and show the parameters used in the original creation of the snapshot). Clicking on the Snapshots button brings up a dialog box to set parameters for the playback. In all other regards the molecular dynamics run from playback is identical to a real-time run. Averaging, graphing and visualization of the evolving molecular system can be performed in the same way at playback time as at creation time. Real-time interactive exploration at the creation of a trajectory is appropriate when the trajectory is computed quickly. Exploration at playback time is appropriate when the trajectory is computed slowly instead.

### Running a Script at Playback

At playback time, it is possible to run a script at every time step. This may have many uses but a simple example might be to execute a quantum mechanical single point and plot a molecular orbital. The playback would then show the molecular orbital as a function of time during the trajectory.

The procedure for running a script involves giving a specific predefined name to the script and placing it in the HyperChem program directory. The name of the script must be **chem-dynamics.scr**. If such a script exists, then it is run at every playback step. If it does not exist in this directory, playback proceeds as normal.

One should be careful to remember to change the name of the script or move it when not desiring this capability.

## Global Minima on a Potential Energy Surface

The HyperChem optimization algorithms only guarantee that the minimum obtained is a local minimum. The global minimum of the potential energy surface is a desirable point to know but can be impractical to find when there are large numbers of degrees of free-

dom (large numbers of atoms). Many techniques are available, however, for exploring the surface and finding additional minima. Knowing that one of these additional minima is the global minimum is not easy, however. You could obviously try to find additional minima by using different starting points for a minimization. As shown in the diagram earlier, a potential surface can be very convoluted and different starting points will lead to different minima.

HyperChem has a facility for a more systematic approach to the global minimum than just choosing random starting points. This facility is associated with the idea of simulated annealing.

### **Simulated Annealing**

The basic concept of simulated annealing is to use the temperature and kinetic energy to get over energy barriers between a particular region with a local minimum and a generally lower energy region where the global minimum might exist. The pragmatic procedure involves heating the system to an artificially high temperature, letting molecular dynamics move the system to a more favorable potential energy region, and then letting the system gradually cool down to seek out the lowest energy without getting stuck in shallow local minima. Simulated annealing is described more extensively in the *Practical Guide*.

## **Simple Reactions on a Potential Energy Surface**

The theory of chemical reactions has many facets including elaborate quantum mechanical scattering approaches that treat the kinetic energy of atoms by proper wave mechanical methods. These approaches to chemical reaction theory go far beyond the capabilities of a product like HyperChem as many of the ideas are yet to have wide-spread practical implementations.

Fortunately, the theory of chemical reactions need not always require such sophistication. Indeed, since nuclei (atoms) are much heavier than electrons you can use classical mechanics to understand many chemical reactions. That is, as long as the potential energy surface is computed quantum mechanically, a classical mechanical treatment of motion on this potential surface can provide an adequate treatment of many chemical reactions. This approach is referred to as classical trajectory analysis and is the methodology for HyperChem's molecular dynamics with a poten-

tial surface that allows molecules to come apart, i.e., to react. In most molecular mechanics potential surfaces, the potential is such that it describes molecules only near their equilibrium geometry but does not describe dissociation or movement into totally new regions of the potential surface that equate to a chemical reaction occurring. Quantum mechanically calculated surfaces such as those available in HyperChem allow such reactions to be described. Basically HyperChem performs molecular dynamics but allows bonds to break and reform.

## Trajectory Analysis

The  $F + H - H \rightarrow F-H + H$  reaction is a common example of a reaction easily studied by classical trajectory analysis. The potential surface we are interested in is that for  $FH_2$ . This potential surface may have many extrema. One of them corresponds to an isolated Fluorine atom and a stable  $H_2$  molecule; these are the reactants. Another extremum of the surface corresponds to an isolated hydrogen atom and the stable H-F molecule; these are the products. Depending on how the potential surface was obtained there may or may not be an extremum corresponding to stable  $H_2F$ , but at the least you would expect an extremum corresponding to the transition state of the reaction being considered.

The reaction in question can be studied by exploring molecular dynamics on the potential energy surface paying particular attention to trajectories that lead from a region of the potential surface characterizing reactants to a region of the potential surface characterizing products. In the simplest scenario you start with  $H_2$  and an isolated F atom but then shoot the F atom at the  $H_2$  and watch the molecular dynamics trajectory to see what happens. If the trajectory leads to a region of the potential surface representing products, a score for *reaction* is tallied. If the trajectory terminates in a region of the potential surface that still corresponds to  $F + H_2$ , then a score for *non-reaction* is tallied. By giving the F atom a distribution of initial positions, velocities, and directions corresponding to experimental conditions, and performing the experiment over and over again, you can calculate the probability that the end result of a trajectory is *react over non-react*, and hence calculate a reaction rate constant. HyperChem includes no particular facilities for setting up these Boltzmann distributions of initial conditions but does allow the exploration of any one particular initial condi-

tion. A script or third-party software package interfaced to HyperChem via DDE could, in principle, do a full trajectory analysis by looping over initial conditions and keeping score.

## Setting Initial Coordinates and Velocities

To study a particular classical trajectory you must set the initial coordinates and velocities. In the above case of  $F + H_2$  you might position an  $H_2$  molecule at the center of the screen, oriented any way you chose, and then place the F atom in the left part of the screen approximately on line with the  $H_2$  molecule. This positioning might be done manually, or by setting the coordinates to specified values, using interactive translation of selected atoms or by actually editing the HIN file. The initial velocities of the two H atoms would probably best be set to values sampled from an isolated  $H_2$  molecule trajectory at the temperature  $T$  of the experiment. The velocity of the F atom might then be set to have only a large positive  $x$  value, say a few hundred Ångstroms/picosecond so as to come in to the center of the screen from the left and collide with the  $H_2$  molecule causing a reaction or possibly bouncing off without reaction. A variety of these initial conditions correspond to the experimental situation.

Coordinates of atoms can be set by normal translation or rotation of HyperChem molecules. To set initial velocities, however, it is necessary to edit the HIN file explicitly. The unit of velocity in the HIN file is Ångstroms/picosecond. A react.hin file and a script react.scr are included with HyperChem to illustrate one simple reacting trajectory. In order to have these initial velocities used in a trajectory the Restart check box of the Molecular Dynamics Options dialog box must be checked. If it is not, the initial velocities in the HIN file will be ignored and a re-equilibration to the temperature  $T$  of the Molecular Dynamics Options dialog box will occur. This destroys any imposed initial conditions on the molecular dynamics trajectory.

## Temperature Considerations

A trajectory analysis of a chemical reaction is less ambiguous if free dynamics is used and constant temperature algorithm is turned off. If Restart is correctly checked, then the temperature setting in the Molecular Dynamics Options dialog box becomes irrelevant. It is used only to determine an initial set of velocities when free dynamics is used. If the velocities come from the HIN file instead

(Restart) then the temperature is related only to the average kinetic energy from the velocities in the HIN file. In a proper trajectory analysis the initial velocities are sampled according to the temperature of the experimental situation being simulated.

If the constant temperature algorithm is used in a trajectory analysis, then the initial conditions are constantly being modified according to the simulation of the constant temperature bath and the relaxation of the molecular system to that bath temperature. The effect of such a bath on a trajectory analysis is less studied than for the simulation of equilibrium behavior.

## RHF/UHF Considerations

As discussed earlier, the RHF scheme may not be appropriate to the situation where open shells are present or bonds are breaking. In this case, a UHF scheme may be the only sensible computational approach. This becomes particularly relevant here in conjunction with chemical reactions where we need to correctly describe molecular dissociation. This can be easily illustrated with the  $H_2$  molecule. At equilibrium distances the UHF and RHF methods give the same answer; the UHF different-orbitals-for-different-spins solution degenerates into the RHF solution. However, at long bond lengths a branch point occurs and the UHF becomes a lower energy solution. As dissociation occurs, the UHF solution converges to a correct description of two isolated H atoms whereas the RHF solution goes to the wrong limit, indicative that a closed shell is an improper description of isolated hydrogen atoms. Thus, if the whole potential surface is to be described correctly, a UHF solution must be used. HyperChem cannot detect that the UHF solution is now lower in energy than the RHF solution and switch to the UHF solution; the UHF solution must be used throughout. HyperChem kicks the UHF solution off the metastable RHF initial guess and is always able to obtain the UHF solution (when it is lower than the RHF one) although, in principle, the RHF solution is also a solution to the UHF equations.

## UV Visible Spectroscopy

In HyperChem, you can now compute the energy difference between the ground electronic state and the first few excited electronic states of a molecular system by using the *ab initio* method

or any of the semi-empirical methods except for the Extended Hückel. To generate a UV-vis spectrum, you must perform a singly excited CI method with the *ab initio* method or semi-empirical method you choose.

Use the Electronic Spectrum dialog box to display and analyze the UV-vis spectrum produced by a singly excited CI calculation. This dialog box is available only after you do a single point CI semi-empirical calculation. Electronic Spectrum is then activated on the Compute menu.

By definition, the transition frequency,  $\nu$ , of the UV spectrum is given by

$$\nu = \frac{E_f - E_i}{h} \quad (200)$$

and the oscillator strength is given by

$$f_{i \rightarrow f} = \left( \frac{8\pi^2 m}{3e^2 h^2} \right) (E_f - E_i) |\mathbf{d}_{fi}|^2 \quad (201)$$

for the transition  $i \rightarrow f$ , where  $i$  is the initial state and  $f$  is the final state and  $\mathbf{d}_{fi}$  is the transition dipole moment, and  $h$  is the Planck constant. Here only the dipole moment transition is considered and all the higher orders of transitions are ignored because of the small probability of the higher order transitions. The oscillator strength is a dimensionless value.

The energies,  $E_i$  and  $E_f$  of the initial and final states of transitions in equations (178) and (179) are determined by the CI eigenvalues and the transition dipole moment  $\mathbf{d}_{fi}$  is obtained by using the CI eigenvectors, that is,

$$\mathbf{d}_{fi} = \langle \Phi_f | \sum_k \mathbf{r}_k | \Phi_i \rangle \quad (202)$$

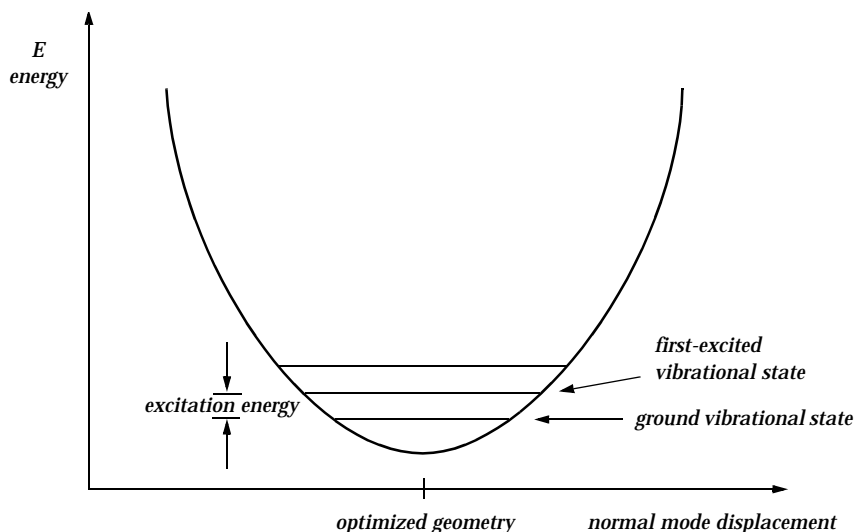
where  $k$  is the sum over all the electrons.

# Vibrational Analysis and IR Spectroscopy

## Vibrational Calculation

You can now use HyperChem to calculate vibration using *ab initio* methods and any of the semi-empirical methods except for Extended Hückel.

The following illustration shows the potential energy surface for vibrational motion along one normal mode:



HyperChem computes the Hessian using numerical second derivative of the total energy with respect to the nuclear positions based on the analytically calculated first derivatives in *ab initio* methods and any of the semi-empirical methods, except the Extended Hückel. Vibration calculations in HyperChem using an *ab initio* method may take much longer than calculations using the semi-empirical methods.

HyperChem performs a vibrational analysis at the molecular geometry shown in the HyperChem workspace, without any automatic pre-optimization. HyperChem may thus give unreasonable results when you perform vibrational analysis calculations with an unoptimized molecular system, particularly for one far from optimized. Because the molecular system is not at a stationary point, neither at a local minimum nor at a local maximum, the vibra-



tional analysis may give strange results including negative vibrational frequencies. Except in very specialized cases, you should always do a geometry optimization before a vibrational analysis.

## Normal Coordinate Analysis

HyperChem models the vibrations of a molecule as a set of  $N$  point masses (the nuclei of the atoms) with each vibrating about its equilibrium (optimized) position. The equilibrium positions are determined by solving the electronic Schrödinger equation.

Defining mass-weighted Cartesian displacement coordinates  $q_i$ ,

$$q_i = m_i^{1/2}(x_i - x_i^e) \quad (203)$$

then the classical kinetic energy,  $T$ , of vibration about the equilibrium positions is

$$T = \frac{1}{2} \sum_{i=1}^{3N} \left( \frac{dq_i}{dt} \right)^2 \quad (204)$$

The potential energy of vibration is a function of the coordinates,  $x_1, \dots, x_N$ ; hence it is a function of the mass-weighted coordinates,  $q_1, \dots, q_{3N}$ . For a molecule, the vibrational potential energy,  $U$ , is given by the sum of the electronic energy and the nuclear repulsion energy:

$$U = U(q_1, \dots, q_{3N}) \quad (205)$$

The vibrational potential may be expanded in a Taylor series about the equilibrium positions of the atoms.

$$U = U_e + \sum_{i=1}^{3N} \left( \frac{\partial U}{\partial q_i} \right)_e q_i + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} \left( \frac{\partial^2 U}{\partial q_i \partial q_j} \right)_e q_i q_j + \dots \quad (206)$$

At the equilibrium geometry of the molecule,  $U$  is a minimum and the gradient vanishes

$$\left( \frac{\partial U}{\partial q_i} \right)_e = 0 \quad (207)$$

If the vibrations are small (e.g., low temperatures, rigid bonding framework), the higher order terms can be neglected. Thus the potential energy simplifies to

$$U = U_e + \sum_{i=1}^{3N} u_{ij} q_i q_j \quad (208)$$

where  $u_{ij}$  is given by

$$u_{ij} = \left( \frac{\partial^2 U}{\partial q_i \partial q_j} \right)_e \quad (209)$$

The classical-mechanical problem for the vibrational motion may now be solved using Newton's second law. The force on the  $x$  component of the  $i^{\text{th}}$  atom is

$$F_{i,x} = m \frac{d^2 x_i}{dt^2} = - \frac{\partial U}{\partial x_i} \quad (210)$$

Transforming to mass-weighted coordinates, equation (210) can be rewritten into a set of  $3N$  simultaneous linear differential equations

$$\frac{d^2 q_i}{dt^2} + \frac{\partial U}{\partial q_i} = 0 \quad (211)$$

Each differential equation contains all the coordinates  $q_i$  since  $\frac{\partial U}{\partial q_i}$  is expressed as a single summation over the  $q_j$ 's.

$$\frac{\partial U}{\partial q_i} = \sum_{j=1}^{3N} u_{ij} q_j \quad (212)$$

Substituting equation (212) into equation (211), it becomes

$$\frac{d^2 q_i}{dt^2} + \sum_{j=1}^{3N} u_{ij} q_j = 0 \quad (213)$$

for  $i = 1, 2, \dots, 3N$ .

Equations (213) are a system of  $3N$  simultaneous linear differential equations in the  $3N$  unknowns  $q_j$ . It can be transformed to a

matrix form, if the mass-weighted coordinates,  $q_i$ , are rewritten into

$$q_i = A_i \sin(\lambda^{1/2} t + b) \quad (214)$$

for  $i = 1, 2, \dots, 3N$ , where  $\lambda$  and  $b$  are constants. Then the matrix form of equation (213) is

$$\mathbf{U}\mathbf{L} = \mathbf{L}\mathbf{\Lambda} \quad (215)$$

where matrix  $\mathbf{U}$  is the mass-weighted Hessian with matrix element  $u_{ij}$  defined in equation (209),  $\mathbf{L}$  and  $\mathbf{\Lambda}$  are eigenvectors and eigenvalues, respectively.

The eigenvalues and eigenvectors of the mass-weighted force matrix can be obtained by diagonalizing equation (215). Then each eigenvalue corresponds to its normal coordinates,  $Q_k$

$$Q_k = \sum_{i=1}^{3N} l_{ki} q_i \quad (216)$$

Because  $\mathbf{U}$  is a unitary transformation matrix, equation (216) can be written in the form

$$q_i = \sum_k l_{ki} Q_k \quad (217)$$

Using normal coordinates  $Q_k$ , equation (213) can be rewritten to

$$\frac{d^2 Q_k}{dt^2} + \lambda_k Q_k = 0 \quad (218)$$

with  $k = 1, 2, \dots, 3N$ . The general solutions of these equations are

$$Q_k = B_k \sin(\lambda_k^{1/2} t + \beta_k) \quad (219)$$

with  $k = 1, 2, \dots, 3N$ .

Motion along each normal coordinate is described by each atom vibrating in phase with one another with the same frequency. The vibration frequency,  $\nu$ , is related to the eigenvalues,  $\lambda$ , by

$$\nu = \frac{\lambda_i^{1/2}}{2\pi} \quad (220)$$

Notice that although the energies and forces are evaluated quantum mechanically in HyperChem, the vibrational analysis has been purely classical.

## Infrared Absorption

The mechanism of infrared (IR) light absorption is conceptually similar to that of UV visible light absorption. In both cases, the oscillating electric dipole of the light beam induces a mirror image oscillating electric dipole in the molecule (transition dipole). Energy (photons) may be exchanged between the molecule and the light beam if the frequency of the light closely corresponds to an energy gap between levels of the molecule.

The connection between transition energy  $\Delta E$  and frequency  $\nu$  is given by Einstein's classic formula

$$\Delta E = h\nu \quad (221)$$

where  $h$  is the Planck constant. UV visible frequencies ( $\sim 10^{15}$  cycles per second) correspond to gaps between different electronic energy levels, whereas IR frequencies ( $\sim 10^{12}$  cycles per second) correspond to gaps between vibrational energy levels. Thus, each line in a UV-visible spectrum corresponds to an excitation of electrons from one electronic state to another. Analogously, each line in an IR spectrum represents an excitation of nuclei from one vibrational state to another.

The frequency of electronic motion in a molecule (classically speaking, the number of orbits per second) is similar to that of UV-visible light, and the frequency of vibrational motion is similar to that of IR light. When IR light passes through a molecule, the electrons see an essentially constant electric field rather than an oscillating dipole. Similarly, when UV-visible light passes through a molecule, the nuclei see an essentially constant electric field rather than an oscillating dipole. In both cases, no mechanism exists for the absorption of light due to the frequency mismatch between the light and the molecule. In other words, IR light cannot polarize electrons and UV-visible light cannot polarize nuclei.

When IR radiation passes through a molecule, the nuclei move so as to create a molecular dipole that is synchronized with that of

the light. To a first approximation, each normal mode of vibration interacts independently with IR light.

A normal mode can absorb IR light if the molecular dipole moment changes during the course of a normal vibration. For example, the symmetric stretch in CO<sub>2</sub> does not entail a changing dipole moment, whereas the asymmetric stretch and bend do induce a changing dipole moment. CO<sub>2</sub> absorbs IR light at frequencies corresponding to the asymmetric stretch and bend because the IR light is “shaking” the atoms of the molecule along the asymmetric stretch and bending coordinates.

Quantitatively, we may express how “shakeable” a given normal mode will be under the influence of IR light, by examining the magnitude of its induced dipoles.

$$\mu = \mu_0 + (d\mu/dq)_0 q + (d^2\mu/dq^2)_0 q^2 + \dots \quad (222)$$

Keeping only the linear term, the transition dipole moment is given by

$$\langle \mu_{fi} \rangle = \int \Psi_f^{\text{vib}} (d\mu/dq) q \Psi_i^{\text{vib}} dq \quad (223)$$

Since we assume that all higher derivatives are essentially zero,  $d\mu/dq = \text{constant}$ . Therefore, it may be taken out of the integral sign as follows:

$$\langle \mu_{fi} \rangle = (d\mu/dq) \int \Psi_f^{\text{vib}} q \Psi_i^{\text{vib}} dq \quad (224)$$

Qualitatively, the selection rule for IR absorption for a given mode is that the symmetry of  $q \Psi_i^{\text{vib}}$  must be the same as  $q \Psi_f^{\text{vib}}$ . Quantitatively, the transition dipole moment is proportional to the dipole derivative with respect to a given normal mode  $d\mu/dq$ .

Generally, IR oscillator strengths (absorption intensities) are on the order of  $10^{-4}$  of those of UV visible lines.

The integrated infrared band intensity for the  $k^{\text{th}}$  fundamental is defined [W. B. Person and K. C. Kim, *J. Chem. Phys.* 69, 1764 (1978)] as

$$A_k = \frac{1}{CL} \int_{\text{band}} \left( \ln \frac{I_0}{I} \right) dv \quad (225)$$

where  $C$  is concentration (in moles liter<sup>-1</sup>),  $L$  is optical path length (in cm),  $\nu$  is wavenumber in cm<sup>-1</sup>, and  $I_0$  and  $I$ , respectively, are the intensities of incident and transmitted light. Assuming electric and mechanical harmonicity,  $A_k$  may be approximated by

$$A_k = \frac{\pi N_A g_k}{3000c^2} \left| \frac{\partial \mu}{\partial Q_k} \right|^2 \quad (226)$$

where  $N_A$  is the Avogadro number (mol<sup>-1</sup>),  $c$  is the light velocity (cm s<sup>-1</sup>),  $g_k$  is the degeneracy factor, and  $\partial \mu / \partial Q_k$  is the dipole-moment derivative with respect to the  $k^{\text{th}}$  normal coordinate  $Q_k$ .

In equation (225), a natural logarithm is used. Another definition [B. A. Hess, Jr., L. J. Schaad, P. Carsky, and R. Zahradnik, *Chem. Rev.* 86, 709 (1986)] uses a common logarithm in equation (225), that is,

$$A_k = \frac{1}{CL} \int_{\text{band}} \left( \log \frac{I_0}{I} \right) d\nu \quad (227)$$

So, the integrated infrared band intensity for the  $k^{\text{th}}$  fundamental is defined as

$$A_k = \frac{\pi N_A g_k}{3000 \ln(10) c^2} \left| \frac{\partial \mu}{\partial Q_k} \right|^2 \quad (228)$$

In HyperChem, equation (226) is used for calculating the integrated infrared band intensities for the *ab initio* method and equation (228) is employed for all the semi-empirical methods. All IR lines correspond to transitions from the ground vibrational state to an excited vibrational state that has one additional quantum deposited in a given vibrational mode.



## Acknowledgments

Hypercube would like to thank the following authors for contributing significant portions of this text:

Dr. Allison Howard, University of California at San Francisco.

Dr. James McIver, State University of New York at Buffalo.

Dr. Jack Collins, Molecular Research Institute, Palo Alto, CA.





# **HyperChem<sup>®</sup> Release 7.0**

## **for Windows**

**Reference Manual Volume 1**

# Copyright © 2002 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperTNDO, HyperDFT, HyperGauss, HyperNMR, HyperChem Data and ChemPlus are trademarks of Hypercube, Inc.

## Third Party Trademarks

MM2 is a trademark of Dr. Norman Allinger, University of Georgia.

CHARMm is a registered trademark of Accelrys Inc.

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows, Windows 95, Windows 98, Windows ME, Windows NT, and Windows XP are trademarks of Microsoft Corporation.

IBM is a registered trademark of International Business Machines, Inc.

ChemDraw is a trademark of CambridgeSoft Corporation.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

## Volume 1

<b>Chapter 1 Introduction</b>	<b>1</b>
What is HyperChem? . . . . .	1
Calculation Methods . . . . .	3
Starting HyperChem . . . . .	3
HyperChem for Windows. . . . .	4
Using Multiple Windows . . . . .	6
Using the Mouse . . . . .	7
Using Keyboard Alternatives for Windows . . . . .	9
Using Keyboard Alternatives for HyperChem . . . . .	10
HyperChem Menus. . . . .	12
Gray Menu Items . . . . .	13
HyperChem Dialog Boxes. . . . .	13
Persistent Dialog Boxes. . . . .	14
Working in Dialog Boxes . . . . .	14
Editing Text . . . . .	14
Editing Keys . . . . .	15
Case Preferences . . . . .	15
Entering Exponents . . . . .	15
<b>Chapter 2 The Basics of HyperChem</b>	<b>17</b>
Levels of Molecular Organization . . . . .	17
Coordinate Systems. . . . .	18
Viewer's Coordinate System . . . . .	19
Molecular Coordinate System. . . . .	20

Inertial Axes . . . . .	21
Constructing a Molecule . . . . .	22

## **Chapter 3 Tools 27**

Pop-up Help for Tool Icons . . . . .	27
Drawing Tool . . . . .	28
Drawing Molecules . . . . .	29
Choosing and Changing Elements . . . . .	29
Adding Hydrogen Atoms . . . . .	30
Selection Tool . . . . .	30
Selection Levels . . . . .	32
Selecting . . . . .	32
Using the Selection Tool . . . . .	32
Using the Select Menu . . . . .	34
Measuring with the Selection Tool . . . . .	35
Selecting Annotations . . . . .	36
XY Translating and Zooming an Annotation . . . . .	37
XY Rotation Tool . . . . .	37
Using the XY Rotation Tool . . . . .	37
Setting the Amount of XY Rotation . . . . .	38
Z Rotation Tool . . . . .	39
Using the Z Rotation Tool . . . . .	39
Setting the Amount of Z Rotation . . . . .	40
XY Translation Tool . . . . .	40
Using the XY Translation Tool . . . . .	41
Fast Translation . . . . .	42
Translation of Annotations . . . . .	42
Z Translation Tool . . . . .	43
Using the Z Translation Tool . . . . .	43
Setting the Amount of Z Translation . . . . .	45
Zoom Tool . . . . .	45
Using the Zoom Tool . . . . .	45
Setting the Zoom Sensitivity . . . . .	46
Zooming Annotations . . . . .	46
Z Clipping Tool . . . . .	47
Using the Z Clipping Tool . . . . .	48
Setting the Amount of Clipping Plane Movement . . . . .	49
Keyboard Equivalents for Tools . . . . .	49
A . . . . .	50
Tools . . . . .	50
Annotations are Text . . . . .	51
Symbols are Text in a Symbol Font . . . . .	51

The Annotation Layer . . . . .	53
Transparency in Printing . . . . .	53
The Annotation Interface . . . . .	54
Creating a Simple Text Annotation . . . . .	54
Creating a Symbol . . . . .	54
Selecting Annotations . . . . .	56
Moving Annotations . . . . .	57
Editing Annotations. . . . .	57
Multiline Text. . . . .	58
Ordering within Annotation Layer . . . . .	59
Ordering the Annotation Layer . . . . .	59
Hiding the Annotation Layer . . . . .	60

## **Chapter 4 Annotations 61**

Definition. . . . .	61
Types of Annotations . . . . .	61
Text . . . . .	62
Lines . . . . .	62
Ellipses . . . . .	62
Rectangles . . . . .	63
An Example . . . . .	63
The Annotation Layer . . . . .	64
Transparency in Printing . . . . .	65
The Annotation Interface . . . . .	65
Text Annotations . . . . .	66
Annotations as Text. . . . .	66
Symbols are Text in a Symbol Font . . . . .	67
Creating a Simple Text Annotation . . . . .	68
Creating a Symbol . . . . .	68
Selecting Text Annotations . . . . .	70
Moving Text Annotations . . . . .	71
Editing Text Annotations . . . . .	71
Multiline Text. . . . .	72
Line, Ellipse, and Rectangle Annotations . . . . .	73
Options. . . . .	74
Ordering Annotations within the Annotation Layer . . . . .	75
Ordering the Annotation Layer . . . . .	75
Hiding the Annotation Layer. . . . .	76

## Chapter 5 Managing Files 77

File Menu . . . . .	77
New . . . . .	78
Open . . . . .	78
Open File Dialog Box . . . . .	79
Reading Non-standard PDB Files . . . . .	81
Reading ChemDraw Files . . . . .	81
Using the Z-Matrix File Filter . . . . .	81
Merge . . . . .	82
Save . . . . .	82
Save As . . . . .	83
Save File Dialog Box . . . . .	83
Editing Comments . . . . .	85
Save As HTML... . . . .	86
Content of HTML Page . . . . .	86
Option Details . . . . .	88
Sample HTML Page . . . . .	89
Start Log . . . . .	90
Open Log Dialog Box . . . . .	90
Stop Log . . . . .	91
Log Comments . . . . .	91
Import . . . . .	92
Export . . . . .	93
Print . . . . .	94
General tips for printing or copying images . . . . .	94
Preferences . . . . .	95
Preferences Property Sheet . . . . .	95
Window Color . . . . .	95
Bond Color . . . . .	96
Selection Color . . . . .	96
Tool. . . . .	97
Key . . . . .	98
Setup Image . . . . .	99
Isosurface Colors . . . . .	102
Paths . . . . .	103
Network License . . . . .	104
Using Black-and-white Output . . . . .	105
Script Menu . . . . .	105
Open Script . . . . .	106
Script One . . . . . Script Ten . . . . .	106
Script Editor. . . . .	107
Overview . . . . .	107
Tutorials . . . . .	108

1. Entering Commands In The Script Editor . . . . .	108
2. Using The Paste Message Dialog Box . . . . .	109
3. Requesting Information From HyperChem . . . . .	109
4. Working With The Message List . . . . .	110
Working With Files . . . . .	111
Reference . . . . .	112
Main Window . . . . .	112
File Menu . . . . .	112
Edit Menu . . . . .	113
Options Menu . . . . .	113
Cancel Menu . . . . .	113
Help Menu . . . . .	113
Paste Message Dialog Box . . . . .	114
Values Dialog Box . . . . .	114
Help Menu . . . . .	115
About HyperChem . . . . .	116

## **Chapter 6 Building Molecules 117**

Build Menu . . . . .	117
Explicit Hydrogens . . . . .	119
Default Element . . . . .	120
Element Table Dialog Box . . . . .	121
Element Properties . . . . .	123
Add Hydrogens . . . . .	123
Model Build/Add H & Model Build . . . . .	124
Constrain Length and Angles . . . . .	125
Allow Arbitrary Valence . . . . .	126
Set Formal Charge . . . . .	126
Set Formal Charge Dialog Box . . . . .	127
United Atoms . . . . .	128
All Atoms . . . . .	129
Calculate Types . . . . .	129
Compile Type Rules . . . . .	130
Set Atom Type . . . . .	130
Set Atom Type Dialog Box . . . . .	131
Set Mass . . . . .	132
Set Atom Mass Dialog Box . . . . .	132
Set Charge . . . . .	132
Set Atom Charge Dialog Box . . . . .	133
Set Custom Label . . . . .	133
Set Custom Label Dialog Box . . . . .	133
Constrain Geometry . . . . .	134



Constrain Geometry Dialog Box . . . . .	134
Constrain Bond Length . . . . .	136
Constrain Bond Length Dialog Box . . . . .	137
Constrain Bond Angle . . . . .	137
Constrain Bond Angle Dialog Box . . . . .	138
Constrain Bond Torsion . . . . .	138
Constrain Bond Torsion Dialog Box . . . . .	139
Databases Menu . . . . .	139
Amino Acids . . . . .	141
Amino Acids Dialog Box . . . . .	142
Make Zwitterion . . . . .	145
Remove Ionic Ends . . . . .	146
Sequence Editor . . . . .	147
Overview . . . . .	147
Tutorial . . . . .	147
Background . . . . .	148
Reference . . . . .	149
File Menu . . . . .	149
Edit Menu . . . . .	150
View Menu . . . . .	151
HyperChem Menu . . . . .	153
Classifying Secondary Structure . . . . .	153
Cancel Menu . . . . .	154
Help Menu . . . . .	154
Dayhoff Comparison Dialog Box . . . . .	155
Sequence Length Dialog Box And XY Plot Dialog Box . . . . .	155
Files . . . . .	156
Hints . . . . .	156
Limitations . . . . .	156
Nucleic Acids . . . . .	157
Nucleic Acids Dialog Box . . . . .	159
Add Counter Ions . . . . .	162
Crystal Builder . . . . .	163
Overview . . . . .	163
Tutorials . . . . .	163
1. Creating A Crystal Of Sodium Metal . . . . .	163
2. Creating A Crystal Of Sodium Chloride . . . . .	165
Background . . . . .	166
Reference . . . . .	167
Main Window . . . . .	167
File Menu . . . . .	168
View Menu . . . . .	168

	HyperChem Menu . . . . .	169
	Help Menu . . . . .	170
	Atom Data Dialog Box . . . . .	170
	Samples Dialog Box . . . . .	171
	View Dialog Box . . . . .	173
	Choose Structure Dialog Box . . . . .	173
	About Dialog Box . . . . .	173
	Toolbar . . . . .	173
	Hints . . . . .	175
Sugar Builder . . . . .		175
	Overview . . . . .	175
	Tutorial . . . . .	176
	Background . . . . .	177
	Reference . . . . .	178
	Main Window . . . . .	178
	File Menu . . . . .	178
	Add Menu . . . . .	179
	Help Menu . . . . .	180
	Aldoses Dialog Box . . . . .	181
	Ketoses Dialog Box . . . . .	182
	Derivatives Dialog Box . . . . .	182
	Blocking Groups Dialog Box . . . . .	182
	User Dialog Box . . . . .	182
	Link Dialog Box . . . . .	185
	Link Chains: Orientation Dialog Box . . . . .	186
	About Dialog Box . . . . .	186
	Limitations . . . . .	186
Polymer Builder . . . . .		187
	Monomer Specification . . . . .	187
	Head and Tail . . . . .	187
	Polymerization by Elimination . . . . .	187
	Specific Torsion Angles . . . . .	187
	Example. . . . .	188
	Limitations. . . . .	190
Mutate . . . . .		190
	Mutate Dialog Box . . . . .	191
	Choices of Nucleotide Residues . . . . .	192
Replace . . . . .		192
	Protein Design. . . . .	192
	Cutting (Deleting) a Residue Sequence . . . . .	193
	Replacing a Residue Sequence . . . . .	194
	Creating a Source Sequence . . . . .	194
	Replacing Target Residues . . . . .	197

Relative Length of Residues . . . . .	198
Relative Orientation of Residues. . . . .	199
Executing the Replace . . . . .	199
Inserting A Residue Sequence . . . . .	199
Invoke Database . . . . .	199

## Chapter 7 Edit, Select, and Display Menus

**201**

Edit Menu . . . . .	201
Clear . . . . .	203
Cut . . . . .	203
Copy . . . . .	203
Copy ISIS Sketch . . . . .	204
Paste . . . . .	204
Pasting Superatoms From ISIS/Draw . . . . .	205
Copy Image . . . . .	206
Invert. . . . .	206
Reflect . . . . .	208
Rotate . . . . .	209
Rotate Dialog Box . . . . .	209
Translate. . . . .	210
Translate Dialog Box . . . . .	211
Zoom . . . . .	212
Zoom Dialog Box . . . . .	212
Z Clip. . . . .	213
Z Clip Dialog Box . . . . .	214
Align Viewer . . . . .	215
Align Viewer Dialog Box . . . . .	215
Align Molecules . . . . .	216
Align Molecules Dialog Box . . . . .	217
Set Bond Length . . . . .	218
Set Bond Length Dialog Box. . . . .	219
Set Bond Angle. . . . .	219
Set Bond Angle Dialog Box . . . . .	220
Set Bond Torsion . . . . .	220
Set Bond Torsion Dialog Box . . . . .	220
Select Menu . . . . .	220
Visible Atoms . . . . .	222
Atoms . . . . .	223
Residues . . . . .	225
Molecules . . . . .	226
Multiple Selections: Accumulating Selections . . . . .	227

Select Sphere and Group Selecting . . . . .	227
Selecting in Two Dimensions . . . . .	228
Selecting in Three Dimensions . . . . .	230
Select All . . . . .	231
Complement Selection . . . . .	232
Select . . . . .	232
Select Dialog Box . . . . .	233
Selecting by String . . . . .	236
Selecting by Number . . . . .	236
Numbering . . . . .	237
Name Selection . . . . .	237
Name Selection Dialog Box. . . . .	238
Naming Selections . . . . .	240
Using Named Selections. . . . .	240
Using Named Selections for Restraints . . . . .	241
Using Named Selections in MD Calculations . . . . .	241
Name Molecule . . . . .	242
Name Molecule Dialog Box. . . . .	242
Extend Ring . . . . .	242
Extend Side Chain . . . . .	243
Extend to sp3 . . . . .	244
Select Backbone . . . . .	245
Display Menu . . . . .	248
Scale to Fit . . . . .	250
Overlay . . . . .	251
RMS Fit . . . . .	252
Overview . . . . .	252
Tutorials. . . . .	252
1. Overlaying Two Structures Of The Same Molecule . . . . .	252
2. Overlaying Selected Atoms Only . . . . .	253
Reference . . . . .	254
RMS Fit Options Dialog Box . . . . .	254
Overlay By Molecule Numbering . . . . .	255
Overlay By Selection Order . . . . .	255
Suppressing The Options Dialog Box . . . . .	256
Hints. . . . .	256
Show All . . . . .	258
Show Selection Only . . . . .	258
Hide Selection . . . . .	258
Rendering . . . . .	258
Rendering: Rendering Method. . . . .	258
Rendering: Sticks . . . . .	260
Rendering: Cylinders. . . . .	262

Rendering: Balls . . . . .	263
Rendering: Overlapping Spheres . . . . .	264
Using Stereo . . . . .	268
Walleyed Stereo . . . . .	268
Cross-eyed Stereo . . . . .	269
Last Rendering . . . . .	269
Raytracing . . . . .	269
Overview . . . . .	269
Obtaining a Raytraced Image . . . . .	270
Default Lighting Scheme . . . . .	270
Types of Images . . . . .	270
Custom Lighting . . . . .	271
Using POV-Ray . . . . .	273
Show Isosurface . . . . .	274
Isosurface . . . . .	274
Show Hydrogens . . . . .	274
Show Periodic Box. . . . .	275
Show Multiple Bonds. . . . .	276
Show Hydrogen Bonds . . . . .	277
Recompute Hydrogen Bonds . . . . .	278
Show Inertial Axes. . . . .	279
Show Dipole Moment . . . . .	280
Labels. . . . .	281
Labels Dialog Box . . . . .	281
Color . . . . .	282
Selection Color Dialog Box . . . . .	283
Element Color . . . . .	284
Element Color Dialog Box . . . . .	285

## **Chapter 8 Chemical Calculations 287**

Setup Menu . . . . .	287
Molecular Mechanics. . . . .	289
Molecular Mechanics Force Field Dialog Box . . . . .	290
MM+ Options Dialog Box . . . . .	291
Force Field Options Dialog Box . . . . .	293
Force Field Components Included Dialog Box . . . . .	295
Semi-empirical . . . . .	296
Semi-empirical Method Dialog Box . . . . .	298
Available Elements. . . . .	300
Extended Hückel . . . . .	301
CNDO, INDO . . . . .	301
MINDO3 . . . . .	301

MNDO . . . . .	302
AM1 . . . . .	302
PM3 . . . . .	302
ZINDO/1. . . . .	303
ZINDO/S. . . . .	303
Semi-empirical Options (Hückel) Dialog Box . . . . .	304
Semi-empirical Options Dialog Box . . . . .	305
Configuration Interaction Dialog Box . . . . .	308
Practical Applications. . . . .	310
Mixed Mode Calculations . . . . .	311
Ab Initio and Density Functional . . . . .	311
Choosing a Basis Set . . . . .	312
DFT Specific Basis Sets . . . . .	312
Generic Basis Sets . . . . .	312
Pople-like Basis Sets . . . . .	313
Other Basis Sets . . . . .	314
Ab Initio and Density Functional Method Dialog Boxes . . . . .	315
Assign Other Basis Set Dialog Box . . . . .	318
Ab Initio and DFT Options Dialog Box . . . . .	318
Ab Initio and DFT Advanced Options Dialog Box. . . . .	322
Initial Guess of MO Coefficients . . . . .	325
Number of d Orbitals. . . . .	326
Ghost-Atoms . . . . .	326
Configuration Interaction . . . . .	327
Configuration Interaction Dialog Box . . . . .	327
Extra Basis Function Dialog Box . . . . .	328
DFT Integration Grids Dialog Box. . . . .	329
Exchange Correlation Potential Dialog Box . . . . .	331
Mixed Mode Calculations . . . . .	334
Periodic Box . . . . .	335
Periodic Boundary Conditions. . . . .	336
Periodic Box Options Dialog Box . . . . .	337
Changing periodic box size. . . . .	338
Removing the periodic box. . . . .	339
Solvents other than water . . . . .	339
Restraints . . . . .	340
Restraint Forces Dialog Box. . . . .	340
Setting Up Restraints. . . . .	342
Applying Restraints . . . . .	343
“Dummy atoms” and Conformational Restraint . . . . .	343
Set Velocity . . . . .	344
Get and Set Velocity Dialog Box . . . . .	344
Applying an Electric or Magnetic Field . . . . .	345

Network . . . . .	346
Network Options Dialog Box . . . . .	347
Edit Parameters. . . . .	347
Atom Parameters . . . . .	348
Bond Parameters . . . . .	349
Angle Parameters . . . . .	350
Torsion Parameters. . . . .	350
Select Parameter Set . . . . .	351
<i>Force field</i> Parameters Dialog Box . . . . .	352
Compile Parameter File . . . . .	353
Reaction Map . . . . .	353
Compute Menu . . . . .	357
Warning Messages . . . . .	359
Single Point . . . . .	360
Using a Molecular Mechanics Method . . . . .	361
Using a Semi-empirical or Ab Initio or DFT Method . . . . .	361
Geometry Optimization . . . . .	362
Molecular Mechanics/Semi-empirical/Ab Initio/DFT Optimization Dialog Box . . . . .	364
Molecular Dynamics . . . . .	368
Changing the Initial Velocity for Calculations . . . . .	369
Molecular Dynamics Options Dialog Box . . . . .	369
Molecular Dynamics Snapshots Dialog Box . . . . .	375
Molecular Dynamics Playback Dialog Box . . . . .	375
Molecular Dynamics Averages Dialog Box . . . . .	377
Molecular Dynamics Results Dialog Box . . . . .	380
Time Periods in a Molecular Dynamics Simulation . . . . .	382
Running a Script During Molecular Dynamics . . . . .	383
Langevin Dynamics . . . . .	383
Monte Carlo . . . . .	384
Vibrations . . . . .	385
Transition State . . . . .	386
Transition State Search Dialog Box. . . . .	388
Properties Display . . . . .	389
Atom Properties. . . . .	389
Bond Properties. . . . .	390
Molecular System Properties. . . . .	390
Plot Molecular Graphs . . . . .	391
Plot Molecular Properties Property Sheet . . . . .	392
Plot Molecular Properties . . . . .	392
Isosurface Grid Dialog Box . . . . .	393
Contour Grid. . . . .	395
Isosurface Rendering . . . . .	396

Transparency Level . . . . .	398
Mapped Function Options . . . . .	399
Plotting Results . . . . .	399
Orbitals . . . . .	401
Orbitals Dialog Box . . . . .	401
Describing Orbitals . . . . .	404
Practical Guidelines for Orbitals . . . . .	405
Vibrational Spectrum . . . . .	406
Vibrational Spectrum Dialog Box . . . . .	408
Practical Guidelines . . . . .	409
Comparing Computed Results with Experiment . . . . .	410
Electronic Spectrum . . . . .	411
Electronic Spectrum Dialog Box . . . . .	412
Practical Guidelines . . . . .	414
Saving Information in a Log File . . . . .	415
Potential Energy Plotting . . . . .	416
One Dimensional Potentials . . . . .	416
Choosing the Independent Variable . . . . .	416
Current Selection . . . . .	417
Named Selection PLOT1 . . . . .	417
Requesting a Plot . . . . .	417
Plotting Options . . . . .	419
Help . . . . .	420
Types of Fitting. . . . .	420
Data . . . . .	421
Exporting and Printing Plots . . . . .	421
Titles, Labels, Etc. . . . .	422
Two-Dimensional Potentials . . . . .	423
Example. . . . .	423
Cautions . . . . .	425
Conformational Search. . . . .	425
Overview . . . . .	425
Tutorial . . . . .	426
Background . . . . .	428
1. Selection Of An Initial Structure . . . . .	429
2. Variation Of Geometric Parameters . . . . .	429
3. Geometry Optimization . . . . .	431
4. Comparison With Previously Accepted Conformations And Energy Criterion . . . . .	431
Options Supported In Conformational Search . . . . .	432
Reference . . . . .	433
Main Window . . . . .	433
File Menu . . . . .	436



Edit Menu . . . . .	437
Options Menu . . . . .	437
Run Menu . . . . .	438
Cancel Menu . . . . .	439
Help Menu . . . . .	439
Auto Save Option Dialog Box . . . . .	439
Choose Torsions Dialog Box . . . . .	439
Define Rings Dialog Box . . . . .	440
Setup Ranges Dialog Box . . . . .	441
Search Method Dialog Box . . . . .	442
Test Options For Search Dialog Box . . . . .	443
Atoms To RMS Compare Dialog Box . . . . .	446
Optimization Options Dialog Box . . . . .	447
Random Number Seed Dialog Box . . . . .	448
Limits For Search Dialog Box . . . . .	449
Files . . . . .	449
HCCONF.INI File . . . . .	449
HCS Files . . . . .	450
Limitations . . . . .	450
QSAR Properties . . . . .	450
Overview . . . . .	450
Tutorials . . . . .	451
1. Calculating Partial Charges . . . . .	451
2. Calculating Log P . . . . .	452
3. Calculating Surface Areas . . . . .	452
Background . . . . .	453
Partial Charges . . . . .	453
Limitations . . . . .	453
Surface Area (Approx.) . . . . .	455
Surface Area (Grid) . . . . .	455
Volume . . . . .	456
Hydration Energy . . . . .	456
Log P . . . . .	456
Refractivity . . . . .	457
Polarizability . . . . .	458
Reference . . . . .	458
Main Window . . . . .	458
File Menu . . . . .	460
Options Menu . . . . .	461
Cancel Menu . . . . .	461
Help Menu . . . . .	461
Calculation Options Dialog Box . . . . .	461
Output Options Dialog Box . . . . .	462

Results Window Dialog Box . . . . .	463
About QSAR Properties Dialog Box . . . . .	463
Batch Calculations Using DDE . . . . .	464
QSAR Properties As DDE Server . . . . .	464
Starting A DDE Conversation . . . . .	464
DDE Message Syntax . . . . .	465
Example . . . . .	466
Hints. . . . .	468
Drawing Heterocyclic Rings For Log P Calculations . . . . .	468
Cancel . . . . .	469

## Volume 2

### Chapter 9 Constructing Molecules 471

Drawing Molecules . . . . .	471
Adding Atoms and Bonds . . . . .	472
Removing Atoms. . . . .	473
Changing the Drawing Element . . . . .	473
While Drawing . . . . .	474
After Drawing . . . . .	475
Element Properties . . . . .	475
Changing the Bond Order . . . . .	475
Aromatic Rings . . . . .	477
Alternative Drawing Mode. . . . .	477
Adding Hydrogen Atoms . . . . .	478
Cutting and Pasting Atoms . . . . .	479
Cut . . . . .	479
Copy . . . . .	480
Paste . . . . .	480
Allowing Arbitrary Valence . . . . .	481
Changing Molecular Conformation . . . . .	481
Using Wedges to Display and Specify Stereobonds . . . . .	482
Wedges Explained . . . . .	484
For Atoms with Four Neighbors . . . . .	484
For Atoms with Three Neighbors . . . . .	485
Changing Chirality . . . . .	485
Converting 2D Drawings into 3D Structures . . . . .	487
Incremental Model Building . . . . .	487
Formal Charge . . . . .	488

Building Inorganic Structures . . . . .	489
Building Molecules from Residues . . . . .	492
The Databases Menu . . . . .	493
Proteins . . . . .	493
Constructing a Protein . . . . .	494
Cross-linking with Cyx . . . . .	495
Choosing Conformations . . . . .	495
Creating Zwitterions . . . . .	495
Mutating Amino Acids . . . . .	496
Adding Unusual Residues. . . . .	496
Nucleic Acids . . . . .	497
Constructing a Polynucleotide . . . . .	497
Choosing Conformations . . . . .	499
Adding Counter Ions . . . . .	499
Mutating Nucleic Acid Residues . . . . .	499
Creating Unusual Base Pairs . . . . .	500
Adding Unusual Residues. . . . .	501
Water Molecules . . . . .	501

## **Chapter 10 Protein Data Bank Files 503**

Obtaining PDB Files . . . . .	503
Reading PDB Files. . . . .	503
Creating PDB-type Files . . . . .	504
Sample PDB Files . . . . .	506

## **Chapter 11 Scripts and DDE 509**

Automatic Operation with Scripts . . . . .	509
Integrating HyperChem with DDE . . . . .	509
Using HyperChem Messages . . . . .	510
Types of Input Messages . . . . .	510
Message Syntax . . . . .	511
Variables . . . . .	511
Behavior of array variables . . . . .	512
Atom Indices . . . . .	512
Two-dimensional Arrays . . . . .	513
One-dimensional Arrays . . . . .	513
Commands . . . . .	516
Arguments . . . . .	516
Menu Activators . . . . .	517
HyperChem Messages . . . . .	518

File Menu . . . . .	519
Edit Menu . . . . .	526
Build Menu . . . . .	529
Select Menu . . . . .	533
Display Menu . . . . .	536
Databases Menu . . . . .	542
Setup Menu . . . . .	545
Compute Menu . . . . .	556
Script Menu . . . . .	568
Cancel . . . . .	573
Data Variables. . . . .	573
Help Menu. . . . .	578
Miscellaneous Commands. . . . .	579
Miscellaneous Variables . . . . .	580
Cross Reference of HyperChem Messages . . . . .	584
Message Cross-Reference . . . . .	584
Scripts . . . . .	602
Dynamic Data Exchange (DDE) . . . . .	604
Other Applications to HyperChem . . . . .	604
HyperChem to Other Applications . . . . .	605
HyperChem and Excel . . . . .	606

## **Chapter 12 Using HyperChem Results 611**

Setting Display Conditions . . . . .	611
Transferring Graphics . . . . .	612
Setting Up a Graphics Transfer . . . . .	612
Choosing the Extent of the Image . . . . .	612
Choosing File Formats . . . . .	613
Choosing the Destination . . . . .	613
Saving Color Images . . . . .	614
Using Copy Image (or [F9]). . . . .	614
Using the Clipboard. . . . .	614
Using BMP or WMF Files . . . . .	615
Printing a Molecular System . . . . .	616
Storing Results in a Log File . . . . .	616
Stopping a Log . . . . .	616
Adding Comments to a Log . . . . .	617
Storing Results of a Molecular Dynamics Simulation . . . . .	617
Analyzing HyperChem Results . . . . .	618

## Appendix A HyperChem Files 619

Water Files . . . . .	621
Program Files . . . . .	621
Essential DLL Files . . . . .	622
Module Executable Files. . . . .	622
Help Text File . . . . .	622
Script Files . . . . .	622
Snapshot Files . . . . .	623
Fixed Sized Header . . . . .	623
Variable Sized Header. . . . .	625
Coordinate and Velocity Data . . . . .	627
Plot and Averages Files . . . . .	629
Graphics Files . . . . .	630
Metafile Graphics . . . . .	630
Bitmapped Graphics . . . . .	631
Log Files. . . . .	631
Extended Hückel Calculations . . . . .	632
NDO Calculations. . . . .	633
Ab initio calculations. . . . .	634
Molecular Mechanics Calculations . . . . .	638
Annotated Example Log File Output . . . . .	639
MM+ . . . . .	639
AMBER . . . . .	643
Extended Hückel . . . . .	644
CNDO. . . . .	645
CNDO Configuration Interaction . . . . .	651

## Appendix B Force Field Files 653

Atom Types . . . . .	654
Atom Typing Rules . . . . .	655
Syntax . . . . .	656
forcefield . . . . .	656
<element-symbol>:. . . . .	656
;<comment>. . . . .	656
<clause> . . . . .	656
<test>? . . . . .	657
Connected to Expressions. . . . .	657
Side Groups . . . . .	659
Rings . . . . .	659
<assignment-statement>.. . . .	659
Force Field Parameters . . . . .	659

Parameter Files . . . . .	660
Parameter Sets . . . . .	661
Newer Force Fields for Amber and BIO+(CHARMM). . . . .	664
Type Parameter File (*typ) . . . . .	665
Bond Length Parameter File (str) . . . . .	665
MM+ Bond Length Parameter file. . . . .	666
Bond Angle Parameter File (ben) . . . . .	666
MM+ Bond Angle Parameter Files. . . . .	667
Urey-Bradley Bend Parameter File . . . . .	668
Torsion Parameter File (tor) . . . . .	669
MM+ Torsion Angle Parameter Files . . . . .	669
Improper Torsion Parameter File (*imp) . . . . .	670
BIO+ Improper Torsion Parameter File . . . . .	671
Out-of-plane Bending Parameter File (mmpoop) . . . . .	672
Nonbonded Parameter Files (nbd) . . . . .	673
Pair-based Nonbonded Parameter Files . . . . .	675
Hydrogen Bond Parameter File (am*hbd) . . . . .	677

## **Appendix C Quantum Mechanics Files 679**

Semi-Empirical Files . . . . .	679
Extended Hückel Parameter File . . . . .	681
exhuckel.abp File Format . . . . .	681
References for Parameters in exhuckel.abp . . . . .	682
CNDO and INDO Parameter File . . . . .	684
cindo.abp File Format . . . . .	684
MINDO/3 Parameter Files . . . . .	685
mind03_1.abp File Format . . . . .	686
mind03_2.abp File Format . . . . .	687
mind03_3.abp File Format . . . . .	688
MNDO and MNDO/d Parameter Files . . . . .	688
mndo_1.abp or mndod_1.abp File Format . . . . .	688
mndo_2.abp or mndod_2.abp File Format . . . . .	689
AM1 and PM3 Parameter Files . . . . .	690
am1_1.abp and pm3_1.abp File Format . . . . .	690
am1_2.abp and pm3_2.abp File Format . . . . .	692
am1_3.abp and pm3_3.abp File Format . . . . .	693
ZINDO/1 Parameter Files . . . . .	693
zindo1_1.abp File Format . . . . .	693
zindo1_2.abp File Format . . . . .	694
zindo1_3.abp File Format . . . . .	695
ZINDO/S Parameter Files . . . . .	696
zindos_1.abp File Format . . . . .	696

zindos_2.abp File Format . . . . .	697
zindos_3.abp File Format . . . . .	698
zindos_4.abp File Format . . . . .	699
TNDO Parameter Files . . . . .	700
The Default TNDO Parameter Set . . . . .	700
Improved Parameter Sets . . . . .	700
default.abp TNDO File Format . . . . .	702
Basis Set Files . . . . .	703
Basis Set File Format . . . . .	703
Basis Set . . . . .	703
Syntax. . . . .	704
Partial List of Basis Set Files . . . . .	705
Creating a New Basis Set. . . . .	709

## **Appendix D HIN Files 711**

Structure of a HIN File . . . . .	711
File Syntax . . . . .	712
; comment . . . . .	713
forcefield. . . . .	714
sys. . . . .	714
seed . . . . .	715
view . . . . .	715
box . . . . .	716
mol . . . . .	716
res. . . . .	716
atom . . . . .	717
Atom Flags . . . . .	718
vel. . . . .	718
mass . . . . .	718
basset . . . . .	719
formalcharge . . . . .	719
endres . . . . .	719
endmol . . . . .	720
selection . . . . .	720
selectrestraint . . . . .	720
Setting Flag Bits for selectrestraint . . . . .	721
selectatom . . . . .	722
endselection . . . . .	722
dynamics . . . . .	722
aromaticring . . . . .	722
text . . . . .	723
line . . . . .	724

circle . . . . .	724
rectangle . . . . .	725

## **Appendix E Residue Template Files 727**

Structure of a TPL File . . . . .	727
Template Syntax. . . . .	729
;<comment> . . . . .	730
[<residue-name>]. . . . .	730
<atom-description> . . . . .	730
<atom-name> <aliases> . . . . .	732
<aliases>. . . . .	732
<element> . . . . .	732
<tag>. . . . .	732
(bonding-info). . . . .	733
<force-field> . . . . .	733
internal . . . . .	734
Amino Acid Residues . . . . .	735
Nucleic Acid Residues . . . . .	737
Adding New Residues . . . . .	739
General Requirements . . . . .	739

## **Appendix F Default Settings in the Registry 743**

Registry Structure . . . . .	743
Registration . . . . .	746
Ab Initio Settings . . . . .	747
AbInitio . . . . .	747
AMBER Settings . . . . .	749
amber . . . . .	749
amber, <i>CustomName</i> . . . . .	750
Basis Set Settings . . . . .	753
BasisSet . . . . .	753
BIO+ Settings. . . . .	755
bio+ . . . . .	755
bio+, <i>CustomName</i> . . . . .	756
Computational Settings . . . . .	759
Compute . . . . .	759
Filters Settings . . . . .	767
Filters . . . . .	768
MM+ Settings . . . . .	769



mm+ . . . . .	769
mm+, <i>CustomName</i> . . . . .	769
Network Settings . . . . .	772
Network . . . . .	772
OPLS Settings . . . . .	773
opls . . . . .	773
opls, <i>CustomName</i> . . . . .	773
Path Settings . . . . .	776
Path . . . . .	776
Preference Settings . . . . .	777
Preferences . . . . .	777
Registration Settings . . . . .	785
Registration . . . . .	785

<b>Credits and Acknowledgments</b>	<b>787</b>
------------------------------------	------------

<b>Glossary</b>	<b>789</b>
-----------------	------------

<b>Index</b>	<b>805</b>
--------------	------------

# Chapter 1

## Introduction

### What is HyperChem?

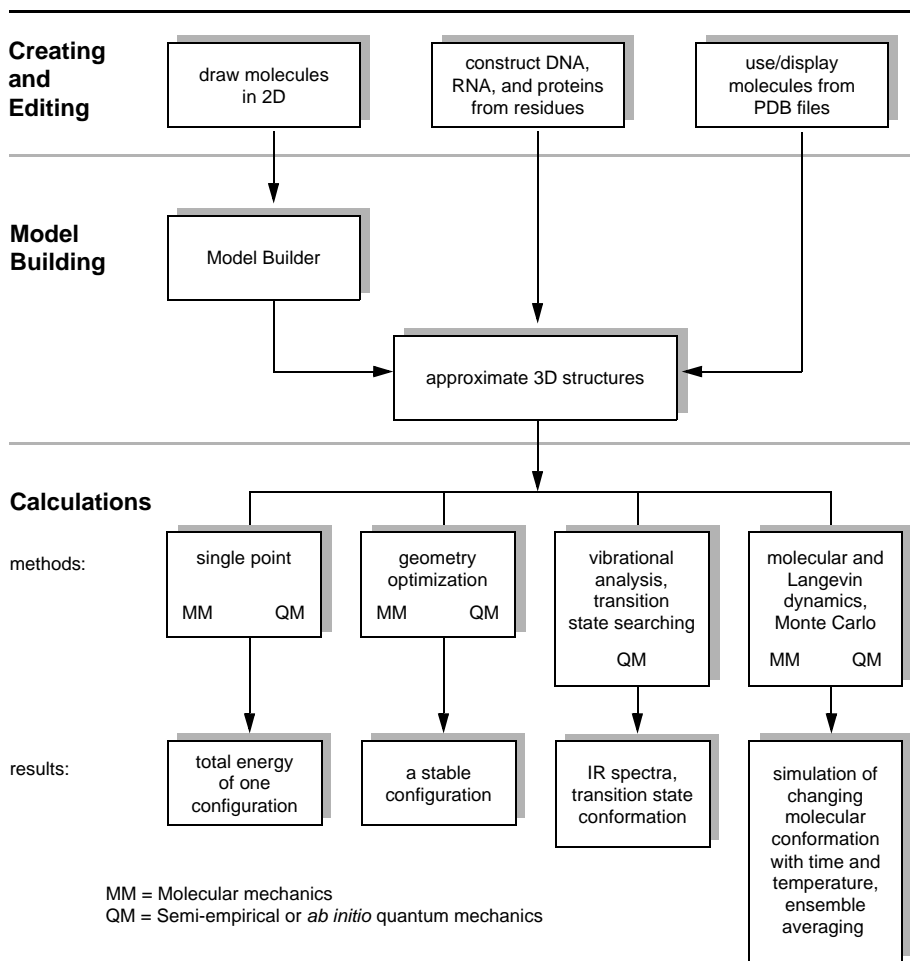
HyperChem® is a molecular modeling and simulation program that lets you perform complex chemical calculations.

Use the Glossary at the back of this book to look up terms and concepts you might be unfamiliar with. Use the master index to find subjects in *Getting Started*, *Computational Chemistry*, and the *HyperChem for Windows Reference Manual*.

HyperChem includes these functions:

- Drawing molecules from atoms and converting them to three-dimensional (3D) models
- Constructing proteins and nucleic acids from standard residues
- Using molecules from other sources; for example, Brookhaven Protein Data Bank (PDB) files
- Rearranging molecules by, for example, rotating and translating them
- Changing display conditions, including stereo viewing, rendering models, and structural labels
- Setting up and directing chemical calculations, including molecular dynamics, by various molecular mechanical or *ab initio* or DFT or semi-empirical quantum mechanics methods
- Determination of isotope effects in vibrational analysis calculations for semi-empirical and *ab initio* SCF methods
- Graphing the results of chemical calculations
- Solvating molecules in a periodic box

## HyperChem: Summary of Major Functions



## Calculation Methods

HyperChem provides these methods for chemical calculations:

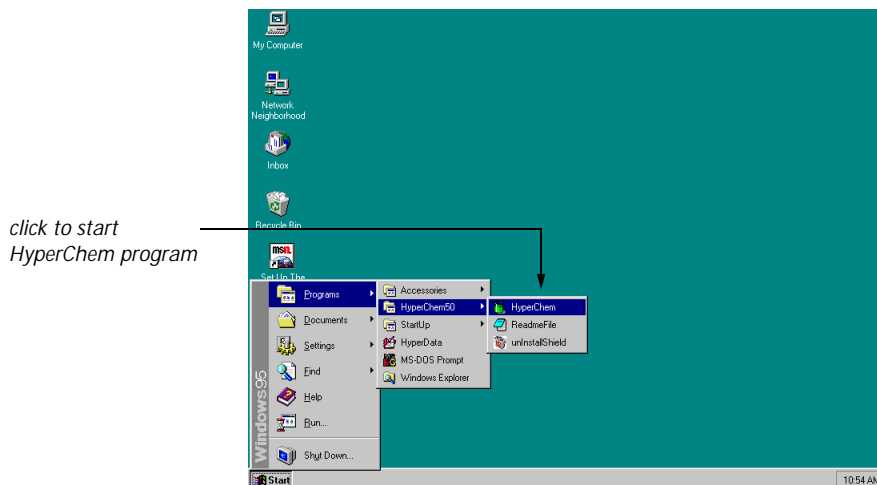
- MM+, AMBER, BIO+(CHARMM), and OPLS force fields for molecular mechanics calculations
- Extended Hückel, CNDO, INDO, MINDO3, MNDO, MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO semi-empirical, and *ab initio* or DFT quantum mechanical calculations

HyperChem can mix molecular mechanics and semi-empirical or *ab initio* or DFT (quantum mechanics) calculations for a molecular system. You can use these two types of methods on selected portions of the same molecule.

## Starting HyperChem

To start HyperChem:

Click on “Start”. Move the mouse pointer across “Programs”, “HyperChem 7”, and “HyperChem Pro”. If you have moved the mouse correctly, you can then click on “HyperChem Pro” to start the program.



**Note:** HyperChem can take a file name as a command line argument. If the file has extension HIN or ENT, HyperChem will read

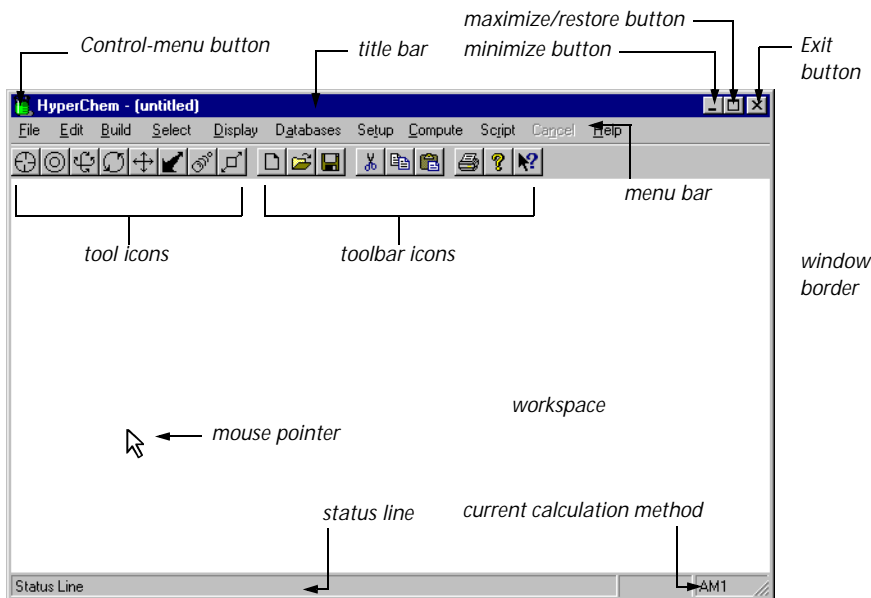
the molecular system in HIN or PDB format respectively (see “Open” on page 78). If the file has the extension SCR, then HyperChem will run the file as a script file (see “Open Script” on page 106).

If you have a file of script commands named **chem.scr** in the HyperChem directory, it will be executed automatically when HyperChem starts.

## HyperChem for Windows

HyperChem works with the various versions of Microsoft® Windows™, including Windows 95™ through Windows XP™, and follows the rules for a Windows application. See Microsoft Documentation for information about Windows organization and operation.

When you start HyperChem, you see an empty HyperChem window. A typical window looks like this:



<b>Control-menu button</b>	Double-click on this box to close the window. Click to see the Control menu, containing items to let you move or reorganize the window you are using. You can also do most of these tasks by using the mouse on other parts of the window.
<b>Title bar</b>	Shows <i>HyperChem</i> plus the filename of the molecular system in the workspace. If there is more than one HyperChem window, the title bar of the active window has a distinct color. (The color depends on your choice in the Windows Control Panel.)
<b>Minimize button</b>	L-click to reduce the window to an icon.
<b>Maximize button</b>	L-click to enlarge the window to fill the whole screen. Once selected, this button turns into a Restore button, which returns the window to its previous size.
<b>Exit button</b>	Click to exit HyperChem. You can also exit HyperChem from the File menu item.
<b>Restore button</b>	L-click to reduce a window from full-screen size to fill part of the screen. This button appears only after you use the Maximize button.
<b>Menu bar</b>	Shows the names of available pull-down menus. L-click on a menu name to pull down the menu.
<b>Tool icons</b>	Controls the HyperChem tools for drawing, selecting, and moving molecules, and for zooming and setting the clipping slab.  With Release 6 of HyperChem Pro and beyond, an additional set of Annotation Icons, not shown above, are also available. These icons, described in Chapter 3 , are used for creating annotations. Annotations may be of type text, line, ellipse, or rectangle.
<b>Toolbar icons</b>	These buttons provide shortcuts for menu items such as file opening and saving, cutting and pasting, etc. Workspace  Contains the molecular system you work on.
<b>Mouse pointer</b>	The mouse cursor for choosing menu items, choosing and operating tools, and working with win-

	dows. When you move this cursor into the workspace, it looks like a HyperChem tool.
<b>Status line</b>	Shows the function of menu items and the results of selections and calculations.
<b>Window border</b>	To change the size of a window, place the mouse pointer on this border, L-click and hold down the button, and then drag. Different parts of the border provide different types of size adjustments. The border for the active window has a distinct color; for example, a gray window border signifies that the window is active, and a white border that it is inactive.

## Using Multiple Windows

You can have several HyperChem windows on screen at once.

To start another HyperChem window:

1. If the HyperChem window is filling the screen, L-click on the Minimize or the Restore button in the active HyperChem window.

The Minimize button reduces the window to an icon named HyperChem - [FILENAME]. The Restore button reduces the window to a smaller window. In either case, the window is smaller but still active.

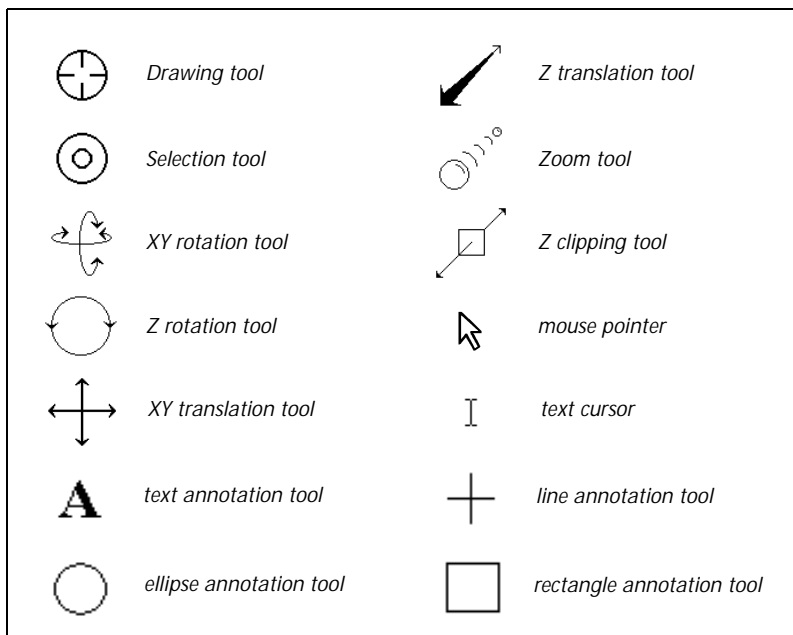
2. Follow the original instructions for starting HyperChem to open a new window named HyperChem - [untitled].
3. Double-click on the original HyperChem icon or window to expand it.
4. L-click anywhere on the window to activate it.

**Warning:** If you are using a network licensed version of HyperChem, you must have sufficient licenses to open multiple HyperChem windows.

Each HyperChem window can contain a different molecular system and have different preference settings (see "Preferences" on page 95), display conditions (see "Display Menu" on page 248), and computational choices.

## Using the Mouse

You can move the mouse cursor and HyperChem tools by moving the mouse. Outside the HyperChem workspace, the cursor becomes a mouse cursor and looks like an arrowhead. Inside the workspace, the cursor becomes a HyperChem tool and looks like one of the following tool icons (see chapter 3, "Tools"):



Use a mouse for these tasks:

- Drawing, selecting, or moving molecules
- Choosing items on the menu bar
- Choosing options in dialog boxes
- Editing text in dialog boxes
- Creating annotations (text and line drawings)

To use the mouse with menus and dialog boxes, follow standard procedures for Microsoft Windows.



To use the mouse with HyperChem, first position the cursor on a molecular structure or icon. Then use one of the following procedures, depending on the task:

Procedure	How to do it	What it does
L-click R-click	Press and release the left or right mouse button	Do this to choose menu items and options in dialog boxes, and to select atoms. L-click and R-click have opposite effects when using tools, for example, when selecting and deselecting atoms
L-drag R-drag	Press and <i>hold down</i> a mouse button (left or right), and then move the mouse to perform an operation; release the mouse button to complete the operation	Use L-drag to choose menu items and text, to select atoms, and to draw molecules (with the Drawing tool). Use R-drag to deselect atoms or to move molecules in the Molecular Coordinate System
LR-drag RL-drag	Hold down both mouse buttons, but press the left first. RL-drag reverses the order of holding down the mouse buttons. With both mouse buttons held down, move the mouse to adjust the size of a selection rectangle or circle (see "Select Sphere and Group Selecting" on page 227). Then release both mouse buttons	Use LR-drag to select groups of atoms; use RL-drag to deselect a group of atoms

Procedure	How to do it	What it does
[Shift]+L-click [Shift]+R-click	Hold down [Shift] while you press and release the <i>left</i> or <i>right</i> mouse button	Use this for changing chirality (with the Drawing tool) or stepwise operation of the HyperChem tools (see chapter 3, "Tools"). The left button moves all molecules, and the right button moves selected molecules
Double-click	Press and release the left mouse button twice rapidly	This is usually a shortcut for a common operation. In some dialog boxes, double-click on the option instead of L-clicking on an option and then on OK. Double-clicking on a bond with the Drawing tool creates an aromatic bond

## Using Keyboard Alternatives for Windows

Microsoft Windows provides keyboard alternatives to using the mouse. The keyboard might be easier and faster for some users. Use these standard Windows procedures to open menus, to choose menu items, or to choose options in dialog boxes:

Opening a menu	Press [Alt], then the key corresponding to the underlined letter in the menu name. For example, to open the <u>S</u> elect menu, press [Alt] and then [S].
Closing a menu	Press [Esc].
Choosing a menu item	Press the key corresponding to the underlined letter. For example, for <u>O</u> pen, press [O].

<b>Closing a dialog box</b>	Press [Enter] to close a dialog box and to accept the choices you made. This usually corresponds to L-clicking on OK.
<b>Choosing a dialog box option</b>	Press the key corresponding to the underlined letter. For example, for <u>N</u> o, press [N].
<b>Moving within a dialog box</b>	Press [Tab] to move between sections of a dialog box.
<b>Moving through a list</b>	Press the [Up] and [Down] keys. The list can be a scroll box containing filenames or a set of check boxes or buttons.
<b>Checking a box</b>	Press [Spacebar] to place an X in a check box or to remove an X. You find these check boxes in dialog boxes. Checking a box turns on an option.

## Using Keyboard Alternatives for HyperChem

HyperChem has special keyboard shortcuts for using HyperChem tools; see “Keyboard Equivalents for Tools” on page 49.

You can also use these keys for HyperChem functions:

Keys	Function	Description
[F1]	Help	Shows the Help contents or, if you highlighted a menu item or have a dialog box active, the Help for that menu item or dialog box.
[Shift]+[F1]	Help	Shows context-sensitive Help. If no menu or dialog is active this will change the cursor to the Help cursor. You click on a Tool icon, menu item or other part of the HyperChem window for Help on that topic.
[Alt]+[F4]	Exit	Duplicates Exit on the File menu

Keys	Function	Description
[Ctrl]+[N]	New	Clears the HyperChem workspace. This duplicates New on the File menu.
[Ctrl]+[O]	Open	Presents the Open File dialog box to read in a molecular system stored in a file. This duplicates Open on the File menu.
[Ctrl]+[S]	Save	Presents the Save File dialog box for storing a molecular system in a file. This duplicates Save on the File menu.
[Ctrl]+[A]	Save As	Presents the Save File dialog box for storing a molecular system in a file. This duplicates Save As on the File menu.
[Del]	Clear	Removes selected atoms from the workspace. This duplicates Clear on the Edit menu. You cannot recover these atoms from the Clipboard.
[Ctrl]+[X]	Cut	Transfers selected atoms from the workspace to the Clipboard. This duplicates Cut on the Edit menu.
[Ctrl]+[C]	Copy	Copies selected atoms from the workspace to the Clipboard. This duplicates Copy on the Edit menu.
[Ctrl]+[V]	Paste	Copies selected atoms from the Clipboard to the workspace. This duplicates Paste on the Edit menu.
[F9]	Copy Image	Places a picture of all or part of the monitor screen in a file or on the Windows Clipboard. [F9] duplicates functions of Copy Image on the Edit menu, and lets you save images of the workspace, pull-down menus, and dialog boxes. Use Setup Image on the Edit menu to tell HyperChem how [F9] should work.
[Esc]	Cancel	Duplicates Cancel on the menu bar. Stops a calculation or reading of a large file.

---

Keys	Function	Description
[Spacebar]	Scale to Fit	Duplicates Scale to Fit on the Display menu (see page 250). This brings the molecular system or selected atoms to the center of the screen so that they fit in the workspace. This is useful after you move molecules with a Translation tool or the Zoom tool. Also sets the clipping planes for Z clipping and adjusts the y axis of the data plot in the Molecular Dynamics Results dialog box.

---

## HyperChem Menus

You select HyperChem commands and settings from pull-down menus on the menu bar above the workspace. When you highlight a menu item, a description of that item and its action appears on the status line (see “HyperChem for Windows” on page 4).

Use menu items to set HyperChem conditions (like the type of labels on atoms) and to start HyperChem procedures and calculations. Menu items work as follows:

- Menu items with ellipsis points, such as Select... on the Select menu, open dialog boxes with buttons, text fields, and check boxes for making choices.
- Some menu items turn on or off (toggle) a condition. Turning on a menu item places a check mark (3) to the left of the item. The condition remains on until you select the same item again or, in some cases, select an alternative item.

Turning on a menu item can have an immediate effect on a molecular system. For example, Show Hydrogens on the Display menu shows hydrogens for the molecules in the workspace and for all molecules that you later draw. Other items, such as Explicit Hydrogens on the Build menu, change the way the Drawing tool works, but do not affect existing molecules.

## Gray Menu Items

HyperChem changes gray menu items to black to show that the item is available. You must satisfy a condition to activate this type of menu item. For example, Invert on the Edit menu is gray until you select atoms and give the selection the predefined name POINT. This activates Invert (and changes it from gray to black) so that you can invert the molecule.

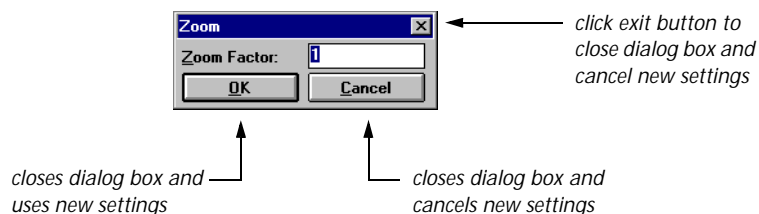
## HyperChem Dialog Boxes

Choosing most menu items displays a dialog box for entering information, choosing options, or starting a HyperChem function. Choosing some dialog boxes displays additional dialog boxes. Message dialog boxes appear showing warning or error conditions.

You can change the position of a dialog box by placing the mouse cursor on its title bar and L-dragging it anywhere on screen.

HyperChem has two ways of removing a dialog box when you have finished with it:

- To close a dialog box without using new settings, L-click on Cancel or click on the Exit button in the upper-right corner of the dialog box.
- To close a dialog box and accept new settings or to start a calculation, L-click on OK or Proceed in the dialog box.



Most dialog boxes temporarily inactivate HyperChem menus and tools. When you close the dialog box, HyperChem regains its full capabilities. Persistent dialog boxes (see the next section) are an exception to this rule.

## Persistent Dialog Boxes

The Default Element, Amino Acids, Nucleic Acids, and Ab Initio dialog boxes are persistent. You can leave these dialog boxes on screen and still use HyperChem menus and tools. When you have finished using one of these dialog boxes, double-click on the Control-menu box to remove it. The structure remains in the workspace.

## Working in Dialog Boxes

HyperChem dialog boxes contain these standard Windows components:

- Buttons and check boxes. L-click to choose the items associated with a button or check box. The center of a button turns black and a 3 appears in a check box. These remain activated until you L-click again.
- Lists with scroll bars. L-click on a list item. A highlight (black background) appears and the text turns white. If the list is longer than the list box, a scroll bar appears on the right to let you move through the list.
- Text boxes. You can enter information into text boxes. See the procedures in the next section.

## Editing Text

HyperChem dialog boxes contain text boxes for entering text or numbers. Follow these rules:

- L-click with the text cursor (I-shaped) anywhere in the text to move the insertion point (a vertical, flashing bar).
- L-drag the text cursor to select text that you want to delete or replace.
- To delete text, select it and press [Delete] or [Backspace].
- To replace text, select it and start entering new text.

## Editing Keys

You can use these keys for editing:

[Enter] Moves the insertion point to a new line.

[Backspace] Erases the character (or selected text) to the left of the insertion point.

[Delete][Del] Erases the character (or selected text) to the right of the insertion point.

[4] [3] Moves the insertion point.

[1] [2]

## Case Preferences

For filenames, upper- and lowercase letters are equivalent. For example, HyperChem recognizes molecule.hin and Molecule.HIN as the same filename.

## Entering Exponents

When entering numbers, use this format for scientific notation (exponents):  $Y \times 10^N = YeN$ .

For example, for  $1.05 \times 10^4$ , enter 1.05e4. HyperChem warns you about numbers that are out of range or in the wrong format.





# Chapter 2

## The Basics of HyperChem

### Levels of Molecular Organization

HyperChem uses four levels of molecular organization. It is important to understand these levels:

**Atoms**            Single atoms, either free or in a chemical compound. The HyperChem Drawing tool (see “Drawing Molecules” on page 29) creates atoms and bonds between them.

**Residues**        Groups of bonded atoms, such as amino acid or nucleic acid subunits of a protein or polynucleotide. Residues come only from database files or HyperChem Input (HIN) files. HyperChem supplies database files of amino acid and nucleic acid residues (see “Databases Menu” on page 139). Proteins and polynucleotides from PDB files (see page “Protein Data Bank Files” on page 503) are also composed of residues. Water molecules in either a periodic box or from a PDB file are residues. They show the residue labels HOH or WAT.

Residues have these properties:

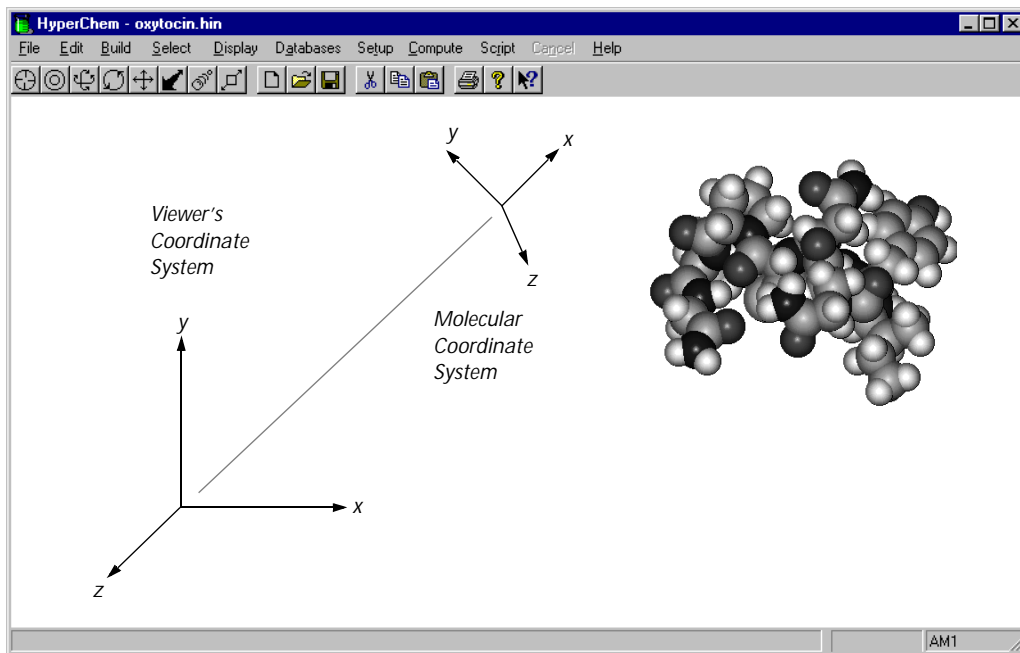
- They have standard structures.
- Their atoms may have partial charges.
- You can select them with the Selection tool in the residues mode.
- You can add labels characteristic of residues (see “Labels” on page 281).

Amino acids or nucleotides that you construct from atoms do not have these properties.

<b>Molecules</b>	<p>A molecule is composed of covalently bonded atoms or residues. You can create molecules with the Drawing tool, construct polymers by using the Databases menu, or display molecules stored in a HIN or PDB file (see “Protein Data Bank Files” on page 503 and “HIN Files” on page 711). HyperChem does not have a fixed limit on structure size. However, the memory capacity or speed of your computer might impose practical limits on the size of molecule you can conveniently manipulate.</p> <p>Because the concept of a “molecule” in HyperChem is that of a “connected graph”, an individual atom is also a “molecule”.</p>
<b>Molecular System</b>	<p>A molecular system is one or more molecules displayed in the HyperChem workspace or stored in a file.</p>

## Coordinate Systems

HyperChem has two coordinate systems for displaying molecules: the Viewer's Coordinate System and the Molecular Coordinate System. Each coordinate system has x, y, and z Cartesian axes.



Each molecular system also has its own internal axes, called inertial axes (primary, secondary, and tertiary). These axes are fixed in the Molecular Coordinate System.

## Viewer's Coordinate System

The Viewer's Coordinate System defines the positions of atoms relative to you, the viewer. The Viewer's Coordinate System is fixed by the screen geometry. The y axis is vertical, the x axis is horizontal, and the z axis is perpendicular to the plane of the screen.

In the Viewer's Coordinate System, you can move molecules without changing their atomic coordinates. With HyperChem tools, use the *left* mouse button to move molecules in the Viewer's Coordinate System. Molecules move as if you were changing your viewpoint and moving past the molecules. This works only for whole molecular systems.

With some HyperChem dialog boxes, like Align Viewer, movement is automatically in the Viewer's Coordinate System if there are no selected atoms.

When you first create a molecule, HyperChem aligns the Viewer's and the Molecular Coordinate System so that the x, y, and z axes correspond. As you move molecules, the coordinate systems might lose this correspondence. Use **Align Viewer** on the Edit menu (see page 215) to move the Viewer's Coordinate System so that it once again corresponds to the Molecular Coordinate System.

When you save a molecular system in a HIN file, HyperChem includes information about the Viewer's Coordinate System (see "view" on page 715). Opening a file restores the previous view of the molecular system stored in the file.

## Molecular Coordinate System

The Molecular Coordinate System is attached to a molecular system and determines its atomic coordinates, which are the coordinates used in all HyperChem chemical calculations.

You can determine the molecular coordinates of any atom by selecting the atom (see "Measuring with the Selection Tool" on page 35). The coordinates appear in the status line.

With HyperChem tools, you move the atoms in the Molecular Coordinate System by using the *right* mouse button. This works only for selected atoms, residues, and molecules. This table summarizes the right mouse button functions:

Tool	Right Mouse Button Function
XY rotation	Rotates selected molecules only
Z rotation	Rotates selected molecules or selected, rotatable side chains
XY translation	Moves selected atoms or whole molecules <sup>a</sup>
Z translation	Moves selected atoms or whole molecules <sup>a</sup>

a. Right button function is limited to whole molecules if Whole molecule translation is on (✓) in the Tool Preferences dialog box.

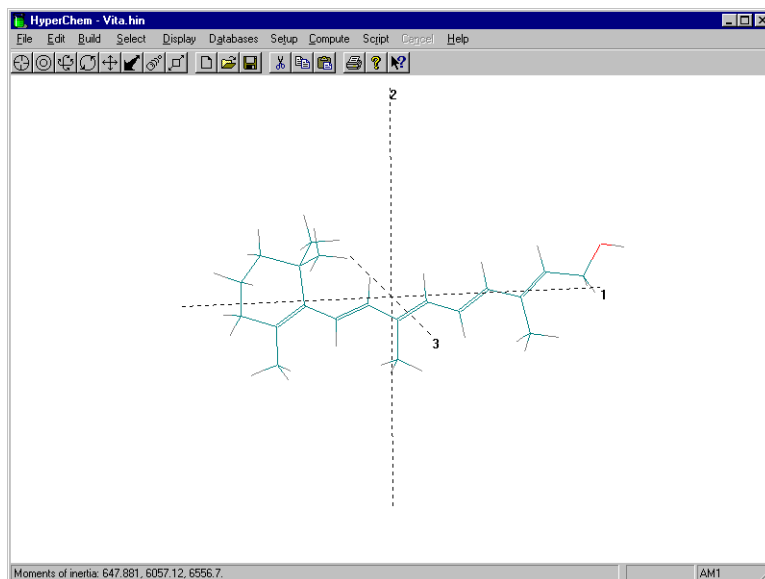
With some HyperChem dialog boxes, like **Align Molecules**, movement is automatically in the Molecular Coordinate System.

## Inertial Axes

For either selected atoms or whole molecular systems, HyperChem can calculate the moments of inertia and inertial axes. HyperChem reports the moments of inertia on the status line and displays the inertial axes as dotted white lines, fixed in the Molecular Coordinate System. These three moments and axes (primary, secondary, and tertiary axes) are functions of the location and mass of atoms. The inertial axes belong to a molecular system and are independent of the Viewer's or Molecular Coordinate System.

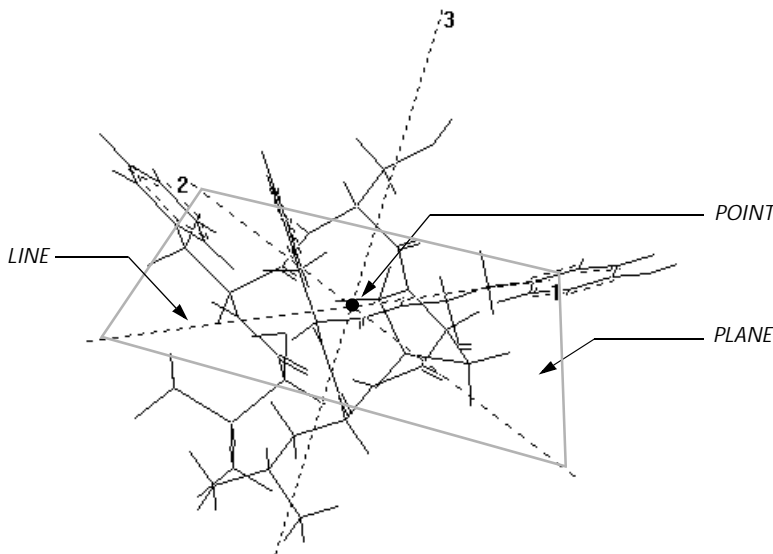
Turn on Show Axes (✓) on the Display menu to show the inertial axes. These are orthogonal and are numbered 1, 2, and 3, for primary, secondary, and tertiary axes respectively. The axes remain in the workspace until you turn off Show Axes (no check mark).

The primary inertial axis usually marks the longest distance through a molecular system, and the tertiary axis marks the shortest distance.



HyperChem uses inertial axes for rearranging molecules (see “Align Molecules” on page 216) in the workspace and when defining POINT, LINE, and PLANE (see “Select Dialog Box” on page 233). These definitions are in terms of inertial axes.

<b>POINT</b>	The intersection of the inertial axes at the center of mass of the selected atoms.
<b>LINE</b>	The primary inertial axis of the selected atoms. <sup>1</sup>
<b>PLANE</b>	The plane formed by the primary and secondary inertial axes, perpendicular to the tertiary axis of the selected atoms.



## Constructing a Molecule

This section describes creating a molecule in HyperChem and indicates where you can find more information. You can draw molecules “from scratch” or construct them from residues.

To draw a molecule:

1. Use the Drawing tool to sketch atoms in a molecule and connect them with covalent bonds. (See “By default, when the cursor pauses over one of the tool icons, a message briefly describing the tool will pop up. You can disable this feature by changing the line ToolHelpPopup=Yes to ToolHelpPopup=No in your Registry.” on page 28.)

1. The primary inertial axis corresponds to the smallest eigenvalue of the moment of inertia matrix.

- Select and change the element from the Default Element dialog box. (See “Changing the Drawing Element” on page 473.)
  - L-click or L-drag to create individual atoms or bonded atoms. (See “Drawing Molecules” on page 471.)
  - Specify the bond order by clicking bonds with the Drawing tool. L-clicking a bond increases the bond order, and R-clicking decreases it. Double-clicking a bond creates a conjugated bond; double-clicking a bond in a ring creates an aromatic ring. (See “Changing the Bond Order” on page 475.)
  - With wedges turned on in stick rendering (see “Rendering” on page 258), specify stereochemistry by [Shift]+L-clicking on bonds before model building. (See “Drawing Molecules” on page 471 and “Changing Chirality” on page 485.)
2. Invoke the Model Builder on the Build menu to create a 3D molecule from the 2D structure you drew. (See “Model Build/Add H & Model Build” on page 124.)

Use any of the following options to make changes to the resulting structure:

1. Edit the system with the Drawing tool and with Cut, Copy, and Paste commands on the Edit menu. (See “Edit Menu” on page 201.)
2. Specify the molecular conformation.
  - If necessary, change the bond length, bond angle, or torsion angles of selected atoms by using one of the Set commands on the Edit menu. (See “Set Bond Length” on page 218, “Set Bond Angle” on page 219, and “Set Bond Torsion” on page 220.)
  - Establish Model Builder constraints for the geometry, bond length, bond angle, or torsion angles by using the Constrain commands on the Build menu. (See “Constrain Geometry” on page 134, “Constrain Bond Length” on page 136, “Constrain Bond Angle” on page 137, and “Constrain Bond Torsion” on page 138.)
  - Set the chirality by shift-clicking on atoms using the Drawing tool or by using Invert and Reflect commands on the



Edit menu. (See “Invert” on page 206 and “Reflect” on page 208.)

*Important:* Run the Model Builder again after you have set the conformation.

3. Choose Geometry Optimization on the Compute menu to improve the conformation of the system or selected subset, using either molecular mechanics or semi-empirical or *ab initio* quantum mechanics. (See “Geometry Optimization” on page 362.)

*Important:* Do not run the Model Builder again, or you will lose the optimization.

To build a polypeptide from residues:

1. Choose Amino Acids on the Databases menu. (See “Amino Acids” on page 141.)
2. Specify the conformation of the first residue in the Conformation section of the dialog box.
3. L-click the first amino acid residue. L-click Ace if you want an N-terminal cap.
4. Continue specifying conformations and clicking residues until the polypeptide is complete.
5. L-click Nme if you want a C-terminal cap.
6. If you did not cap the ends of the polypeptide, you can make it a zwitterion by choosing Make Zwitterion on the Databases menu. (Remove Ionic Ends reverses this action.) (See “Make Zwitterion” on page 145.)
7. Change an individual amino acid within the protein by selecting it and choosing Mutate on the Databases menu. (See “Mutate” on page 190.)

To build a polynucleotide from residues:

1. Choose Nucleic Acids on the Databases menu. (See “Nucleic Acids” on page 157.)
2. Specify the conformation of the polynucleotide in the Conformation section of the dialog box.
3. L-click the first nucleic acid residue. L-click on 5CAP if you want a 5' cap.

4. Continue specifying conformations and clicking residues until the polynucleotide is complete.
5. L-click 3CAP if you want a 3' cap.
6. If you want sodium ions to balance the negative charge on each phosphate group, choose Add Counter Ions on the Databases menu. (See "Add Counter Ions" on page 162.)
7. Change an individual nucleic acid within the polynucleotide by selecting it and choosing Mutate on the Databases menu. (See "Mutate" on page 190.)

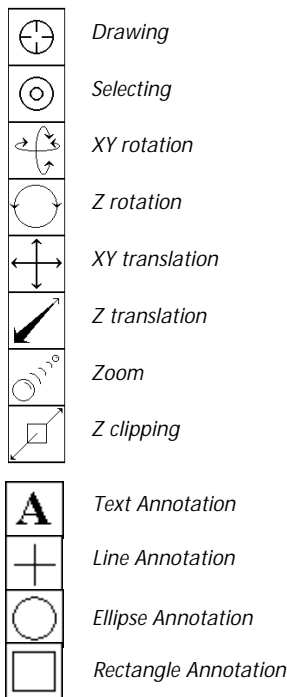
To add water molecules:

1. If you want the molecular system in a periodic box containing water molecules (rather than *in vacuo*), choose Periodic Box on the Setup menu. (See "Periodic Box" on page 335.)
2. Specify the options for the periodic box in the Periodic Box Options dialog box.



# Chapter 3

## Tools



*Drawing*

*Selecting*

*XY rotation*

*Z rotation*

*XY translation*

*Z translation*

*Zoom*

*Z clipping*

*Text Annotation*

*Line Annotation*

*Ellipse Annotation*

*Rectangle Annotation*

This chapter explains the eight HyperChem molecule tools you use to construct, model, select, and move molecules plus the four annotation tools that are used to create text, lines, ellipses and rectangles serving as annotations to the molecules in the workspace. A Toolbar containing these 12 tools appears on top of the HyperChem workspace. An additional standard Windows toolbar for “file open”, etc. accompanies the HyperChem specific toolbar.

Each tool is represented by an icon, or symbol. When you use the tool, the cursor takes the shape of the icon (except for the line tool).

The Selecting, XY Translation, and Zoom tool may also be used to manipulate annotations as well as atoms and molecules.

You can use the tool icons in these ways:

**L-click** Place the mouse cursor on a tool icon and click the left mouse button to turn the cursor into a tool. When you move the cursor to the workspace, the cursor changes to a tool, with the appearance of the tool icon.

**Double-click** Place the cursor on a tool icon and click the left mouse button twice, rapidly, to begin a procedure or to show a dialog box related to the tool.

For mouse fundamentals, see “Using the Mouse” on page 7. The following descriptions give you more information on HyperChem tools.

### Pop-up Help for Tool Icons

By default, when the cursor pauses over one of the tool icons, a message briefly describing the tool will pop up. You can disable

this feature by changing the line ToolHelpPopup=Yes to ToolHelpPopup=No in your Registry.

## Drawing Tool



With the Drawing tool, you can draw and construct molecules in two dimensions (2D). You supply information about types of atoms and their bonding, without concern for accurate geometry. You can later ask HyperChem to turn the 2D drawing into a 3D structure. You can then use a geometry optimization calculation (see “Geometry Optimization” on page 362) to improve the 3D structure.

You can do the following with the Drawing tool:

- Create molecules containing any element from the periodic table
- Add or remove atoms from existing molecules
- Create covalent bonds and change bond order
- Change chirality and local conformation

To use the Drawing tool icon:

- |              |  |
|--------------|--|
| L-click      | L-click to turn the cursor into the Drawing tool.  |
| Double-click | Double-click on the Drawing tool to see the Element Table dialog box. You can set the default element (the type of atom you are using for drawing, for example, carbon) or turn on or off Allow Ions and Explicit Hydrogens. You can also reach this dialog box by choosing Default Element on the Build menu. |

## Drawing Molecules

To draw molecules:

1. L-click on the Drawing icon and move the Drawing tool into the workspace.
2. To create individual atoms, L-click anywhere in the workspace.
3. To remove an atom, R-click near the atom.
4. To create two singly bonded atoms, L-drag then release the mouse button.
5. To add another bond and atom, L-click near an atom, L-drag, and then release the mouse button.
6. To change an element, choose the new element from the Element Table dialog box (see page 121), and then L-click on the atom you want to change.
7. To create a double bond, L-click on a single bond. To create a triple bond, L-click on a double bond.
8. To create a conjugated bond, double-click on a bond. To create an aromatic ring, double-click on a bond in the ring.
9. To decrease the order of a bond (conjugated to single; triple to double; double to single), R-click on it.
10. To remove a single bond and an atom connected only through that bond, R-click on the bond.
11. To specify the chirality or local conformation of an atom, [Shift]+L-click on one of its bonds (with sticks and wedges rendering turned on) to specify constraints for the model builder. You can also [Shift]+L-click on an atom to change the chirality of that atom after model building.

For more information about drawing and revising molecules, see chapter 8, "Constructing Molecules."

### Choosing and Changing Elements

Normally, the Drawing tool places only heavy (nonhydrogen) atoms in a molecule. You choose the heavy atom (from helium through lawrencium) from the Element Table dialog box. This choice is also known as the default element. To see the Element

Table dialog box, double-click on the Drawing icon or choose Default Element on the Build menu.

You can change the default element any time *while drawing* by changing the choice in the Element Table dialog box. Then continue drawing.

To change an element *after drawing* a molecule:

1. From the Element Table dialog box, choose a new element. For example, if you drew your molecule with carbon as the default element and you want to change a carbon atom to an oxygen, click on oxygen in the dialog box.
2. With the Drawing tool, L-click on the atom you want to change.

*Note:* You can change atoms in 2D drawings or, after using the Model Builder, in 3D structures.

## Adding Hydrogen Atoms

HyperChem can automatically add the hydrogen atoms needed to complete the normal valence structure of heavy atoms if you draw the molecule with Explicit Hydrogens off. Use the Model Builder (Add H & Model Build on the Build menu) to add hydrogen atoms, and then create a 3D structure.

You can also add hydrogens manually while drawing. Choose Explicit Hydrogens from the Element Table dialog box or on the Build menu. When Explicit hydrogens is on, Add H & Model Build on the Build menu changes to Model Build. The Model Builder does not automatically add hydrogens.

You can always ask HyperChem to add required hydrogens to a structure by choosing Add Hydrogens on the Build menu. This choice is independent of whether the Explicit Hydrogens setting is on or off.

## Selection Tool



The Selection tool lets you mark atoms or molecules, preparing them for specific operations or procedures, such as labeling or calculations, for example. The tool may also be used to select something in the annotation plane when manipulating an annotation rather than atoms or molecules.

**Remember:** If there is a selection, HyperChem usually acts only on the selected atoms or annotations. If atoms and annotations are both selected, HyperChem acts in preference only on the atoms.

You can select these atomic and molecular entities (structures):

- Atoms
- Bonds
- Bond angles
- Residues (in proteins or polynucleotides and some water molecules)
- Backbones (in proteins or polynucleotides)
- Shortest path between atoms, bonds or residues
- Side chains
- Rings
- Groups of atoms
- Several or all molecules in the workspace

You can also use the Selection tool to measure these values:

- Bond lengths and distances between nonbonded atoms
- Bond angles and torsion angles

To use the Selection icon:

**L-click**            L-click on the Selection icon to turn the cursor into the Selection tool.

**Double-click**    Double-click on the Selection icon to start the Model Builder. The Model Builder turns selected atoms into a 3D chemical model. Double-clicking is the same as choosing Model Build or Add H & Model Build on the Build menu. For a discussion of the Model Builder, see “Converting 2D Drawings into 3D Structures” on page 487.

**Note:** As you select structures, information about these structures appears in the status line.



## Selection Levels

The Selection tool always works at one of these levels of molecular organization:

- |                  |   |
|------------------|---|
| <b>Atoms</b>     | The smallest unit of selection is an atom or bond. L-clicking an atom selects the atom. L-clicking a bond selects the two atoms that the bond connects. |
| <b>Residues</b>  | The smallest unit of selection is a residue from a database or, if there are no residues, an atom or bond.  |
| <b>Molecules</b> | The smallest unit of selection is a whole molecule.   |

These are the first three items on the Select menu. Only one of these items can be active (✓) at a time. The remaining items on the Select menu make selection easier. For a complete discussion of this menu, see page 220.

## Selecting

HyperChem provides several ways to select atoms or molecules:

- Use the Selection tool to select atoms, residues, molecules, and groups of atoms.
- Use the Select menu to select classes of structures and named structures.

## Using the Selection Tool

For these procedures, you must first L-click on the Selection tool icon to obtain the Selection tool.

Procedure	What to do
Selecting single structures	Choose the selection level (Atoms, Residues, or Molecules) on the Select menu. This determines the smallest unit you can select. Choose Multiple Selections on the Select menu to turn it off (no check mark). L-click on a structure to select it. Each selection replaces the previous selection

Procedure	What to do
Accumulating selections	Choose Multiple Selections on the Select menu to turn it on (✓). Then use the procedure for selecting single structures (above). Each selection adds to previous selections
Canceling a selection	R-click on a structure to deselect it
Selecting all atoms	L-click on an empty part of the workspace. Or choose Select All on the Select menu
Deselecting all atoms	R-click on an empty part of the workspace
Selecting a group of atoms	LR-drag (hold both mouse buttons, first the left, then the right) a selection rectangle around a group of structures. Release both mouse buttons to select all atoms that project onto the 2D plane of the screen. To select all structures in a sphere (3D) instead of a rectangle (2D), first choose Select Sphere on the Select menu (✓). Then starting at an atom, LR-drag a circle to the radius that you want. Release both mouse buttons to complete the selection. Group selection works with Multiple Selections either on or off. For more information, see "Select Sphere and Group Selecting" on page 227
Deselecting a group of atoms	RL-drag (hold both mouse buttons, first the right, then the left) a selection rectangle or sphere around a group of atoms (see above). Release both mouse buttons to complete the deselection
Selecting backbone atoms in any molecule	With Atoms on (✓), L-drag from one atom to a distant atom in the same molecule. This selects all atoms and bonds in the shortest continuous path connecting these atoms, but not side groups. With Residues on in the Select menu, this action selects all residues that the shortest path of bonds pass through
Deselecting backbone atoms	With Atoms on (✓), R-drag from one atom to a distant atom in the same molecule. This deselects atoms and bonds in the continuous path connecting these atoms. With Residues on, the residues along the path are deselected

Procedure	What to do
Selecting side chains	With Atoms on (✓), double-click on the side of the bond that is connected to the side chain. This selects all atoms on the side of the bond that you clicked as well as the atom on the other side of the bond
Selecting rings	With Atoms on (✓), double-click on an atom or bond in a ring. This selects the atoms in the smallest ring. With Residues on, double-clicking a ring selects all residues the ring passes through

## Using the Select Menu

These items on the Select menu select structures directly, without the Selection tool.

Procedure	What to do
Selecting all structures	Choose Select All to select all atoms in a molecular system
Complementing a selection	Choose Complement Selection to deselect the existing selection and to select all atoms previously not selected
Saving and recalling selections	Use Name Selection to name selected atoms and to store this selection. HyperChem provides three pre-defined selection names (POINT, LINE, and PLANE) that store the selection and add geometric properties to the selection. You can also store a selection with any name that you provide (see "Name Selection Dialog Box" on page 238). Use Select, also on this menu, to recall a named selection for a molecular system
Selecting atoms and molecules by name or number	Choose Select . . . to select atoms by a previously defined name or to select atoms, residues, and molecules by sequence number (see "Name Selection Dialog Box" on page 238)
Selecting a ring	Choose Extend Ring to select all atoms in a ring, if one bond or atom in the ring is already selected

Procedure	What to do
Selecting a rotatable side chain	Choose Extend Side Chain to select all atoms in a rotatable side chain if the first two atoms in the side chain are already selected. The order of selection for the first two atoms determines the direction of side chain selection (see page 243)
Selecting all atoms up to the sp <sup>3</sup> -sp <sup>3</sup> bonds	Choose Extend to sp <sup>3</sup> to select all atoms, starting from an existing selection, until all bonds with only one atom selected are single bonds between sp <sup>3</sup> atoms. Use this to prepare a selection for a mixed mode quantum mechanics calculation
Selecting a polypeptide or polynucleotide backbone	Choose Select Backbone to select only the backbone atoms and bonds in polynucleotides and polypeptides constructed with residues from the Databases menu or from PDB files. For polypeptides, disulfide bridges are also selected

## Measuring with the Selection Tool

Selecting atoms, residues, molecules, and bonds gives you information about these structures. You can measure bond lengths, angles, and torsion angles, and the equivalent structural factors among nonbonded atoms. Measurements appear on the status line.

Procedure	What to do
Identifying an atom	Turn on Atoms (✓) on the Select menu, and then L-click on an atom. You see the number of the atom in a molecule, the element name, the PDB atom name (if available), the atom type (for the active force field), the atomic charge, and its coordinates in the Molecular Coordinate System
Identifying a residue	Turn on Residues (✓) on the Select menu, and then L-click on a residue. You see the residue name, number, and charge of the residue. A number in parenthesis, if present, is the residue number from the PDB file

Procedure	What to do
Identifying a molecule	Turn on Molecules (✓) on the Select menu, and then L-click on a molecule. You see the molecule number and charge

These measurements use the Selection tool with Atoms on (✓) and Multiple Selections on (✓):

Procedure	What to do
Measuring bond lengths	Select two atoms or the bond between them. Alternatively, select the atom at one end of the bond, and L-drag to the atom at the other end of the bond
Measuring bond angles	Select two adjacent bonds or the three bonded atoms. Alternatively, select the atom at one end of the angle, and L-drag to the atom at the other end of the angle
Measuring torsion angles	Select three bonds or four atoms that define a torsional rotation. Alternatively, select an atom at one end of the four-atom torsion, and L-drag to the atom at the other end of the torsion
Measuring distances between atoms	Select one atom, and then select the second atom. The atoms can be in different molecules. Do <i>not</i> L-drag. Multiple Selections must be on
Measuring improper angles	Select any two bonds or three atoms. The “improper angle” depends on the order of selection
Measuring improper torsion angles	Select any three bonds or four atoms. The “improper torsion angle” depends on the order of selection

## Selecting Annotations

The selection tool can be used to select annotations as well as atoms, residues, and molecules. To select an annotation, L-click on text, a line, an ellipse or a rectangle to give it an outline in the “Selection Color” previously chosen via File/Preferences. The

default selection color is green. To deselect an annotation, R-click on the annotation. If you have any difficulty selecting specific annotations among a number of closely-spaced annotations, use the menu item Annotations/Select Next or Annotations/Select Previous (or the equivalent keys Ctrl+K or Ctrl+L) to select the correct annotation.

## XY Translating and Zooming an Annotation

Once an annotation is selected, operations on that annotation are enabled as operations on atoms would be if atoms were selected. Most of these operations are part of the Annotation Menu. However, two of the tools are also enabled for a selected annotation. The two tools that can be used on a selected annotations are the XY Translation tool and the Zoom tool. The XY Translation tool is used to move annotations around in the workspace. the zoom tool can be used to enlarge or shrink an annotation. The other tools, such as rotation do not apply to an annotation.

## XY Rotation Tool



With the XY rotation tool, you can rotate one or more molecules around the x or y axis, or around both axes at once.

To use the XY rotation tool icon:

- |              |  |
|--------------|--|
| L-click      | L-click on the XY rotation icon to turn the cursor into the XY rotation tool. You can then move the XY rotation tool into the workspace and rotate molecules by dragging.  |
| Double-click | Double-click on the XY rotation icon to see the Rotate dialog box. Use this dialog box for setting exact rotation conditions (see page 209). You can also see this dialog box by choosing Rotate on the Edit menu. |

## Using the XY Rotation Tool

To use the XY rotation tool:

1. L-click on the XY rotation icon and move the XY rotation tool into the workspace.

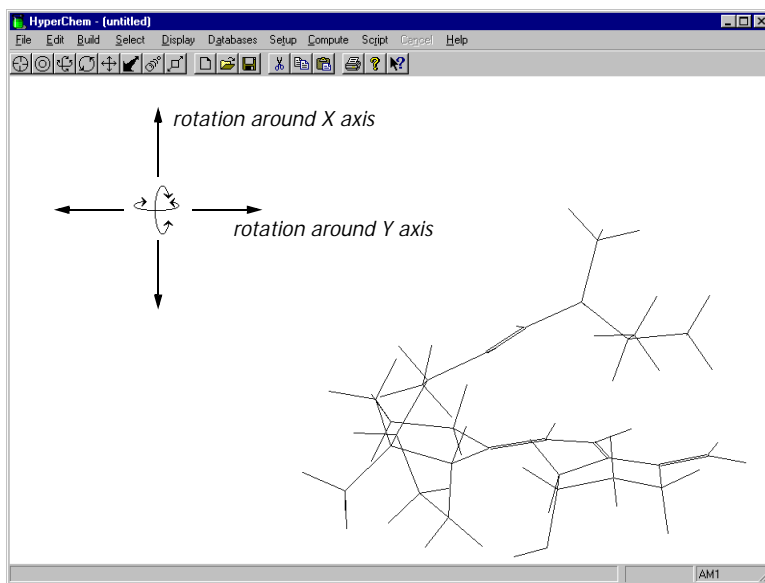
2. L-drag to rotate the molecular system.

This rotation leaves the atomic coordinates unchanged, as if the viewer is moving around the molecules.

3. R-drag to rotate whole, selected molecules.

These molecules rotate around a center of atomic mass, changing individual atomic coordinates.

4. Move the XY rotation tool horizontally to rotate around the y axis, and vertically to rotate around the x axis. Movement in other directions produces rotation around both the x and y axes.



## Setting the Amount of XY Rotation

Normally, moving the XY rotation tool horizontally from one edge of the workspace to the other rotates a molecule by 360 degrees. You can change this amount by entering a number from 0 to 3600 degrees for Tool, XY rotation, in the Tool Preferences dialog box (see page 95). Thirty-six-hundred degrees increases the sensitivity of the rotation tool 10-fold compared to the default of 360 degrees.

## Z Rotation Tool



With the Z rotation tool, you can rotate molecules around the z axis or rotate a selected side chain around a bond. The side chain cannot be part of a ring.

To use the Z rotation tool icon:

- |              |  |
|--------------|--|
| L-click      | L-click on the Z rotation icon to turn the cursor into the Z rotation tool. You can then move the Z rotation tool into the workspace to rotate molecules by dragging.  |
| Double-click | Double-click on the Z rotation icon to see the Rotate dialog box. Use this dialog box for setting exact rotation conditions (see “Rotate Dialog Box” on page 209). You can also see this dialog box by choosing Rotate on the Edit menu. |

### Using the Z Rotation Tool

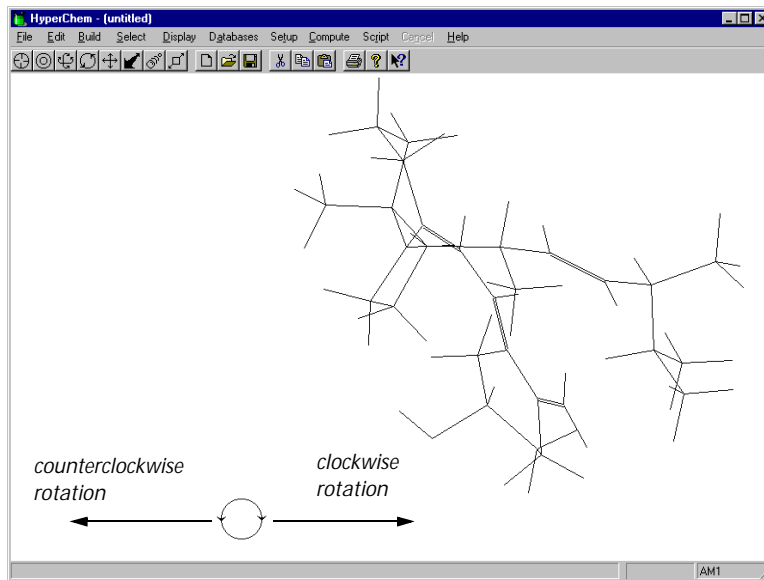
To use the Z rotation tool:

1. L-click on the Z rotation icon and move the Z rotation tool into the workspace.
2. L-drag horizontally to rotate the molecular system.  
This rotation leaves the atomic coordinates unchanged, as if the viewer is moving around the molecules.
3. R-drag horizontally to rotate whole, selected molecules or a selected side chain that is not part of a ring. If there is no selection, this rotates the whole molecular system.

Molecules rotate around their centers of atomic mass. Side chains rotate around the selected bond connecting to the rest of the molecule. Individual atomic coordinates change during this rotation.

**Note:** If you try to rotate part of a molecule that cannot rotate, HyperChem beeps, shows a warning message on the status line, and temporarily inactivates the Z rotation tool until you release the mouse button. Change the selection and try again.





## Setting the Amount of Z Rotation

Normally, moving the Z rotation tool horizontally across the workspace rotates a molecule by 360 degrees. You can change this amount by entering a number from 0 to 3600 degrees for Tool, Z rotation, in the Tool Preferences dialog box (see page 95). Thirty-six-hundred degrees increases the sensitivity of the rotation tool 10-fold compared to the default of 360 degrees.

## XY Translation Tool



The XY translation tool lets you move molecules or selected atoms along the x and y axes (in the plane of the computer screen).

To use the XY translation tool icon:

- L-click**            L-click on the XY translation icon to turn the cursor into the XY translation tool. You can then move the XY translation tool into the workspace and translate molecules by dragging.
- Double-click**    Double-click on the XY translation icon to see the Translate dialog box. Use this dialog box for setting exact translation conditions (see “Translate Dialog

Box” on page 211). You can also see this dialog box by choosing Translate on the Edit menu.

## Using the XY Translation Tool

To use the XY translation tool:

1. L-click on the XY translation icon and move the XY translation tool into the workspace.
2. L-drag to move the molecular system.

This translation leaves the coordinates of all molecules unchanged, as if the viewer is moving around the molecules.

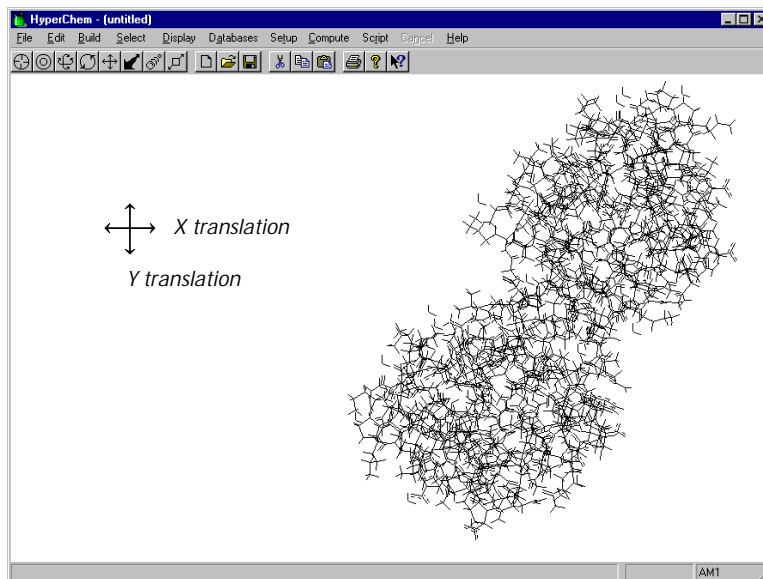
3. R-drag to move selected molecules, residues, and atoms. You can move selected residues or atoms only if you turn off Whole molecule translation (see page 95). The default setting is Whole molecule translation on (✓).

Atomic coordinates change during this translation.

*Caution:* When translating selected residues or atoms in a larger molecule, HyperChem might stretch bonds and angles to accommodate the new atomic positions. These bonds are often unnatural. To avoid distorting bonds, turn Whole molecule translation on (see page 95).

4. Molecules move in the same direction as the XY translation tool.
5. You can drag molecules only to the edges of the workspace.

To increase the relative range for XY translation, use the Zoom tool to reduce the apparent sizes of the molecules. This provides a relatively larger workspace for moving molecules.



## Fast Translation

To translate molecules faster, turn on Fast translation in the Tool Preferences dialog box (see page 95). This choice substitutes a picture of the molecules, called a bitmap. HyperChem can move a bitmap faster because it does not need to compute new atomic coordinates for each step in the translation. When you finish translating and release a mouse button, HyperChem computes the new atomic coordinates and converts the bitmap back to a normal molecular rendering.

## Translation of Annotations

Provided only annotations are selected the XY Translation tool can also be used to move annotations around in the workspace. Select any number of annotations, then L-drag or R-drag the XY Translation tool in the workspace to move annotations. This is the default way to position annotations. If any atoms are selected the translation will apply to them not the selected annotations.

## Z Translation Tool



With the Z translation tool, you can move molecules or selected atoms along the z axis (toward you or away from you). Molecules appear larger and the degree of perspective increases as they move closer to you.

*Note:* Perspective (see “Rendering: Sticks” on page 260) affects Z translation. With Perspective on, molecules change size. With Perspective off, molecules remain the same size, but atoms disappear from the display as you move them out of the clipping slab (see page 47).

To use the Z translation tool icon:

- |              |  |
|--------------|--|
| L-click      | L-click on the Z translation icon to turn the cursor into the Z translation tool. You can then move the Z translation tool into the workspace and L-drag or R-drag to move molecules.  |
| Double-click | Place the cursor on the Z translation icon and double-click to see the Translate dialog box. Use this dialog box for setting exact translation conditions (see “Translate Dialog Box” on page 211). You can also see this dialog box by choosing Translate on the Edit menu. |

*Caution:* If atoms disappear after a z axis translation, you have moved them outside the clipping slab. Use the Z clipping tool to adjust the clipping planes (see page 48) or press [Spacebar], making the atoms visible in the workspace.

## Using the Z Translation Tool

To use the Z translation tool:

1. L-click on the Z translation icon and move the Z translation tool into the workspace.
2. L-drag to translate the molecular system.

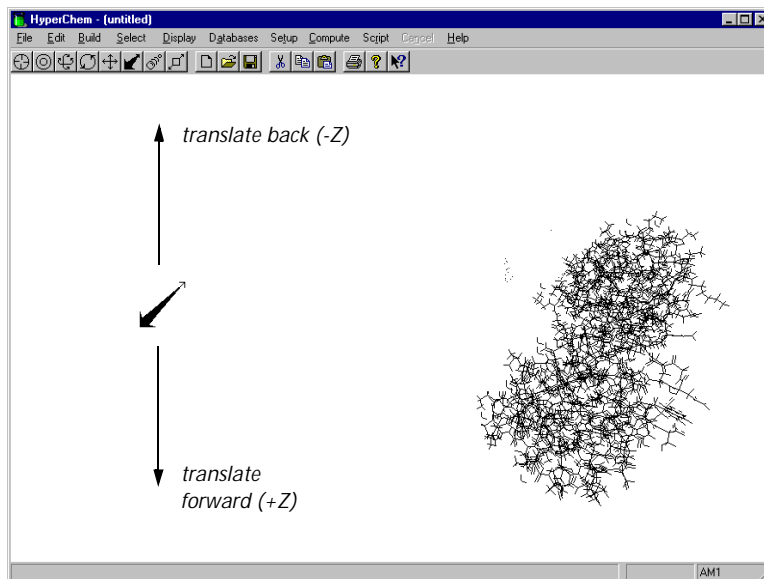
This translation leaves the coordinates of all molecules unchanged, as if you were moving toward or away from the molecules.

3. R-drag to translate selected molecules, residues, and atoms. You can move selected residues or atoms only if you turn off Whole molecule translation (see page 95). The default setting is Whole molecule translation on (✓).

Atomic coordinates change during this translation.

**Caution:** To avoid distorting bonds, turn on Whole molecule translation (see page 95). When translating selected residues or atoms in a larger molecule, HyperChem stretches bonds to accommodate the new atomic positions. These bonds are often unnatural.

4. Move the Translation tool from the top to bottom of the workspace to translate molecules toward you. Move the tool from bottom to top to translate molecules away from you. Moving the tool the vertical height of the workspace moves molecules the default distance of 10 Ångstroms (see “Setting the Amount of Z Translation” on page 45).



In this example, Z translation (back) moved the molecule through the back clipping plane (see page 47), hiding the back of the molecule.

## Setting the Amount of Z Translation

Normally, moving the Z translation tool vertically across the workspace translates molecules by 10 Ångstroms. You can change this amount by entering a number between >0 and ≤1000 in the Tool Preferences dialog box (see page 95).

## Zoom Tool



The Zoom tool lets you change the magnification of the molecular system. For example, you can magnify large molecules to observe atomic organization. It can also be used to enlarge or shrink selected annotations.

To use the Zoom tool icon:

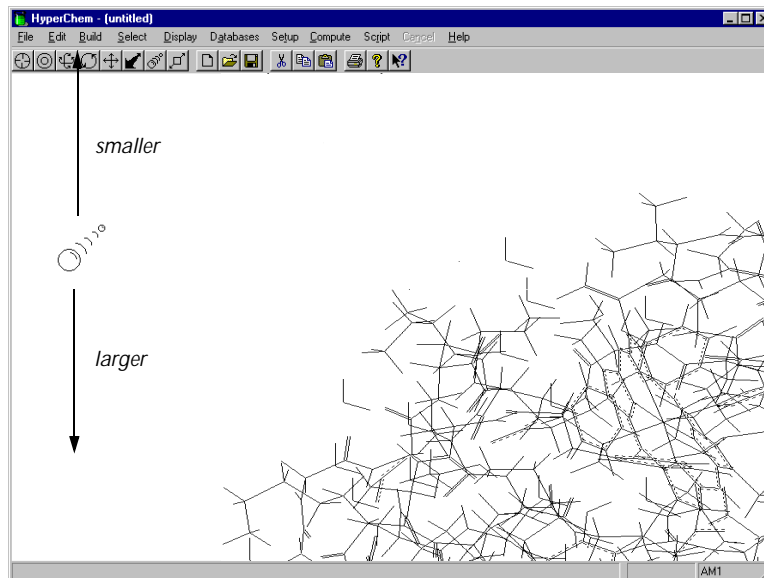
- |              |  |
|--------------|--|
| L-click      | L-click on the Zoom icon to turn the cursor into the Zoom tool. Then move the Zoom tool into the workspace, and L-drag to change the apparent size of the molecular system.  |
| Double-click | Double-click on the Zoom icon to see the Zoom dialog box. Use this dialog box for setting exact zoom conditions (see “Zoom Dialog Box” on page 212). You can also see this dialog box by choosing Zoom on the Edit menu. |

*Caution:* Use a Sticks rendering for bonds with high magnifications of molecular systems. High magnifications (Zoom Factor) of all molecular renderings *except* Sticks might require more memory than your computer has available.

## Using the Zoom Tool

To use the Zoom tool:

1. L-click on the Zoom icon and move the Zoom tool into the workspace.
2. L-drag the Zoom tool vertically, from the top to the bottom of the workspace, to enlarge molecules. L-drag the tool from the bottom to the top to reduce the size of molecules. Moving the tool the vertical height of the workspace magnifies molecules by the default factor of 10 (see below).



**Important:** If you accidentally lose sight of a molecule or increase its size beyond the edges of the workspace, press [Spacebar] to bring it back to the center of the workspace.

The right mouse button has no effect on the Zoom tool.

## Setting the Zoom Sensitivity

Normally, moving the Zoom tool vertically across the workspace changes the magnification by a factor of 10. You can change this amount by entering a number  $>1$  and  $\leq 1000$  in the Tool Preferences dialog box (see page 95).

## Zooming Annotations

Selected annotations can be enlarged or shrunk using the zoom tool. Just select certain annotations by themselves and then drag the annotation tool as with selected atoms and molecules.

## Z Clipping Tool



With the Z clipping tool, you can look inside molecules by showing only the atoms within a “slice.” The slice is called the *clipping slab*.

The clipping slab is present for every molecular system. The clipping slab has front and back planes that are parallel to each other and to the plane of the computer screen. Only atoms between the planes are visible. HyperChem first sets the clipping planes so that you can see a complete molecular system.

With the Z clipping tool, you can move each clipping plane relative to the molecular system; the atomic coordinates are fixed. The status line shows the positions of the front and back clipping planes as distances, in Ångstroms, from the viewer (along the Viewer’s z axis). The viewer is at zero relative to the clipping planes.

The Z clipping tool can show you successive slices through molecules. You can do this two ways:

- Use the Z clipping tool to move the clipping planes through a molecular system.
- Use the Z clipping tool to set the depth of the clipping slab, and then use the Z translation tool to move molecules through the clipping slab.

To use the Z clipping tool icon:

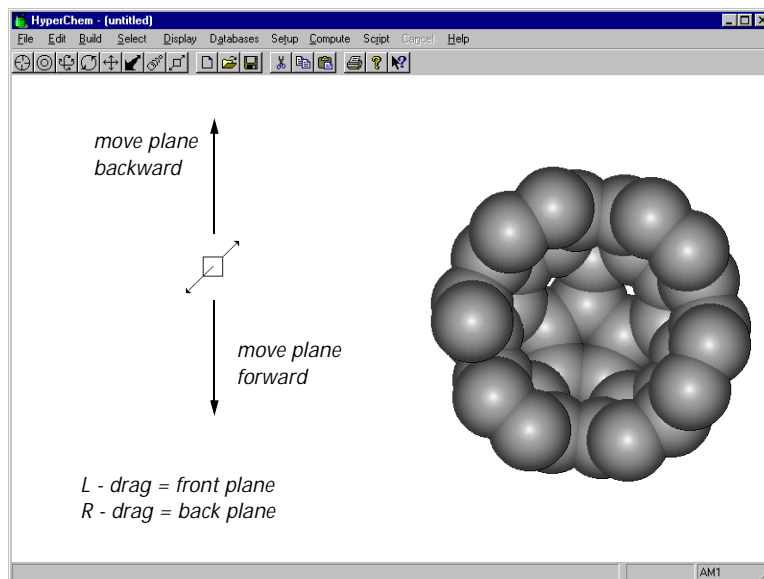
- |                     |  |
|---------------------|--|
| <b>L-click</b>      | L-click on the Z clipping icon to turn the cursor into the Z clipping tool. You can then move the Z clipping tool into the workspace and L-drag or R-drag to adjust the front or back planes of the clipping slab.   |
| <b>Double-click</b> | Double-click on the Z clipping icon to see the Z Clip dialog box (see page 214). Use this dialog box to set exact positions of the clipping planes, either graphically or numerically. You can also see this dialog box by choosing Z Clip on the Edit menu. |



## Using the Z Clipping Tool

To use the Z clipping tool:

1. L-click on the Z clipping icon and move the Z clipping tool into the workspace.
2. L-drag vertically to adjust the position of the front clipping plane.
3. R-drag vertically to adjust the position of the back clipping plane.



4. Dragging from the top to the bottom of the screen moves a clipping plane forward. Dragging from the bottom to the top of the screen moves a clipping plane back. Moving the tool the vertical height of the workspace moves a clipping plane by the default setting of 10 Ångstroms (see the next section).

**Important:** If the molecular system disappears from the workspace while you are using the Z clipping tool, press [Spacebar] to bring it back to the center of the clipping slab.

## Setting the Amount of Clipping Plane Movement

Normally, moving the Z clipping tool vertically across the work-space moves a clipping plane by 10 Ångstroms. You can change this amount by entering a number > 0 and ≤ 1000 in the Tool Preferences dialog box (see page 95).

## Keyboard Equivalents for Tools

You can use keys on the numeric keypad instead of using a mouse to operate HyperChem tools. The keyboard operations apply only to the eight molecular tools and not to the 4 annotation tools. The Key settings in the Tool Preferences dialog box give the size of the change for each keystroke. Pressing a key or a combination of keys gives these results:

	Key	[Shift]+ Key <sup>a</sup>	[Ctrl]+ Key or [Ctrl]+ [Shift]+ Key <sup>b</sup>
[4]	X rotates down	Y translates down	XY translates or rotates selection, if any, down
[3]	X rotates up	Y translates up	XY translates or rotates selection, if any, up
[1]	Y rotates to left	X translates to left	XY translates or rotates selection, if any, left
[2]	Y rotates to right	X translates to right	XY translates or rotates selection, if any, right
[Home]	Z rotates clockwise	Z translates into screen	Z translates or rotates selection, if any
[End]	Z rotates counterclockwise	Z translates out of screen	Z translates or rotates selection, if any

a. [Shift] changes the operation from rotation to translation.

b. Pressing [Ctrl] is like using the right mouse button to move selected atoms. [Ctrl] modifies the effect of either pressing a key or [Shift] plus a key.

## A

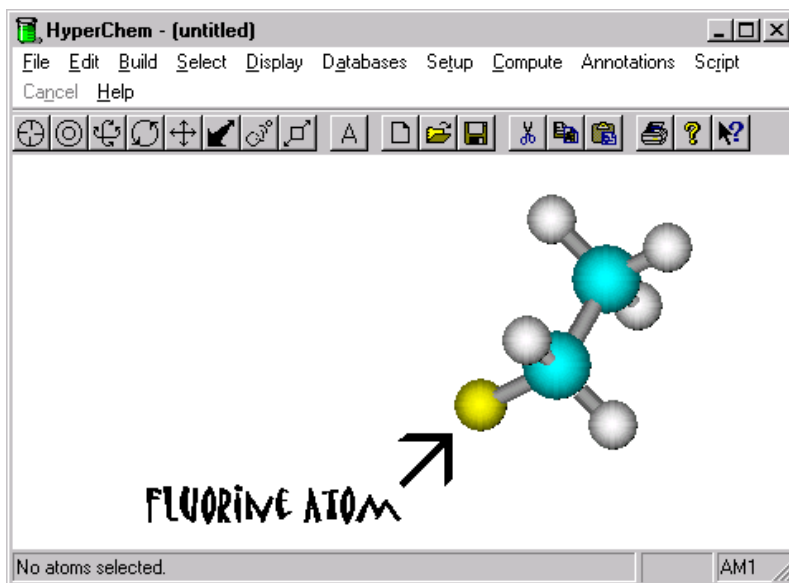
	Key	[Shift] + Key <sup>a</sup>	[Ctrl] + [Shift] + Key
[PgUp]	Zooms smaller	Moves front clipping plane into screen	Moves back clipping plane into screen
[PgDn]	Zooms larger	Moves front clipping plane out of screen	Moves back clipping plane out of screen

a. [Shift] changes the operation from zooming to clipping plane movement.

# Chapter 3

## Tools

HyperChem 6 introduces the concept of *annotations*. As shown in the window below, it is now possible to annotate any molecular system by adding specific pieces of text or “text symbols” to the window. These text elements are referred to as annotations.

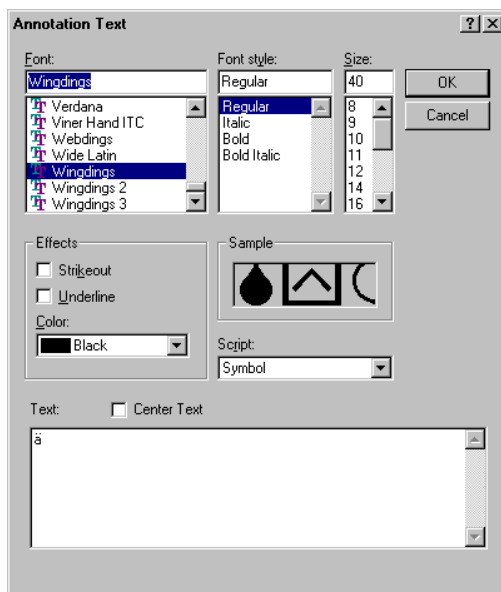


## Annotations are Text

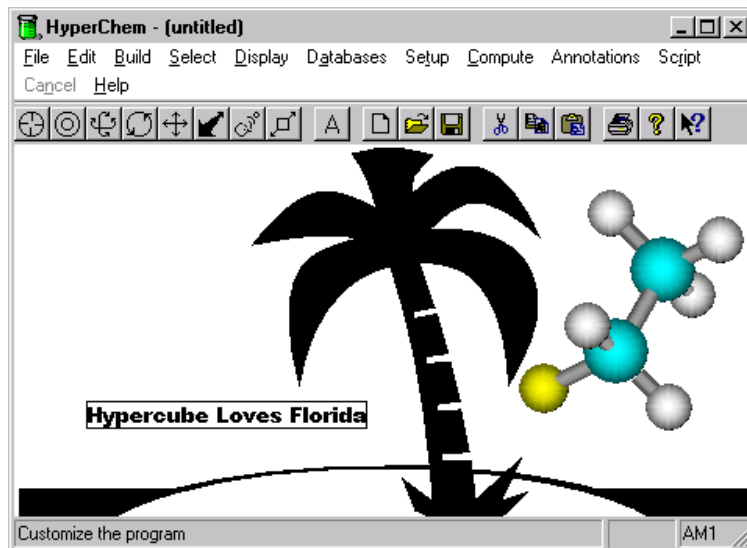
The “Fluorine Atom” annotation above is text that uses a font appropriate to ordinary text, in this case the Matisse ITC font. The arrow, on the other hand is a one-character piece of text that uses a symbol-like font, in this case the Wingdings font. All annotations are just simple text. By using symbol-like fonts, it is possible to have annotations that include a wide variety of symbols such as arrows, lines, circles, rectangles, and other *clip art*.

## Symbols are Text in a Symbol Font

An annotation can have a font, a font style, a size, and a color. The “arrow” above is actually an “a with an umlaut” character (in the Wingdings font) as shown below.

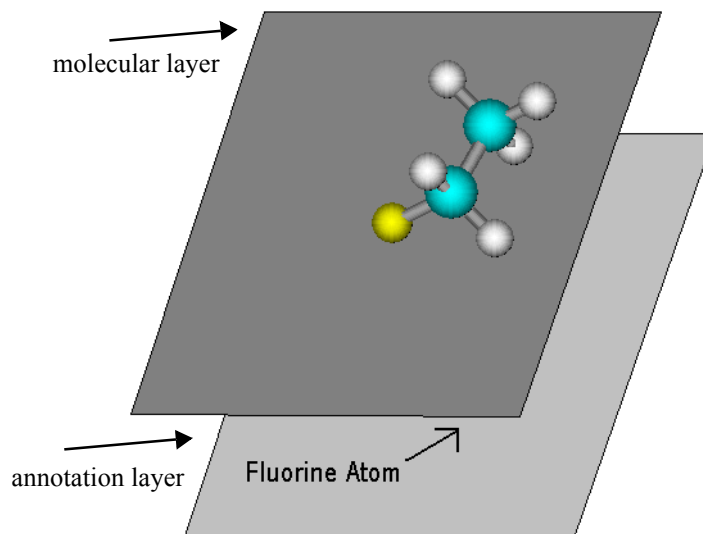


Using various symbol-like fonts such as Symbol, Wingdings, Webdings, etc. with a large font size, it is possible to create a variety of annotation effects such as the following:



## The Annotation Layer

Annotations exist in a discrete layer different from the *molecular layer*. The two layers are completely disjoint and annotations do not affect any of the molecular modeling operations of HyperChem. The annotation layer can even be hidden so as not to provide a visual distraction for molecular modeling. The annotation layer is nominally in front of the molecular layer although it can be pushed back at any time so that it is behind the molecular layer as shown below.

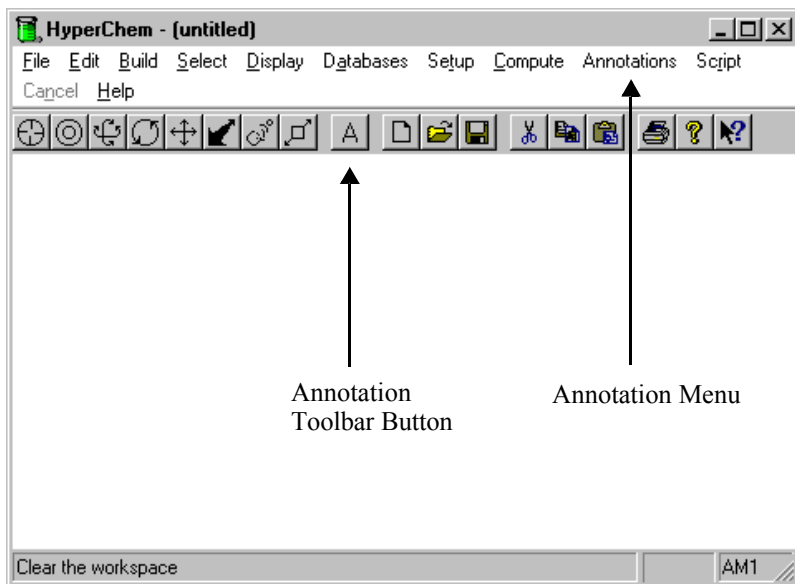


## Transparency in Printing

When printing directly from HyperChem, the molecular layer is only transparent if a sticks rendering is being used. When the molecular layer is transparent, the annotation layer will show through as it always does on the screen. However, when directly printing a solid rendering as illustrated above, the annotation layer will not appear on the printed output. The reason is that the printing of any solid rendering is done as a bitmap for the whole page and obscures any annotations. If a direct printing is to show the annotations, the annotation layer must be in front (except for stick-only renderings of molecules).

## The Annotation Interface

The new annotation layer is controlled by a new toolbar button and a new menu as shown below:



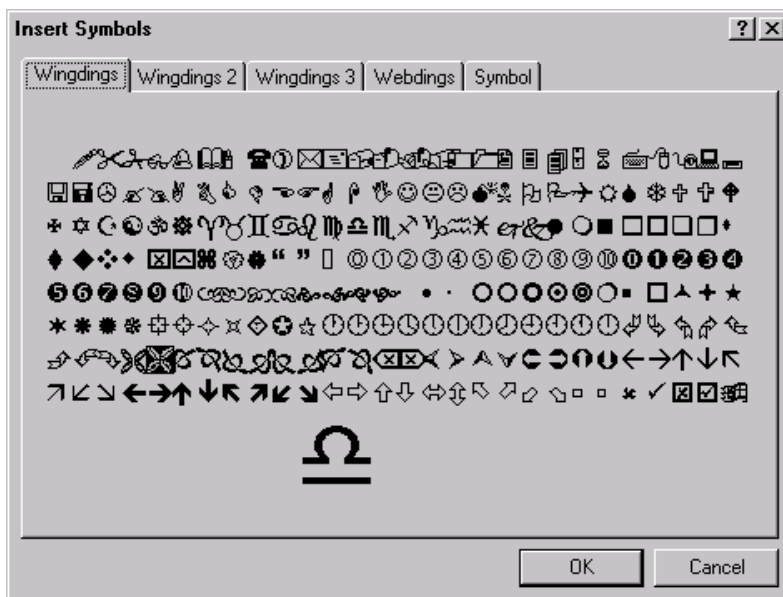
## Creating a Simple Text Annotation

The toolbar button is used to create simple text annotations. You first need to click on the toolbar button to select annotations. Then click anywhere in the workspace to indicate the starting point of a text annotation. The **Annotation Text** dialog box, shown earlier, will appear allowing you to type in the appropriate annotation text in the edit field at the bottom of the dialog box. At this point you can choose the font, size, color, etc. Closing the dialog box will cause the text to appear in the HyperChem workspace.

## Creating a Symbol

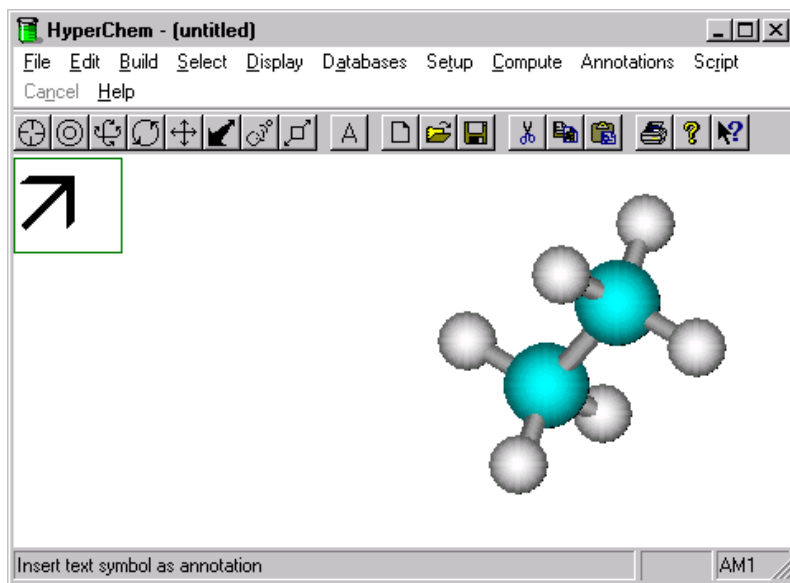
Symbols, although they are just text, are normally created by the Symbol... menu item. This is because the text corresponding to the particular symbol required will generally not be known. Symbols

are identical to any other text but have an alternative way of creating them allows you to visually select the symbol. The <Annotations/Symbol...> menu item will bring up a dialog box that looks as follows:



Clicking on any symbol will render an enlarged view of the symbol at the bottom of the dialog box. When you find the symbol you want, click on OK to add the symbol to the HyperChem workspace. Initially, the symbol will be a standard size and will be placed in the upper left hand corner of the workspace as show below:





The symbol can then be placed where you like with the size, color, etc. that you want. To edit symbols they must first be selected.

## Selecting Annotations

A selected annotation has a bounding box. Symbols are always initially selected as shown above. Selecting and deselecting symbols is done in a manner (almost) identical to the selecting of atoms, residues, and molecules. Thus, one also uses the selection cursor,



to select annotations. Clicking directly on an icon with the selection cursor will select the annotation. The selecting of annotations is always accumulative, independent of the state of Multiple Selections menu item, which is reserved for the molecular plane. Annotations are deselected all at once by clicking with the right mouse button in empty space when selection is active, i.e. a right mouse button deselects all.

Two menu items, <Annotations/Select All> and <Annotations/Select Next> are provided to assist in selecting the desired annotations.

To ensure that one is in a pure molecular modeling mode, no annotation should be selected.

## Moving Annotations

Annotations can be moved or positioned anywhere in the 2D workspace. Unlike molecules in the molecular plane, annotations are strictly two-dimensional. They can only be selected and translated; they cannot be rotated. They are moved (translated) by selecting them and then using the translation cursor,



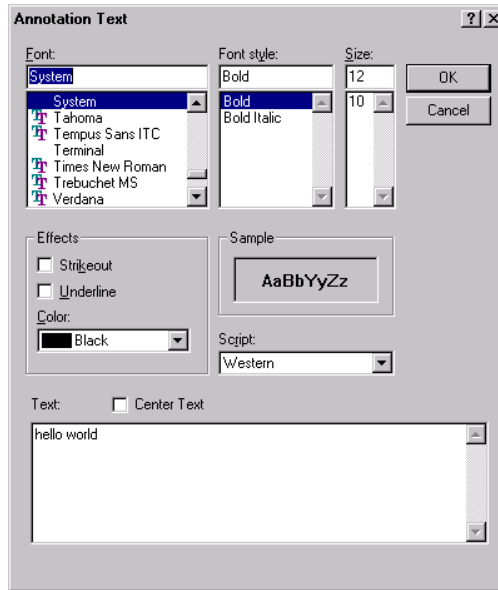
to translate all selected annotations together. If anything is selected in the molecular plane, those translations are given priority and no annotations are moved. To move a specific annotation, only that annotation must be selected of all elements in the workspace. If more than one annotation is selected with no atoms, residues, or molecules selected, then the translation cursor will translate all selected annotations together.

## Editing Annotations

To edit an annotation, the Annotations/Edit Text...> menu item is used. To edit an annotation, only a single annotation (but any atoms, residues, or molecules) can be selected. If this is not the case, the <Annotations/Edit Text...> menu item will be grayed. When the menu item is not grayed then the currently selected annotation can be edited. Whether it is a simple text annotation or a symbol does not affect the editing.

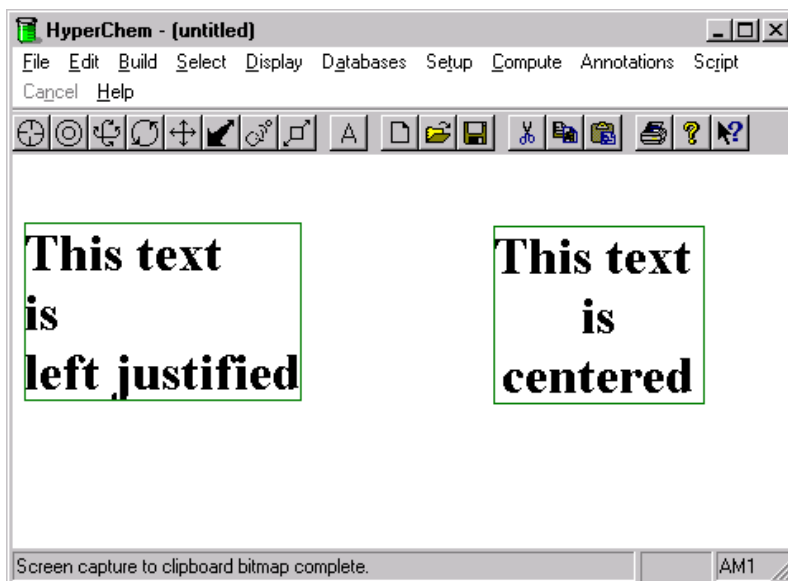
Editing is performed in the same way as the text associated with an Annotation cursor click is first chosen. For symbols, the text that you see in the dialog box may be somewhat cryptic, i.e. a single, perhaps unrecognizable character. You should leave the text as it is if you want one of the selected symbols.

Apart from the actual text, the other relevant properties of an annotation are clearly evident from the Annotation Text dialog box brought up by clicking on the <Annotations/Edit Text...> menu item.



## Multiline Text

An annotation can consist of many text lines if they are typed into the dialog box with carriage returns. Clicking on the <Annotations/Edit Text...> menu item allows you to choose whether this text is left justified or centered in the bounding box.



## Ordering within Annotation Layer

Within the annotation layer a z-ordering is imposed and annotations are drawn from back to front. The later an annotation is created, the more forward in z it is positioned. Sometimes it will require you to modify the z-order of individual annotations. Two menu items, <Annotations/Move Forward> and <Annotations/Move Backward> are provided for this. They allow you to move a single selected annotation either back or forward in individual steps.

## Ordering the Annotation Layer

The ordering within the Annotation layer is separate from the more global ordering of the annotation layer and the molecular layer. This later ordering of the two layers is controlled by the <Annotations/Options...> menu item rather than the <Annotations/Move Forward> and <Annotations/Move Backward> menu items. As indicated above the annotation layer should be in front of the molecular layer to guarantee that it is seen on printing.

There are other times, however, when you may want a backdrop for molecules and the annotation layer is best put to the back.

## **Hiding the Annotation Layer**

To avoid visual distractions and to concentrate on molecular modeling efforts it may be useful to hide all annotations temporarily. This can be accomplished via the <Annotations/Options...> menu item.

# Chapter 4

## Annotations

### Definition

HyperChem contains the concept of an *Annotation*. An annotation is text and/or line drawings used to describe or illustrate a molecular system. Annotations are created by drawing tools that are distinct from the drawing tool used to draw a molecule used in molecular modeling. These annotations are not part of the molecular system *per se*, but are contained in a layer or plane that is distinct from the molecular layer or plane. Thus annotations are completely disjoint from the molecular system. Annotations, although they exist in different, non-interacting layer from the molecular system, can be superimposed with the molecular layer for printing and presentation purposes.

HyperChem thus contains a primitive but useful drawing program that is independent of HyperChem's molecular modeling capabilities. This drawing program creates only annotations. Annotations have their own records in a molecular HIN file and can thus be stored in a file system along with (or without) a molecular system.

### Types of Annotations

Annotations can be one of 4 types:

- Text
- Line
- Ellipse
- Rectangle

## Text

A

A text annotation is simply a piece of text with a particular font , size, etc. Using available Windows fonts, symbols such as arrows, mathematical symbols, greek letters, etc. can be used in annotations. A symbol in HyperChem is a text character with a symbol-like font such as “Wingdings”.

## Lines

+

A line annotation is simply a line drawn with the line drawing tool. The line can be dotted, dashed, etc. and have a width (only solid lines) and a color. A line may have “gravity” so that it is easy to be sure that lines join properly.

A line annotation may sometimes look like part of a “stick” rendering of a molecule, i.e. a line draw with the molecule drawing tool, but it exists in a different plane and is not affected by nor relevant to any molecular modeling operation.

## Ellipses

○

An ellipse can be drawn by an L-drag in the workspace using the ellipse tool. It is centered at the initial position for the L-drag. Ellipses are rectilinear, i.e. the principle axes of the ellipse are parallel to the X (horizontal) and Y (vertical ) axes. A circle may be drawn as a degenerate ellipse with equal X and Y axes.

Ellipses can be either filled or not filled (outline only). The fill and the outline of an ellipse can be any color but must be the same

color. Menu items and dialog boxes are available for choosing the color and fill option for a newly drawn ellipse or an existing one that is currently selected.

## Rectangles



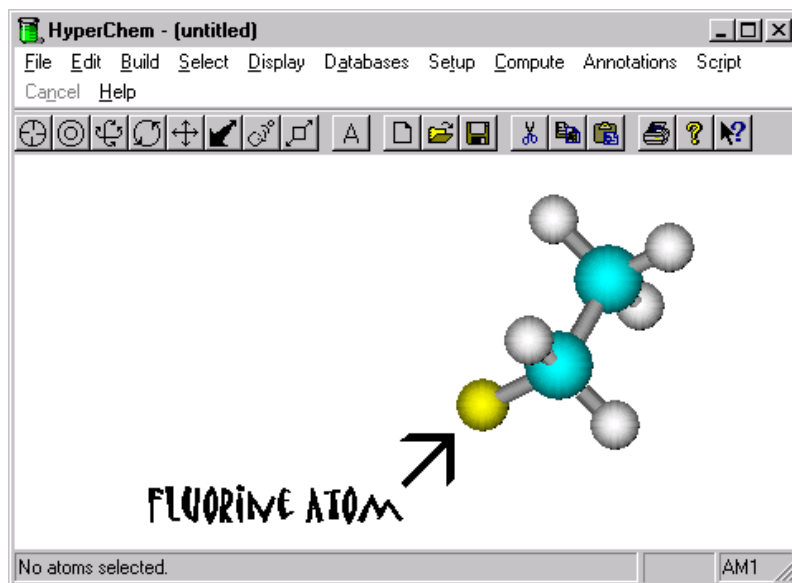
A rectangle can be drawn by an L-drag in the workspace using the rectangle tool. Its upper left hand corner is at the initial position for the L-drag. Rectangles are rectilinear, i.e. the principle axes of the rectangle are parallel to the X (horizontal) and Y (vertical) axes. A square may be drawn as a degenerate ellipse with equal X and Y axes.

Rectangles can be either filled or not filled (outline only). The fill and the outline of an ellipse can be any color but must be the same color. Menu items and dialog boxes are available for choosing the color and fill option for a newly drawn rectangle or an existing one that is currently selected.

## An Example

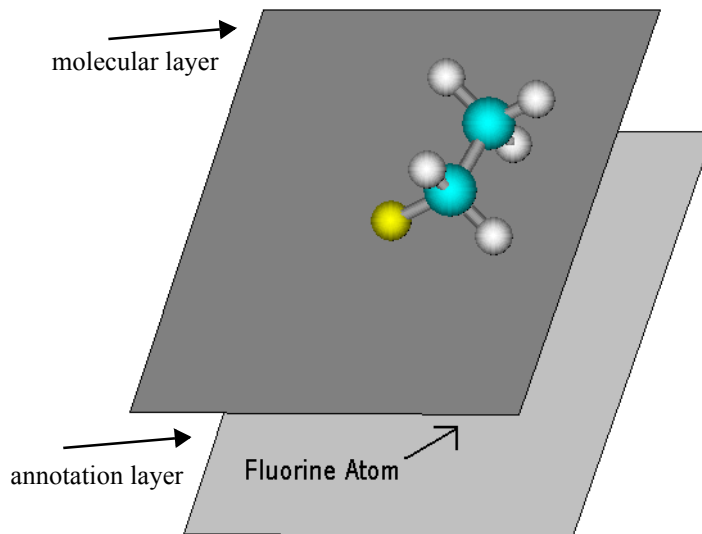
As shown in the window below, it is possible to annotate any molecular system by adding specific pieces of text or “text symbols” to the window. These text elements are referred to as text annotations.





## The Annotation Layer

Annotations exist in a discrete layer different from the *molecular layer*. The two layers are completely disjoint and annotations do not affect any of the molecular modeling operations of HyperChem. The annotation layer can even be hidden so as not to provide a visual distraction for molecular modeling. The annotation layer is nominally in front of the molecular layer although it can be pushed back at any time so that it is behind the molecular layer as shown below.

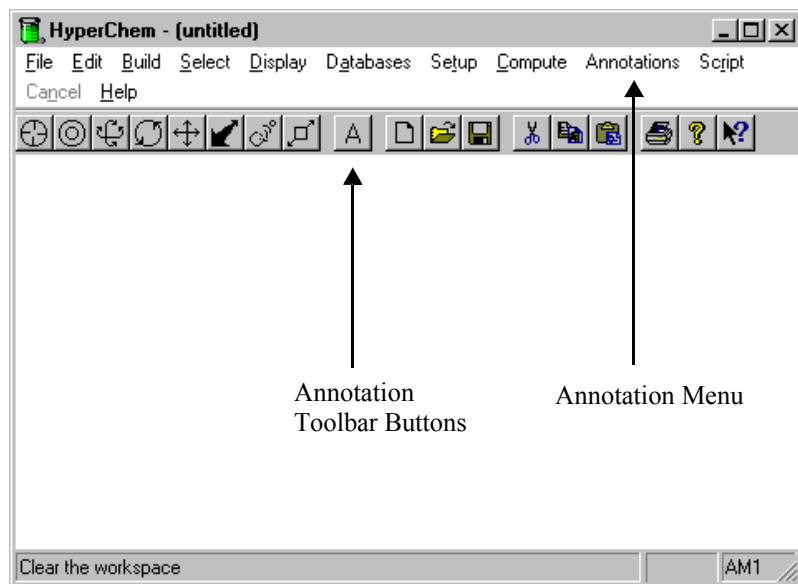


## Transparency in Printing

When printing directly from HyperChem, the molecular layer is only transparent if a sticks rendering is being used. When the molecular layer is transparent, the annotation layer will show through as it always does on the screen. However, when directly printing a solid rendering as illustrated above, the annotation layer will not appear on the printed output. The reason is that the printing of any solid rendering is done as a bitmap for the whole page and obscures any annotations. If a direct printing is to show the annotations, the annotation layer must be in front (except for stick-only renderings of molecules).

## The Annotation Interface

The annotation layer is controlled by the four annotation toolbar buttons and a menu as shown below:



## Text Annotations

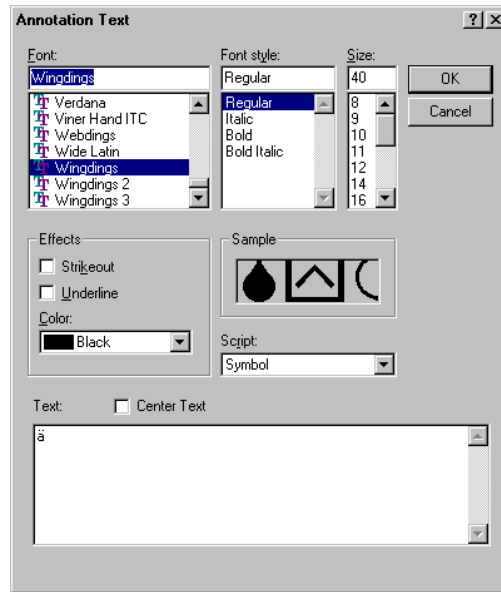
Of the four types of annotations, text annotations behave somewhat differently from the drawing primitives for lines, ellipses, and rectangles.

### Annotations as Text

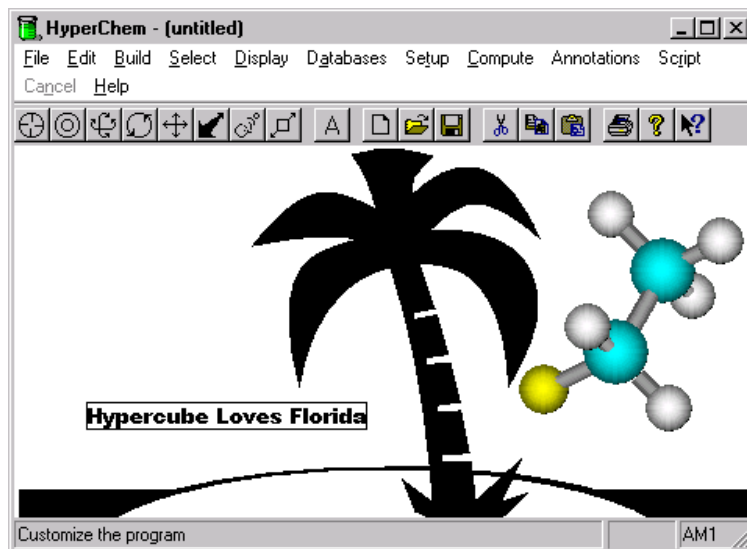
The "Fluorine Atom" annotation in the example above is text that uses a font appropriate to ordinary text, in this case the Matisse ITC font. The arrow, on the other hand is a one-character piece of text that uses a symbol-like font, in this case the Wingdings font. All text annotations are just simple text. By using symbol-like fonts, it is possible to have annotations that include a wide variety of symbols such as arrows, lines, circles, rectangles, and other *clip art*.

## Symbols are Text in a Symbol Font

A text annotation can have a font, a font style, a size, and a color. The “arrow” above is actually an “a with an umlaut” character (in the Wingdings font) as shown below.



Using various symbol-like fonts such as Symbol, Wingdings, Webdings, etc. with a large font size, it is possible to create a variety of annotation effects such as the following:

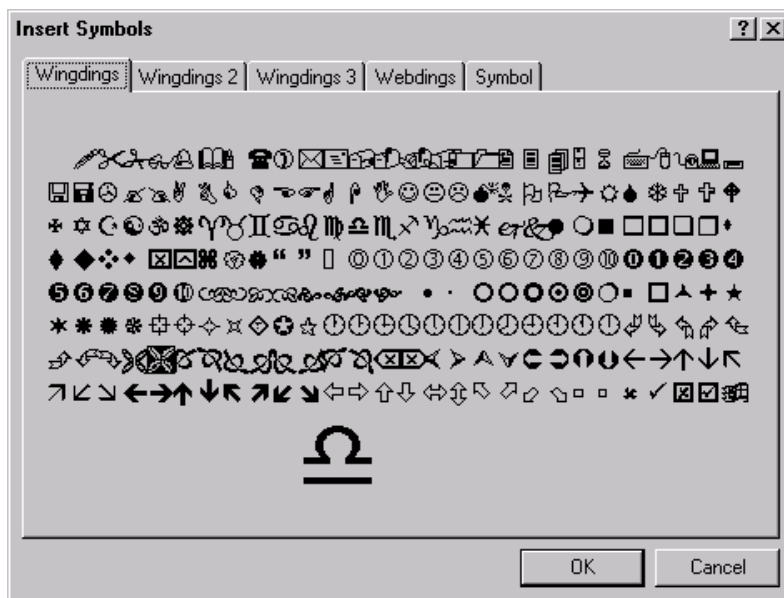


## Creating a Simple Text Annotation

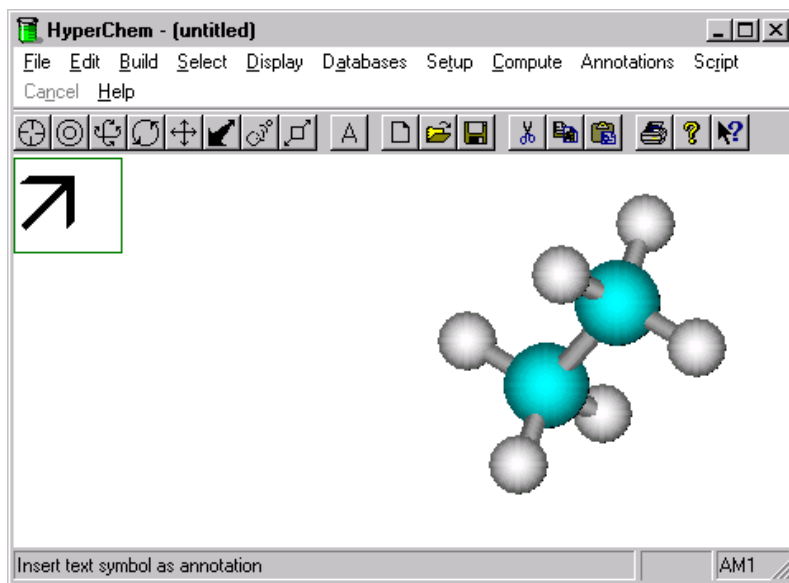
The text annotation toolbar button is used to create simple text annotations. You first need to click on the toolbar button to select annotations. Then click anywhere in the workspace to indicate the starting point of a text annotation. The **Annotation Text** dialog box, shown earlier, will appear allowing you to type in the appropriate annotation text in the edit field at the bottom of the dialog box. At this point you can choose the font, size, color, etc. Closing the dialog box will cause the text to appear in the HyperChem workspace.

## Creating a Symbol

Symbols, although they are just text, are normally created by the Symbol... menu item. This is because the text corresponding to the particular symbol required will generally not be known. Symbols are identical to any other text but have an alternative way of creating them allows you to visually select the symbol. The <Annotations/Symbol...> menu item will bring up a dialog box that looks as follows:



Clicking on any symbol will render an enlarged view of the symbol at the bottom of the dialog box. When you find the symbol you want, click on OK to add the symbol to the HyperChem workspace. Initially, the symbol will be a standard size and will be placed in the upper left hand corner of the workspace as show below:



The symbol can then be placed where you like with the size, color, etc. that you want. To edit symbols they must first be selected.

## Selecting Text Annotations

A selected text annotation has a bounding box. Symbols are always initially selected as shown above. Selecting and deselecting symbols is done in a manner (almost) identical to the selecting of atoms, residues, and molecules. Thus, one also uses the selection cursor,



to select annotations. Clicking directly on an annotation with the selection cursor will select the annotation. The selecting of annotations is always accumulative, independent of the state of Multiple Selections menu item, which is reserved for the molecular plane. Annotations are deselected all at once by clicking with the

right mouse button in empty space when selection is active, i.e. a right mouse button deselects all.

Three menu items, <Annotations/Select All> , <Annotations/Select Next> and <Annotations/Select Previous> are provided to assist in selecting the desired annotations.

To ensure that one is in a pure molecular modeling mode, no annotation should be selected.

## Moving Text Annotations

Annotations can be moved or positioned anywhere in the 2D workspace. Unlike molecules in the molecular plane, annotations are strictly two-dimensional. They can only be selected and translated; they cannot be rotated. They are moved (translated) by selecting them and then using the translation cursor,



to translate all selected annotations together. If anything is selected in the molecular plane, those translations are given priority and no annotations are moved. To move a specific annotation, only that annotation must be selected of all elements in the workspace. If more than one annotation is selected with no atoms, residues, or molecules selected, then the translation cursor will translate all selected annotations together.

## Editing Text Annotations

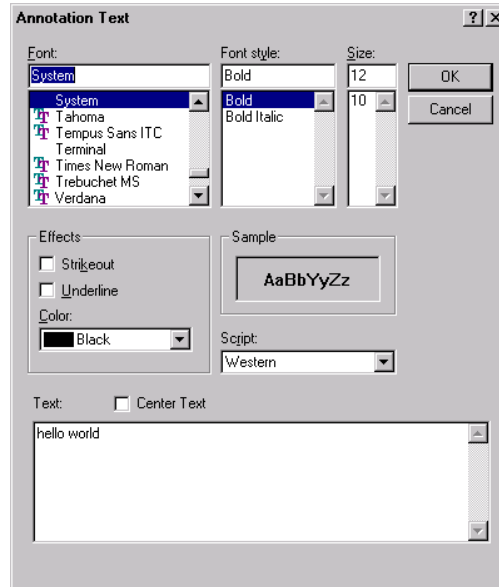
To edit a text annotation, the <Annotations/Edit Text...> menu item is used. To edit an annotation, only a single text annotation (but any atoms, residues, or molecules) can be selected. If this is not the case, the <Annotations/Edit Text...> menu item will be grayed. When the menu item is not grayed then the currently selected text annotation can be edited. Whether it is a simple text annotation or a symbol does not affect the editing.

Editing is performed in the same way as the text associated with an Annotation cursor click is first chosen. For symbols, the text that you see in the dialog box may be somewhat cryptic, i.e. a single,



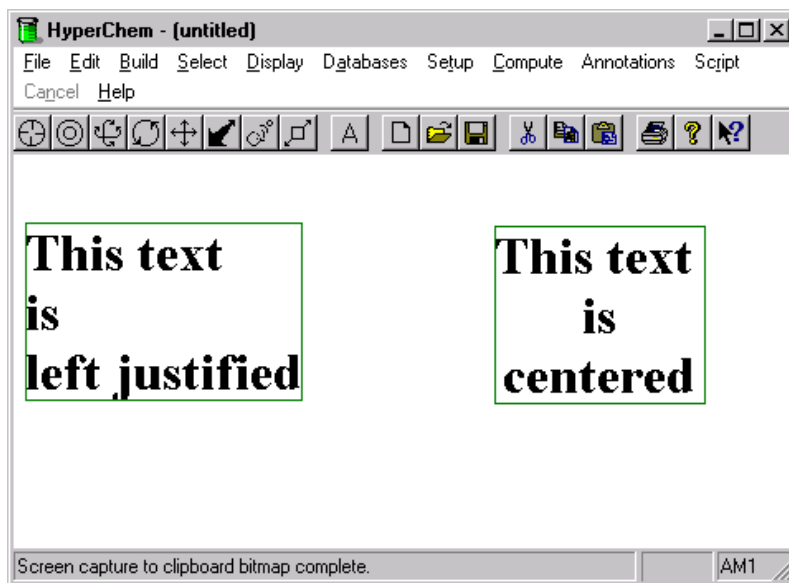
perhaps unrecognizable character. You should leave the text as it is if you want one of the selected symbols.

Apart from the actual text, the other relevant properties of an annotation are clearly evident from the Annotation Text dialog box brought up by clicking on the <Annotations/Edit Text...> menu item.



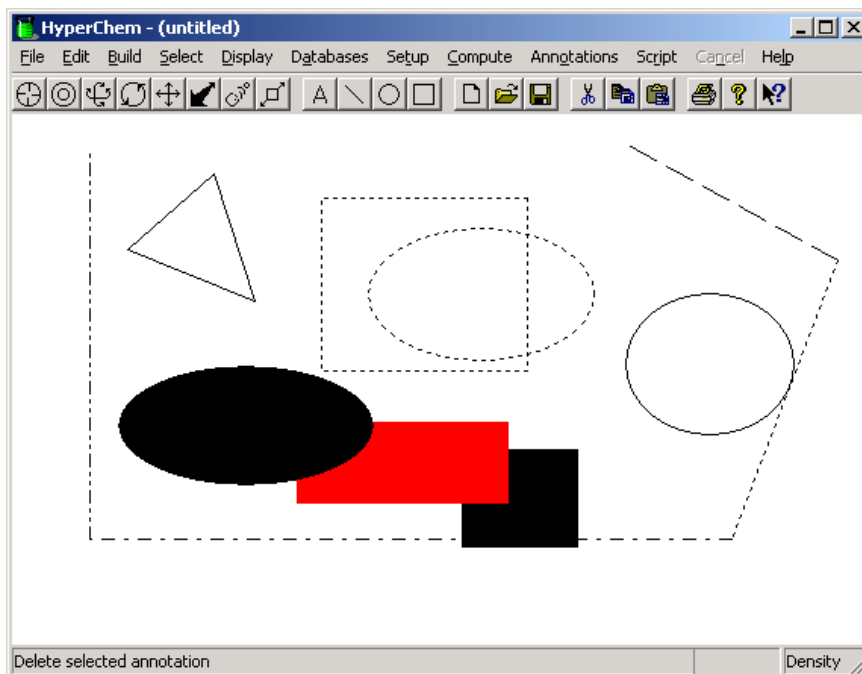
## Multiline Text

An annotation can consist of many text lines if they are typed into the dialog box with carriage returns. Clicking on the <Annotations/Edit Text...> menu item allows you to choose whether this text is left justified or centered in the bounding box.



## Line, Ellipse, and Rectangle Annotations

Unlike Text Annotations, line, ellipse, and rectangle annotations are simple drawing elements as per a simple drawing program. To draw one simply, chooses the correct cursor by clicking on the appropriate tool of the toolbar and then draws lines, ellipses, or rectangles with the tool, as shown below:

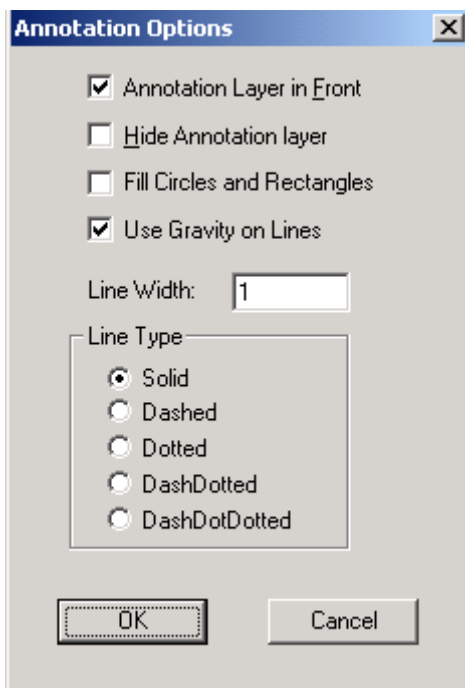


## Options

Annotations have various options available. A variety of types of lines can be drawn although only solid lines can be thick. Ellipses and rectangles can be filled or not.

To make it convenient for lines to join, gravity is an option.

The annotation options dialog box is shown below. The drawing program in HyperChem is primitive but allows one to quickly prepare appropriate annotations for molecule-oriented transparencies and slides. For more elegant presentations, bitmaps or line drawings of molecules can be copied from HyperChem into more professional drawing programs such as those from Adobe, Corel, Microsoft, etc.



## Ordering Annotations within the Annotation Layer

Within the annotation layer a z-ordering is imposed and annotations are drawn from back to front. The later an annotation is created, the more forward in z it is positioned. Sometimes it will require you to modify the z-order of individual annotations. Two menu items, <Annotations/Move Forward> and <Annotations/Move Backward> are provided for this. They allow you to move a single selected annotation either back or forward in individual steps.

## Ordering the Annotation Layer

The ordering within the Annotation layer is separate from the more global ordering of the annotation layer and the molecular layer. This later ordering of the two layers is controlled by the <Annotations/Options...> menu item rather than the <Annotations/Move Forward> and <Annotations/Move Backward> menu

items. As indicated above the annotation layer should be in front of the molecular layer to guarantee that it is seen on printing. There are other times, however, when you may want a backdrop for molecules and the annotation layer is best put to the back.

## **Hiding the Annotation Layer**

To avoid visual distractions and to concentrate on molecular modeling efforts it may be useful to hide all annotations temporarily. This can be accomplished via the <Annotations/Options...> menu item.

# Chapter 5

## Managing Files

This chapter describes the commands on the File, Script, and Help menus. It explains opening and saving files, setting up HyperChem display conditions, running script files to automate HyperChem functions, and obtaining help while using HyperChem.

For accessing files, HyperChem uses the standard Windows File dialog boxes. Click on Help in the Start button for information about specifying drives, directories and files with Windows File dialog boxes.

### File Menu

The File menu contains these file management and utility commands:

<b>New</b>	Clears the workspace for drawing new molecules. You can also press [Ctrl]+[N]
<b>Open . . .</b>	Clears the workspace and shows a molecular system stored in a file. You can also press [Ctrl]+[O]
<b>Merge . . .</b>	Adds the contents of a file (molecular system) to the molecules in the workspace
<b>Save</b>	Stores the molecules now in the workspace to a file. You can also press [Ctrl]+[S]
<b>Save As . . .</b>	Stores the molecules now in the workspace in a file, with a new filename. You can also press [Ctrl]+[A]
<b>Save as HTML...</b>	Create dynamic content for the web from HyperChem.

<b>Start Log . . .</b> <sup>a</sup>	Stores the results of chemical calculations in a log file
<b>Stop Log</b> <sup>a</sup>	Turns off the log for recording calculation results in a log file
<b>Log Comments . . .</b> <sup>a</sup>	Adds comments to a log file
<b>Import . . .</b>	Import orbital data
<b>Export . . .</b>	Export orbital data
<b>Print . . .</b>	Sends a copy of the molecules in the workspace to a printer
<b>Preferences . . .</b>	Sets HyperChem operating conditions. HyperChem remembers these settings from one session to the next
<b>Exit</b>	Ends your session with HyperChem. You can also press [Alt]+[F4]

a. When this item is gray, it is temporarily unavailable. You must change some condition in HyperChem to make it available. You must start a log before you can stop it or log comments, and you must stop a log before you can start a new one.

## New

New clears the workspace and prepares HyperChem for creating new molecules. This option asks if you want to save the molecules present in the workspace if you made changes to the molecular system.

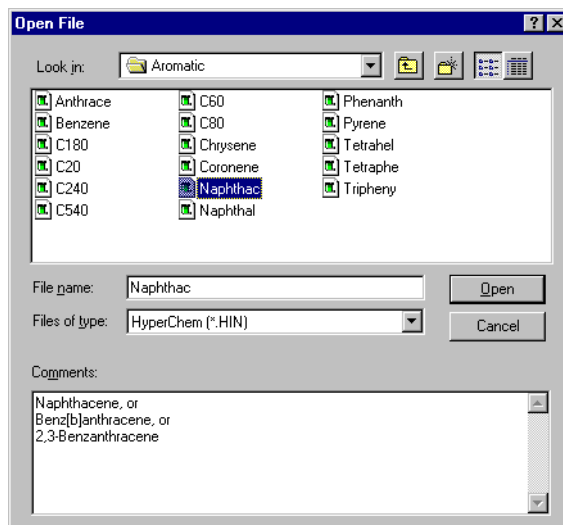
## Open

Open clears the workspace and lists the molecules stored in files. The Open File dialog box lets you choose a filename and copy the molecules from this file to the workspace.

Before you can see the contents of a file, HyperChem asks if you want to save the molecular system presently in the workspace if you made changes since last saving it.

## Open File Dialog Box

Use the Open File dialog box to copy a molecular system from a file to the workspace. The dialog box provides several ways to specify a filename.



- |                          |  |
|--------------------------|--|
| <b>File Name</b>         | You can enter the filename, with or without path, in this text box and L-click on Open. If you choose a file name from the file list, the name will appear in this box.            |
| <b>Directories</b>       | This reports the current directory. The files appearing in the file list are in this directory. The directory list box can be used to change the current directory.                |
| <b>Drives</b>            | This reports the current drive. The files appearing in the file list are on this drive in the current directory. The drives list box can be used to change the current disk drive. |
| <b>File Type</b>         | Use this list to choose the file type and the extension filter for listing file names. The standard file types that HyperChem can read and write are:                              |
| <b>HyperChem (*.HIN)</b> | Files containing complete information about molecules created with HyperChem (see page 711). These filenames have the  |



	<i>.hin</i> extension. HIN stands for HyperChem INput file.
Brookhaven PDB (*.ENT)	Files stored in the format used by the Brookhaven Protein Data Bank (see page 503). Before showing you these chemical structures, HyperChem uses information in the chem.tpl file to add hydrogen atoms and bonding when necessary.
ISIS Sketch (*.SKC)	The Sketch file format used by ISIS/Draw from MDL Information Systems. These filenames have the <i>.skc</i> extension.
Cartesian Coords (*.XYZ)	The Cartesian file format consists of simple records consisting only of atomic number and x, y, and z coordinates for the easiest possible input. These filenames have the <i>.xyz</i> extension.
MDL MOL (*.MOL)	The format used by many programs from MDL Information Systems. These filenames have the <i>.mol</i> extension.
MOPAC Z-Matrix (*.ZMT)	The molecular definition portion of a MOPAC file, with geometry given by internal coordinates. HyperChem expects files in this format to have the <i>.zmt</i> extension.
Tripos MOL2 (*.ML2)	The format used by many programs from Tripos Associates. HyperChem expects files in this format to have the <i>.ml2</i> extension.
ChemDraw (.CHM)	The format used by ChemDraw from CambridgeSoft. Files in this format usually have the <i>.chm</i> extension.

**Caution:** If you choose an inappropriate file format, for example HIN instead of PDB, and then open a file, HyperChem might not succeed in reading the file. HyperChem does not automatically detect the file format that you are reading.

**Comments** This box shows any comments stored in the file currently selected, which might describe the molecular system and its source. You can scroll through the comments in this box. For a PDB file,

only some of the comments appear. You cannot enter new comments here. Use *Save As* on the *Files* menu to add new comments. Comments are not available from Z-Matrix format files.

OK L-click to copy the contents of a file, appearing in the File Name text box, to the workspace. You can also double-click on the filename in the file list.

## Reading Non-standard PDB Files

PDB files written by some other programs may not conform strictly to the Brookhaven Protein Databank specifications. In particular, PDB files written by some other programs may have left-justified atom names. In order to read these files in, you may try running a script containing the command “non-standard-pdb-names = true”. HyperChem will then try interpreting the PDB file according to a non-standard form. You should reset HyperChem by running a script with the command “non-standard-pdb-names = false” in order to re-enable reading of standard PDB files.

## Reading ChemDraw Files

Since ChemDraw is a drawing program and not a modelling program, it handles some structural elements in ways that must be corrected manually in order that they be read properly by HyperChem.

When an “aromatic” ring is drawn in ChemDraw, the result is a circle — a picture element — attached to a structure with single bonds. When a ChemDraw file (type .CHM) is read by HyperChem, it ignores the picture elements and processes only the chemical structures, so an “aromatic” ring shows up with single bonds. To have the ring brought into HyperChem drawn with aromatic bonds, you must change the bond type for the ring bonds to “aromatic” in ChemDraw by selecting the ring and using the *Object/Bond Properties* menu item. Alternately, you can change the bonding to aromatic after the structure is imported to HyperChem by double-clicking on a bond.

## Using the Z-Matrix File Filter

You should note that the algorithm for reading and writing Z-matrix files involves a step proportional to the square of the number of atoms present, which takes a long time for large molecules.

The algorithm for inferring connectivity and bond order from a Z-matrix file is heuristic, and may not always give the bond orders you want. You should check the connectivities and bond-orders yourself before carrying out operations that require connectivity information, such as model building or molecular mechanics calculations.

Saving a file in Z-Matrix format and then reading it back in to HyperChem is a way of automatically generating bonds between unconnected atoms. This can be useful for processing structures imported from other sources. Also, a structure which is imported from a Z-Matrix file is reoriented such that the first atom in the file is placed at the origin, the second atom on the positive x axis, and the third atom in the xy plane. This can be useful for such operations as vibrational analysis.

## Merge

Merge adds molecules from a file to the molecular system in the workspace. Merge is useful for comparing or “docking” molecules. Docking involves positioning molecules to simulate possible interactions, for example, enzyme-substrate interactions.

The added molecules appear a few Ångstroms away from the existing molecular system. (You can use a script command to prevent this movement. See the script command `translate-merged-systems`.)

When you choose Merge, the Open File dialog box (see page 79) appears for you to supply the name of a file to merge. HyperChem retains the first file name as the current file name.

You can repeat Merge, adding several files to the workspace.

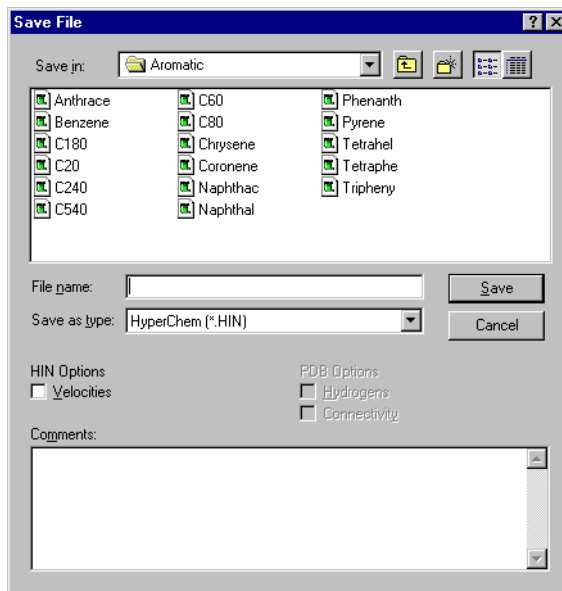
## Save

Save copies the molecular system from the workspace to a file. If this is a new file, you enter a filename in the Save File dialog box that appears (see page 83). If the file already has a name, Save stores the molecular system in this file without showing you the Save File dialog box, replacing the last version of the molecular system. The filename has a limit of eight characters plus a three-character extension.

## Save As

Save As stores the molecular system in a file. This option is the same as Save, except that the Save File dialog box always appears. Use Save As to copy a file or to store new versions of a molecular system in different files.

### Save File Dialog Box



**File Name** Enter the path and name of the file in this text box using any combination of upper- and lowercase letters. If you choose a file name from the file list, the name will appear in this box.

*Note:* You must replace the filename wild card (\*) in the File Name text box.

**Directories** This reports the current directory. The files appearing in the file list are in this directory. The directory list box can be used to change the current directory.

**Drives** This reports the current drive. The files appearing in the file list are on this drive in the current direc-

tory. The drives list box can be used to change the current disk drive.

<b>File Type</b>	Use this list to choose the file type and the extension filter for listing file names. The standard file types that HyperChem can write are:
<b>HyperChem (*.HIN)</b>	Files containing complete information about molecules created with HyperChem (see page 711). These filenames have the <i>.hin</i> extension. HIN stands for HyperChem INput file.
<b>Brookhaven PDB (*.ENT)</b>	Files stored in the format for the Brookhaven Protein Data Bank (see page 503). Before showing you these chemical structures, HyperChem uses information in the <i>chem.tpl</i> file to add hydrogen atoms and bonding when necessary.
<b>ISIS Sketch (*.SKC)</b>	The Sketch file format used by ISIS/Draw from MDL Information Systems. These filenames have the <i>.skc</i> extension.
<b>Cartesian Coords (*.XYZ)</b>	This Cartesian coordinate file consists only of records with atomic number and x, y, and z coordinates. It is created when the simplest possible molecule file is required. These filenames have the <i>.skc</i> extension.
<b>MDL MOL (*.MOL)</b>	The format used by many programs from MDL Information Systems. These filenames have the <i>.mol</i> extension.
<b>MOPAC Z-Matrix (*.ZMT)</b>	The molecular definition portion of a MOPAC file, with geometry given by internal coordinates. HyperChem expects files in this format to have the <i>.zmt</i> extension.
<b>Tripos MOL2 (*.ML2)</b>	The format used by many programs from Tripos Associates. HyperChem expects files in this format to have the <i>.ml2</i> extension.
<b>HIN Options</b>	This area becomes active when File Type is set to HyperChem HIN format. For a complete description of a HIN file, see page 711.

Velocities	Check this box to store atomic velocities calculated in a molecular dynamics simulation. This lets HyperChem start another molecular dynamics simulation where the last one ended.
PDB Options	This area becomes active when File Type is set to Brookhaven PDB format. PDB files have the extension <i>.ent</i> (see page 503 for further information about PDB files).
Hydrogens	Check this box to add coordinates for hydrogen atoms to the file. Most PDB files do not have coordinates for hydrogen atoms.
Connectivity	Check this box to add information on bonding. HyperChem creates CONECT records in the file. PDB files normally use CONECT records only for nonstandard residues (see “Protein Data Bank Files” on page 503). If you use PDB files to store molecules <i>not</i> composed of standard residues, turn on this option.
Comments	This text box is only active if File Type is set to HyperChem HIN format or Brookhaven PDB format. Enter notes here about the molecular system. When you have clicked in this text box you see a flashing vertical bar, the insertion point for text, and a standard I-shaped text cursor. Text you enter appears to the left of the insertion point.
OK	L-click to copy the system in the workspace to a file, using the name appearing in the File Name text box. You can also double-click on the filename in the file. list.

## Editing Comments

To add or change text, remember these rules:

- L-click with the text cursor (I-shaped) anywhere in the text to move the insertion point.
- L-drag the text cursor to select text that you want to delete or replace.

- To delete text, select it and press [Del] or [Backspace].
- To replace text, select it and enter new text.

You can use these keys for editing:

[↵]	Moves the insertion point to a new line.
[Backspace]	Erases the character (or selected text) to the left of the insertion point (text cursor).
[Del]	Erases the character (or selected text) to the right of the insertion point (text cursor).
[⌘] [A]	Moves the insertion point (text cursor).
[⌘] [Z]	Moves the insertion point (text cursor).

## Save As HTML...

HyperChem can directly create dynamic content for the web. If you have created a molecule in HyperChem or calculated molecular orbitals, vibrational spectra, or electronic spectra, then HTML files that display dynamic versions of these can be written out for immediate inclusion into any web site. They are dynamic in that they are not simply static pictures of molecules, orbitals, etc. but the molecules can be rotated, the orbitals selected, the vibrational modes animated, and so on.

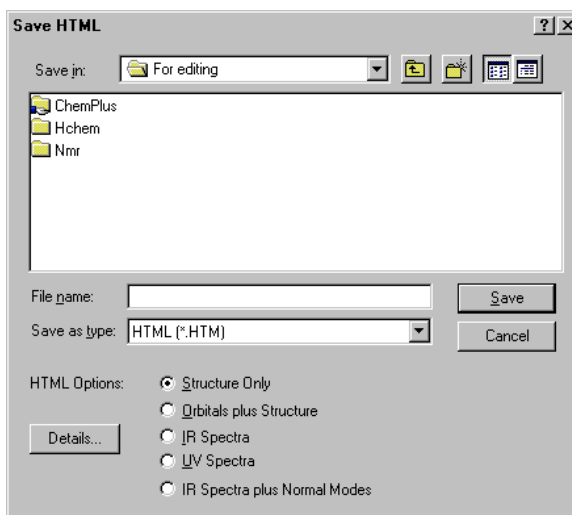
The web pages are fully dynamic but require a user to have a free Netscape plug-in installed on their machine in order to view the content. This free plug-in is included with HyperChem for distribution to people who wish to view your web pages or it may be downloaded at no cost from the Hypercube web site, <http://www.hyper.com>.

## Content of HTML Page

At this stage (the automatic creation of HTML pages is a quickly evolving option), 5 options are available:

- Structure only
- Orbitals plus Structure
- IR spectra
- UV spectra

- IR spectra plus normal modes



If you select *Structure Only* then a web page will be created that displays a molecule in a variety of selectable renderings such as ball and cylinder, overlapping spheres, etc. The structure of the molecule can be explored by shift-clicking on the atoms of a bond, torsion, etc. to ask for the value of the bond length, torsion angle, etc. If the molecule has residue structure, it is also possible to ask to see the ribbon structure of the backbone. The created dynamic content uses new DirectX technology to display the molecule, ribbons, etc. and the rendering may be rotated, translated, zoomed, etc. at high speed since they are rendered locally using the plug-in.

Selecting *Orbitals plus Structure* will create a web page that includes the ability to select, display, and manipulate all the molecular orbitals of the molecule.

Selecting *IR Spectra* will display a plot of the calculated spectra allowing the user of the browser to pick a peak and enquire about the frequency and intensity, zoom and pan on the spectra, and display an envelope representing a chosen line width for the spectra.

Selecting *UV Spectra* will display these spectra in essentially an identical fashion to vibrational spectra above but with values derived from the electronic spectra calculation option of HyperChem.

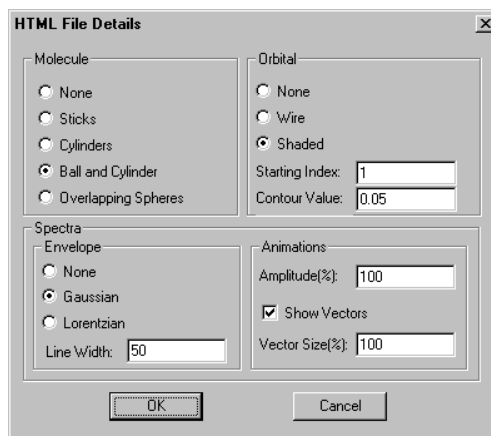


Finally, selecting *IR spectra plus Normal Modes* will create a web page with two windows, one showing the spectra and one showing the molecule. Selecting a peak in the spectra window will animate that normal mode in the molecule window.

## Option Details

Associated with the creation of an HTML page are the details of the actual renderings requested, e.g. whether a ball and stick rendering is created or a CPK rendering, etc. This choice is only for the initial rendering of the HTML page. Since the HTML page is dynamic, the rendering can be changed by the user of the page; the Details... button describes the initial state of the page only.

Pushing the Details... button shown above leads to the following dialog box,

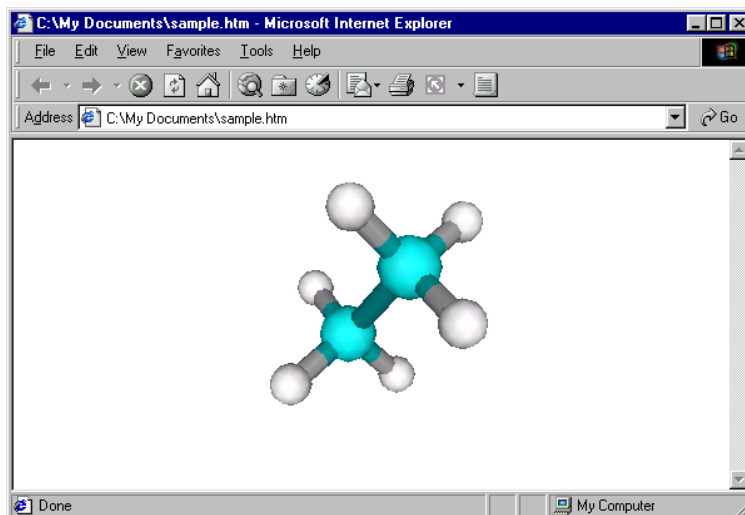


Apart from the animation settings, these choices are easily modified in the browser using a dialog box that is obtained by double-clicking with the right mouse button in the client area of the browser.

The HTML file that is produced can immediately be placed on a web page and makes your HyperChem molecular modeling results immediately available to anyone on the web, provided they have installed the WATATL.DLL ActiveX control on their machine plus the NPHIN32.DLL Netscape plug-in. These controls are included with your copy of HyperChem and can be freely passed to anyone

in the world. Alternatively, the controls can be downloaded from Hypercube, Inc.'s web site at <http://www.hyper.com> or, alternatively, from <http://www.MolecularModeling.com>.

## Sample HTML Page



The above picture shows a browser containing a live molecule from HyperChem. This molecule can be rotated, translated, zoomed, etc. in the web page using a mouse. Double-clicking with the right mouse button on the page results in a pop-up dialog box that changes the rendering, the backdrop color, etc. One can even select atoms, bonds, torsions, etc. and the value of the relevant structural parameter will be displayed in the status line of the browser. Thus, structures can be very cleanly and easily published in this fashion.

In the same way that a structure is shown published on the web in the above screen shot, orbitals, and spectra can also be published. With a web page that contains a vibrational spectra, the normal mode animations can be seen by anyone by just clicking on the relevant spectral line. Thus one is really publishing active content rather than just pictures. All that is required for anyone to see this active content is for them to have installed or downloaded the appropriate free control, i.e. WATATL.DLL plus an associated

Netscape plug-in (NPHIN32.DLL) that works with either Netscape or Microsoft browsers.

## Start Log . . .

Start Log stores the results of HyperChem chemical calculations in a log file. Log files contain comments appearing on the status line during a calculation, plus starting conditions and additional results.

When you choose Start Log, enter a log filename in the Start Log dialog box that appears. You can accept the default filename (chem.log) or enter another filename, which can have any extension. If you use an existing filename, HyperChem lets you either add (append) results to this file or replace the contents of the file.

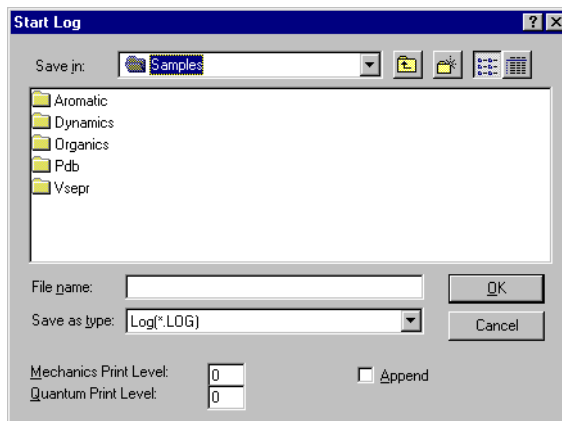
Using the Start Log dialog box, you can have only one log file active at a time. The amount of information stored in a log file depends on the settings used for Mechanics Print Level and Quantum Print Level (see page 631). Allowed values range from 1 to 9.

For more information on using log files see “Storing Results in a Log File” on page 616 and “Log Files” on page 631.

*Note:* Start Log is gray and unavailable if HyperChem is already using a log file.

## Open Log Dialog Box

Use this dialog box to name a log file. L-click OK to accept chem.log as the log filename or a different filename you've entered, and start the log recording.



If the log file is already present, HyperChem asks if you want to append to it or replace it. The log file is usually located in the current directory. The “Append” option specifies that you wish to append to the current file.

To add comments to a log file, choose Log Comments on the File menu. To stop recording, choose Stop Log on the File menu (see the next section).

## Stop Log

Stop Log turns off the recording of a log file and also adds the date and time you stopped recording. Use Start Log to begin recording.

To see or print the log contents, you can use Windows Write or Notepad. Enter the filename (for example, \hyper\chem.log), L-click OK, and then choose No conversion in the next dialog box that appears (for Windows Write, only).

For more information on using log files see “Storing Results in a Log File” on page 616 and “Log Files” on page 631.

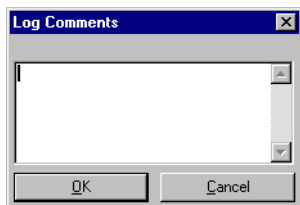
**Note:** Stop Log is gray and unavailable unless HyperChem is already using a log file.

## Log Comments

Use Log Comments to add notes to a log file when a log file is open and recording results. Use Start Log to begin recording in a log file.

You see the Log Comments dialog box to enter your comments.

Enter your comments in text box of the Log Comments dialog box that appears and press OK to save them in the open log file.:

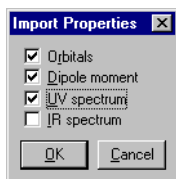


For more information on using log files see “Storing Results in a Log File” on page 616 and “Log Files” on page 631.

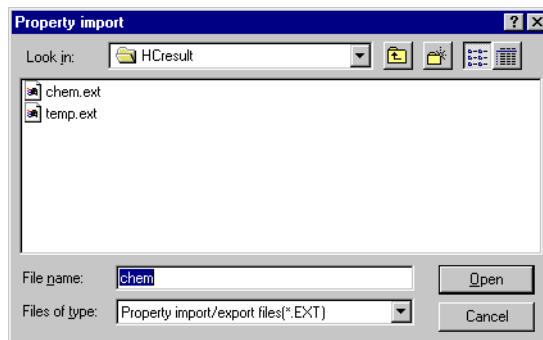
*Note:* Log Comments is gray and unavailable unless HyperChem is using a log file.

## Import . . .

This dialog box allows you to import orbital data that have been calculated by HyperChem and stored, or have been provided by other software. The data must be consistent with the structure in the workspace.



- |                      |  |
|----------------------|--|
| <b>Orbitals</b>      | If this box is checked (✓) and the data are in the file, the orbital data are imported.                                |
| <b>Dipole moment</b> | If this box is checked (✓) and the data are in the file, the dipole moment vector is imported.                         |
| <b>UV spectrum</b>   | If this box is checked (✓) and the data are in the file, the UV/visible spectrum is imported.                          |
| <b>IR spectrum</b>   | If this box is checked (✓) and the data are in the file, the IR spectrum is imported.                                  |
| <b>OK</b>            | This opens the Property Import dialog box, which allows you to specify the file that the data should be imported from. |



The default extension for these Property files is .EXT.

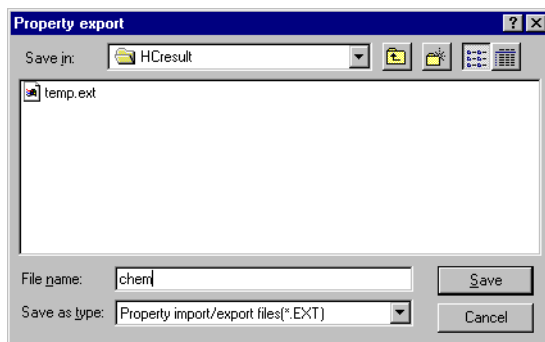
## Export . . .

This dialog box allows you to save orbital data that have been calculated by HyperChem, so that they can be read back in later or sent to other software. Items in the dialog box are gray and inactive unless the relevant data have been calculated with a semi-empirical or *ab initio* calculation (or imported) and are available for storage.



- |               |   |
|---------------|---|
| Orbitals      | If this box is checked (✓), the orbital data are saved in a file.   |
| Dipole moment | If this box is checked (✓), the dipole moment vector is saved in a file.  |
| UV spectrum   | If this box is checked (✓), the UV/visible spectrum is saved in a file. For information about calculating a UV/visible spectrum, see “Electronic Spectrum” on page 411. |
| IR spectrum   | If this box is checked (✓), the IR spectrum is saved in a file. For information about calculating an IR spectrum, see “Vibrations” on page 385.                         |

**OK** This opens the Property Export dialog box, which allows you to specify the file that the data should be imported from.



The default file extension for saving electronic data in a property file is .EXT.

## Print

Print reproduces the molecules in the workspace on your printer.

This command displays the standard Windows Print dialog box, where you can set page orientation and other printer options.

**Note:** You can speed up printing by choosing Sticks from the Rendering option on the Display menu.

Be sure that your printer is on and ready. HyperChem does not inform you of printer problems, but relies on Windows to manage printing.

### General tips for printing or copying images

Some experimentation may be required to obtain the best quality molecular images, depending on your display and printer settings. Some suggestions are:

- For spheres renderings, use at least 256 colors if available for your display. This will improve monochrome as well as color printing or clipboard images.
- For spheres renderings or images copied as bitmaps, using the highest resolution of your display will produce the smoothest images (and printed atom outlines for spheres rendering). For

spheres it is usually best to use the highest resolution display settings that also have at least 256 colors.

- For printers with “dots” that are not square (i.e. not a 1:1 aspect ratio), best results may be obtained using the lowest resolution settings in the printer's Setup dialog box.
- The quality of black and white images copied to the clipboard on 256-color display devices can be improved by using the “window-color = monochrome” script command.

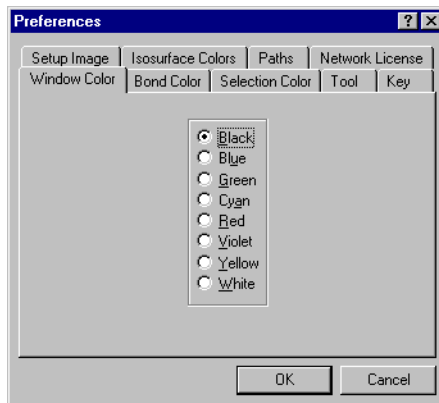
## Preferences

Using Preferences, you can set some conditions for using HyperChem. When you leave HyperChem, it stores your choices in the Registry or in a file (chem.ini) and uses these settings for your next session. The settings that will be used when you start HyperChem will be the same as when you last exited the program, and not (necessarily) the same as the settings of any other HyperChem windows you may have open at the same time.

### Preferences Property Sheet

When you choose Preferences, the Preferences dialog box appears. It has nine property sheets, marked by “tabs”, which you use to set the different kinds of HyperChem options.

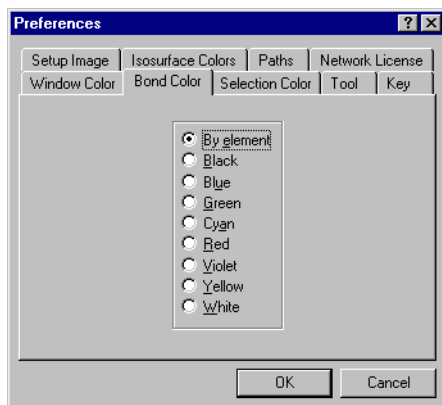
### Window Color



Choose a color for the workspace background. The default is black.



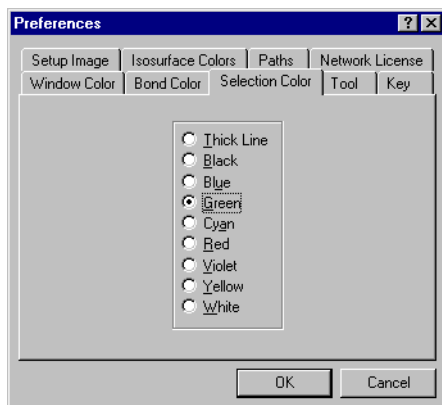
## Bond Color



Choose a color for atoms and covalent bonds. The default is By element, which colors each element and half bond in a Sticks rendering according to the settings in the Element Color dialog box (see page 284). Bond color does not affect hydrogen bonds, which are drawn as dotted white lines.

*Note:* See “Using Black-and-white Output” on page 105 for information about working without color.

## Selection Color

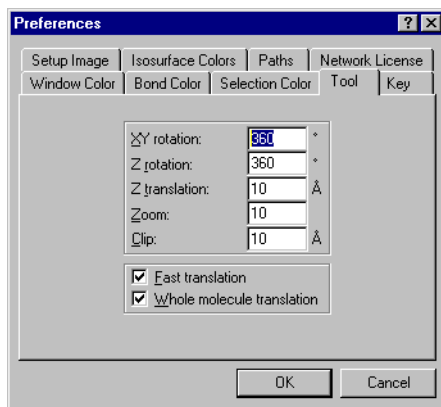


Choose a color that HyperChem applies to selected atoms and bonds. The color returns to Bond color (see above) when you deselect the atoms. The default selection color is green. The choice of Thick line does not change the color, but replaces the lines repre-

senting selected bonds with heavier lines. Thick line works only for Sticks renderings.

**Note:** The Thick line option slows the movements of molecules (translation and rotation) when you are using HyperChem tools.

## Tool



These settings regulate the behavior of a tool in the workspace, whether you are dragging to rotate or translate molecules, to change the Zoom Factor, or to set the clipping planes.

**XY rotation** Normally, moving the XY rotation tool completely across the workspace rotates a molecule by 360 degrees. You can change this amount by entering a number >0 and up to 3600.

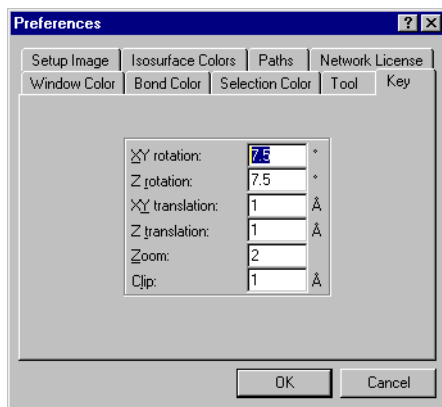
**Z rotation** Normally, moving the Z rotation tool completely across the workspace rotates a molecule by 360 degrees. You can change this amount by entering a number >0 and up to 3600.

**Z translation** Normally, moving the Z translation tool vertically completely across the workspace translates molecules by 10 Å. You can change this amount by entering a number >0 and up to 1000.

**Zoom** Normally, moving the Zoom tool vertically completely across the workspace magnifies molecules by 10×. You can change this amount by entering a number >0 and up to 1000.

- Clip** Normally, moving the Z clipping tool vertically completely across the workspace moves a clipping plane by 10 Å. You can change this amount by entering a number >0 and up to 1000.
- Fast translation** Check this box (✓) to increase the speed of moving a molecule with a translation tool. This substitutes a picture of the molecule, called a bitmap, for the molecular structure on the display. HyperChem can move bitmaps more quickly because it does not need to compute atomic coordinates. When you complete a translation and release the mouse button, HyperChem computes the new atomic coordinates and discards the bitmap.
- Whole molecule translation** Check this box (✓) to restrict translation to whole molecules. If you use the Translation tool to try to translate a selected portion of a molecule, HyperChem will translate the entire molecule. If you use the Translate dialog box to move a selected whole molecule to a point, the center of mass moves to the new point. When this option is off (no ✓), you can translate selected parts of molecules, distorting bond geometry.

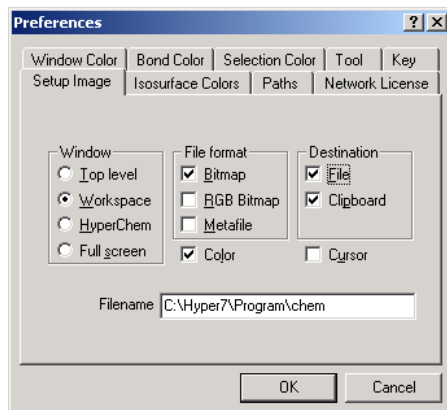
## Key



These settings regulate the stepwise rotation and translation of the contents of the workspace in response to your using the [3], [4], [1], [2], [PgUp], [PgDn], [Home], and [End] keys (see “Keyboard Equivalents for Tools” on page 49.)

- XY rotation** This is the amount that the system rotates around the x or y axis when you use one of the [3], [4], [1], and [2] keys.
- Z rotation** This is the amount that the system rotates around the z axis when you use the [Home] and [End] keys.
- XY translation** This is the amount that the system translates in the x or y direction when you use [Shift], and one of the [3], [4], [1], and [2] keys.
- Z translation** This is the amount that the system translates in the z direction when you use [Shift], and one of the [Home] and [End] keys.
- Zoom** This is the factor that the system grows or shrinks by if you use the [PgUp] or [PgDn] key.
- Clip** This is the amount that the clipping plane moves by if you use [Shift], and one of the [PgUp] and [PgDn] keys.

## Setup Image



The settings in this section control how an image is saved when you use the “Copy Image” command or press the [F9] key. (See page 206.) You can set the extent of the picture from the full screen to a specific menu or dialog box. You also choose the destination (Clipboard or File), file format (Bitmap or Metafile), filename, and whether the image should be color or monochrome. You can use these images with other programs to prepare reports or publications.

<b>Window</b>	This setting determines the part of the screen to copy.
<b>Top level</b>	This copies any active pull-down menu or visible dialog box. If these are not visible, it copies the active HyperChem window. You must use [F9] instead of Copy Image for menus and dialog boxes.
<b>Workspace</b>	This copies only the workspace of the active HyperChem window.
<b>HyperChem</b>	This copies the entire active HyperChem window, including its borders.
<b>Full screen</b>	This copies everything on screen, including other program windows.
<b>File format</b>	You can choose either bitmapped (BMP) or Windows metafile (WMF) formats, or both, to store a picture. HyperChem can store only one picture at a time in each file format. The next time you use the same file format, the new picture replaces the last one. You can also use a 24-bit RGB Bitmap (BMP) file format. Because this operation disables the standard color set used for display, specifying this format disables most other Setup Image options, including saving to the other formats and saving to the Clipboard.
<b>Bitmap</b>	This saves the picture as a reproduction of the pixels on screen in a file named by <i>Filename</i> . The extension .bmp is used with <i>Filename</i> . The resolution of this image is the same as the resolution of your monitor, which you chose when setting up Windows. Use Bitmap to save Spheres renderings or contour plots from a single point, semi-empirical calculation (see “Single Point” on page 360).
<b>RGB Bitmap</b>	This is like the regular Bitmap format above, but instead of using the standard color set which is used for ordinary display, the image pixels are stored in full 24-bit color. The appearance of regular bitmap pictures can change if you import them to

other software that is using a different color set, but an RGB bitmap does not change because it does not depend on an external color set. However, an RGB bitmap can be displayed properly only by software that can process 24-bit color. Software that cannot work in 24-bit color may be able to read an RGB bitmap but the resulting picture will look very strange. These pictures cannot be saved to the Clipboard.

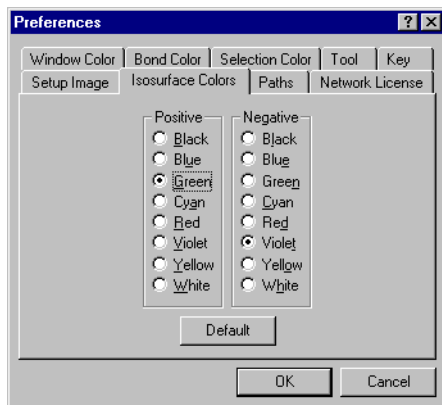
Metafile	This saves the picture as a line drawing in the file name by <i>Filename</i> . The extension .wmf is used with <i>Filename</i> . Metafile is best for Sticks, Ball and Cylinder, Dots and Stick & Dots renderings. Metafiles require about 10 times less storage space than bitmap files containing the same renderings. The resolution of a metafile image does not depend on the monitor, but on the resolution of the output device. For example, if you print a metafile on a laser printer, the resolution might be 300 dpi.
Destination	You can choose one or both locations to store a picture.
File	HyperChem stores the picture in the files <i>Filename.bmp</i> and/or <i>Filename.wmf</i> where the <i>Filename</i> is entered in the edit field of the dialog box described below. The extension .bmp or .wmf is added automatically and should not be typed in the edit box. You can use either or both files. Each file can hold only one picture at a time.
Clipboard	HyperChem stores the picture as a bitmap or a metafile, or both, on the Windows Clipboard. The Display menu in the Windows Clipboard program refers to a Metafile as a Picture. The Clipboard can hold only one image of each type at a time.
Color	Check this box (✓) to store a picture in full color. Turn Color off to produce a black-and-white pic-

ture and to substitute dotted lines for certain colored lines (see “Using Black-and-white Output” on page 105). For a BMP file format, color results in a file at least four times larger. For a WMF file, Color produces only a small difference in file size. For an RGB bitmap file, Color makes no difference in file size.

**Cursor** Check this box (✓) for the cursor to appear in the picture. The cursor can be the mouse pointer, a HyperChem tool, or a text cursor (in a dialog box).

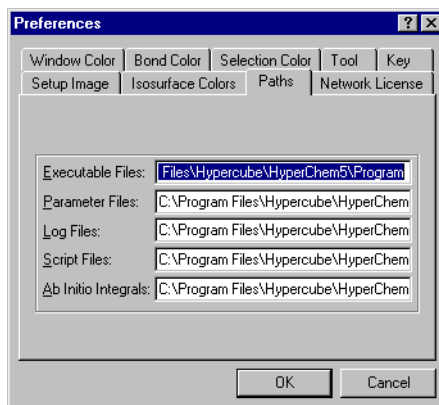
**Filename** Use this edit field to enter a Filename for any bitmap file or metafile file that will store a captured image when the Destination for the capture is File as described above. The Filename should include a path but not use an extension, which will be automatically added as .bmp or .wmf. This filename is programmable via scripts if a sequence of image captures is required for a movie.

## Isosurface Colors



These settings show the colors used for drawing the positive and negative lobes of isosurfaces. For Gouraud shaded surfaces and translucent surfaces, these colors are the two endpoints for the range of colors used for rendering.

## Paths

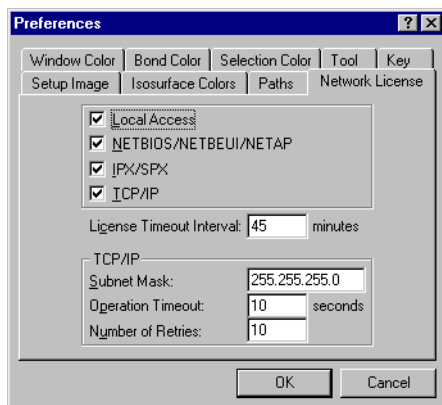


These settings show the directories which are used to hold HyperChem's files.

<b>Executable files</b>	This is the directory that stores HyperChem's executable (program) files.
<b>Parameter files</b>	This is the directory that holds the molecular mechanics and semi-empirical parameter files, and the <i>ab initio</i> and DFT basis set files.
<b>Log files</b>	This is the directory in which, by default, HyperChem saves a log file which you open.
<b>Script files</b>	This is the directory which, by default, HyperChem looks in to find script files.
<b>Ab initio integrals</b>	This is the directory in which HyperChem stores the temporary files that hold the 2-electron integrals during <i>ab initio</i> and DFT calculations.



## Network License



These settings control HyperChem's network operations if you are using a network (rather than a stand-alone) version of HyperChem.

- |   |  |
|---|--|
| <b>Local Access</b>                     | If this option is marked (✓), HyperChem will try to get a license to run from a network hardware lock installed on the machine that it is running on   |
| <b>NETBIOS/<br/>NETBEUI/<br/>NETAP</b>  | If this option is marked (✓), HyperChem will try to get a license to run over a NETBIOS/NETBEUI/NETAP network.   |
| <b>IPX/SPX</b>                          | If this option is marked (✓), HyperChem will try to get a license to run over an IPX/SPX network.  |
| <b>TCP/IP</b>                           | If this option is marked (✓), HyperChem will try to get a license to run over a TCP/IP network.  |
| <b>License<br/>Timeout<br/>Interval</b> | If HyperChem is left running without user interaction (that is, use of menu items), it will eventually release its network license so that other people can use it. This prevents people who forget to exit HyperChem from using up the licenses indefinitely. This option allows you to specify how long (in minutes) HyperChem should wait before releasing its license for others to use. Even if you lose your network license due to timing out, the next time you try a menu operation, HyperChem will attempt to obtain another license. The license timeout interval should be set for a time somewhat |

longer than you expect that your calculations will take, since you cannot use a menu item while a calculation is in progress.

TCP/IP options      These settings affect HyperChem's interaction with TCP/IP networks.

## Using Black-and-white Output

If you set Window color to White and Bonds to Black, HyperChem uses dotted and solid lines when it would normally use two colors. For example, when HyperChem draws contour lines for quantum mechanics calculations, positive contours appear solid and negative contours appear dashed (see "Plot Molecular Graphs" on page 391).

Use Thick line to distinguish selected atoms from unselected atoms.

## Exit

Choose Exit or press [Alt]+[F4] to end the HyperChem session. HyperChem asks if you want to save changes, if any, to the molecular system in the workspace. As HyperChem closes, it stores information about its present state (for example, display conditions and calculation settings) in the Registry or in the file chem.ini. When you start HyperChem again, it uses these settings.

## Script Menu

Open Script . . .	Starts a script from a file that you choose
Script One	These items are replaced by names of scripts that you can choose
. . .	
Script Ten	
Script Editor	Script Editor provides an environment for developing and editing HyperChem scripts

Scripts use instructions from a file to automate HyperChem procedures. These instructions include all normal HyperChem operations, including selecting menu items. You can use a script file to

manipulate molecules, automate chemical calculations, and perform structural analyses.

You see the Run Script dialog box (see the next section), which lets you choose a script file and start running the script.

You must use an editor, for example Windows Notepad, to write a script file. See “Automatic Operation with Scripts” on page 509 for the rules and procedures for writing a script.

## Open Script

When you choose Open Script, you see the Run Script dialog box to start a set of instructions, stored in a script file (usually with the extension *.scr*), that tells HyperChem what to do.

The Run Script dialog box has the same components as the Open File dialog box (see page 79), except that the only file type listed is Script (\*.SCR). If the script begins with comments, these will appear in the Comments text box when the file is selected in the list.

*Note:* When you start HyperChem, it automatically opens and runs the script in *chem.scr*. This script can contain any operations that you need. For example, you can use the script command *change-user-menuitem* to place scripts on the Script menu in place of Script One, and so on. *Chem.scr* must be in the Program directory, with the program executable files.

## Script One . . . Script Ten

These items, inactive and in gray, indicate the locations for script names. To replace these items with names of scripts, you must write a script containing specific instructions (see the next examples). To start a script from this menu, double-click on the script name.

*Note:* In Release 6 of HyperChem and beyond these menu items do not appear except implicitly. That is, they are there but are effectively invisible. One can still replace them as described below and the replaced version will be visible.

*Example:* :

This script adds the script name *RedSelectionColor* to the Script menu in place of Script One:

```
change-user-menuitem 1 RedSelectionColor color1.scr
script-menu-enabled(1) = true
```

RedSelectionColor appears in place of Script One on the Script menu. The second command is necessary in order to enable the command — that is, to make it active instead of gray and inactive.

*Example:*

This script adds the script name “Torsion 180” to the second entry in the script menu, and specifies that when it is selected by the user, a script command is executed:

```
script-menu-caption (1) "Torsion 180"
script-menu-enabled (1) yes
script-menu-command (1) "set-bond-torsion 180"
```

The new menu item executes the “set-bond-torsion” command. The command could instead be an instruction to run a script file, load a structure file, etc.

*Note:* If you name a script file chem.scr and place it in the HyperChem main directory, HyperChem runs the script automatically when you start up HyperChem. You can use this feature to customize the script menus every time you start HyperChem.

## Script Editor

### Overview

Script Editor provides an environment for developing and editing HyperChem scripts (\*.SCR). It also provides a convenient way of sending script messages to HyperChem without creating a script file: it acts as a command-line interface to HyperChem.

The script editor is useful only for scripts written in the HyperChem command language (Hcl scripts with the \*.SCR extension). It is not useful for writing Tool Command Language (Tcl) scripts (\*.TCL). Thus, it is most likely to be used for immediate execution of individual Hcl script commands. Tcl scripts require a normal text editor of some sort to prepare them.

Script Editor provides the following features:

- A command line for sending messages directly to HyperChem.

- An editable command message list so that commands can be saved for later editing.
- A cut and paste facility for saving the commands to the clipboard.
- User-selectable fonts for a more pleasing interface.
- A sizable window, so that you can view as much, or as little, of the message list as you wish.
- The ability to load, save and merge text files.
- A paste function that provides you with a list of valid HyperChem messages.

The Script Editor executable file is HCSCRIPT.EXE, installed in your Program directory. A list of all messages is stored in MESSAGES.TXT in the LIBRARY subdirectory.

## Tutorials

The tutorials demonstrate the different way that Script Editor can be used.

### 1. Entering Commands In The Script Editor

The ability to send script messages directly to HyperChem is convenient in many cases. It allows simple access to many initialization settings (such as the log file print level) that are otherwise cumbersome to change.

To send commands directly to HyperChem:

1. Start HyperChem, and start Script Editor. Make sure that the Options/Command Line menu item is checked (to use Script Editor in command line mode).
2. Click on the Command Line entry box. Then type a HyperChem script command. For example, type "window-color green" (without the quotation marks).
3. Click on the Execute button or press the Enter key to send the command to HyperChem. The HyperChem screen turns green.

## 2. Using The Paste Message Dialog Box

Typing in commands is fine as long as you know HyperChem script messages well. Script Editor provides a "Paste Message" facility to make it easier to find the HyperChem script messages. The commands are stored in the MESSAGES.TXT file.

To use the Paste Message facility:

1. Choose the File/New menu item to clear the Script Editor screen.
2. Choose the Edit/Paste Message menu item to bring up the Paste Message dialog box..
3. Click on the "File" line in the Menu list, to bring up a list of all the script messages associated with the HyperChem File menu.
4. Scroll down the Message list until you find "window-color black" and click on it to select it. Notice that "window-color black" appears as a message underneath the Menu list, previewing the message to be pasted.

*Note:* Script message with arguments (such as window-color) are accompanied by either a description of the argument type (such as int for an integer) or by the "factory setting" in other cases. The window-color message has an argument that is a color.

5. If you do not want to have example message arguments pasted, uncheck the Paste Arguments check box. Notice that the argument ("black") disappears from the string under the Menu list.
6. Click on the OK button to paste the message into the Script Editor command line and to dismiss the dialog box. (If you click on Apply the dialog box remains for further use.) The message now appears in the command line with the first argument highlighted for editing. Type "red" to change the window-color setting, and click on the Execute button to send the message to HyperChem. The HyperChem screen will turn red.

## 3. Requesting Information From HyperChem

In addition to sending commands to HyperChem, Script Editor can request information, by asking for the value of state variables.

To find the current value of a HyperChem state variable:

1. Make sure the Paste Message dialog box is dismissed, and select File/New to clear the Script Editor.
2. Type "mouse-mode?" in the Script Editor and execute the command. The Values dialog box appears with the message "mouse-mode = selecting" (or another mode) displayed.
3. Now send the command "default-element ?" to HyperChem. The value of the current default element appears beneath the mouse-mode message. Each query to HyperChem will add more to the list.
4. To clear the list, double click on the top left corner of the Values dialog box to clear and dismiss it, or press the Escape key.
5. To delete some items from the list, select them and press the Delete key or the Backspace key.
6. The responses come back with a "tag" attached. The "tag" for "default-element = 6" is "default-element = ". If you just want to see just the value of your request, rather than the whole statement, uncheck the . Toggling this item on or off specifies whether to retrieve information with or without the tag. An alternative way of turning the tag off or on is to send the command "query-response-has-tag Yes (or No)".

#### **4. Working With The Message List**

The Message List can be used in several ways. This exercise takes you through some of them. The following instructions assume that the Options/Command Line menu item is unchecked and the bottom text box is labeled Edit Line.

To send commands from the Message List:

After the previous exercises, you should have a selection of commands in your Message List. To send these commands to HyperChem again, simply select them by dragging the cursor across the relevant lines, and click on the Execute button. You can send several commands at once.

To edit an item in the Message List:

Either:

1. Click on the item you wish to edit.

2. Use the backspace key to erase from the end of the line to the place where you wish to make the change.
3. Type the remainder of the line again.

Or:

1. Double click on the item you wish to edit. The item now appears in the Edit Line.
2. Make your changes in the Edit Line. The changes will be echoed in the Message List as you make them.

To insert a new command into the Message List:

1. Click on the line which you wish to insert a new command above.
2. Click on the Insert button.
3. Type the command into the empty place in the list that appears.
4. If you wish to add extra messages, press the Enter key to open a new line in the Message List.

The Append button is similar to the Insert button except the new line is added below the highlighted line.

To delete commands from the Message List:

1. Select the commands you wish to delete.
2. Click on the Delete button.

## **Working With Files**

If you have developed a script file that you want to save, choose File/Save to save the file.

You can Open or Merge a script file into the Message List by selecting the appropriate items under the File menu.

Alternatively, you can use the Cut or Copy items from the Edit menu to add selected items from the Message List into the clipboard. Paste will add text from the clipboard to whichever box has the focus.



## Reference

This section provides detailed descriptions of the main window, menus and dialog boxes.

### Main Window

The Script Editor main window displays a list of script messages and allows you to edit them and send them to HyperChem. The following controls are present on the main window:

Message List	This contains the list of script messages being edited.
Edit Line or Command Line	Here is where the script messages are edited. The label says <b>Edit Line</b> if the "File/Command Line" is unchecked and <b>Command Line</b> if it is checked.
Execute	Sends the selected messages from the message list to HyperChem.
Insert	Inserts a blank line in the message list before the currently selected message.
Append	Opens a blank line in the message list after the currently selected message.
Delete	Deletes the currently selected messages from the message list.

### File Menu

Use the file menu to read and write script files.

<b>O</b> pen...	Brings a file into the Message List.
<b>M</b> erge...	Merges a file with the contents of the Message List.
<b>S</b> ave	Saves the contents of the Message List into a named file.
<b>S</b> ave <b>A</b> s...	Invokes the Save As.. dialog box for saving files.
<b>E</b> xit	Ends this session of Script Editor.

## **Edit Menu**

Use this menu to interact with the Windows clipboard and to paste HyperChem messages into Script Editor.

<b>C</b> ut	Cuts the selected text into the clipboard.
<b>C</b> opy	Copies the selected text into the clipboard.
<b>P</b> aste	Pastes from the clipboard into the control with focus.
<b>D</b> elete	Delete the selection without copying it to the clipboard.
<b>P</b> aste <b>M</b> essage...	Invokes the Paste Message dialog box to prompt you for script messages.

## **Options Menu**

Use this menu to set options for Script Editor.

<b>R</b> espond With Tag	If this is checked, queries sent to HyperChem will be returned with a tag. If it is not checked then no tag will be attached.
<b>F</b> ont...	Invokes the standard Fonts dialog box to alter the appearance of the user interface.
<b>C</b> ommand Line	This button toggles the behavior from command-line mode to script-editor modes. If this is checked, then using Enter after editing automatically sends the commands in the <b>Command Line</b> box to HyperChem.

## **Cancel Menu**

Use this menu to cancel a set of script messages that is being sent to HyperChem.

## **Help Menu**

Use this menu to access information about this module.

---

<b>About Script Editor...</b>	Brings up the About Script Editor dialog box.
-------------------------------	---

---

## Paste Message Dialog Box

The paste message dialog box prompts you for script messages.

<b>Menu List</b>	Lists the menus and categories for script commands. Select one to fill up the message list.
<b>Message List</b>	Lists the script messages associated with the selected menu item.
<b>Paste Arguments</b>	If checked, includes the arguments with the script message.
<b>Query value</b>	If checked, the script message is sent as a query.
<b>OK button</b>	Pastes the message string into the Script Editor command line and dismisses the Paste Message dialog box.
<b>Apply button</b>	Pastes the message string into the Script Editor command line, but does not dismiss the Paste Message dialog box.
<b>Cancel button</b>	Dismisses the Paste Message dialog box without taking any action

## Values Dialog Box

The Values Dialog box provides a listing of the results of "queries" to HyperChem. It allows you to save these results to a file for later reference, or to paste them into other applications via the clipboard..

Here are the menu items for the Values dialog box:

<b>File/New</b>	Clears the current Values list.
<b>File/Save</b>	Saves the listing to a file or, if there is no current file name, invokes the File/Save common dialog box.
<b>File/Save As..</b>	Invokes the File/Save As common dialog box.
<b>File/Exit</b>	Closes the Values window.
<b>Edit/Clear</b>	Clears the current listing of values.

<b>Edit/Cut</b>	Cuts the currently selected items of the Values list to the clipboard.
<b>Edit/Delete</b>	Deletes the currently selected items from the Values list.

## Help Menu

<b>Index</b>	Lists topics available in the Help window
<b>Keyboard</b>	Provides information for using keyboard keys with HyperChem
<b>Commands</b>	Explains the commands (items) in each menu, with step-by-step instructions for using commands and descriptions of options in all dialog boxes
<b>Tools</b>	Explains how to use HyperChem tools
<b>Scripts &amp; DDE</b>	Defines terms used in HyperChem commands, dialog boxes, and reference materials
<b>Glossary</b>	Defines terms used in HyperChem commands, dialog boxes, and reference materials
<b>Using Help</b>	Explains the HyperChem Help system
<b>Tutorials</b>	On-line Tutorials for Understanding HyperChem
<b>About Hyper-Chem . . .</b>	Shows the release information (including version number) and serial number for HyperChem

HyperChem Help is similar to Help for Microsoft Windows and all other Windows applications. See *Microsoft Windows User's Guide* for information about using Help.

You can use these keyboard shortcuts as alternatives to using the Help menu:

[F1]	Shows the Help contents or, if you highlighted a menu item or have a dialog box active, the Help for that menu item or dialog box.
[Shift] + [F1]	Shows context-sensitive Help. If no menu or dialog is active this will activate the Help cursor. You click

on the Tool icon, menu item or other part of the HyperChem window for Help on that topic.

## **About HyperChem**

Choosing this item shows the About HyperChem dialog box with the version number of the HyperChem software, your serial number and the copyright notice.

# Chapter 6

## Building Molecules

This chapter describes menu items and associated dialog boxes on the Build and Databases menus for building molecules.

### Build Menu

Use the Build menu to draw molecules and operate the HyperChem Model Builder. The Model Builder converts 2D drawings into 3D chemical models. The Build menu has these commands:

---

✓ Explicit Hydrogens <sup>a</sup>	You add hydrogen atoms while drawing. HyperChem does not automatically add hydrogens during model building
Default Element...	You choose the element that the Drawing tool uses
Add Hydrogens	Adds hydrogen atoms to a drawing to satisfy the valence requirements of heavy atoms
Model Build /Add H & Model Build	Creates a 3D model from a 2D drawing. Acts on selected atoms or on all atoms in the molecular system. Calculates atom types for molecular mechanics calculations. If Explicit Hydrogens is on, "Add H &" disappears from this item, and the Model Builder does not add hydrogen atoms to the structure

---

---

<b>✓ Constrain Lengths and Angles<sup>a</sup></b>	Allows easier drawing of symmetric rings and 2D structures by constraining all bond lengths to be the same and all bond angles to be a multiple of 30 degrees.
<b>✓ Allow Arbitrary Valence<sup>a</sup></b>	Allow any number of bonds (up to 12) to an atom. If not flagged then the normal rules of valence apply and warnings are made of attempts to have too many bonds. Commonly flagged to allow special situations such as for coordination compounds.
<b>Set Formal Charge<sup>c</sup></b>	Sets the formal charge on an atom. Influences the model builder and the number of hydrogens automatically added. For example, a formal charge of +1 on a Nitrogen adds four hydrogens rather than the usual three. Does not directly affect other than the model builder.
<b>United Atoms<sup>b</sup></b>	Reverses All Atoms (see below) and includes bonded hydrogen atoms as part of atom types for certain heavy atoms, simplifying molecular mechanics calculations. This item applies to selected atoms or, if there are no selections, to the whole molecular system
<b>All Atoms<sup>b</sup></b>	Reverses United Atoms (above) and represents all hydrogens as individual atoms for molecular mechanics calculations. This item applies to selected atoms or, if there are no selections, to the whole molecular system
<b>Calculate Types</b>	Assigns the atom types necessary for molecular mechanics calculations. The Model Builder includes this function. This item applies to selected atoms or, if there are no selections, to the whole molecular system
<b>Compile Type Rules</b>	If you change the atom types in the text file chem.rul, use this item to update the binary file, typerule.bin, that HyperChem uses for molecular mechanics calculations

---

<b>Set Atom Type . . . <sup>c</sup></b>	Changes the atom type for all selected atoms
<b>Set Mass . . . <sup>c</sup></b>	Sets the mass of selected atoms.
<b>Set Charge . . . <sup>c</sup></b>	Sets the atomic charge for all selected atoms
<b>Set Custom Label . . . <sup>c</sup></b>	Sets arbitrary text as the custom label for selected atoms. Used only as annotation-like feature for atom label.
<b>Constrain Geometry . . . <sup>c</sup></b>	Specifies the geometry around a single, selected atom. The Model Builder can use this constraint when constructing a 3D model
<b>Constrain Bond Length . . . <sup>c</sup></b>	Specifies the bond length for a single, selected bond. The Model Builder can use this constraint when constructing a 3D model
<b>Constrain Bond Angle . . . <sup>c</sup></b>	Specifies the bond angle for a single, selected bond angle. The Model Builder can use this constraint when constructing a 3D model
<b>Constrain Bond Torsion . . . <sup>c</sup></b>	Specifies the bond torsion angle for a selected torsion. The Model Builder can use this constraint when constructing a 3D model

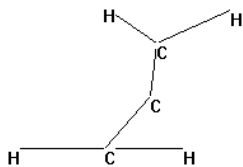
- a. A check mark appears next to this item when it is on. Turning it on affects all further drawing, until you choose it again to turn it off.
- b. United Atoms and All atoms apply to selected atoms in a molecular system, and are not alternatives for a whole molecular system.
- c. When this item is gray, it is temporarily unavailable. You must select an appropriate group of atoms to make it available.

## Explicit Hydrogens

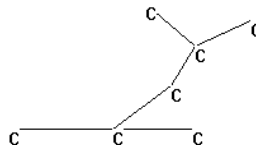
If Explicit Hydrogens is on (✓), you can add hydrogen atoms while you draw molecules. As you use the Drawing tool, all singly bonded atoms appear first as hydrogens. Adding another bond changes a hydrogen atom to the default element (for example, carbon).



When Explicit Hydrogens is on, Add H & Model Build changes to Model Build. The Model Builder does not add additional hydrogen atoms.

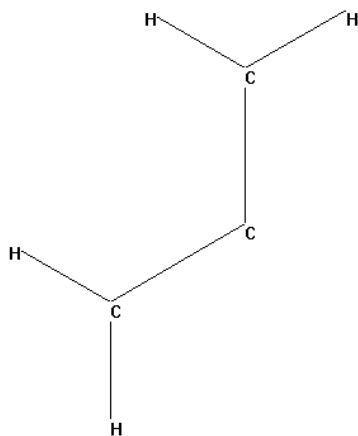


*drawn with Explicit Hydrogens on*

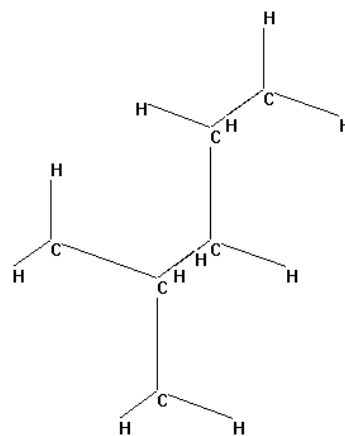


*drawn with Explicit Hydrogens off*

When Explicit Hydrogens is off (no check mark), the Drawing tool places only the default element. Use the Model Builder or Add Hydrogens option (see page 123) later to add the necessary hydrogen atoms to complete the valence requirements of heavy atoms.



*drawn with Explicit Hydrogens on, then used Model Build*



*drawn with Explicit Hydrogens off, then used Add H & Model Build*

## Default Element

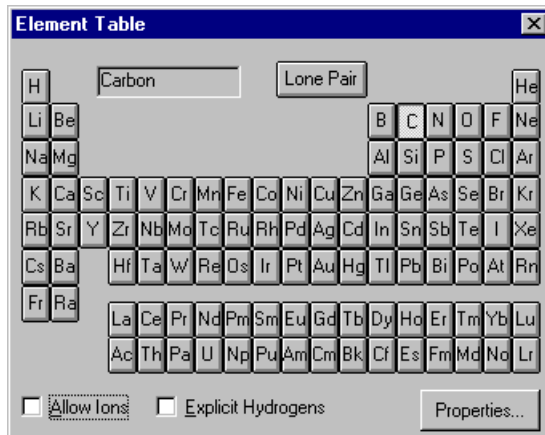
When you choose Default Element, you see the Element Table dialog box for choosing the element for drawing. The Drawing tool places atoms of the default element into the workspace. You can also display this dialog box by double-clicking the Drawing tool.

Choose an element in the periodic table (hydrogen through lawrencium) or a lone pair of electrons (useful for certain molecular mechanics calculations). The default element is changed to the element that you clicked on and the drawing tool is selected so that you can immediately draw in the HyperChem workspace. The dialog box remains on screen until you dismiss it. You can change the default element while you are drawing a molecule. Choose a new element from the Element Table dialog box and continue drawing. Double clicking on an element will change the default to that element and dismiss the dialog box.

You can also change an element after you draw it. Choose a new element from the Element Table dialog box and, using the Drawing tool, L-click on the atom you want to change. For example, if you are drawing with carbon atoms, choose oxygen from the Element Table dialog box; then L-click on a carbon to change it to an oxygen.

## Element Table Dialog Box

Double-click on the Drawing icon or choose Default Element on the Build menu to display this dialog box.



The dialog box lets you select the default element by clicking on the atomic symbol. This will change the cursor to the Drawing Tool and you can immediately start drawing in the workspace with the new default element. Single clicking will leave the dialog box open and double clicking will close it. This dialog box also lets you turn these drawing options on or off:

### Allow Arbitrary

#### Valence

When Allow Arbitrary Valence is on (✓), you can draw up to twelve bonds to an atom. With Allow Arbitrary Valence off, you can draw only four bonds to carbon, three to nitrogen, two to oxygen, and so on. This is modified by a formal charge on an atom. For example, with carbon,  $\text{bonds} = 4 + \text{formal charge}$ .

HyperChem uses only formal charge to affect valence. Other charges (atom charges) affect molecular mechanics interactions; for example, you can see this by using Set Charge (see page 132). The total quantum charge affects quantum mechanical calculations such as a semi-empirical or *ab initio* quantum mechanical calculation (see “Semi-empirical” on page 296, and “Ab Initio and Density Functional” on page 311).

This option is accessible via either the above dialog box or via the Build menu.

#### Explicit hydrogens

Lets you add hydrogen atoms with the Drawing tool. Singly bonded atoms appear first as hydrogens, then change to the default element when you add the next bond. If this setting is off, you draw only heavy atoms, and HyperChem adds hydrogens during model building. You can also turn this condition on or off by choosing Explicit Hydrogens on the Build menu (see page 119). See also “Add Hydrogens” in the next section.

#### Properties

Shows the Element Properties box for the current default element (see below).

You can dismiss the Element Table dialog box by double clicking on the control-menu box or by double clicking on an element. The latter will also change the default element.

**Note:** To draw a hydrogen with multiple bonds, choose hydrogen as the default element and turn Allow Arbitrary Valence on.

**Caution:** You can use any elements from the periodic table for drawing. However, for some elements the Model Builder might not provide appropriate bond lengths or angles for all atoms. Also, semi-empirical and *ab initio* calculations cannot accommodate all elements.

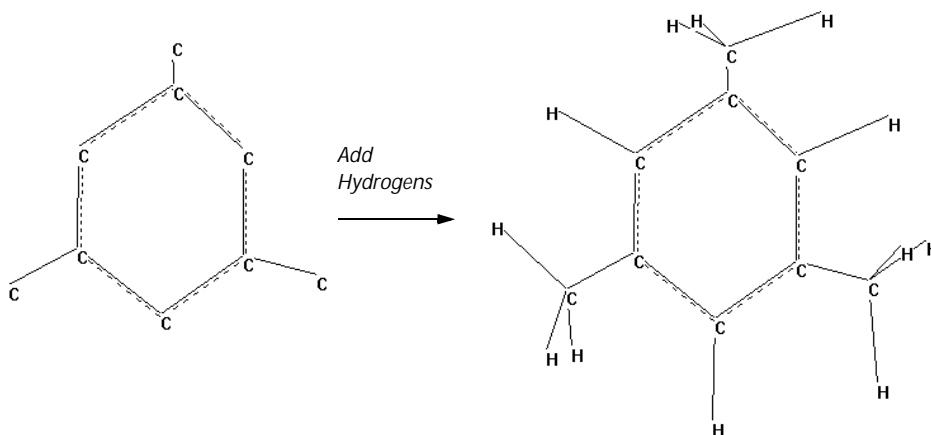
## Element Properties

If you click on the Properties... button in the Element Table dialog box, the Element Properties box will appear, showing the physical properties of the current default element. Alternately, if you shift-click on an element button in the Element Table dialog box, the properties of the element that you selected will be shown. This does *not* change the default drawing element.

Element Properties			
6	C	Carbon	
Atomic Weight:	12.011	Electronegativity:	2.55
Electron Configuration:	1s <sup>2</sup> 2s <sup>2</sup> 2p <sup>2</sup>	1st Ionization Potential:	11.260 V
Atomic Radius:	0.70 Å	2nd Ionization Potential:	24.383 V
Melting Point (1 atm):	n/a	3rd Ionization Potential:	47.888 V
Boiling Point (1 atm):	3825.00 °C	Density (298K):	2.27 g/cm <sup>3</sup>
Heat of Fusion:	n/a	Specific Heat (STP):	0.71 J/gK
Heat of Vaporization:	n/a		

## Add Hydrogens

Add Hydrogens supplies hydrogen atoms to complete the valence shells of heavy atoms. This item works on selected atoms or on all atoms in the molecular system, and works with either a 2D drawing or a 3D model.



Add Hydrogens works with Explicit Hydrogens turned either on or off. If Explicit Hydrogens is on, Add Hydrogens provides hydrogen atoms in addition to the hydrogens already in the molecule.

Add Hydrogens uses a limited portion of the Model Builder (see the next section) to determine the number and location of hydrogen atoms. However, it does not convert a 2D drawing into a 3D model.

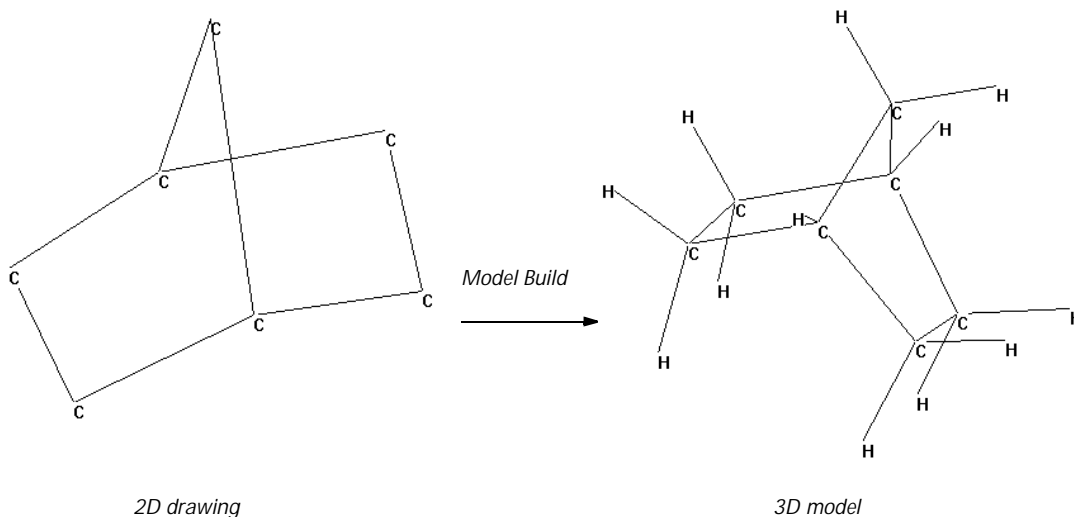
## Model Build/Add H & Model Build

Model Build uses the Model Builder to convert your 2D drawing into a 3D molecular model. Model Build works on selected atoms or, if there is no selection, on all atoms in the molecular system.

**Caution:** The Model Builder acts only on selected atoms, if you select any. This can lead to distortion of the molecule if the selection is not a functional group.

Model Build appears on the Build menu if Explicit Hydrogens is on. HyperChem does not add hydrogen atoms while building a 3D model. This item changes to Add H & Model Build if Explicit Hydrogens is off. The Model Builder then adds all needed hydrogen atoms.

The Model Builder uses a set of rules to construct an approximate 3D model. These rules might have limitations with some structures. For a better 3D model, first use the Model Builder, then use Geometry Optimization on the Compute menu, with either a molecular mechanics or semi-empirical quantum mechanics method.



The Model Builder does not add charges (ionic or partial) to molecules. You can add charges to selected atoms using Set Charge on the Build menu. Or, you can use a semi-empirical or *ab initio* quantum mechanical calculation to determine atomic charges (see “Single Point” on page 360).

The Model Builder assigns atom types (see page 130). Sometimes these types need to be applied manually.

For a discussion of the Model Builder, see “Converting 2D Drawings into 3D Structures” on page 487.

## Constrain Length and Angles

HyperChem drawing has traditionally been very free form in that one's goal was only drawing a 2D sketch (graph) that represented the connections in a molecule which was then to be immediately converted into a 3D structure via HyperChem's model builder. This computational model is convenient and simple if one's only goal is to create the three-dimensional structure. However, if one also wanted a “pretty” 2D picture as well, this mode of free-hand drawing did not make it easy to create well-formed six-sided rings, for example. HyperChem 6 includes the ability to constrain the bond lengths and angles in your drawing so that the result drawing, while still convertible into a 3D structure in an identical fashion to earlier releases, provides a “prettier” 2D drawing for potential printing and inclusion into a document.

HyperChem should still be thought of as a molecular modeling program rather than a drawing program but these constraints along with other changes in release 6 (annotations, in particular) allow you to use HyperChem to produce drawings in conjunction with your molecular modeling.

To constrain your drawing, you simply set the appropriate flag on the drawing menu and from then on all lines drawn are the same length; angles are restricted to multiples of 30 degrees. With these constraints it is easy, for example, to draw a regular six-sided ring. When you want to relieve the drawing constraints and draw as in earlier versions of HyperChem, you just reset the constraint flag on the drawing menu. Thus, you can mix constrained and unconstrained drawing in the same molecule. You may want to create a 2D drawing of a molecule, for example, that has certain bonds all the same length in a regular structure but has other bonds that require an arbitrary length or drawing angle.

This option has no effect whatsoever, on the operation of the model builder or other parts of HyperChem. It simply allows you to control some facets of how you create your 2D drawing so that the drawing is “prettier”.

## Allow Arbitrary Valence

When Allow Arbitrary Valence is on (✓), you can draw up to twelve bonds to an atom. With Allow Arbitrary Valence off, you can draw only four bonds to carbon, three to nitrogen, two to oxygen, and so on. This is modified by a formal charge on an atom. For example, with carbon,  $\text{bonds} = 4 + \text{formal charge}$ .

HyperChem uses only formal charge to affect valence. Other charges (atom charges) affect molecular mechanics interactions; for example, you can see this by using Set Charge (see page 132). The total quantum charge affects quantum mechanical calculations such as a semi-empirical or *ab initio* quantum mechanical calculation (see “Semi-empirical” on page 296, and “Ab Initio and Density Functional” on page 311).

Allow Ions remains on until you turn it off by choosing this item again. This option is accessible via either the above dialog box or via the Build menu.

## Set Formal Charge

HyperChem, and the model builder in particular, understand and include the concept of *formal charge on an atom*. The formal charge on a atom can be set by the user and this formal charge influences the 3D structure created by the model builder.

For example, a single carbon atom which is given a formal charge of +1 and submitted to the model builder results in trigonal bipyramidal  $\text{CH}_5^+$  system being produced. That is, 5 hydrogens are added. If the formal charge on a single carbon atom is set to -1, then the model builder will produce  $\text{CH}_3^-$ . If the formal charge is the default value of zero, then ordinary  $\text{CH}_4$  is created by the model builder. In earlier releases, the model builder always assumed zero for formal charges and created structures appropriate for neutral systems.

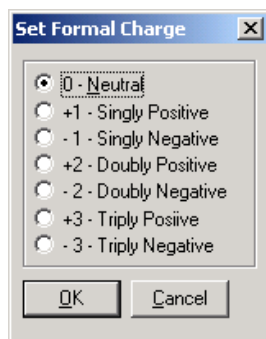
The formal charge on an atom is set by selecting an atom and clicking on the menu item <Set formal charge..> to choose the value of

the formal charge, a positive or negative integer. These formal charges result in the total system charge (also an integer) having a default value that derives from the formal charges, if any. The total system charge which is what determines the number of electrons included in quantum mechanical calculations can be forced to a value that overrides the formal charges if necessary but the total system charge will normally derive from the individual formal charges.

HyperChem also includes the concept of atomic charges which are not integers and are normally the output of various quantum mechanical calculations. The atomic charges, in the absence of quantum mechanical calculations, the use of templates for amino and nucleic acids, or other procedures which set the atomic charges, will reflect the formal charges. For example, setting the formal charge on a Nitrogen atom to +1 and invoking the model builder will result in  $\text{NH}_4^+$  with a formal charge +1 on the Nitrogen and atomic charges of 1.0 for the Nitrogen and 0.0 for the Hydrogens. A quantum mechanical calculation might then change the atomic charges to 0.96 on the Nitrogen and 0.01 for each of the hydrogens while leaving the formal charges unchanged.

## Set Formal Charge Dialog Box

This dialog box list the values of formal charge which you can set for the selected atoms.

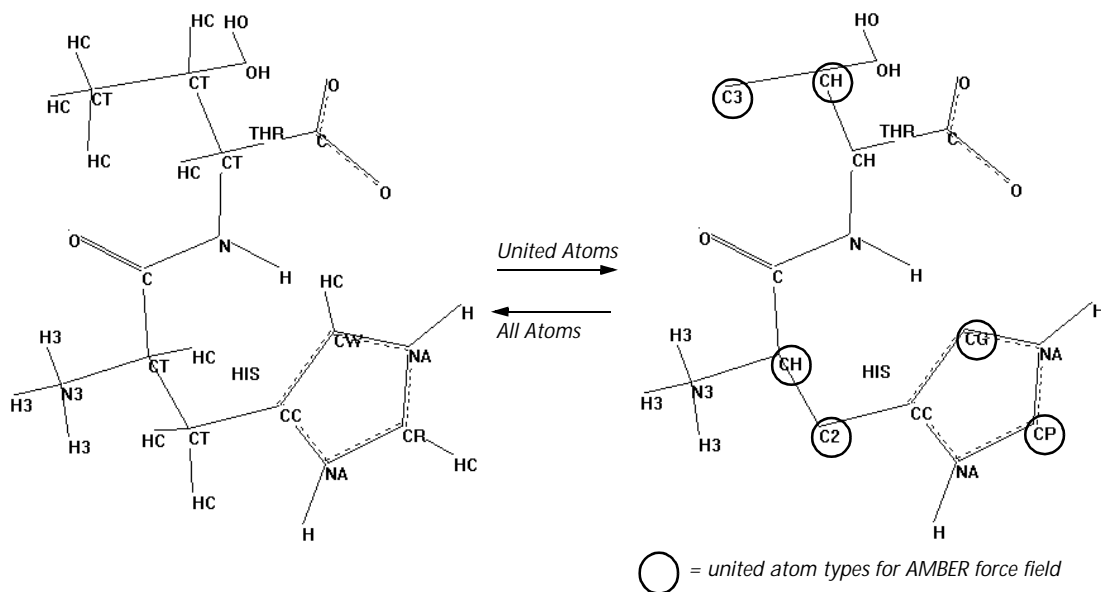




## United Atoms

United Atoms simplifies a molecular structure and calculations on this structure by including hydrogen atoms in the definition of a carbon atom. Force fields have different atom types (see “Atom Types” on page 654) for these composite atoms. United Atoms acts on selected atoms, or, if there is no selection, on the whole molecular system. This choice remains on until you select the atoms again and choose All Atoms (see the next section).

**Note:** United Atoms applies to selected atoms, so you can use both a united-atom and an all-atom representation in the same molecular system.



**Note:** When you change between United Atoms and All Atoms, HyperChem changes atom types, as needed, for the force field in effect.

**Note:** United Atoms and All Atoms are available only after you choose a molecular mechanics force field on the Setup menu (see “Force Field Options Dialog Box” on page 293).

The united atom rules for simplifying a structure depend on the force field that you choose in the Setup menu (see “Force Field Options Dialog Box” on page 293) and the residue templates for

the force field (see “Residue Template Files” on page 727). In the literature on force fields, united atoms are also known as extended atoms or extended hydrogens.

## All Atoms

All Atoms reverses the United Atoms choice, so that the molecular system represents every hydrogen as a separate atom. All Atoms applies to selected atoms or, if there is no selection, to the whole molecular system.

*Note:* All Atoms applies to selected atoms, so you can use both a united atom and an all atom representation in the same molecular system.

In the literature on force fields, ‘all atoms’ is also known as ‘explicit hydrogens’.

## Calculate Types

Calculate Types assigns atom types to selected atoms or to all atoms in the molecular system. Atom types represent the stereochemical properties of each atom (for example, CO for  $sp^2$  carbonyl carbon, CT for tetrahedral carbon, and CH for a united atom representation of an  $sp^3$  carbon with no separate hydrogen). Each molecular mechanics force field uses a different set of atom types. Calculate Types applies types for the last force field you chose in the Molecular Mechanics Force Field dialog box (see page 290).

Molecular mechanics calculations require atom types. Since the Model Builder (see “Model Build/Add H & Model Build” on page 124) automatically makes these assignments, you need only to use Calculate Types when you do not plan to use the Model Builder. If you change force fields, HyperChem automatically assigns new types. In some cases it may be necessary to assign atom types manually (see page 130).

To check that all atoms in the workspace have an atom type, choose Labels on the Display menu and then choose Type for the label. Atoms without assigned types appear with a \*\* label.

If you are using a new or altered set of atom types (a new chem.rul file or type parameter file), use Compile Type Rules on the Build menu before you do a molecular mechanics calculation.

## Compile Type Rules

Use Compile Type Rules if you change the rules for assigning atom types that HyperChem uses for molecular mechanics calculations (see “Atom Types” on page 654). You can change the rules for assigning atom types in the file `chem.rul` by using a text editor. Then use Compile Type Rules to create the new binary version of this file, `typerule.bin`, which HyperChem uses for calculations.

*Caution:* If you change the file `chem.rul`, be sure to back up or save copies of the original files. Errors in the new `chem.rul` can prevent HyperChem calculations.

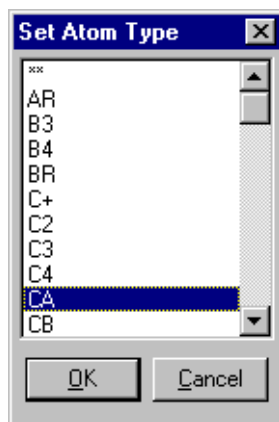
## Set Atom Type

When you select one or more atoms and then choose Set Atom Type, a dialog box appears for changing the type for selected atoms. HyperChem uses atom types for molecular mechanics calculations.

HyperChem assigns atom types when you choose Model Build or Calculate Types on the Build Menu (see page 124 and page 129), when you use a residue from the Databases menu, when you add water molecules to a periodic box, or when you open a PDB or MOL2 file. The types depend on the molecular mechanics method that you choose (see page 289).

*Note:* If you select multiple atoms, Set Atom Type assigns the same atom type to each of the selected atoms.

## Set Atom Type Dialog Box



The Set Atom Type dialog box lists atom types for the active force field. The selected atom appears highlighted. For example, the Model Builder might show the type CT for a carbon atom. You can change this to another atom type in the list.

**Caution:** HyperChem does not check your choice. Be sure to pick an atom type appropriate for the element.

To check existing atom types in a molecular system, choose Labels on the Display menu and then choose Type for the label. Atoms without assigned atom types have a \*\* label.

The MM+ calculation method can work with unknown atom types using its Default Force Field (see the *Computational Chemistry* reference manual). (The other molecular mechanics force fields require that types be assigned to all atoms.) The MM+ Default method is able to assign force field parameters on the basis of bonding characteristics. This can be used to work around some limitations of the molecular mechanics methods. For example, in structures like biphenyl, all of the carbon atoms are assigned types appropriate for  $sp^2$  atoms. The bond between the rings joins two  $sp^2$ -type atoms, and so is given the characteristics of a double bond: it is relatively short, and all of its neighboring atoms are forced towards a single plane, even though the bond actually is single, and the rings should be tilted with respect to each other. If the two atoms where the rings are joined are set to the unknown (\*\*) type, an MM+ calculation can give the proper results.

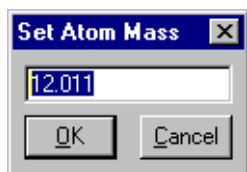
**Note:** Set Atom Type is gray unless you select one or more atoms.

## Set Mass

The Set Mass Dialog box is used to set the mass for selected atoms. By default, when an atom is drawn it is assigned the average atomic mass for that element. For vibrational analysis (see “Vibrations” on page 385) it is sometimes desirable to use different values.

**Note:** If you select multiple atoms, Set Mass assigns the same atom mass to each of the selected atoms.

### Set Atom Mass Dialog Box



In the Set Atom Mass dialog box, enter a positive real number, the mass of the selected atom in atomic units.

**Caution:** HyperChem does not check to see if the mass you assign is a reasonable value for the atom(s) selected.

To check existing atomic masses, choose Labels on the Display menu and then choose Mass for the labels. When you assign a new mass, the label changes to show the new mass, but only after you deselect the atom.

**Note:** Set Mass is gray unless you select one or more atoms.

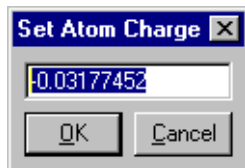
## Set Charge

You use Set Charge to change the net charge on selected atoms. Atoms you draw in HyperChem do not have atomic charges. To add charges to the atoms, you can use Set Charge or run a semi-empirical or *ab initio* calculation.

For amino acid and nucleic acid residues and for water molecules in a periodic box, HyperChem provides atomic charges. Molecular mechanics calculations use net atomic charges to predict electrostatic interactions.

*Note:* If you select multiple atoms, Set Charge assigns the same atom charge to each of the selected atoms.

## Set Atom Charge Dialog Box



In the Set Atom Charge dialog box, enter a positive or negative charge, either a fraction or a whole number. For example,  $-0.20$  indicates a slight excess of electrons;  $+2.0$  indicates a cation deficient in two electrons.

*Caution:* HyperChem does not check the sign or magnitude of the charge. Be sure it is appropriate for an atom. Also, HyperChem does not check the charge distribution in a molecule.

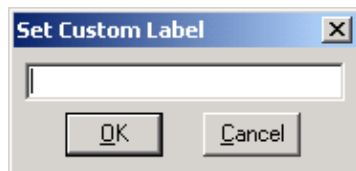
To check existing atomic charges, choose Labels on the Display menu and then choose Charge for the labels. Atoms without a charge have a 0.000 label. When you assign a new charge, the label changes to show the new charge, but only after you deselect the atom.

*Note:* Set Charge is gray unless you select an atom.

## Set Custom Label

This will allow arbitrary text to be associated with selected atoms as a label. To display assigned custom labels, ensure "Custom" is selected from the "Display/Labels".

## Set Custom Label Dialog Box



In the Set custom Label dialog box enter a string that will be the custom label for the selected atoms (or all atoms if none are selected).

## Constrain Geometry

Constrain Geometry specifies the geometry of the bonds around a selected atom when you use the Model Builder. (It does not affect the geometry during optimizations; see “Restrains” on page 340.) When you choose Constrain Geometry, you see the Constrain Geometry dialog box with choices for geometry. To constrain the chirality of an atom for the Model Builder, [Shift]+L-click on one of its bonds with the drawing tool (see “Changing Chirality” on page 485).

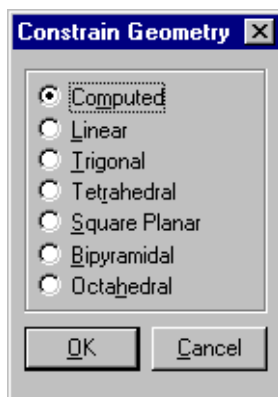
HyperChem uses the Computed option as the default setting for geometry. With Computed, the Model Builder decides on the geometry.

*Note:* This constraint is a high priority for the Model Builder (see “Converting 2D Drawings into 3D Structures” on page 487). However, additional constraints can affect the model and might alter the final results.

You can eliminate the constraint and return to the Model Builder’s geometry by choosing Computed in this dialog box, and then by choosing Model Build on the Build menu.

*Note:* Constrain Geometry is gray unless you select an atom.

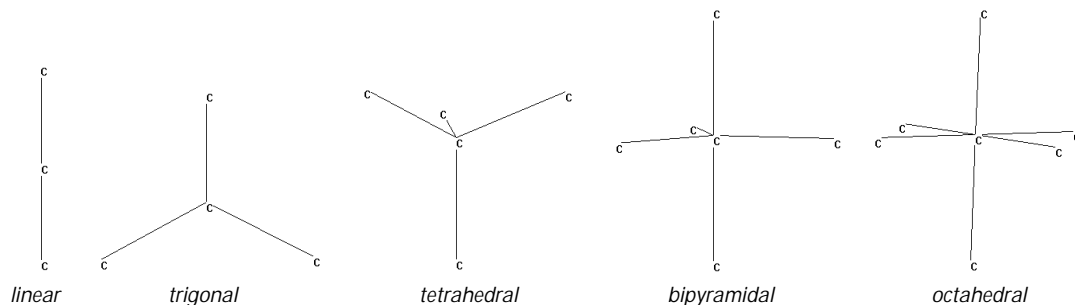
### Constrain Geometry Dialog Box



These are the choices for geometry. All bond angles are relative to one selected atom:

<b>Computed</b>	No constraint. The Model Builder decides on the geometry.
<b>Linear</b>	Two bonds at 180 degrees. If the selected atom has more than two bonds, the Model Builder constrains two bonds only to 180 degrees.
<b>Trigonal</b>	Three bonds in a plane, with 120 degrees between each pair of bonds. If the selected atom has more than three bonds, the Model Builder constrains three bonds only to a plane and 120 degrees. If the selected atom has only two bonds, the Model Builder constrains the two bonds to a 120 degree angle.
<b>Tetrahedral</b>	Four bonds, in a tetrahedral arrangement, with 109.47 degrees between each pair of bonds (in every pair-wise combination). If the selected atom has more than four bonds, only four are in a tetrahedral arrangement. If the selected atom has three bonds, all pairs have 109.47 degree angles.
<b>Square Planar</b>	Four bonds in a plane, with 90 degrees between each <i>cis</i> pair of bonds. If the selected atom has more than four bonds, the Model Builder constrains only four of them to the plane and to angles of 90 degrees. If it has two or three bonds, then one or two bond angles, respectively, will be constrained to the plane and to 90 degree angles.
<b>Bipyramidal</b>	Five bonds, with three bonds in a plane (120 degrees from each other), and two bonds perpendicular to the plane and 180 degrees apart. If the selected atom has six bonds, the Model Builder constrains four bonds to a plane, but only three are separated by 120 degrees. If the selected atom has two or three bonds, these atoms lie in a plane with 120 degree angles. The fourth bond is orthogonal to this plane at 90 degrees to the other bonds.
<b>Octahedral</b>	Six bonds, with all angles at 90 or 180 degrees. If the selected atom has less than six bonds, all bond angles are 90 or 180 degrees.





### To use Constrain Geometry:

1. Select an atom.
2. Choose Constrain Geometry on the Build menu.
3. Choose one of the geometries provided. This geometry remains with this atom during this HyperChem session until you choose another geometry. A HIN file does *not* store information about geometry constraints.
4. Deselect the atoms or select all atoms in the molecule.
5. Choose Model Build on the Build menu.

## Constrain Bond Length

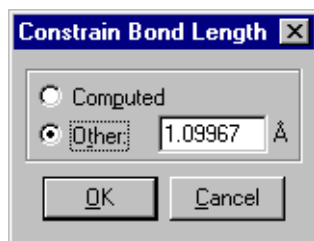
Constrain Bond Length limits the length, in Ångstroms, of a selected bond when you use the Model Builder. (It does not affect the bond length during optimizations; see “Restrains” on page 340.) First select a bond or two bonded atoms, then choose Constrain Bond Length. The Constrain Bond Length dialog box appears for entering a bond length.

**Note:** This constraint is a high priority for the Model Builder (see “Converting 2D Drawings into 3D Structures” on page 487). However, additional constraints can affect the model and might alter the final result.

You can eliminate the constraint and return to the Model Builder’s default bond length by selecting the bond, choosing Computed in this dialog box, and then choosing Model Build on the Build menu.

**Note:** Constrain Bond Length is gray unless you select two covalently bonded atoms.

## Constrain Bond Length Dialog Box



**Computed** No constraint. The Model Builder decides on the bond length.

**Other** Enter a positive value  $> 0$  to 1000 Ångstroms. The practical range is 0.75 to 4 Ångstroms.

You can return to the Model Builder's bond length by choosing Computed in this dialog box and then choosing Model Build on the Build menu.

## Constrain Bond Angle

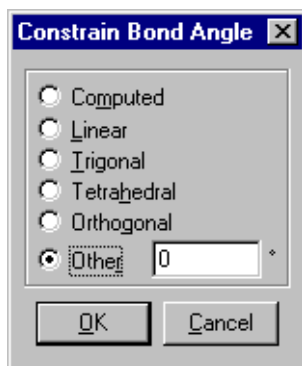
Constrain Bond Angle specifies the angle between two adjacent bonds when you use the Model Builder. (It does not affect the bond angle during optimizations; see "Restrains" on page 340.) First select three atoms or the two included bonds, then choose Constrain Bond Angle. The Constrain Bond Angle dialog box appears for you to choose one of the angles provided or to enter a new bond angle, in degrees.

*Note:* This constraint is a high priority for the Model Builder (see "Converting 2D Drawings into 3D Structures" on page 487). However, additional constraints can affect the model and might alter the final result.

You can eliminate the constraint and return to the Model Builder's default bond angle by selecting the bond, choosing Computed in this dialog box, and then choosing Model Build on the Build menu.

*Note:* Constrain Bond Angle is gray unless you select three atoms that form a bond angle.

## Constrain Bond Angle Dialog Box



These are the choices for a bond angle:

Computed	No constraint. The Model Builder decides on the bond angle.
Linear	180 degrees.
Trigonal	120 degrees.
Tetrahedral	109.47 degrees.
Orthogonal	90 degrees.
Other	Enter a value in the range of $\pm 360$ degrees.

## Constrain Bond Torsion

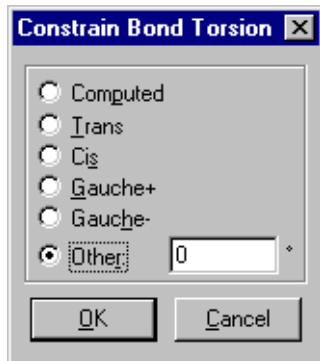
Constrain Bond Torsion specifies a torsion angle when you use the Model Builder. (It does not affect the torsion angle during optimizations; see “Restrains” on page 340.) First select the four atoms or the three included bonds that define a torsion angle. Then choose Constrain Bond Torsion. When the dialog box appears, choose one of the standard torsion angles or enter a new torsion angle.

**Note:** This constraint is a high priority for the Model Builder (see “Converting 2D Drawings into 3D Structures” on page 487). However, additional constraints can affect the model and might alter the final result.

You can eliminate the constraint and return to the Model Builder's default torsion angle by selecting the angle, choosing Computed in the dialog box, and then choosing Model Build on the Build menu.

*Note:* Constrain Bond Torsion is gray unless you select four atoms that form a torsion.

### Constrain Bond Torsion Dialog Box



These are the choices for a torsion angle:

Computed	No constraint. The Model Builder decides on the torsion angle.
Trans	180 degrees.
Cis	0 degrees.
Gauche+	60 degrees.
Gauche-	-60 degrees.
Other	Enter any value $\pm 180$ degrees.

## Databases Menu

Use the Databases menu to create proteins and nucleic acids (DNA and RNA) from a library of residues. The residues have a standard chemical structure. HyperChem provides residue structures (in

template files) and rules for connecting these residues to make macromolecules. The Databases menu has these commands:

<b>Amino Acids...</b>	Creates polypeptides, starting from the N-terminus. The Amino Acids dialog box appears for selecting amino acid residues and polypeptide conformation
<b>Make Zwitterion</b>	Changes a polypeptide, constructed from database residues, into a zwitterion, with charged C-terminal (COO <sup>-</sup> ) and N-terminal (NH <sub>3</sub> <sup>+</sup> ) groups
<b>Remove Ionic Ends</b>	Reverses Make Zwitterion, returning the C- and N-terminal groups to -NH and -CO
<b>Sequence Editor</b>	Creates or reads one-letter amino acid sequences.
<b>Nucleic Acids...</b>	Creates single- or double-stranded DNA or RNA. You can select nucleotide residues, polynucleotide conformations, and sugar conformations
<b>Add Counter Ions</b>	Adds sodium ions to the polynucleotides in the workspace, neutralizing the phosphate groups' charges
<b>Crystals...</b>	A module for reading, creating and viewing crystal structures.
<b>Saccharides...</b>	This sugar builder will build polysaccharide structures.
<b>Polymers...</b>	Builds linear and branched polymers from any specified monomer.
<b>Mutate...<sup>a</sup></b>	You can change one selected amino acid or nucleotide residue into another. This item applies only to peptides or polynucleotides constructed from database residues or to molecules from PDB files
<b>Replace...<sup>a</sup></b>	Replaces one continuous set of amino acid residues with another contiguous set of amino acid residues from the clipboard. It combines a cut and paste operation into a single operation for homologous protein design.

---

<b>Invoke Database...</b>	Invokes the independent HyperChem Data module. HyperChem Data is a database to be used in conjunction with HyperChem. It allows 2D and 3D search of a database of molecules for input to molecular modeling and can store results of molecular modeling.
---------------------------	--

---

a. When this item is gray, it is temporarily unavailable. You must change some condition in HyperChem to make it available.

Proteins and nucleic acids from Brookhaven PDB files (see page 503) are composed of residues. You can use many of the Database menu items to modify molecules from those files.

## Amino Acids

Choose Amino Acids to construct polypeptides from amino acid residues. These amino acids have standard chemical structures.

In the Amino Acids dialog box, choose the secondary conformation of the polypeptide (Alpha helix, Beta sheet, or Other for your own Phi, Psi, and Omega angles). Then choose from these residues, starting at the N-terminus:

- 20 standard amino acids.
- Two neutral and one protonated form of histidine (Hid, Hie, and His). At physiological pH, the side group of histidine is either positively charged or neutral, depending on its local environment.
- Half-cystine residues (Cyx) for constructing disulfide links (see “Cross-linking with Cyx” on page 495).
- One N-terminal blocking group (Ace). Use this as the first residue only.
- One C-terminal blocking group (Nme). Use this as the last residue only.

*Note:* Residues from this dialog box are not equivalent to amino acids you construct atom by atom. You can add new residues to the choices in the Amino Acids dialog box or modify residues obtained from this dialog box (see “Adding Unusual Residues” on page 496).

HyperChem links residues together in the order you select them. A C-terminal blocking group (cap) prevents further addition. After

adding a cap, choosing additional residues begins another polypeptide.

If the polypeptide has no cap, you can later add more residues to the C-terminus. If you turn the polypeptide into a zwitterion (see the following section), you cannot add more residues to this molecule; choosing additional residues begins another polypeptide.

As you extend the polypeptide, you can change these properties of the next residues that you add:

- Secondary structure (Alpha helix, or Beta sheet, or Other)
- Conformation of the peptide bond (Omega)
- D or L isomer of each residue

These choices remain in effect until you make other choices. You cannot change these properties after you add the residues.

**Note:** Do not use the Model Builder on a whole constructed polypeptide. The Model Builder discards the optimized geometry in these molecules and cannot accurately reproduce the structure. You can, however, use the Model Builder on selected atoms; for example, for small modifications of standard residues.

For detailed instructions on building molecules from residues, see “Building Molecules from Residues” on page 492.

## Amino Acids Dialog Box



HyperChem provides these amino acid residues:

Ala	Alanine, neutral aliphatic, nonpolar.
Gly	Glycine, neutral, polar.
Ser	Serine, neutral polar.

Thr	Threonine, neutral, polar.
Leu	Leucine, neutral, aliphatic, nonpolar.
Ile	Isoleucine, neutral, aliphatic, nonpolar.
Val	Valine, neutral, aliphatic, nonpolar.
Asn	Asparagine, neutral, polar.
Gln	Glutamine, neutral, polar.
Arg	Arginine, positively charged.
Hid	Histidine, neutral, with hydrogen in the delta nitrogen.
Hie	Histidine, neutral with hydrogen in the epsilon nitrogen.
Hip, His	Histidine, positively charged, with a hydrogen on both the delta and epsilon nitrogens.
Trp	Tryptophan, neutral aromatic, nonpolar.
Phe	Phenylalanine, neutral aromatic, nonpolar.
Tyr	Tyrosine, neutral aromatic, polar.
Glu	Glutamic acid, negatively charged.
Asp	Aspartic acid, negatively charged.
Lys	Lysine, positively charged.
Pro	Proline, neutral, nonpolar. <sup>1</sup>
Cys	Cysteine, neutral, polar.
Cyx	Half-cystine, neutral, polar. Cyx in a molecule from a PDB file is a full (cross-linked) cystine residue. Cyx stands for a whole cystine residue if a molecule comes from a PDB file.
Met	Methionine, neutral, nonpolar.
Ace	Acetyl, N-terminal blocking group.
Nme	N-methyl amino; C-terminal blocking group.

HyperChem gives you these choices for the conformation of the next residues that you add. Each amino acid that you add to a

1. When you incorporate proline into a polypeptide, HyperChem cannot use the Phi or Psi torsional angles in this dialog box. Instead, it uses the angles required by the pyrrolidine ring.



polypeptide can have a different secondary conformation. In this way, the polypeptide can have stretches of alpha-helical structure, beta-pleated structure, and other transitional conformations.

**Alpha helix** Sets standard Phi and Psi torsional angles of  $-58$  and  $-47$  degrees for an alpha helix, with a transpeptide bond (Omega = 180 degrees).

**Beta sheet** Sets standard Phi and Psi torsional angles of 180 and 180 degrees for a beta pleated sheet, with a transpeptide bond (Omega = 180 degrees).

Conformation	Phi	Psi	Other
Alpha	-58.00	-47.00	180
Beta	180	180	180
Other	$\pm 180$	$\pm 180$	$\pm 180$

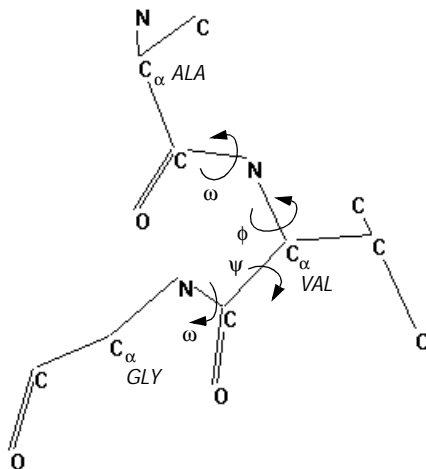
**Other** Accepts nonstandard values for Phi, Psi, and Omega torsional angles.

**Phi** Torsional angle ( $\phi$ ) for C-N-C $\alpha$ -C. Range:  $\pm 360$  degrees.

**Psi** Torsional angle ( $\psi$ ) for N-C $\alpha$ -C-N. Range:  $\pm 360$  degrees.

**Omega** The peptide bond torsional angle ( $\omega$ ), normally trans, from 107 to 180 degrees. Applies to the next amino acid residue. Range:  $\pm 360$  degrees.

**Isomer** Lets you choose the usual L stereoisomer (S chirality) or the D stereoisomer (R chirality).



*Note:* The Amino Acids dialog box stays on screen. You can leave this dialog box in the workspace while using tools or other menu items (for example, Labels on the Display menu).

## Make Zwitterion

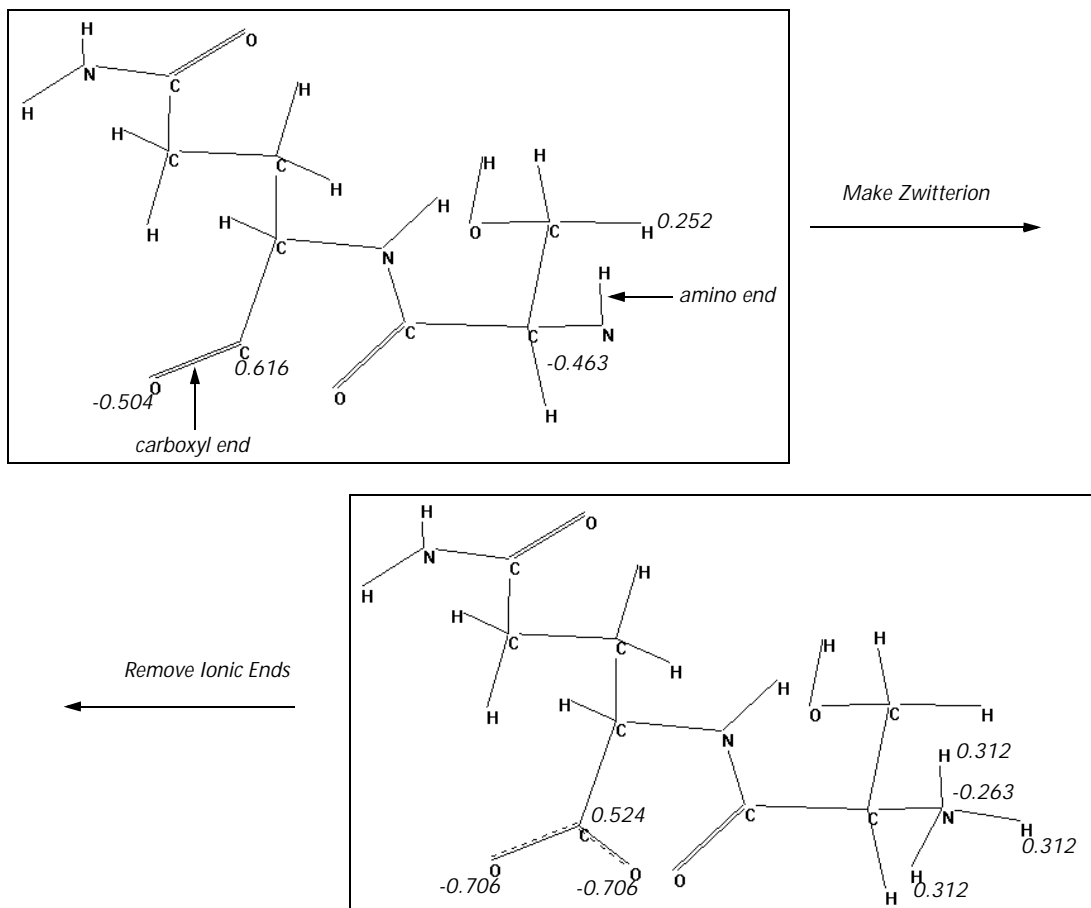
At neutral pH in solution, polypeptides exist as zwitterions, with  $-\text{NH}_3^+$  and  $-\text{COO}^-$  terminal groups. Make Zwitterion converts a selected polypeptide, constructed from the Databases menu or a PDB file, into a zwitterion. If there are no selected molecules, Make Zwitterion acts on all polypeptides in a molecular system. The polypeptide must have the usual terminal groups ( $-\text{NH}$  and  $-\text{CO}$ ), without terminal caps (Ace or Nme).

When you construct polypeptides with HyperChem, the ends are incomplete ( $-\text{N}-\text{H}$  and  $-\text{C}-\text{O}$ ). Make Zwitterion converts the ends to  $-\text{NH}_3^+$  and  $-\text{COO}^-$ .

*Note:* Amino acid residues from a database (the Databases menu or a PDB file) are composed of atoms with partial charges. Make Zwitterion changes the partial charges on the terminal atoms. It does not change the charges on side groups of amino acids (for example, acidic, basic, or imidazole side groups).

Make Zwitterion prevents adding more amino acid residues to existing polypeptides. If you choose additional residues from the Amino Acids dialog box, they start a new polypeptide.

Remove Ionic Ends (see the next section) reverses the results of Make Zwitterion.



## Remove Ionic Ends

Remove Ionic Ends reverses the results of Make Zwitterion. This converts  $-\text{NH}_3^+$  and  $-\text{COO}^-$  terminal groups in a polypeptide to  $-\text{NH}$  and  $-\text{CO}$ .

# Sequence Editor

## Overview

The Sequence Editor is used to create or read in one-letter amino acid sequences, possibly from a sequence database such as those provided by the National Center for Biotechnology's *Entrez* system on CD-ROM (or on-line at <http://www.ncbi.nlm.nih.gov/entrez>). The primary sequence can then be assigned secondary structures that are either alpha helix, beta sheet, beta turn, or random coil. The three-dimensional structure can then be sent to HyperChem for molecular modeling. The Sequence Editor can also be used to view certain properties of a sequence, such as the percentage of individual residues, the hydrophobic character of residues, etc. It can also be used to view two sequences and compare them in a simple fashion.

The Sequence Editor module is the executable file, HCSE-QUEN.EXE, installed in your Program directory. Several sample files in FASTA format (FAST\*.SEQ) are provided in the Sequences sub-directory of the Samples directory.

## Tutorial

This tutorial illustrates how to generate a secondary structure for a small protein obtained from Entrez.

### Reading a Sequence for Modeling

1. Start the Sequence Editor. This may be done by clicking on the Sequence Editor menu item of the Databases menu in HyperChem.
2. Click on the menu item File/Open to open the file SAMPLES\SEQUENCES\FAST3.SEQ and see it displayed in the top Sequence window. The bottom window displays the secondary structure of the sequence. This sequence has not had a secondary structure assigned so that it is all random coil as designated by "r". Right-handed alpha helices are designated by "h", left-handed alpha helices by "l", 3-10 helices by "e", pi helices by "i", extended beta sheets by "x", parallel beta sheets by "p", anti-parallel beta sheets by "a", beta I turns by "12", beta II turns by "34", and beta III turns by "56".

3. Click on the menu item View/Polarity to display the polarity of each residue in the bottom window rather than the secondary structure. A non-polar residue is designated by "n", a polar residue by "p", a negatively-charged residue by "-", and a positively-charged residue by "+". Click on the menu item View/Secondary to return to viewing the secondary structure of the sequence.
4. Select a portion of the total sequence by dragging the mouse along the text in either window. The corresponding text in the other window will automatically be also selected. Click on the menu item Edit/Alpha helix and note that the bottom window shows that this portion of the sequence now has an alpha helical secondary structure.
5. Select other portions of the sequence and designate them as alpha helices, beta sheets, beta turns, or random coil as desired until the operation of interactively setting the secondary structure (displayed in the bottom window) of a primary sequence (displayed and manipulated in the top window) is mastered. Two residues must be selected for the beta turns. Note that all changes to the contents of the bottom window must be made through menu commands acting on selected portions of the sequence. This means that, for example, the secondary structure cannot be typed directly into the bottom window.
6. Click on the menu item HyperChem/Put to create your sequence and its secondary structure in HyperChem. Since this protein is rather large, and would take some time to send to HyperChem, press the Cancel menu item to stop this command. Once the protein corresponding to the sequence has been built by HyperChem, it can be used for visualization, analysis and modeling using other features of HyperChem. The random coil portion of the protein is built using a random number generator to specify each phi ( $\phi$ ), psi ( $\psi$ ) angle.

## Background

A subscription to *Entrez*, which includes sequence databases on CD-ROM, is available from:

National Center for Biotechnology Information  
 National Library of Medicine  
 National Institutes of Health, Bldg. 38A  
 8600 Rockville Pike  
 Bethesda, MD, USA 20894  
 Phone: (301) 496-2475  
 Email: [info@ncbi.nlm.nih.gov](mailto:info@ncbi.nlm.nih.gov)  
 Web: <http://www.ncbi.nlm.nih.gov/entrez>

Protein secondary structure is discussed in:

T. E. Creighton, *Proteins*, W.H. Freeman (New York, 1984)  
 ISBN 0-7167-1566-X, Chapters 5 and 6.

Dayhoff matrices are discussed in:

J. E. Ash, W. A. Warr, and P. Willett, *Chemical Structure Systems*, Ellis Horwood (New York, 1991) ISBN 0-13-126699-3,  
 Chapter 9.

## Reference

This section describes the menus and dialog boxes available in the Sequence Editor.

### File Menu

The File menu is used for reading and writing files and for copying the primary sequence to the bottom window. It contains these items:

<u>N</u> ew	Clears both windows and resets the Sequence Editor
<u>O</u> pen...	Selects a sequence file (FASTA format with SEQ as the extension) to be read in. If there is also a file with the same name but the extension SEC, then a secondary structure is read in as well. If there is no such file then the secondary structure is initialized as all random coil.

<b>S</b> ave	Saves the sequence in the top window as a FASTA sequence file (with extension SEQ). The secondary structure is saved in a file with the same name except for using SEC as the extension. This is the same as <b>Save As</b> when a file name has not been set.
<b>S</b> ave <b>A</b> s...	Use this menu to set a file name and then save the sequence in the top window and its secondary structure as in <b>Save</b> .
<b>T</b> ransfer	Use this menu to copy the sequence in the top window to the other (or second) sequence. This may be displayed in the bottom window via the menu item " <b>Display/Other Sequence</b> ". A subsequent reading or creation of a different sequence into the top window allows two sequences to be compared. The sequence that has been transferred to the bottom window cannot be transferred back to the top window.
<b>E</b> xit	Exits and terminates the Sequence Editor.

## **Edit Menu**

The Edit menu contains the standard Windows clipboard-related commands of Cut, Copy, Paste and Delete as well as commands for changing secondary structure. Each of the commands for secondary structure apply to the selected residues in the top window.

<b>C</b> ut	Removes selected residues from the top window and puts them into the clipboard.
<b>C</b> opy	Copies selected residues into the clipboard.
<b>P</b> aste	Pastes residues from the clipboard into the top window.
<b>D</b> elete	Removes selected residues from the top window without placing them in the clipboard.
<b>S</b> earch...	Searches for a specified subsequence in the sequence shown in the top window. The subsequence is specified by the dialog box brought up by this menu item.
<b>S</b> earch <b>N</b> ext	Continues the search for the next occurrence of the subsequence.
<b>A</b> lpha <b>H</b> elix	Changes the secondary structure of the selected residues to alpha helix (denoted "h") with $\phi=-58^\circ$ , $\psi=-47^\circ$ .

<b>Left-Handed Alpha Helix</b>	Changes the secondary structure of the selected residues to left-handed alpha helix (denoted "l") with $\phi=58^\circ$ , $\psi=47^\circ$ .
<b>3-10 Helix</b>	Changes the secondary structure of the selected residues to 3-10 helix (denoted "e") with $\phi=-49^\circ$ , $\psi=-26^\circ$ .
<b>Pi Helix</b>	Changes the secondary structure of the selected residues to pi helix (denoted "i") with $\phi=-57^\circ$ , $\psi=70^\circ$ .
<b>Extended Beta Sheet</b>	Changes the secondary structure of the selected residues to extended beta sheet (denoted "x") with $\phi=180^\circ$ , $\psi=180^\circ$ .
<b>Parallel Beta Sheet</b>	Changes the secondary structure of the selected residues to parallel beta sheet (denoted "p") with $\phi=-119^\circ$ , $\psi=113^\circ$ .
<b>Anti-Parallel Beta Sheet</b>	Changes the secondary structure of the selected residues to anti-parallel beta sheet (denoted "a") with $\phi=-139^\circ$ , $\psi=135^\circ$ .
<b>Beta I Turn</b>	Changes the secondary structure of two selected residues to beta I turn (denoted "12") with $\phi_1=-60^\circ$ , $\psi_1=-30^\circ$ , $\phi_2=-90^\circ$ , $\psi_2=0^\circ$ . This requires that two residues are selected.
<b>Beta II Turn</b>	Changes the secondary structure of two selected residues to beta II turn (denoted "34") with $\phi_1=-60^\circ$ , $\psi_1=120^\circ$ , $\phi_2=80^\circ$ , $\psi_2=0^\circ$ . This requires that two residues are selected.
<b>Beta III Turn</b>	Changes the secondary structure of two selected residues to beta III turn (denoted "56") with $\phi_1=-60^\circ$ , $\psi_1=-30^\circ$ , $\phi_2=-60^\circ$ , $\psi_2=-30^\circ$ . This requires that two residues are selected.
<b>Random Coil</b>	Changes the secondary structure of the selected residues to random coil (denoted "r"). If this is chosen then $\phi$ and $\psi$ are assigned random values for each residue when the system is sent to HyperChem.

## **View Menu**

The view menu contains these items:



<b><u>P</u>olarity</b>	Displays the polarity or hydrophobicity of each residue in the bottom window. A non-polar residue is designated "n", a polar residue as "p", a negatively-charged residue as "-", and a positively-charged residue as "+".
<b><u>S</u>econdary <u>S</u>tructure</b>	Displays the secondary structure of each residue in the bottom window. An alpha helical structure is designated by "h", a left-handed alpha helix by "l", a 3-10 helix by "e", a pi helix by "i", an extended beta sheet by "x", a parallel beta sheet by "p", an anti-parallel beta sheet by "a", a beta I turn by "12", a beta II turn by "34" a beta III turn by "56" and a random coil by "r". These secondary structures are realized when the sequence is put into Hyper-Chem.
<b><u>O</u>ther <u>S</u>equence</b>	Displays a second sequence, possibly the same sequence, in the bottom window. Only active after a <b>File/Transfer</b> operation has been performed. If a new sequence is read in (the top window), then the two sequences are different and can be compared in the two windows.
<b><u>B</u>ar <u>C</u>hart...</b>	Displays the Residue Percentages dialog box with a bar graph showing the percentage of each of the 20 amino acids in the sequence of the top window. Double Clicking on the bar graph places it into the clipboard.
<b><u>P</u>ie <u>C</u>hart...</b>	Displays the Residue Percentages dialog box with a pie chart showing the percentage of each of the 20 amino acids in the sequence of the top window. The information displayed is the same as for Bar Graph. Double clicking on the pie chart places it into the clipboard.
<b><u>D</u>ayhoff <u>M</u>atrix...</b>	This menu is only active after <b>File/Transfer</b> . It displays the Dayhoff Comparison Matrix dialog box.
<b><u>D</u>iagonal <u>P</u>lot...</b>	This menu is only active after <b>File/Transfer</b> . It displays the Sequence Length dialog box and then the XY Plot dialog box.

## HyperChem Menu

<b>C</b> lear	Equivalent to a <b>File/New</b> in HyperChem. Clears the workspace in HyperChem for a new protein to be sent to it.
<b>G</b> et	Gets the primary and secondary sequence of the system currently displayed in HyperChem. See below for a description of how the secondary structure is classified.
<b>P</b> ut	Sends a protein, described by its primary and secondary structure in Sequence Editor, to HyperChem. The secondary structure is used to create the 3D atom coordinates in HyperChem.

**Note:** **Put** will set all backbone dihedral angles to canonical values, except for random coil, and will use HyperChem's templates for all other internal coordinates. HyperChem structural information can be lost if you use **Get** followed by **Put**.

## Classifying Secondary Structure

After a protein is read from HyperChem via the menu HyperChem/Get, the secondary structure for the sequence is classified from the values of  $\phi$  and  $\psi$  in the protein. The following (somewhat arbitrary) rules are followed:

Structure	$\phi$	$\psi$	Other Restrictions
x	$165 \leq \phi \leq 195$	$165 \leq \psi \leq 195$	also minus
a	$-180 \leq \phi \leq -50$	$80 \leq \psi \leq 180$	$\phi - \psi > 253$ , not x or 34
p	$-180 \leq \phi \leq -50$	$80 \leq \psi \leq 180$	$\phi - \psi < 253$ , not x or 34
h	$-130 \leq \phi \leq -10$	$-90 \leq \psi \leq -10$	not e, i, 12 or 56

l	$10 \leq \phi \leq 90$	$10 \leq \psi \leq 130$	not 34
e	$-59 \leq \phi \leq -39$	$-36 \leq \psi \leq -16$	not 12 or 56
i	$-67 \leq \phi \leq -47$	$-80 \leq \psi \leq -60$	
1	$-80 \leq \phi \leq -40$	$-50 \leq \psi \leq -10$	next is 2 and previous is p, x, a
2	$-110 \leq \phi \leq -75$	$-15 \leq \psi \leq 20$	previous residue is 1
3	$-80 \leq \phi \leq -40$	$100 \leq \psi \leq 140$	next is 4, previous is p, x, or a, and residue is G
4	$60 \leq \phi \leq 100$	$-20 \leq \psi \leq 20$	previous is 3 and residue is G
5	$-80 \leq \phi \leq -40$	$-50 \leq \psi \leq -10$	next is 6 and previous is p, x, a
6	$-80 \leq \phi \leq -40$	$-50 \leq \psi \leq -10$	previous residue is 5
r	any	any	if is none of the above

## **Cancel Menu**

This command is only active during a HyperChem/Get or HyperChem/Put. Clicking on Cancel will stop Sequence Editor immediately, but will not destroy any of the information which has already been sent or received before the Cancel command is executed.

## **Help Menu**

<b>A</b> bout...	Invoke the About dialog box which displays version and copyright information for Sequence Editor.
------------------	---

## Dayhoff Comparison Dialog Box

The Dayhoff Comparison Matrix Dialog Box shows a matrix of values which qualitatively reflect the probability or ease of substitution of one amino acid with another (in nature). Positive values correspond to common substitutions, and negative values to rare ones. The values in this table can be edited by clicking on a value and typing in a substitute number.

Clicking on the Plot command button will display a three-dimensional bar graph of the values contained in the Dayhoff matrix. When the graph is displayed, clicking on the Exit Plot command button will return the display to the Dayhoff matrix.

Click on the Save command button to save any changes made to the values in the Dayhoff matrix. These values are kept in the file DAYHOFF.DAT in the LIBRARY directory.

## Sequence Length Dialog Box And XY Plot Dialog Box

The Sequence Length dialog box is used to specify the sequence lengths to be used when comparing the two sequences and the threshold Dayhoff score for displaying a point on a diagonal plot. Enter a sequence length as a number greater than one, and the Dayhoff score for success as any number. When you click OK, the XY Plot Dialog Box will be displayed.

The XY Plot dialog box displays a diagonal plot for the current two sequences, based on the current Dayhoff matrix. The module will then compare two fragments, one from each sequence, of length equal to the sequence length value entered above, adding the Dayhoff score for each pair of residues in the fragments, and if the score for these fragments are greater than the desired Dayhoff score, then a point is displayed on the diagonal plot, at a location corresponding to the residue numbers of the two fragments. Sections of the two protein sequences which are similar will produce a solid diagonal line on the plot. Sections which are dissimilar will produce empty space. Insertions or deletions in the sequence of one protein as compared to another will result in lines which are not located on the central diagonal of the plot, but will be above or below it.

## Files

Sequence files use the FASTA file format with SEQ as the file extension. This format is a text file with a single line of comments followed by one or more lines containing one or more one-letter residue labels. The list of residues may run over several lines, but should not contain characters other than valid one-letter amino acid labels. The valid labels are: A, C, D, E, F, G, H, I, K, L, M, P, Q, R, S, T, V, W and Y.

The secondary structure file, with the extension SEC, is a text file containing one character for each residue with the characters being h, l, e, i, x, p, a, 1, 2, 3, 4, 5, 6 or r. The file may have one or more lines. The number of characters should match the number of residues in the SEQ file with the same name except for the extension.

## Hints

You can create peptides with dihedral angles other than the canonical or random values if you also use the Amino Acids dialog box of HyperChem. Starting from the beginning of the sequence, select residues that you want to use canonical values for and then use HyperChem/Put to create those residues in HyperChem. For the residues that you want to have non-canonical or non-random values, switch to the HyperChem Amino Acids dialog box (displayed via the Databases/Amino Acids menu in HyperChem) and add those residues. You can switch back to Sequence Editor when you again want canonical angles.

You can also switch in the above way to use end groups or non-standard residues that you might have added to the HyperChem template files.

As with the HyperChem Amino Acids dialog box, only the repeating units are added. The ends can be "finished" either by starting and ending with appropriate blocking end groups (from the Amino Acid dialog box of HyperChem) or by using the Database/Make Zwitterion menu in HyperChem.

## Limitations

The diagonal plot should be used only for proteins with about 50 residues or more. With smaller proteins, there is not enough data to generate a meaningful plot.

You should note that the secondary structure obtained from the HyperChem workspace by the menu command HyperChem/Get will be only an approximation. In particular, the beta III turn is two residues of a 3-10 helix. Sequence editor distinguishes these two by looking at the previous residue. If it is a beta sheet, then the two residues are labeled '56'. If not, then they are labeled 'ee'. Also, since one of the backbone angles is unavailable for each of the end residues, their secondary structures are poorly defined.

Another example is glucagon, from the PDB file 1gcn (PDB1GCN.ENT in the HyperChem SAMPLES directory). Although most of this protein is alpha helix (approx. from T5 to M27), many of the backbone angles are far from the canonical values. Sequence Editor estimates the secondary structure to be: rhrrhrhrhrhrhrhrhrhrhr.

Isolated residues with h, l, e or i structure, for example: rrrhrrr, actually provide little useful information in general. It is common practice to not consider a section of protein to be in a helical conformation unless 4 or more consecutive residues have the necessary backbone angles, for example: hhhh.

## Nucleic Acids

Choose Nucleic Acids to construct polynucleotides (DNA and RNA) from nucleic acid residues. These residues have standard chemical structures.

In the Nucleic Acids dialog box, you can choose the secondary structure for added residues (A, B, Z, or Other for your own conformational angles).

You can create a polynucleotide, starting at the 5' or 3' end, by choosing from these nucleic acid residues:

- Four deoxyribonucleotides (monophosphate)
- Four ribonucleotides (monophosphate)
- A 3' cap to block elongation
- A 5' cap to block elongation

**Note:** Residues from this dialog box are not equivalent to nucleic acids that you construct atom by atom. You can add new residues to the Nucleic Acids dialog box or modify residues obtained from this dialog box (see "Adding New Residues" on page 739).

HyperChem links residues together in the order that you select them, starting from the 5' end. This is the natural order for DNA or RNA synthesis. If you want to build the polymer from the 3' end, turn Backward on (✓) in the dialog box.

Caps prevent further addition of residues to either the 5' or 3' ends. After adding a cap, additional residues begin another polynucleotide. If the polynucleotide has no cap, you can add more residues later.

As you extend a polynucleotide, you can change these properties of the next residues that you add:

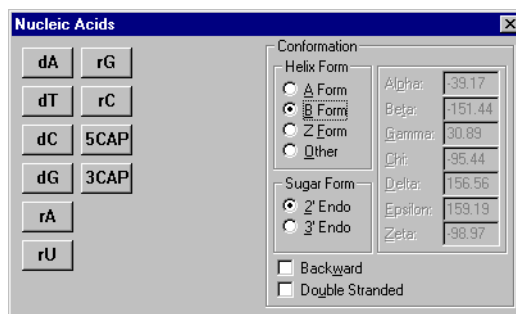
- Secondary conformation (A, B, Z, or Other).
- Either a 2' Endo or a 3' Endo pucker for the sugar ring. The A form of DNA normally contains deoxyribose sugar in the more stable C3'-endo form. The B form of DNA contains the C2'-endo sugar.
- Double-stranded. This adds the residue you select, plus its complementary, hydrogen-bonded, base. HyperChem is primarily concerned with base pairing; some bonds in the complementary strand might have incorrect bond lengths and some base pairs might not have the correct twist angles. Use geometry optimization to correct any errors in structure.

*Caution:* Do not use the Model Builder on a constructed polynucleotide. The Model Builder discards the geometry in these molecules and cannot accurately reproduce the structure.

These choices remain in effect until you make other choices. You cannot change these properties after you add the residues.

For detailed instructions on building molecules from residues, see "Building Molecules from Residues" on page 492.

## Nucleic Acids Dialog Box



HyperChem provides these nucleotide residues:

dA	Deoxyadenosine 5'-monophosphate, a component of DNA.
dT	Deoxythymidine 5'-monophosphate, a component of DNA.
dC	Deoxycytidine 5'-monophosphate, a component of DNA.
dG	Deoxyguanosine 5'-monophosphate, a component of DNA.
rA	Adenosine 5'-monophosphate, a component of RNA.
rU	Uridine 5'-monophosphate, a component of RNA.
rG	Guanosine 5'-monophosphate, a component of RNA.
rC	Cytidine 5'-monophosphate, a component of RNA.
5CAP	A cap (-OH) for terminating a strand at its 5' end.
3CAP	A cap (-OH) for terminating a strand at its 3' end.

HyperChem gives you these choices for the next residues that you add:

Helix Form	This gives the secondary conformation of polynucleotide strands. Each helix form has characteristic torsional angles, twist (amount of rotation for each residue), and pitch (the height of one turn).
------------	--



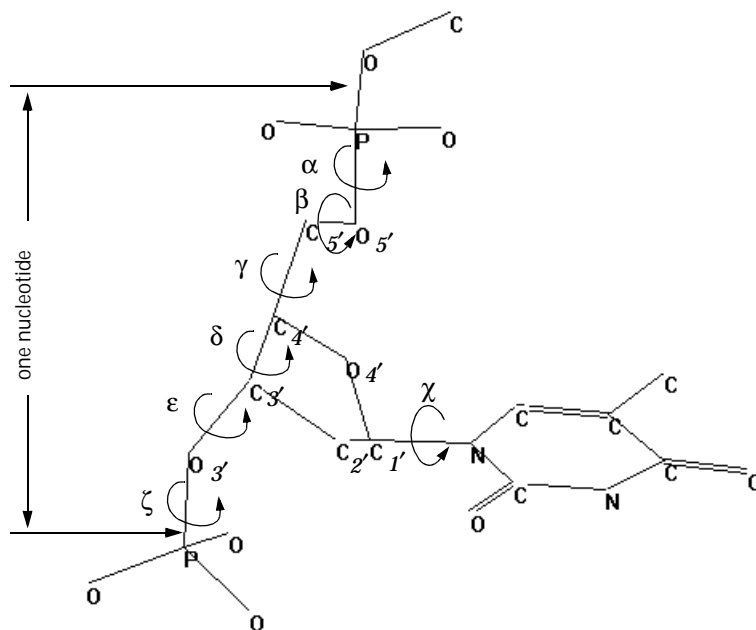
- A Form**      A right-handed helical conformation found in dehydrated, double-stranded DNA, with about 11 residues per turn.
- B Form**      A right-handed helical conformation found in hydrated, double-stranded DNA, with about 10 residues per turn.
- Z Form**      A left-handed helical conformation found in double-stranded DNA, with about 12 residues per turn. Alternate residues (odd- and even-numbered) on one strand of Z DNA have different torsion angles (see the next table).
- Other**        Accepts your values for torsion angles in the polynucleotide backbone (see following table).

Conf.	Alpha	Beta	Gamma	Chi	Delta	Epsilon	Zeta
A	-74.82	-179.1	58.9	-158.97	78.42	-154.98	-67.21
B	-39.17	-151.44	30.89	-95.44	156.56	159.19	-98.97
Z <sup>a</sup> (odd)	47	179	-165	68	99	-104	-69
Z <sup>a</sup> (even)	-137	-139	56	-159	138	-94	80
Other	± 180	± 180	± 180	± 180	± 180	± 180	± 180

a. Torsion angles for odd- and even-numbered residues in a strand of Z DNA.

- Alpha**      ( $\alpha$ ) Torsion angle (see diagram following). Acceptable range:  $\pm 360$  degrees.
- Beta**        ( $\beta$ ) Torsion angle (see diagram following). Acceptable range:  $\pm 360$  degrees.
- Gamma**      ( $\gamma$ ) Torsion angle (see diagram following). Acceptable range:  $\pm 360$  degrees.
- Chi**         ( $\chi$ ) Torsion angle (see diagram following). Acceptable range:  $\pm 360$  degrees.
- Delta**        ( $\delta$ ) Torsion angle (see diagram following). Acceptable range:  $\pm 360$  degrees.

- Epsilon** ( $\epsilon$ ) Torsion angle (see diagram following).  
Acceptable range:  $\pm 360$  degrees.
- Zeta** ( $\zeta$ ) Torsion angle (see diagram following).  
Acceptable range:  $\pm 360$  degrees.



**Sugar Form** Conformations of the ribose ring.

- 2' Endo**  $C_2'$  is out of the plane of the ring, on the same side as  $C_5'$ .
- 3' Endo**  $C_3'$  is out of the plane of the ring, on the same side as  $C_5'$ .

**Double Strand** When this option is on ( $\checkmark$ ), HyperChem adds the complementary residue, according to Watson-Crick base-pairing rules, and constructs an antiparallel polynucleotide. You can switch between double strand and single strand to construct partially base-paired molecules.

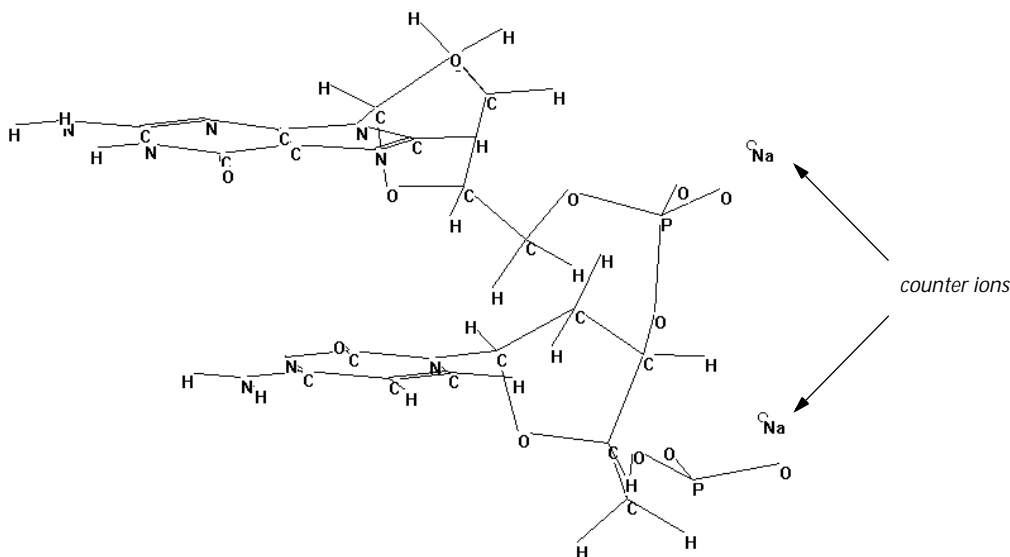
**Backward** When this option is on ( $\checkmark$ ), HyperChem starts polymerization from the 3' end, adding nucleotides to the 5' end. This is opposite to the normal direction of DNA or RNA synthesis. This choice

affects an entire polynucleotide and you must set it before starting a molecule.

**Note:** The Nucleic Acids dialog box stays on screen. You can leave this dialog box in the workspace while you use the Tools or other menu items (for example, Labels on the Display menu).

## Add Counter Ions

Add Counter Ions places a sodium ion near each phosphate group on a polynucleotide backbone. The  $\text{Na}^+$  ion is 1.688 Ångstroms equidistant from each oxygen atom in a phosphate group.



If you choose Add Counter Ions and then add more residues, only the original residues have counter ions. Choose Add Counter Ions again to add  $\text{Na}^+$  ions to the new residues.

**Note:** To remove  $\text{Na}^+$  ions, use Clear or Cut on the Edit menu.

If you store the system in a HIN file, the file also contains the counter ions.

# Crystal Builder

## Overview

Crystal Builder is a module for reading, creating and viewing crystal structures within HyperChem. These structures can be saved, edited or calculations can be performed on them, just as for any other HyperChem system.

The module works by writing a HIN file, and then sending a command to HyperChem to read in the file. This file is set to: HCCRYSTL.\*\*\* in your LIBRARY directory.

Crystal Builder features:

- Easy-to-use interface for building atomic crystals of the basic crystal types: cubic, tetragonal, orthorhombic, rhombohedral, hexagonal, monoclinic and triclinic, plus fcc, hcp and bcc.
- Sample crystal structures, such as NaCl, CsCl, CaF<sub>2</sub>, PbS, MgO, ZnS, etc.
- Ability to edit the atom data, including their fractional coordinates, before creating the crystal.
- Ability to view various crystal planes, defined by their Miller indices.
- Ability to read and write files in the Cambridge Structural Database format (FDAT) as well as HyperChem's HIN format.

The Crystal Builder executable file is HCCRYSTL.EXE, installed in your Program directory. The help file is HCCRYSTL.HLP.

## Tutorials

The following tutorials demonstrate some of the features of Crystal Builder.

### 1. Creating A Crystal Of Sodium Metal

This tutorial illustrates the procedure for creating a crystal structure of sodium:

1. Start HyperChem, then start Crystal Builder by selecting Crystals... from the Databases menu.

2. Under Crystal Type, click on Cubic to select a cubic unit cell.
3. Enter the number of unit cells:  $3 \times 3 \times 3$  by entering the value 3 into each of the three edit boxes.
4. Under Unit Cell Parameters, enter the unit cell length for Na: 4.225 in the first box next to "a, b, c =". The values of b and c will be automatically set to 4.225 also.
5. Enter the number of atoms in the Na (bcc) unit cell: 2 in the appropriate box.
6. Click on the Edit Atom Data... command button to bring up the Atom Data dialog box.
7. In the Atom Data dialog box, the data for atom number 1 should be displayed. Change the atom name from "-" to "NA1". Change the element to "Na".
8. Press the up arrow in the spin button or enter 2 as the atom number, and the data for atom 2 will be displayed. Change a, b and c each from 0 to 0.5. Change the atom name from "-" to "NA2". Click on the Close command button. The module will return to the main window.
9. Choose the menu command: HyperChem/Put to send the structure to HyperChem, or alternatively, press the Toolbar button with an arrow pointing towards a beaker of green fluid.
10. The  $3 \times 3 \times 3$  sodium crystal will appear in the HyperChem window.
11. To view the [1 1 1] crystal plane, choose the View/Crystal Face command in Crystal Builder or click on the View button of the toolbar. A dialog box will appear. Enter the value 1 into each of the edit boxes for h, k and l, and press OK, and HyperChem will rotate the crystal and set the clipping plane to display the [1 1 1] face. Use the Display/Rendering command in HyperChem to change the rendering to spheres (CPK model).
12. Save the unit cell by choosing the File/Save As... menu command. Enter a name for the new file: NA.CSD. Press OK, and the unit cell will be saved in the Cambridge Crystallographic Database format.
13. Save the entire crystal by clicking in the HyperChem window, and selecting the File/Save As... menu command in HyperChem. Save the file as: NA.HIN.

14. Clear Crystal Builder by choosing the File/New menu command. Read in the file NA.HIN with the File/Open command. The Open File dialog box will display a list of CSD files. Change to HIN format by clicking on the down arrow under "List Files of Type:" and clicking on "HyperChem (\*.HIN)". Then click on OK. Crystal Builder reads the data in the HIN file and from it makes a unit cell containing 54 atoms. Select the HyperChem/Put command to send this single unit cell crystal to HyperChem. It should look the same as the crystal created above in step 9.

## 2. Creating A Crystal Of Sodium Chloride

This tutorial illustrates the procedure for creating a crystal structure of sodium chloride:

1. Start HyperChem, then start Crystal Builder by selecting Crystals... from the Databases menu.
2. Press the Samples... command button to bring up the Sample Crystal Structures dialog box.
3. Select the crystal type NaCl by clicking on the appropriate button, then press OK.
4. Enter the number of unit cells:  $3 \times 3 \times 3$  by entering the value 3 into each of the three edit boxes. Do not edit the unit cell parameters: Crystal Builder has set them to appropriate values for NaCl.
5. Choose the menu command: HyperChem/Put to send the structure to HyperChem, or alternatively, press the Toolbar button picturing an arrow pointing towards a beaker of green fluid.
6. The  $3 \times 3 \times 3$  NaCl crystal will appear in the HyperChem window.
7. To view the [1 1 1] crystal plane, choose the View/Crystal Face command in Crystal Builder or click on the View button of the toolbar. A dialog box will appear. Enter the value 1 into each of the edit boxes for h, k and l, and press OK, and HyperChem will rotate the crystal and set the clipping plane to display the [1 1 1] face. Use the Display/Rendering command in HyperChem to change the rendering to Spheres (CPK model).

Please see the Reference section for more detailed descriptions of the various commands and dialog boxes.

## Background

The structure of a crystal is that of an infinitely repeating unit cell. Thus, to describe a crystal structure, it is only necessary to specify the structure of a unit cell. A unit cell has the following components:

a) *shape*: defined by the lengths of its sides (**a**, **b**, **c**) and its internal angles  $\alpha$  (the angle between the **b** and **c** directions),  $\beta$  (the angle between the **a** and **c** directions) and  $\gamma$  (the angle between the **a** and **b** directions).

b) *contents*: defined by the locations of the atoms within the unit cell. These locations are usually reported in fractional coordinates ( $x$ ,  $y$ ,  $z$ ) where each value is between zero and one, in units of (**a**, **b**, **c**).

Crystal Builder can build the following crystal types:

<b>Cubic</b>	The cubic crystal unit cell has $a = b = c$ and $\alpha = \beta = \gamma = 90^\circ$ .
<b>Tetragonal</b>	The tetragonal unit cell has $a = b \neq c$ and $\alpha = \beta = \gamma = 90^\circ$ .
<b>Orthorhombic</b>	The orthorhombic unit cell has $a \neq b \neq c$ and $\alpha = \beta = \gamma = 90^\circ$ .
<b>Rhombohedral</b>	The cubic crystal unit cell has $a = b = c$ and $\alpha = \beta = \gamma \neq 90^\circ$ .
<b>Hexagonal</b>	The hexagonal unit cell has $a = b \neq c$ and $\alpha = \beta = 90^\circ$ , $\gamma = 120^\circ$ .
<b>Monoclinic</b>	The monoclinic unit cell has $a \neq b \neq c$ and $\alpha = \gamma = 90^\circ \neq \beta$ .
<b>Triclinic</b>	The triclinic unit cell has $a \neq b \neq c$ and $\alpha \neq \beta \neq \gamma \neq 90^\circ$ .
<b>Hexagonal Close-Packed</b>	The hexagonal close-packed (hcp) unit cell has $a = b$ , $c = a \cdot \sqrt{8/3}$ , $\alpha = \beta = 90^\circ$ , and $\gamma = 120^\circ$ , with 2 atoms per unit cell. Crystal Builder uses an orthorhombic non-primitive unit cell to represent hcp, with 4 atoms per unit cell, $b = a \cdot \sqrt{3}$ , $c = a \cdot \sqrt{8/3}$ , and $\alpha = \beta = \gamma = 90^\circ$ .
<b>Face-Centered Cubic</b>	The face-centered cubic (fcc, ccp) crystal type has 4 atoms per unit cell, with $a = b = c$ and $\alpha = \beta = \gamma = 90^\circ$ .

<b>Body-Centered Cubic</b>	The body-centered cubic (bcc) crystal type has 2 atoms per unit cell, with $a = b = c$ and $\alpha = \beta = \gamma = 90^\circ$ .
----------------------------	---

The crystal information used in Crystal Builder comes from the following sources:

F. A. Cotton, and G. Wilkinson, *Advanced Inorganic Chemistry*, 5<sup>th</sup> Ed., John Wiley & Sons Ltd. (New York, 1988) ISBN 0-471-84997-9, Chapter 1.

J. D. H. Donnay, Editor, *Crystal Data, Determinative Tables*, 2<sup>nd</sup> Ed., American Crystallographic Association (Washington, 1963).

C. Kittel, *Introduction to Solid State Physics*, 5<sup>th</sup> Ed., John Wiley & Sons (New York, 1976) ISBN 0-471-49024-5, Chapter 1.

I. N. Levine, *Physical Chemistry*, 3<sup>rd</sup> Ed., McGraw-Hill (New York, 1988) ISBN 0-07-037474-0, Chapter 24.

H. D. Megaw, *Crystal Structure*, Saunders (1973).

C. N. R. Rao, Editor, *Solid State Chemistry*, Marcel Dekker (New York, 1974) Chapter 1.

A. F. Wells, *Structural Inorganic Chemistry*, 5<sup>th</sup> Ed., Clarendon Press (Oxford, 1984) ISBN 0-19-855370-6.

## Reference

This section describes the details of the main window, menus, dialog boxes and the toolbar.

## Main Window

Use the main window to specify the type of crystal to be created in HyperChem.

<b>Crystal Type</b>	Click on the button next to the type of crystal you wish to create. (See Background section for a description of the available crystal types.)
<b>Number of unit cells</b>	Enter the desired number of unit cells in each direction.



<b>Place Atoms inside Periodic Box</b>	Click to toggle whether or not to place the atoms of the crystal inside a periodic box in HyperChem. Not available for rhombohedral, hexagonal, monoclinic or triclinic unit cells.
<b>a, b, c</b>	Enter the unit cell lengths in Ångstroms.
<b><math>\alpha</math>, <math>\beta</math>, <math>\gamma</math></b>	Enter the unit cell angles in degrees.
<b>Atoms per unit cell</b>	Enter the number of atoms per unit cell. This number is not adjustable if one of the hcp, fcc or bcc unit cell types is selected.
<b>Edit Atom Data</b>	Press this command button to activate the Atom Data dialog box, where you can enter information about each atom in the unit cell.
<b>Samples...</b>	Press this command button to invoke the Sample Crystal Structures dialog box, where you can select one of the sample crystals for display.

## File Menu

Use the File menu for tasks involving the handling of files. The File menu contains these items:

<b><u>N</u>ew</b>	This command deletes any previous values for the atoms per unit cell and atomic positions.
<b><u>O</u>pen...</b>	This command brings up the File dialog box, where you can select a file (containing unit cell data) to load into Crystal Builder. Crystal Builder can read and write files in HIN format or Cambridge Structural Database (CSD) format.
<b>Save <u>A</u>s...</b>	This command brings up the File Save dialog box, where you can select a file name into which the unit cell data will be saved. Crystal Builder can read and write files in HIN format or Cambridge Structural Database format.
<b><u>E</u>xit</b>	Use this command to exit Crystal Builder.

## View Menu

Use the View menu to control the direction of view and rendering of your crystal in HyperChem. The View menu contains these items:

<b>C</b> ystal Face...	This command brings up the View dialog box in which the user can enter the Miller indices of a crystal face to be viewed.
<b>D</b> isks	This command allows the user to specify disks rendering for the <i>next</i> crystal. The change only takes effect when a new crystal is created. To change the rendering of the crystal currently displayed in HyperChem, use the Display/Rendering command in HyperChem.
<b>S</b> pheres	This command allows the user to specify spheres rendering for the <i>next</i> crystal. The change only takes effect when a new crystal is created. To change the rendering of the crystal currently displayed in HyperChem, use the Display/Rendering command in HyperChem.
<b>P</b> eriodic Box	This command toggles the display of the periodic box in HyperChem (if present). The change only takes effect when a new crystal is created. To toggle the display or hiding of the periodic box in HyperChem with immediate effect, use the Display/ Show Periodic Box command in HyperChem. Note that HyperChem can only display cubic, tetragonal and orthorhombic boxes.
<b>T</b> oolbar	This command toggles the display of the Toolbar. Each button on the Toolbar serves as a shortcut to the Menu commands. See below for a complete description.

## HyperChem Menu

Use the HyperChem menu to exchange atoms and crystals with HyperChem. The HyperChem menu contains these items:

<b>G</b> et	Receive from HyperChem the atom positions and box lengths of the currently displayed system, and create a crystal unit cell from them.
<b>P</b> ut	Send the crystal that is currently specified in Crystal Builder to HyperChem.

## Help Menu

Use the Help menu to gain access to the on-line Help system for Crystal Builder. The Help menu contains these items:

<b>C</b> ontents	Use this command to display the contents page for on-line help in Crystal Builder.
<b>O</b> verview	Use this command to display an overview of the Crystal Builder module, including an example session.
<b>U</b> sing Help	Use this command to display help on how to use on-line help.
<b>A</b> bout...	This command activates the About dialog Box, which will display information about your copy of Crystal Builder.

## Atom Data Dialog Box

Pressing the Edit Atom Data command button on the main window brings up the Atom Data dialog box. Use this dialog box to enter data about the atoms contained in the unit cell.

<b>atom number</b>	Enter the number of the atom you wish to edit, or press the up/down arrows on the spin control to increment/decrement the atom number.
<b>a, b, c</b>	Enter the atom's position within the unit cell in fractional coordinates. Each value should be greater than or equal to zero and less than one. (Crystal Builder allows a greater range of values for CSD files to allow for symmetry-related positions.)
<b>Atom Name</b>	Enter a name for the atom, or accept the default value of "-". The name can have up to four characters, and must not contain any spaces or quotes.
<b>Element</b>	Enter the atomic symbol for the element (up to three characters).

<b>A</b> tom <b>T</b> ype	Enter the MM+ atom type for this atom (up to four characters). The default is "***", which means an unknown type. If you wish to perform a calculation in HyperChem on a crystal, but are not sure of appropriate atom types, build the crystal without types, then use the Build/Calculate Types command in HyperChem to assign types.
<b>C</b> harge	Enter the charge associated with the atom, in units of electron charge. These charges will be used by molecular mechanics calculations, and do not need to be integers.
<b>C</b> lose	Use this command button to stop editing atom data and exit this dialog box.
<b>H</b> elp...	Invoke On-line Help about the Atom Data dialog box.

## Samples Dialog Box

Clicking on the Samples... button in the main window brings up the Samples dialog box. This dialog box lists a number of sample crystal structures which can be loaded into HyperChem. To use this dialog box: select a crystal type and press OK.

The following crystal structures are available:

<b><u>N</u>aCl</b>	The NaCl (rock salt) crystal has a cubic unit cell with $a = 5.63 \text{ \AA}$ , 2 atom types and 8 atoms per unit cell: four of each element. The atoms of each type form a bcc lattice.
<b><u>C</u>sCl</b>	The CsCl crystal has a cubic unit cell with $a = 4.11 \text{ \AA}$ and 2 atoms per unit cell: one of each type.
<b><u>C</u>aF<sub>2</sub></b>	The CaF <sub>2</sub> (fluorite) crystal has a cubic unit cell with $a = 5.43 \text{ \AA}$ and 12 atoms per unit cell.
<b><u>D</u>iamond</b>	The diamond crystal has 8 atoms per unit cell, with $a = 3.56 \text{ \AA}$ . In diamond, the atoms are linked by covalent bonds. In Crystal Builder, these bonds have been omitted.
<b><u>P</u>bS</b>	PbS (galena) crystallizes in an NaCl-type structure, with $a = 5.92 \text{ \AA}$ .
<b><u>M</u>gO</b>	MgO crystallizes in an NaCl-type structure, with $a = 4.2 \text{ \AA}$ .
<b><u>C</u>uZn (beta <u>b</u>rass)</b>	CuZn (beta brass) crystallizes in a CsCl-type structure, with $a = 2.94 \text{ \AA}$ .

<b>RbCl (190°C)</b>	RbCl crystallizes in a CsCl-type structure at 190°C, with $a = 3.74 \text{ \AA}$ .
<b>ZnS (zincblende)</b>	ZnS (zincblende) crystallizes in a cubic unit cell with 8 atoms and $a = 5.4093 \text{ \AA}$ .
<b>ZnS (wurtzite)</b>	ZnS (wurtzite, sphalerite) crystallizes in a hexagonal unit cell with 8 atoms per unit cell, and $a = 3.823 \text{ \AA}$ , $c = 6.2565 \text{ \AA}$ .
<b>AgI</b>	AgI crystallizes in a zincblende-type structure, with $a = 6.47 \text{ \AA}$ .
<b>CuCl</b>	CuCl crystallizes in a zincblende-type structure, with $a = 5.41 \text{ \AA}$ .
<b>Argon</b>	Ar (argon) crystallizes in an fcc lattice, with $a = 5.31 \text{ \AA}$ at 4°K.
<b>Copper</b>	Cu (copper) crystallizes in an fcc lattice, with $a = 3.61 \text{ \AA}$ .
<b>Gold</b>	Au (gold) crystallizes in an fcc lattice, with $a = 4.0786 \text{ \AA}$ .
<b>Sodium</b>	Na (sodium) crystallizes in a bcc lattice, with $a = 4.225 \text{ \AA}$ .
<b>NbO</b>	NbO (niobium oxide) crystallizes in a NaCl-type structure with vacancies at the corners and at the cell center, with 6 atoms per unit cell, and $a = 4.2108 \text{ \AA}$ .
<b>Zinc</b>	Zinc (Zn) crystallizes with the hcp structure, with $a = 2.66 \text{ \AA}$ , $c = 4.94 \text{ \AA}$ .
<b>Rutile</b>	Rutile ( $\text{TiO}_2$ ) has 6 atoms per unit cell, with $a = 4.5937 \text{ \AA}$ , $c = 2.9618 \text{ \AA}$ . The Ti atoms form a body-centered tetragonal crystal array, with each Ti surrounded by an octahedron of O ions. Each O ion is surrounded by 3 Ti atoms in a plane.
<b>Spinel</b>	Spinel ( $\text{MgAl}_2\text{O}_4$ ) has 56 atoms per unit cell, with $a = 8.124 \text{ \AA}$ . The O atoms form a bcc lattice, with 1/8 of the tetrahedral holes (there are 2 per O atom) containing Mg atoms, and 1/2 of the octahedral holes (there is 1 per O atom) containing Al atoms.
<b>Perovskite</b>	Perovskite ( $\text{CaTiO}_3$ ) has 5 atoms per unit cell, with $a = 5.367 \text{ \AA}$ , $c = 5.4439 \text{ \AA}$ . The Ca and O atoms form a bcc lattice, with Ti atoms in the octahedral holes between O atoms.
<b>PtS</b>	PtS (platinum sulfide) crystallizes in a tetragonal unit cell with 8 atoms per cell, and $a = 3.47 \text{ \AA}$ , $c = 6.1096 \text{ \AA}$ .

<b>OK</b>	Accept the current crystal type, and exit this dialog box.
<b>Help...</b>	Invoke help about the Samples dialog box.

## View Dialog Box

Choosing View/Crystal Face from the menu bar in the main window, or clicking on the View button on the toolbar, brings up the View dialog box. Use this dialog box to choose a crystal face for display by entering the values of its Miller indices [h k l]. Crystal Builder will then rotate the crystal in the HyperChem display, and set the clipping plane to pass through the center of the displayed system, so that the desired crystal face is shown.

## Choose Structure Dialog Box


If the Cambridge Structural Database file which has been chosen for reading has more than one structure within it, then the Choose Structure Dialog Box will appear. Choose which structure to read by clicking on its name in the list, and then pressing the OK button.










## About Dialog Box

Choosing Help/About from the menu bar in the main form brings up the About dialog box. This form displays information about your copy of Crystal Builder.

## Toolbar

This section provides an explanation for each of the Toolbar Command Buttons in Crystal Builder. The toolbar command buttons provide shortcuts to commands accessible in the menus.

Button	Corresponding Menu Command:	Action:
	File/New	Clears unit cell.

	File/Open	Open HIN or CSD file.
	File/Save As	Save unit cell as HIN or CSD.
	(none)	Switch to sticks rendering.
	View/Disks	Switch to disks rendering.
	View/Spheres	Switch to spheres rendering.
	View/Crystal Face	Invoke View dialog box.
	HyperChem/Get	Get unit cell from HyperChem.
	HyperChem/Put	Send crystal to HyperChem.
	Help/Contents	Start the on-line help, in the Contents section.

## Hints

1. Remember that the size of a crystal will increase rapidly with the number of unit cells.
2. If you build a crystal by entering atomic positions, Crystal Builder does not guess where bonds should go. You can, however, use HyperChem's facilities to make a guess at the bonding in your structure. Once you have put your structure into HyperChem, save it as a Z-matrix (ZMT) file, and read this file in again. The HyperChem Z-matrix file reader includes a bond-inferring routine. HyperChem also allows you to estimate the position of hydrogen atoms, often not present in crystal structures, by using the Add hydrogens item under the Build menu.

## Sugar Builder

### Overview

The Sugar Builder will allow you to build polysaccharide structures in a manner similar to the amino acids and nucleic acids databases of HyperChem. You may create your own residues, which need not be saccharides, and use Sugar Builder to link them together.

Sugar Builder features:

- Templates for 32 common monosaccharides including aldoses and ketoses, including D, L, alpha, beta and acyclic forms.
- Ability to add common blocking groups, including OAc and NAc.
- Ability to chain monosaccharides together to form polysaccharides.
- A parameterized AMBER force field for polysaccharides (aldoses).
- Ability to connect two chains together so as to form branched polysaccharides or proteoglycans. Two chains of any type of residue can be linked, so this could even be used to create a branched polypeptide.
- Ability to add up to 21 user-defined residues to the database.



The following files are included with your software distribution:

HCSUGAR.EXE	The Sugar Builder module, in your Program directory.
*.HIN	A number of hin files, each of which contains a monosaccharide residue. These are stored in the LIBRARY subdirectory of your HyperChem directory.
AM90S*.TXT	Parameter files for the polysaccharide force field, held in the same directory as your other HyperChem parameter files.
AMBERS.PAR	A compiled parameter file for the polysaccharide force field, held in the same directory as your other HyperChem parameter files. If you change <b>AMBERTYP.TXT</b> to other than the original supplied with HyperChem, you must recompile this file.

## Tutorial

This tutorial illustrates the procedure for creating poly(1→4)- $\alpha$ -D-Glucose:

1. Start HyperChem, then start Sugar Builder. This may be done by clicking on the Saccharides menu item of the Databases menu in HyperChem.
2. Use the Add/Aldoses menu command to bring up the Aldoses dialog box.
3. Select the D and  $\alpha$  options to select the D isomer and  $\alpha$  anomer.
4. Choose the Glucose command button to create a glucose residue in HyperChem.
5. Choose the 1→4 connection type by selecting it from the list.
6. Change  $\phi$  and  $\psi$  to 48.05 and -20 by entering these numbers in their edit boxes, respectively.
7. Repeatedly click on the Glucose command button, once for each residue you wish to add to the chain.
8. Switch over to HyperChem and view the polysaccharide.

For more detailed information, please see the Reference section for descriptions of the various commands and dialog boxes.

## Background

Polysaccharides are naturally occurring macromolecules containing building blocks of cyclic sugars connected by O-glycosidic linkages. The building blocks, or monomers of polysaccharides are called monosaccharides. The most common monosaccharide in nature is D-glucose ( $\text{OCH}(\text{HCOH})_4\text{CH}_2\text{OH}$ ). The basic monosaccharides all have the formula  $(\text{CH}_2\text{O})_n$ , where  $n \geq 3$ . The carbon atoms form a linear chain, all but one of them connected to an OH group. The one without an OH is bonded to a carbonyl oxygen instead. The location of the carbonyl in the chain determines the type of residue: if it is at the end of the chain then the monosaccharide is an aldehyde derivative, and called an aldose; if it is not at the end, then it is a ketone derivative, and called a ketose. The carbon atoms of the chain are numbered 1 to  $n$ , starting from the end closest to the carbonyl group.

Monosaccharides can exist in a cyclic form, known as a hemiacetal or hemiketal. In the case of glucose, the alcoholic hydroxyl group at carbon atom 5 can react with the aldehydic carbon atom 1 to form a 6-membered ring structure known as a pyranose. The pyranose form of D-glucose is called D-glucopyranose. For ketoses, the carbonyl carbon 2 may react with the alcoholic hydroxyl at carbon atom 5 to form a five-membered ring, called a furanose. The cyclization to pyranoses and furanoses creates an extra chiral center at carbon 1. Thus, cyclic monosaccharides can exist as two different stereo isomers, or anomers, known as  $\alpha$  and  $\beta$ , and carbon 1 is known as the anomeric carbon.

The linking of monosaccharide units to form polysaccharides always occurs through the anomeric carbon. The residue at the end of the chain containing an unlinked anomeric carbon is known as the reducing sugar, or the reducing end. The anomeric carbon atom of the reducing end can react with any one of the hydroxyl groups of another monosaccharide, to form what is called an O-glycosidal linkage. The linkage is named after the numbers of the associated carbon atoms, for example, if the anomeric carbon of one aldose residue reacts with the hydroxyl group at carbon atom number 4 of a second residue (a common type of linkage), then this link is called 1 $\rightarrow$ 4. So, for example, maltose, which has two  $\alpha$ -D-glucopyranose residues linked 1 $\rightarrow$ 4, is also

known as O- $\alpha$ -D-glucopyranosyl-(1 $\rightarrow$ 4)- $\alpha$ -D-glucopyranose. Starch (amylose) is a long unbranched chain of 1 $\rightarrow$ 4 linked  $\alpha$ -D-glucose residues.

Although monosaccharides have only one reducing end, they do have several locations which can link to the reducing end of another monosaccharide, and this leads to branched chains. Amylopectin is an example: the backbone is made of 1 $\rightarrow$ 4 linked poly- $\alpha$ -D-glucose, and at random points along the chain are branch points where the reducing end of another poly-(1 $\rightarrow$ 4)- $\alpha$ -D-glucose chain has linked to carbon 6, forming an  $\alpha$ (1 $\rightarrow$ 6) link.

Most of the information used to create this module was taken from:

G. O. Aspinall, Editor, *The Polysaccharides, Vol. 1*, Academic Press (New York, 1982) ISBN 0-12-065601-9 (v. 1).

A. L. Lehninger, *Biochemistry, 2<sup>nd</sup> Ed.*, Worth Publishers Inc. (New York, 1979) ISBN 0-87901-047-9, Chapter 10.

The AMBER force field for polysaccharides (aldoses) used here is described in: S. W. Homans, *Biochemistry* **29**, 9110 (1990).

## Reference

This section describes the main window, menus and dialog boxes of the Sugar Builder module.

### Main Window

The main Sugar Builder window contains menus of commands (see below for descriptions), text boxes showing the number of residues which have so far been added to HyperChem, and the sequence of the residues added to HyperChem.

### File Menu

The File menu is used for clearing HyperChem, invoking the Link dialog box and restoring the HyperChem display. It contains these items:

---

<b>New</b>	Clears the main window and sends a File/New command to HyperChem.
------------	---

---

<b><u>L</u>ink Molecules</b>	Invokes the Link dialog box, in which you can select the two atoms which will link two molecules together. This may be used to link any two atoms using a geometry that you can specify.
<b><u>R</u>efresh Screen</b>	Restores the display of the HyperChem workspace. Use this to redisplay the HyperChem workspace if a DDE or other error occurs in Sugar Builder, and the display in HyperChem remains disabled. (Sugar Builder temporarily disables the HyperChem display for improved performance when using DDE.)
<b><u>E</u>xit</b>	Use this command to exit the Sugar Builder module.

## **Add Menu**

The Add menu is used to bring up the dialog boxes for choosing and simple linking of saccharides, end groups and user-defined residues.

<b><u>A</u>ldoses</b>	Starts the Aldoses dialog box, containing buttons for creating and linking aldose residues.
<b><u>K</u>etoses</b>	Starts the Ketoses dialog box, containing buttons for creating and linking ketose residues.
<b><u>D</u>erivatives</b>	Starts the Derivatives dialog box, containing buttons for creating and linking various residues of saccharide derivatives.
<b><u>E</u>nd Groups</b>	Starts the Blocking Groups dialog box, containing buttons for creating and linking various blocking groups to replace the OH groups of saccharide residues.
<b><u>U</u>ser</b>	Starts the User dialog box, containing buttons for adding user-defined residues.

The following residues are available in the D isomeric form only:

2-deoxy-D-ribose	allose	altrose
apiose	erythrose	erythrulose
fructose	galactosamine	galactose
glucosamine	glucose	gulose
idose	lyxose	mannose

N-acetyl-galac- tosamine	N-acetyl-glu- cosamine	psicose
ribose	ribulose	sorbose
tagatose	talose	threose
xylose	xylulose	

The following residues are available in the L isomeric form only:

fucose	rhamnose
--------	----------

The following residues are available in the both the D and L isomeric forms:

arabinose	galactose
-----------	-----------

It is possible to create the missing isomer of one of the above residues so that Sugar Builder will recognize it, without using up one of the user-defined buttons.

The following example will illustrate how to create an L-glucose residue file:

1. Read the file ADGLC.HIN in the LIBRARY directory into HyperChem.
2. Shift-click with the HyperChem drawing tool on the atom named "C1" (atom number 6) to change its chirality.
3. Click on File/Save As... to save the file. Give the new file the name: "ALGLC.HIN", and edit the comment field to say: "alpha-L-glucofuranose".
4. The L-glucose residue will now be available from the Add Aldoses dialog box.

## **Help Menu**

This menu is used to access the About dialog box.

---

<b>About</b>	This command activates the About dialog box, which will display version and copyright information about Sugar Builder.
--------------	--

---

## Aldoses Dialog Box

This dialog box contains command buttons to create the following saccharide residues: glucose, galactose, mannose, allose, altrose, gulose, idose, talose in aldopyranose form, plus ribose, arabinose, xylose and lyxose in aldofuranose form, and erythrose and threose in open-chain form. The dialog box will stay in place until you dismiss it, by using the command button in the top left corner of the window or by using ALT+F4 on the keyboard.

<b>Pyranoses</b>	Command buttons for sending an aldopyranose residue to HyperChem. The residue will have the conformation currently specified in the Conformation area of the dialog box.
<b>Furanoses</b>	Command buttons for sending an aldofuranose residue to HyperChem. The residue will have the conformation currently specified in the Conformation area of the dialog box.
<b>Acyclics</b>	Command buttons for sending an acyclic residue to HyperChem. These residues can be linked together by using the <b>Link/Molecules</b> menu item.
<b>Conformation</b>	Enter the conformation of the residue to be added, including: acyclic, $\alpha$ , or $\beta$ .
<b>Isomer</b>	Choose either the D or L isomer. Some rare L isomers are not available, but templates for these can be easily created from the D isomers (see below for details).
$\phi$ , $\psi$ , $\omega$	Enter the orientation angles. $\phi$ is the $H_1-C_1-O_1-C_n$ angle and $\psi$ is the $C_1-O_1-C_n-H_n$ angle. $\omega$ exists only for 1 $\rightarrow$ 6 links, and is defined as the $O_1-C_6-C_5-H_5$ angle.
<b>Type of Link</b>	Select the linkage type from the list.

## Ketoses Dialog Box

This dialog box contains command buttons to create the following saccharide residues: psicose, fructose, sorbose and tagatose in ketofuranose form, plus ribulose, xylulose and erythrulose in open-chain form. This dialog box functions in the same manner as the Aldoses dialog box (see above for description). The dialog box will stay in place until you dismiss it.

## Derivatives Dialog Box

This dialog box contains command buttons to create the following saccharide residues: glucosamine, galactosamine, N-acetyl-nuraminic acid, N-acetyl muramic acid, N-acetyl glucopyranoside, N-acetyl galactopyranoside, inositol, 2-deoxy-ribose, rhamnose, fucose and apiose. This dialog box functions in the same manner as the Aldoses dialog box (see above for description). The dialog box will stay in place until you dismiss it.

## Blocking Groups Dialog Box

This dialog box contains command buttons to create the following blocking groups: H, NH<sub>2</sub>, =O, COO<sup>-</sup>, methyl, lactyl, O-methyl, N-methyl, O-acetyl, N-acetyl, phosphoric acid, sulphate, N-sulphonic acid.

- |                        |  |
|------------------------|--|
| <b>Location</b>        | Choose the location for the blocking group by selecting first the residue (from the list) and then the atom (from the list). This location must contain an OH group, which will be replaced by the blocking group. |
| <b>Blocking Groups</b> | Add the desired blocking group to the position selected in Location, by pressing the button containing the name of the blocking group.   |

## User Dialog Box

This dialog box contains command buttons for user-defined residues. Definitions for the buttons are kept in the file: USER.BTN in the LIBRARY directory.

- Conformation** Enter the desired conformation of the residue when it will be added to the system in HyperChem. See the Aldoses dialog box for a description. Note that the isomer options are not available. Only the isomer contained in the user file can be added.
- Define New Residue** Assign a command button to a residue, and add it to the USER.BTN file.
- Edit Residue Name** Change the name displayed on a button and/or the name of the file containing the residue associated with a button.

To create a new residue:

1. Draw and build the residue or modify an existing residue in HyperChem.
2. Create the following named selections:

Name	Atom Contents
_Cn	Cn (where n is 1...6)
_Hn	Hn
_On	On
_HOn	HOn

Note that if, for example, H3 has been substituted by a CH<sub>2</sub>OH group, then the named selection \_H3 must contain the C atom of the CH<sub>2</sub>OH group.

3. Save the new residue in a HIN file. Edit the file to give the residue a name. If the residue was created by modifying an existing residue, simply change the line:

```
res 1 <old residue name>
```

to:

```
res 1 <new residue name>
```

where <new residue name> is not more than four characters long. If the residue was not created from an existing residue, you will have to add a res line and an endres line — see the HIN



Files appendix in the HyperChem Reference Manual. Make sure that there is only one residue in the file.

4. Read the new file back into HyperChem.
5. In Sugar Builder, choose the menu command Add/User to bring up the User dialog box.
6. Choose the Define New Residue command button.
7. A button for the new residue will be added to the User dialog box.

To edit the contents of a residue command button:

1. Choose the Edit Residue Name command button.
2. Specify which button you wish to change by entering the text shown on its face. Note that case is significant.
3. Enter the name of the file which contains the residue you wish to associate with this particular button.
4. Enter a name for the residue (up to four characters) which will appear on the face of the button.
5. Enter the residue type number, or enter -1 to have this number computed by Sugar Builder. The type number is created as follows. Take the one of the following numbers which corresponds to the residue type:

Residue Type	Type Number
Aldose, Acyclic	2,000
Ketose, Acyclic	3,000
Heptose, Acyclic	4,000
Octose, Acyclic	5,000
Nonose, Acyclic	6,000
Ketofuranose	7,000
Aldofuranose	8,000
Heptafuranose	9,000
Octafuranose	10,000
Nonafuranose	11,000
Aldopyranose	12,000
Heptapyranose	13,000
Octapyranose	14,000
Nonapyranose	15,000

Other 0

then add to it as many of the numbers below which are appropriate:

Residue Type	Type Number
1-substituted	1
2-substituted	2
3-substituted	4
4-substituted	8
5-substituted	16
6-substituted	32
7-substituted	64
8-substituted	128
9-substituted	256

Here are some examples:

Glucose	12,000
Fructose	7,000
N-acetyl Glucosamine	12,002
N-acetylneuraminic acid	14,010
2-deoxy-ribose	8,002

## Link Dialog Box

Choosing the Link menu command in the main window will bring up the Link dialog box. Sugar Builder will scan the HyperChem workspace and display two lists of atoms, one for each chain, each displaying an atom name and its corresponding residue. Choose one atom from the list of atoms in chain 1 and one atom from the list for chain 2. A bond will be drawn between these two atoms. If necessary, a hydrogen atom will be deleted from each atom. If one of the atoms has a full complement of neighbors, none of which is a hydrogen, then the link command will fail, with the "I don't know how to connect atom ..." message. If it is necessary to make a link to an atom of this type, then one of the atoms bonded to it will have to be deleted by hand within HyperChem before the Link command will be successful. Choosing OK command button

in the Link dialog box will bring up the Link Chains: Orientation dialog box.

### **Link Chains: Orientation Dialog Box**

When the Link Chains: Orientation dialog box is first displayed, Sugar Builder will send a command to HyperChem to connect the two atoms defining the link, and then will query HyperChem for all of the bond bends and bond torsions that pass through the new link bond. These are displayed in the dialog box, along with the new bond. You can choose any one of the angles by clicking on it in the list box. The current value of the angle will be displayed, and you can enter a new desired value. Pressing the Apply command button next to the value will send a command to HyperChem to change the angle to the new, desired value. Pressing the Apply All command button will cause HyperChem to build the link using all of the desired values.

### **About Dialog Box**

Choosing the Help/About menu item brings up the About dialog box, which will display version and copyright information about Sugar Builder.

### **Limitations**

Sugar Builder adds residues to the chain by sending a DDE command to HyperChem to execute a File/Merge on the file containing the next residue. This means that when the user is finished building a polysaccharide, the current file name is set to the file name of the first residue in the chain. If a File/Save command is then performed in HyperChem, the original residue file could be overwritten. Therefore, the residue files have been made read-only. If you wish to change any of these files, you can use the File/Properties menu command in the File Manager to remove the read-only restriction.

The force field parameters supplied with this module are sufficient to simulate only a small fraction of the residues supplied with this module. You may encounter a number of missing parameters when performing a calculation. However, it will often be possible to create the missing parameters by comparison with parameters for similar atom types.

## Polymer Builder

To build a polymer, one regards the arbitrary single molecule in the workspace as a potential monomer. One takes the workspace molecule, defines head and tail parts of the molecule and defines a torsional definition and angle that will be used to connect head to tail repeatedly until a polymer of the required length is obtained.

The polymer builder only builds polymeric structures, it does not analyze their properties beyond what HyperChem does for any molecule. Once the polymeric structure of chosen length is built it constitutes an ordinary molecule as far as HyperChem is concerned.

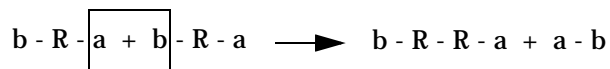
### Monomer Specification

#### Head and Tail

The monomers that can be used to build a polymer must have a head, referred to as "a", and a tail, referred to as "b". That is the monomer must have the form, b - R - a. Any molecule with at least three atoms can be a monomer.

### Polymerization by Elimination

The polymerization reaction is one which creates ...R - R - R - R... by the repeated elimination (condensation) of a-b. That is,



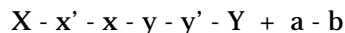
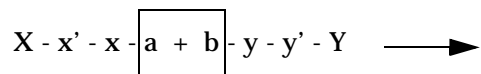
If the R-R torsion angle is considered to be randomized, then the head (a) and tail (b) are all that must be specified to build a polymer.

### Specific Torsion Angles

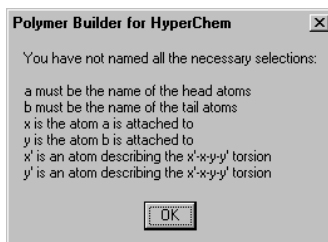
When eliminating "a" and "b", the internal geometry of R is considered to be preserved. However, there is still the rotation about the R-R bond to be considered. One option is that this is randomized during the polymerization and an option radio button is pro-

vided for that. The other possibility is that this angle is specified by the user as a constant value for monomer-monomer connection. In this case we need to consider the geometry of the R-R bond further.

The assumption is that "a" is connected to atom "x" which in turn is connected to atom "x'". similarly, atom "b" is connected to atom "y" which in turn is connected to atom "y'". That is, an elaboration of the monomer condensation is,

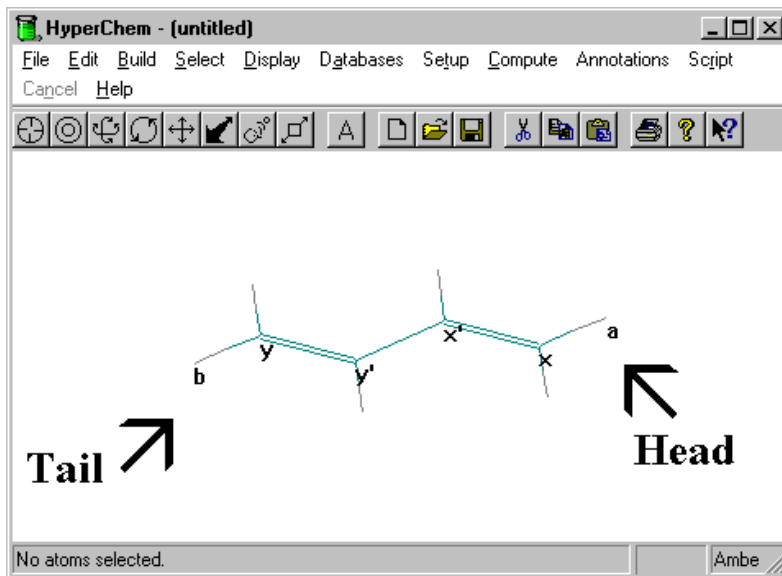


In this case, the torsional angle to be specified is the x'-x-y-y' torsion. Each of these, i.e. a, b, x, x', y, y', are to be specified in the monomer by named selections. If they are not, then then you will see the following error message,

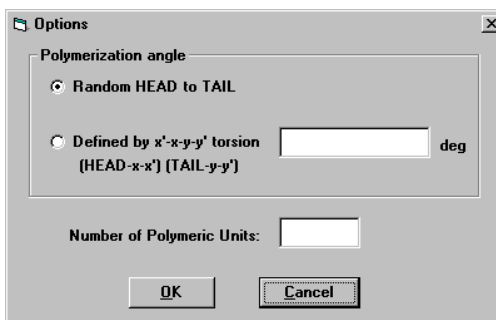


## Example

As an example we will build a polymer from the 1-3 butadiene monomer. The monomer looks as follows, where we have used custom labels and annotations to indicate the named selections.

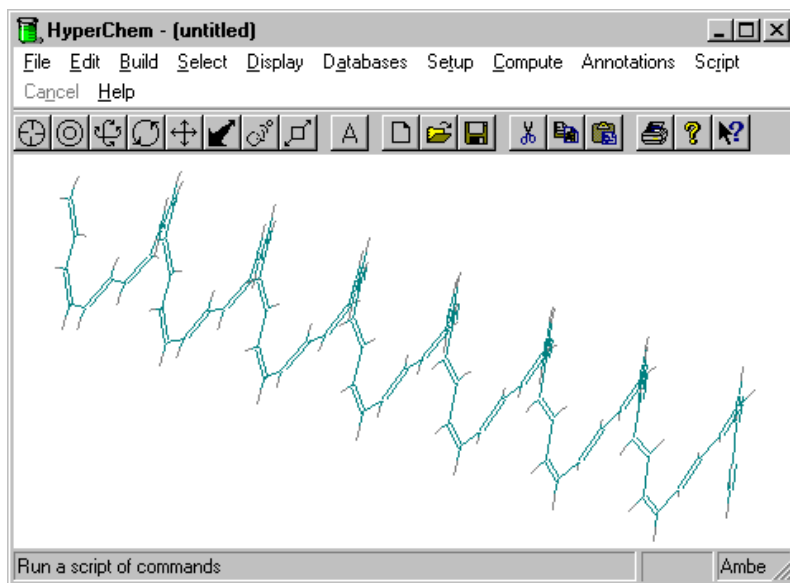


If the appropriate named selections have been made, then clicking on the <Databases/Polymers...> menu item will bring up the following dialog box,



The choices that must be made are whether to randomize the  $x-x'-y'-y$  angles or to give them a specified constant value. In addition one must choose the length of the polymer to build.

If one were to choose 30 degrees for the torsion above and build 20 units of a polymer, one would obtain the following structure:



## Limitations

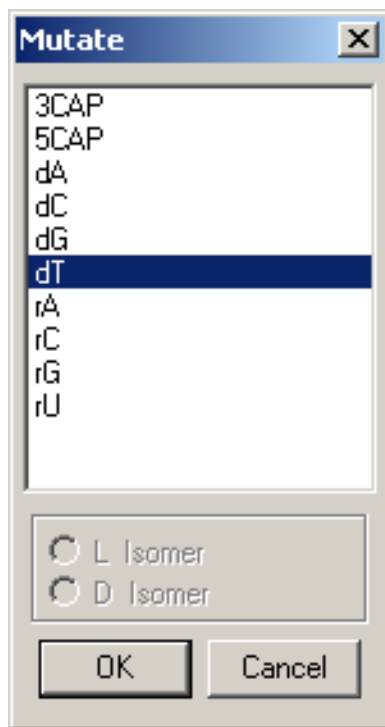
In the above fashion you can use any monomer you like as long as you properly define the required atoms, a, b, x, x', y, and y'. For short head to tail paths, the definitions for x and x' may overlap the definitions of y and y' but this will not matter as long as a, x, and x' (or b, y, and y') refer to bonded sequences of atoms. Thus, for example, x' and y or x' and b might be the same atom. The named selections a (or b) can refer to a group of atoms, rather than a single atom, as long as these atoms are bonded to x (or y) by one bond.

## Mutate

Mutate substitutes a residue for a residue already in a polypeptide or polynucleotide. This biopolymer must contain residues from the Databases menu or it must come from a PDB file (see page 503) or MOL2 file. The Mutate dialog box lists amino acids or nucleotides for substitution.

**Note:** Mutate is gray until you select a residue.

## Mutate Dialog Box



The dialog box shows a list of the available residues (see Appendix E). If you are mutating an amino acid residue, you can specify whether you want the new residue to be the Levorotatory Isomer or the Dextrorotatory Isomer. If you are mutating a nucleotide residue, these buttons are gray and inactive.

To change a residue:

1. Set the Selection tool to Residues (L-click on the Selection icon, then choose Residues on the Select menu).
2. Select a single residue from the molecular system.
3. Choose Mutate on the Databases menu.
4. In the Mutate dialog box, L-click on a new residue and L-click OK.

HyperChem substitutes the new residue for a selected residue.



## Choices of Nucleotide Residues

The Mutate dialog box provides residues listed in the Nucleic Acids dialog box, plus residues from the template (TPL) files that are not available in the Nucleic Acids dialog box. The additional residues (see “Nucleic Acid Residues” on page 737) include less common nucleotides for constructing tRNA molecules. To use these residues, build a polynucleotide—by using the Nucleic Acids dialog box—using temporary nucleic acids. Then use the Mutate dialog box to substitute the final nucleotide. For example, to construct a rU–rG base pair, start with rU–rA, then mutate rA to rG.

## Replace

The Mutate operation above replaces a single residue with a new single residue. The generalization of this is to replace a contiguous sequence of residues with a new sequence that is potentially of different length. This is a generic cut and paste type of operation. In HyperChem this is accomplished with a single Replace operation—with the pasted sequence coming from the clipboard. This operation is particularly useful in the design of homologous proteins.

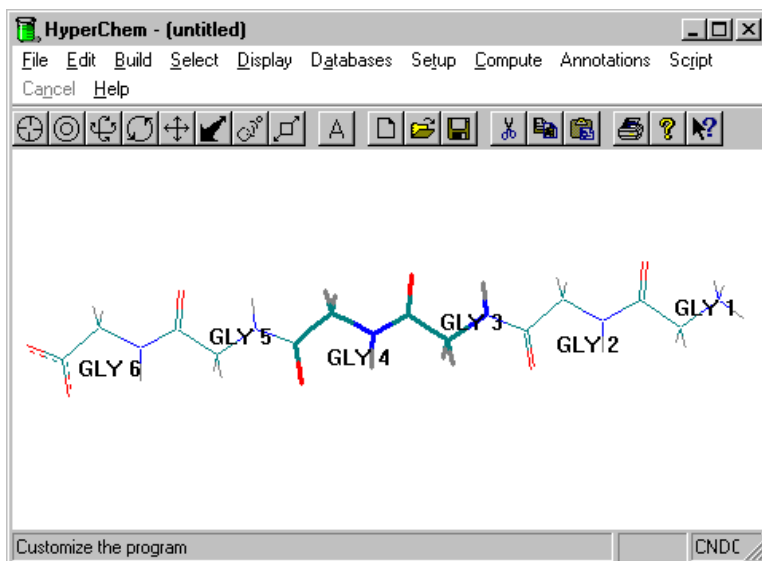
## Protein Design

Protein design and the study of homologous proteins requires the ability to start with a specified protein and make required changes to the primary (and secondary structure) and hence to the native structure, with hopefully beneficial results. Protein Design or the creation of homologous proteins requires one to make more general modifications than just point mutations. For example one may want to insert a short new sequence with defined secondary structure into a specific place in the starting structure.

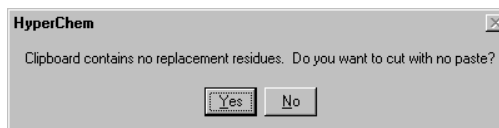
With HyperChem one has the ability to *cut and paste* arbitrary portions of a protein structure. Rather than have separate cut and paste operations, however, the new functionality is organized as a single Replace operation. That is, one specifies a contiguous sequence of amino acids (residues) and *replaces* that sequence with a new sequence. The new sequence is obtained from the clipboard and must be placed there prior to the Replace operation. The old sequence is selected using normal HyperChem selection operations prior to the request to replace it with the contents of the clipboard.

## Cutting (Deleting) a Residue Sequence

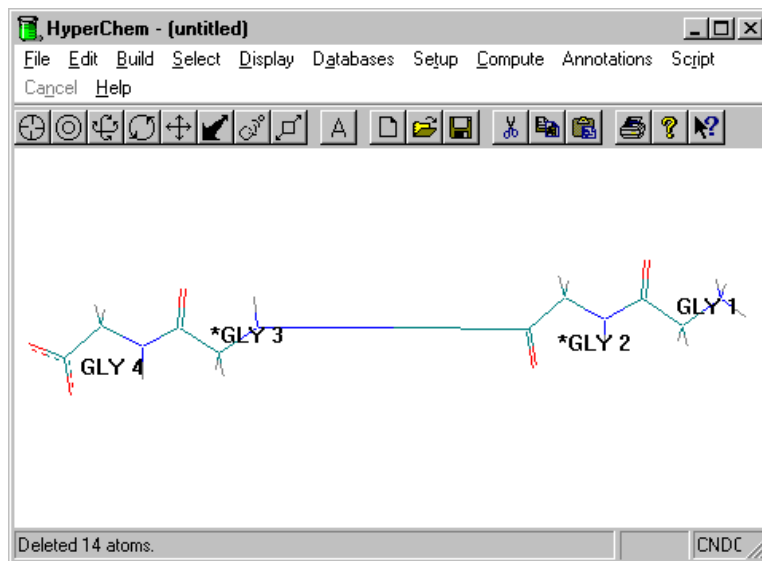
A simple deletion of a residue sequence occurs if the contents of the clipboard are null at the time of the Replace operation. To illustrate this and other operations, we will use simple peptides rather than complex proteins but the same operations apply to bigger systems. Assume we have a short poly-GLY sequence and select the two middle residues, GLY 3 and GLY 4 (we use Thick Lines to indicate the selection here).



If the clipboard is empty or does not contain HyperChem amino acid residues and we now click on the menu item <Databases/Replace...>, we will see the following:



If we hit Yes, then we will obtain the following:



The two selected residues have been deleted. The remaining residues are not altered or moved and this results in a very long peptide bond between the new GLY 2 and GLY 3 residues. This long bond will have to be annealed by an optimization or by molecular dynamics to finalize the effect of eliminating the two residues.

The two adjacent residues, GLY 2 and GLY 3 are at this point considered *dirty* (indicated by the \*) and cannot be mutated, etc. They can be *cleaned* by saving and re-opening the structure.

## Replacing a Residue Sequence

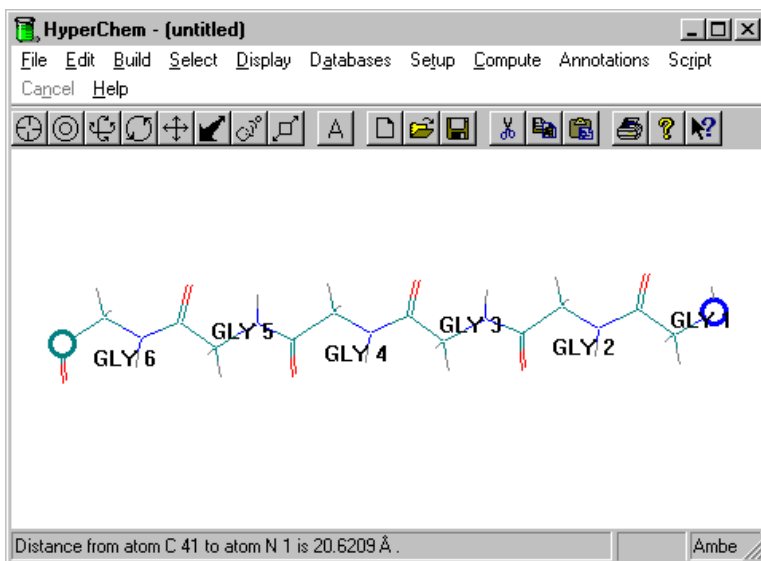
A more ambitious operation is the actual replacement of a set of residues with a new set. In this simple example, we will prepare a loop and replace a single residue with the loop. We will refer to the loop as the source and the single residue as the target, i.e. a target set of residues is replaced by a source set of residues.

## Creating a Source Sequence

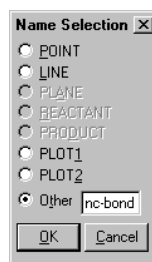
First we create a sequence of six GLY residues using the peptide builder in HyperChem.

### Preparing a loop of residues:

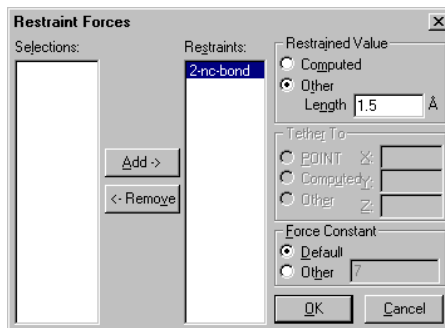
1. Click on GLY six times using the beta sheet conformation to create a short linear set of residues.
2. Select the N of the N-terminal end and the C of the C-terminal end as follows:



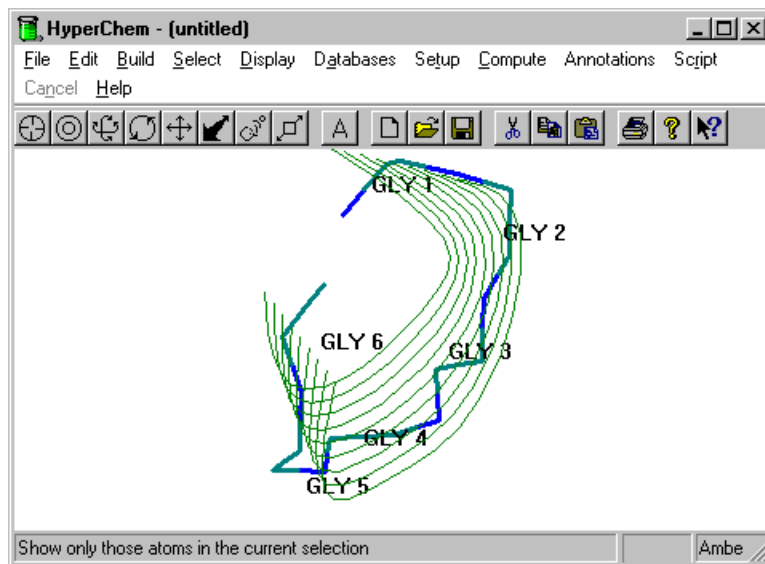
3. Use the menu item Named Selection menu item to create a *named selection*, called "nc-bond".



4. Use the <Setup/Restrains...> menu item to add a restraining force keeping the nc-bond at a restrained value of 1.5 Angstroms, as shown below:



- Use the AMBER 3 force field to optimize the structure. Don't forget to deselect the two atoms prior to the optimization. this will create a loop something like the following:



- Select the whole set of residues and place them in the clipboard by an <Edit/Copy> operation. [Note: be sure all atoms are displayed (unlike above) and selected before doing the copy.]

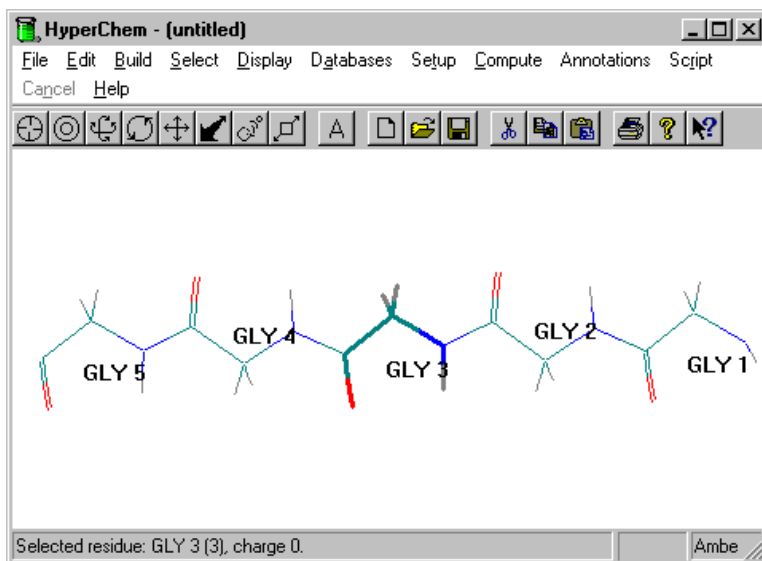
You have now created an appropriate loop to add to a protein or other residue structure. This could be saved and restored to the clipboard later but must be in the clipboard for the following operations. The next step is to ready a target.

## Replacing Target Residues

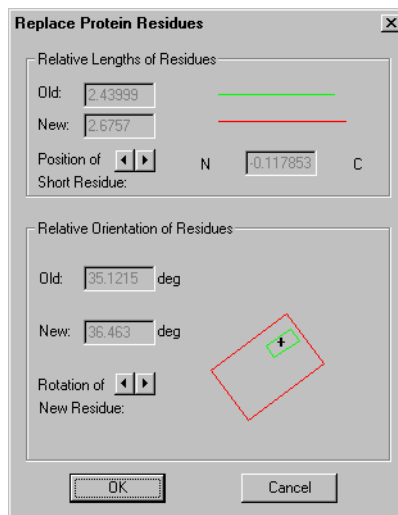
We next need to illustrate how to replace a set of residues with a new set that is of different shape, length, composition, etc. We are using all GLY here for simplicity but any properly constituted residue structure can be used.

Replacing one set of residues with another:

1. Create a simple, length 5, set of GLY residues in their beta sheet conformation.
2. Using residue selection, select the GLY 3 residue as shown below:



Click on the <Databases/Replace...> menu item to bring up the following dialog box:



This dialog box shows the size and orientation of the source and target residues and gives you the ability to fit the source residues in the the space being vacated by the target residues prior to the actual replace being invoked.

### Relative Length of Residues

This section of the dialog box shows you the old (green) and new (red) residues according to their length, i.e. the distance from the N-terminal Nitrogen to the C-terminal Carbon. The numerical values are these total lengths in Angstroms. The diagram shows the translational position where the new (source or green) residues will be positioned relative to the old (target or red) residues along the line from the N-terminal Nitrogen to the C-terminal Carbon.

If they are not the same length then the new residues can be positioned relative to the old residues by moving the shortest sequence via the scroll buttons provided. The position of the short sequence relative to the long sequence is given in Angstroms and should be modified til the value is zero indicating a position in the "middle". This positioning will have the short sequence positioned in the middle of the long sequence. If the new sequence is much longer or much shorter than the old sequence, a considerable amount of annealing will be necessary to relieve the mismatch.

Each of the two sequences is characterized by a vector from N-terminal Nitrogen to C-terminal Carbon. The new sequence will have its vector co-linear with the old sequence. The remaining parameter is the rotation of the new sequence around the line of this vector. This rotation to provide an optimal orientation of the new sequence relative to the old is provided by the lower part of the dialog box.

### **Relative Orientation of Residues**

This part of the dialog box shows the old and new sequences looking along their N-terminal to C-terminal axes which have been made co-linear. The small cross shows the position of the N-terminal to C-terminal axis of the two sequences. The remaining operation is to rotate the new sequence relative to the old sequence for optimal positioning of the new sequence. This is accomplished via the scroll button provided and the display shows the cross-sectional size and orientation of each of the two sequences.

If the N-terminal to C-terminal line is referred to as the z axis, then the rectangles shown represent the minimum and maximum atom coordinates (bounding box) of each sequence in an xy plane perpendicular to the z axis.

### **Executing the Replace**

When the new sequence is translated and rotated to your satisfaction then the OK button will perform the replace operation.

### **Inserting A Residue Sequence**

A pure insertion operation is not possible and must be accomplished via a replacement operation where, for example, the N-terminal residue of the source and target sequences are identical.

## **Invoke Database**

This menu item invokes the separate HyperChem Data module. This module provides database services for HyperChem. HyperChem Data is described in the HyperChem Modules manual.





# Chapter 7

## Edit, Select, and Display Menus

This chapter describes the functions on the Edit, Display, and Select menus. These functions let you change the structure of molecules, alter their appearance in the workspace, and regulate how the Selection tool works.

### Edit Menu

The Edit menu contains items using the Clipboard (for example, Cut and Paste) and items for manipulating molecules in the workspace. The Edit menu has these commands:

---

Clear <sup>a</sup>	Removes selected atoms from the workspace. You cannot recover these structures with Paste. You can also press [Del]
Cut <sup>a</sup>	Moves selected atoms from the workspace onto the Clipboard. You can also press [Ctrl] + [X]
Copy <sup>a</sup>	Copies selected atoms to the Clipboard in HIN format. You can also press [Ctrl] + [C]
Copy ISIS Sketch <sup>a</sup>	Copies selected atoms to the Clipboard in MDL ISIS/Draw clipboard format.
Paste <sup>a</sup>	Copies atoms from the Clipboard to the workspace. You can also press [Ctrl] + [V]
Copy Image	Takes a picture of a HyperChem window. You can also press [F9]

---

<b>Invert<sup>a</sup></b>	Turns all molecules or selected atoms upside down, through a POINT that you choose. Coordinates x, y, and z (relative to a point) become -x, -y, and -z
<b>Reflect<sup>a</sup></b>	Produces a mirror image of all molecules or selected atoms relative to a PLANE that you choose. For the XY plane, coordinates x, y, and z become x, y, and -z
<b>Rotate . . .</b>	Provides rotation by an exact amount (in degrees) about an axis (x, y, and z)
<b>Translate . . .</b>	Provides translation by an exact amount (in Ångstroms) along each axis (x, y, and z) or to a defined point
<b>Zoom . . .</b>	Provides an exact amount of magnification
<b>Z Clip . . .</b>	Shows the molecular system from the top (along the y axis) and provides exact settings of the clipping planes
<b>Align Viewer . . .</b>	Changes the viewer's vantage point to any other axis or line in the Molecular Coordinate System
<b>Align Molecules . . .</b>	Changes the alignment of whole molecules so that their internal (primary, secondary, or tertiary) axes correspond to the viewer's x, y, or z axes
<b>Set Bond Length<sup>a</sup> . . .</b>	Changes the length of a selected bond in a molecule
<b>Set Bond Angle<sup>a</sup> . . .</b>	Sets a selected bond angle in a molecule
<b>Set Bond Torsion<sup>a</sup> . . .</b>	Sets a selected torsion angle in a molecule

a. When this item is gray, it is temporarily unavailable. An appropriate entity must be selected or defined to make it available.

## Clear

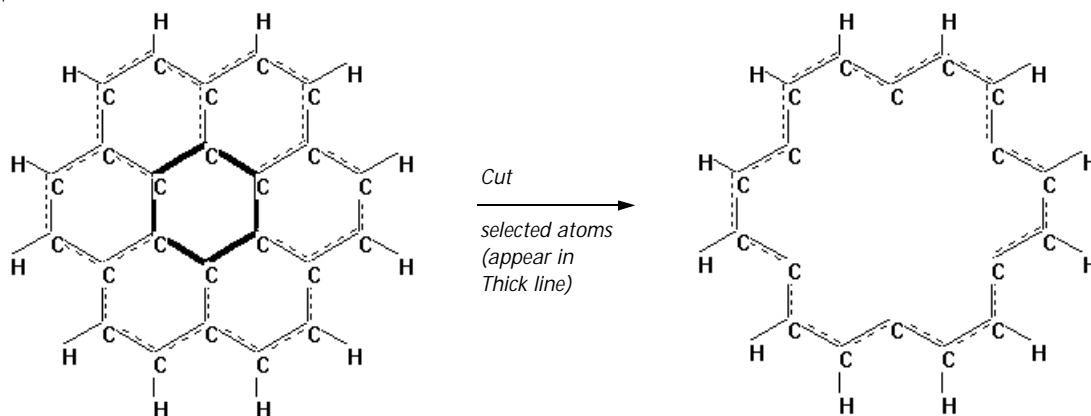
Clear permanently removes selected atoms from the workspace. You cannot recover these atoms. HyperChem asks you to confirm your decision.

## Cut

Cut moves selected atoms from the workspace to the Clipboard, replacing its previous contents. You can then use Paste to copy the atoms back to the workspace of the same HyperChem window or to a different HyperChem window. HyperChem asks you to confirm your decision.

**Note:** Cut is gray and inactive unless you select an atom in the molecular system.

Cut removes selected atoms plus all bonds connecting these atoms to each other and to the rest of the molecular system. For Cut, Copy, and Paste, HyperChem keeps track of atoms rather than bonds.



## Copy

Copy places a duplicate of selected atoms onto the Clipboard in HIN format, replacing its previous contents. You can use Paste to copy the atoms back to the workspace of the same HyperChem window or to a different HyperChem window.

**Note:** Copy duplicates selected atoms and the bonds connecting these atoms, but does not remember the bonds between this selection and the rest of the molecular system.

## Copy ISIS Sketch

Copy ISIS Sketch places a duplicate of selected atoms onto the Clipboard in MDL ISIS/Draw Sketch format, replacing its previous contents. HyperChem can Paste using ISIS/Draw Sketch format, so you can use Paste to copy the atoms back to the workspace of a HyperChem window or to ISIS/Draw.

You can paste structures back and forth between HyperChem and ISIS/Draw. ISIS™ (Integrated Scientific Information System) by MDL Information Systems provides drawing, storage, search, and retrieval capabilities for 2D and 3D chemical structures and related data. For example, you can use HyperChem to analyze or modify a structures, and then transfer the structures into ISIS for registration into a 3D database.

To copy a structure from HyperChem to ISIS Draw:

1. Build a molecule in HyperChem.
2. Choose Molecule as the selection level, and L-click on the molecule.
3. Choose Copy ISIS Sketch on the Edit menu.

A copy of the molecule in ISIS/Draw format (*.skc*) is stored in the Clipboard.

4. Open ISIS/Draw.
5. Choose Paste on the Edit menu.

The molecule appears in the ISIS/Draw window.

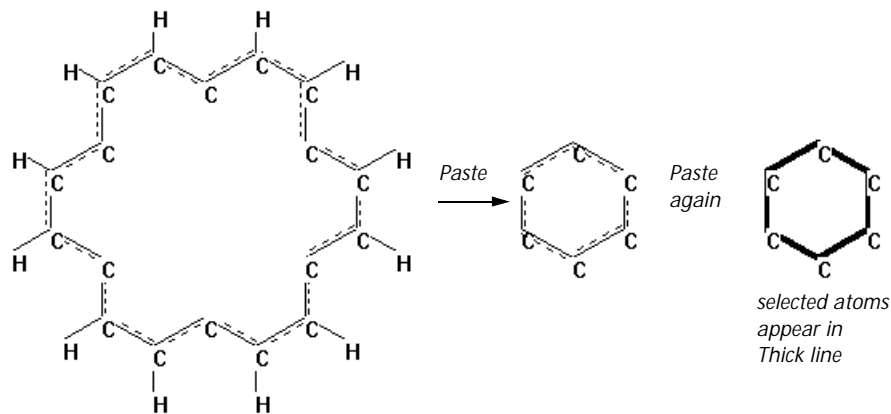
## Paste

Paste copies the contents of the Clipboard to the workspace of a HyperChem window. Use Paste to duplicate atoms and molecules in the same window or to transfer them to another HyperChem window.

**Caution:** Paste does not transfer a molecular model (graphic) to another Windows application (for example, Windows Write). If

you use Paste to move information to another application, you see only a text version of the molecular structure, as it appears in a HIN file. Use Copy Image (see the next section) to transfer graphic images to other applications.

The atoms from the Clipboard appear 1 Ångstrom to the right of the center of mass of the molecular system previously in the workspace.



**Note:** Paste might not be the exact reverse of Cut. Cut removes selected atoms and all bonds connecting these atoms to each other and to the rest of the molecular system. Paste provides only the atoms and the bonds connecting these atoms to each other. For Cut, Copy, and Paste, HyperChem keeps track of atoms rather than bonds.

**Note:** Paste is gray and unavailable until you use Cut or Copy to place atoms on the Clipboard.

## Pasting Superatoms From ISIS/Draw

When cutting or copying molecules from ISIS/Draw for pasting into HyperChem, you should expand any superatoms in the structure before transferring the molecules to the clipboard from ISIS.

If you are unable to do this, (for instance, if you have an ISIS sketch file embedded in a word-processing document), HyperChem expands each superatom in the correct position, but considers it as a separate molecule. Two atoms of the superatom are superimposed on two atoms of the main structure around the connection point. You will need to delete one atom of each superimposed pair

and reconstruct the proper connectivity in order to make a single molecule.

## Copy Image

Copy Image takes a picture of all or part of the screen and places it either in a file or on the Windows Clipboard (see “Setup Image” on page 99). By default, HyperChem stores a Windows metafile picture in the file chem.wmf and a bitmap picture in the file chem.bmp. However, you can select a Filename other than these in the <Preferences/Setup Image> dialog box.

*Note:* Press [F9] for Copy Image. You must use [F9] to save images of pull-down menus and dialog boxes (see page 99).

Each time you use Copy Image, HyperChem replaces the picture that is already on the Clipboard or in a file. To collect several pictures, you must use one of these methods:

- |                          |   |
|--------------------------|---|
| <b>For a file</b>        | If the picture is in a file, select a new name for Filename every time you take a picture. Alternatively, use a script or custom menu item to change the Filename in a sequential way to collect a set of pictures. |
| <b>For the Clipboard</b> | If the picture is on the Clipboard, paste it into a file, using another program that accepts metafile or bitmap images. For example, a Word® for Windows file can store both bitmap and metafile graphics.          |

Use the Setup Image section of the Preferences dialog box (see page 99) to give instructions for copying an image.

Use Print on the File menu to send a copy of the workspace directly to a printer.

## Invert

Invert carries out a molecular symmetry operation (inversion) through a POINT. POINT is the center of mass of one or more selected atoms. Use the Name Selection dialog box to define POINT. If the coordinates of an atom are x, y, and z (relative to the POINT), Invert changes them to -x, -y, and -z.

Once you establish a POINT, you can invert any selected atoms or, if there is no selection, a whole molecular system.

**Note:** Invert is gray unless the molecular system contains a POINT selection.

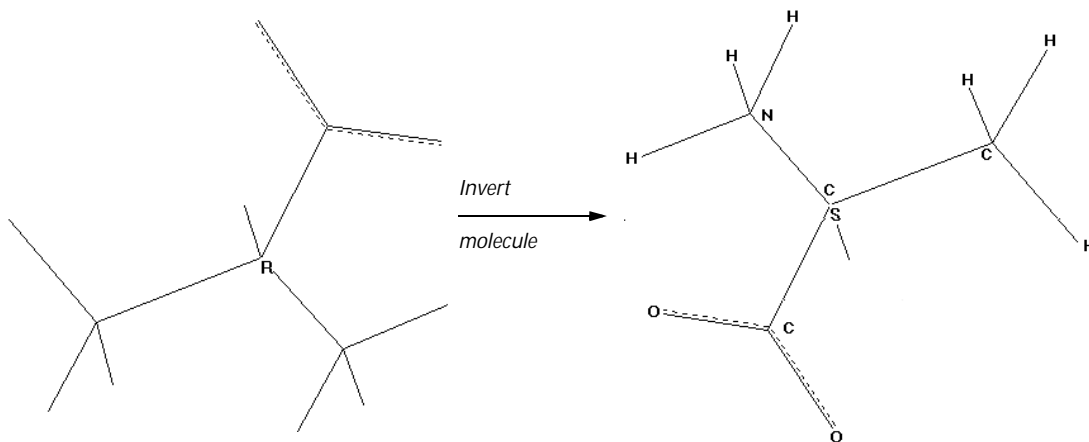
To invert atoms and molecules:

1. Select one or more atoms to define the inversion POINT.
2. Choose Name Selection on the Select menu (see 237). Then choose POINT in the dialog box. L-click OK.
3. Deselect all atoms.
4. Select atoms or molecules to invert. Without a selection, HyperChem inverts the whole molecular system.
5. Choose Invert on the Edit menu.

**Caution:** If you invert selected atoms in a molecule, HyperChem inverts only these atoms and might distort the bonds connecting those atoms to the rest of the molecule.

**Example:**

This example shows HyperChem inverting the amino acid alanine. Since the alpha carbon is a chiral center, inversion changes the chirality of this atom.



You can also, for example, use inversion to convert a chair to a boat conformation.



## Reflect

Reflect carries out a molecular symmetry operation (reflection) through a PLANE. This molecular operation can produce a mirror image of selected atoms or of a whole molecule. If PLANE is the XY plane, then Reflect changes an atom's coordinates from x, y, and z (relative to PLANE) to x, y, and -z.

Use the Name Selection dialog box (see page 238) to define PLANE. PLANE is the surface defined by the primary and secondary inertial axes (see "Inertial Axes" on page 21) of the atoms used to name PLANE.

Reflect acts on selected atoms or, if there is no selection, the whole molecular system. If the reflected atoms contain one or more chiral centers, HyperChem changes their chirality.

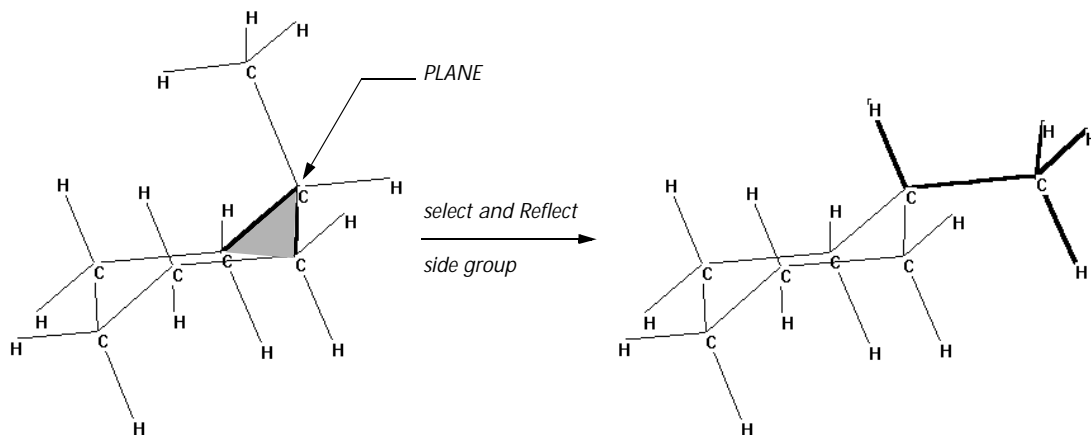
**Note:** Reflect is gray unless the molecular system contains a PLANE selection.

To reflect atoms and molecules:

1. Select at least three atoms to define a reflection PLANE.
2. Choose Name Selection on the Select menu (see page 237). Then choose PLANE in the dialog box. L-click OK.
3. Deselect all atoms.
4. Select the atoms or molecules to reflect. Without a selection, HyperChem reflects the whole molecular system.
5. Choose Reflect on the Edit menu.

**Example:**

This example shows HyperChem reflecting the selected side group of cyclohexane.

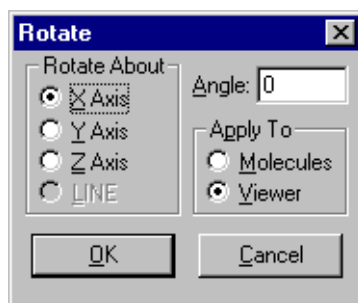


## Rotate

Choosing Rotate displays the Rotate dialog box to produce an exact rotation of molecules. This dialog box duplicates the effects of a rotation tool (see “Using the XY Rotation Tool” on page 37 and “Using the Z Rotation Tool” on page 39).

*Note:* You can also double-click on a Rotation tool icon to reach the Rotate dialog box.

## Rotate Dialog Box



**Rotate About** You can rotate selected atoms or, without a selection, a whole molecular system, around an axis in either the Viewer’s or Molecular Coordinate System or around a LINE. LINE is a named selection (see “Name Selection” on page 237) that belongs to a molecular system. If you cannot rotate selected

atoms, HyperChem shows a message on the status line.

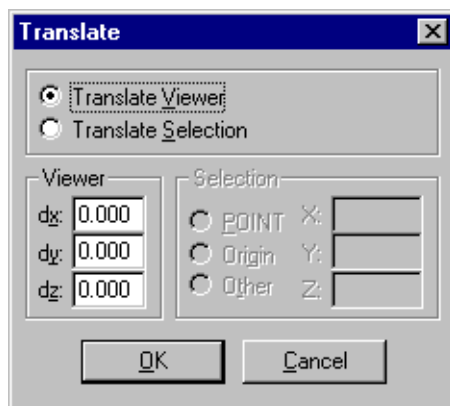
X Axis	This axis is in the horizontal plane of the screen.
Y Axis	This axis is in the vertical plane of the screen.
Z Axis	This axis is perpendicular to the screen.
LINE	LINE is a named selection. Use Name selection on the Select menu to define a LINE for a molecular system.
Angle	Enter a rotation angle. The practical range is $\pm 360$ degrees.
Apply to	Rotates selected atoms or the whole molecular system in either of these ways:
Molecules	Rotates selected molecules in the Molecular Coordinate System, changing atomic coordinates. This works for selected whole molecules, for selected side chains that can rotate, or, if there is no selection, for the whole molecular system.
Viewer	Rotates all molecules in the workspace in the Viewer's Coordinate System, leaving the atomic coordinates unchanged.

## Translate

Translate displays the Translate dialog box to produce an exact translation of molecules. This dialog box duplicates the effects of a Translation tool (see "XY Translation Tool" on page 40 and "Z Translation Tool" on page 43), moving the center of mass of selected atoms or molecules to another position in the Molecular Coordinate System.

*Note:* You can also double-click on the Translation tool icon to display the Translate dialog box.

## Translate Dialog Box



**Translate Viewer** Moves all molecules in the workspace, but leaves the atomic coordinates unchanged, as if the viewer is moving. This is the same as using a translation tool in the workspace with the left mouse button.

- dx** Amount of translation to the right, in Ångstroms.
- dy** Amount of translation up, in Ångstroms.
- dz** Amount of translation toward you, in Ångstroms.

**Translate Selection** Moves the center of mass of selected atoms to a new point, changing the atomic coordinates. This is the same as using a translation tool in the workspace, with the right mouse button. Translation places the center of mass of the selected atoms at a new point (POINT, Origin, or Other) in the Molecular Coordinate System (see “Coordinate Systems” on page 18). If the selection is part of a molecule, HyperChem can move only the selected atoms if Whole Molecule translation is off (see “Preferences Property Sheet” on page 95).

**Note:** You can translate selected residues or atoms away from the molecule that they comprise by turning off Whole molecule translation (see “Tool” on page 97). To accomplish this translation, HyperChem stretches bonds beyond their normal lengths and

angles. The default setting, Whole molecule translation on (✓), prevents this distortion.

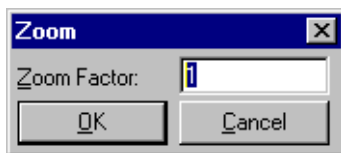
POINT	Moves molecules or selected atoms to a POINT defined in the Name Selection dialog box (see page 238).
Origin	Moves molecules or selected atoms to the origin of the Molecular Coordinate System.
Other	Moves molecules or selected atoms to the same point you chose the last time you used this dialog box, or to new coordinates that you enter for X, Y, Z (see below).
X, Y, Z	For POINT, this shows the location associated with the predefined, named selection, POINT. For Origin, this shows 0, 0, 0, for the origin of the Molecular Coordinate System. For Other, you can set a point in the Molecular Coordinate System.

## Zoom

Zoom displays the Zoom dialog box to produce an exact magnification of molecules in the workspace. This dialog box duplicates the effects of the Zoom tool (see page 45). The dialog box reports the present zoom factor (magnification) for the molecular system in the workspace. You can enter a new value, from 0.01 to 50. A Zoom Factor of 1 produces the same magnification as Scale to Fit (see page 250).

*Note:* You can also double-click on the Zoom tool icon to reach the Zoom dialog box.

### Zoom Dialog Box



**Zoom Factor** This shows the present magnification of the molecular system, if there is no selection, or of selected atoms. A Zoom Factor of 1 is the same as Scale to Fit, which fits the selection or the whole

molecular system into the workspace (see “Scale to Fit” on page 250). Smaller values for Zoom Factor decrease the apparent size and larger values magnify the molecules. For example, a Zoom Factor of 2 doubles the size.

**Caution:** Use a Sticks rendering with high magnifications of molecular systems. High magnifications (Zoom Factor) of all other molecular renderings (*except* Sticks) might require more memory than your computer has available.

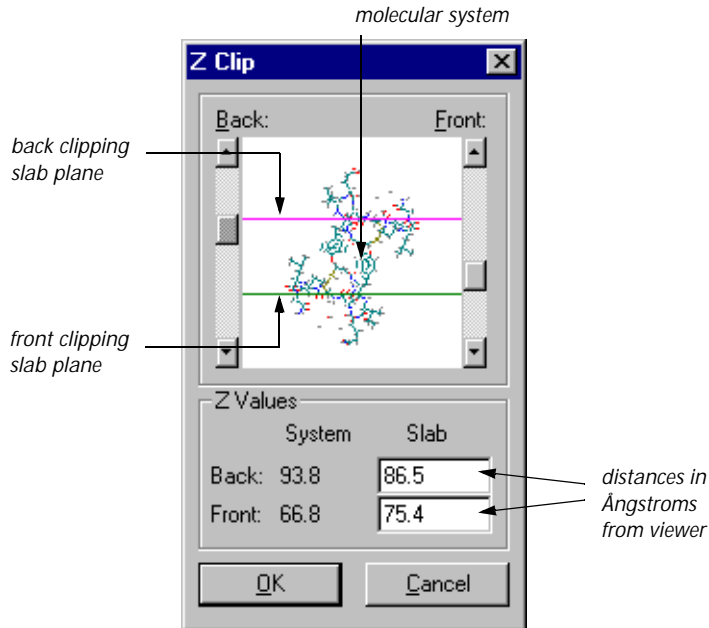
## Z Clip

Z Clip shows the Z Clip dialog box (see page 214), which you use to move the clipping planes. This dialog box duplicates the effects of the Z clipping tool (see page 47) and shows a top view of the molecular system (along the y axis) to aid in setting the clipping planes. Use this dialog box to set the clipping planes, either numerically or graphically.

The dialog box shows a Sticks rendering of the molecules in the workspace. You view the molecules from above, down the y axis of the Viewer’s Coordinate System. A green bar represents the front clipping plane; a violet bar represents the back clipping plane. Whenever you open this dialog box, the clipping planes are set 1/3 and 2/3 down from the top of the display.

**Note:** You can also double-click on the Z clipping tool icon to reach the Z Clip dialog box.

## Z Clip Dialog Box



The dialog box provides this information:

- System**      The extreme front and back positions of all molecules in the workspace. The viewer is at zero. A new molecular system is at 40.
- Slab**         The positions of the front and back clipping planes relative to the viewer, who HyperChem takes to be at zero. As you move the clipping planes by using the scroll bars, these numbers change. Entering numbers changes the position of a clipping plane on screen.

Press [Tab] or L-click to move between Slab/Front and Slab/Back. This highlights the number. Enter a new value from 0 to 3200 Ångstroms. The front slab must remain in front of the back slab.

**Note:** If a molecular system disappears from the workspace while you are using the Z clipping tool, press [Spacebar] or use the numeric settings in this dialog box to display the molecules again. Adjust the slab positions (Front and Back) to include the System (Front and Back).

You can adjust the positions of the clipping planes using two methods:

- Use the scroll bars at the sides of the molecular model to move the clipping planes. Movement is along the z axis of the Viewer's Coordinate System.
- Enter values, to the right, for the positions of the clipping planes. Values represent the distance from the viewer in Ångströms.

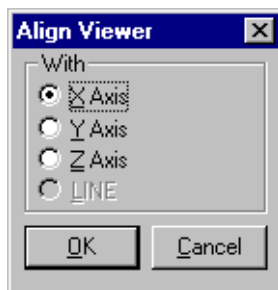
## Align Viewer

Align Viewer modifies the Viewer's Coordinate System so that it has a simple correspondence with the Molecular Coordinate System.

The Viewer's Coordinate System is relative to the plane of the computer screen (see "Viewer's Coordinate System" on page 19). You always view the screen and the HyperChem workspace along the viewer's z axis. The molecular system has its own coordinate system (see "Molecular Coordinate System" on page 20).

When you first construct a molecular system, the viewer's z axis aligns with the molecular z axis. If you rotate the viewer's position (for example, using the Rotation tools and the left mouse button), this alignment is lost. Align Viewer returns the workspace to the initial state or to another standard orientation.

### Align Viewer Dialog Box



X Axis            Aligns the viewer's z axis and the molecular x axis.

Y Axis            Aligns the viewer's z axis and the molecular y axis.



<b>Z Axis</b>	Re-aligns the viewer's z axis and the molecular z axis.
<b>LINE</b>	Aligns the viewer's z axis with a LINE in the Molecular Coordinate System. A LINE is the best-fit to two or more atoms in a molecular system or to the primary inertial axis. <sup>1</sup> This choice is inactive (gray) until you set a LINE in the Name Selection dialog box (see page 238).

For more information about coordinate systems, see page 18.

## Align Molecules

Align Molecules rotates selected whole molecules so that their inertial axes correspond to the viewer's x, y, or z axes. You can also align an inertial axis with a LINE in the Molecular Coordinate System.

If you do an alignment with at least one atom selected, the alignment "moves" the atoms, changing their coordinates in the Molecular Coordinate System. If there is no selection, the alignment maintains the same atomic coordinates, but moves your viewpoint (the Viewer's Coordinate System).

A molecular system has an inertial axis system independent of the Molecular or Viewer's Coordinate Systems. Inertial axes are fixed in a molecular system and depend only on the number, type (mass), and positions of atoms. The inertial axes have their origin at the center of mass and are orthogonal (at right angles to each other). The primary inertial axis is often the longest distance from the center of mass to the edge of the molecular system. Inertial axes are computed for the selected atoms, or if there is no selection, for all atoms in the molecular system.

You can see the inertial axes of a molecular system by choosing Show Axes on the Display menu. See "Coordinate Systems" on page 18 for more information about coordinate systems and axes.

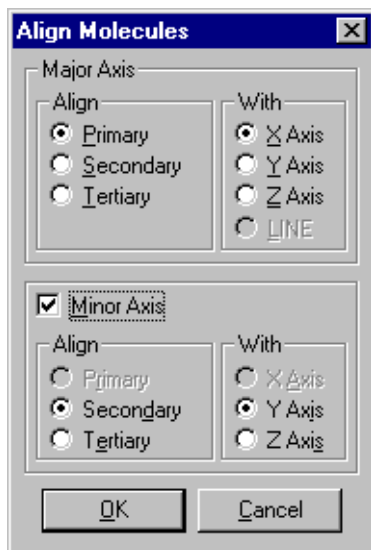
With the Align Molecules dialog box, you can do one or two alignments of a molecular system:

- The first alignment is known as the major alignment. This positions one inertial axis with one viewer axis.

1. The primary inertial axis corresponds to the smallest eigenvalue of the moment of inertia matrix.

- You can then choose to do a second, minor alignment. This rotates the molecules around the first alignment axis, until a second inertial axis aligns with a second axis in the Viewer's Coordinate System.

## Align Molecules Dialog Box



<b>Primary</b>	This is the primary inertial axis <sup>1</sup> , passing through the center of mass of a molecular system. This is often the longest distance from the center of mass to the edge of a molecular system.
<b>Secondary</b>	This axis is orthogonal (at right angles) to the primary and tertiary axes. If the molecular system is planar, the secondary axis is in this plane.
<b>Tertiary</b>	This axis is orthogonal (at right angles) to the primary and secondary axes. If the molecular system is planar, the tertiary axis is perpendicular to this plane.
<b>X Axis</b>	Aligns the viewer's x axis with an inertial axis.
<b>Y Axis</b>	Aligns the viewer's y axis with an inertial axis.
<b>Z Axis</b>	Aligns the viewer's z axis with an inertial axis.

1. The primary inertial axis corresponds to the smallest eigenvalue of the moment of inertia matrix.

- LINE** Aligns the chosen inertial axis with a LINE in the Molecular Coordinate System. A LINE is the best-fit to two or more atoms in a molecular system, or to the primary inertial axis. This choice is inactive (gray) until you set a LINE in the Name Selection dialog box (see page 238).
- Minor** Turn on Minor (✓) if you want to align a second inertial axis with a second coordinate axis. This alignment rotates the molecular system around the first alignment axis to make the second alignment. You have the same choices as for the first alignment, except you cannot use the same inertial or coordinate axes that you used for the major alignment.

**To use Align Molecules:**

1. Select atoms in a molecular system, or deselect all atoms for the whole molecular system. This selection determines the inertial axes for alignment.
2. Choose Align Molecules on the Edit menu.
3. Choose the inertial axis for the major alignment from the upper-left section of the dialog box.
4. Choose an axis or LINE from the lower-left section of the dialog box.
5. Turn on Minor (✓) if you want to align a second inertial axis with a second coordinate axis. This alignment rotates the molecular system around the first alignment axis to make the second alignment.
6. For a minor alignment, repeat steps 2 and 3 using the right-half of the dialog box.

You can see the inertial axes of a coordinate system by choosing Show Inertial Axes on the Display menu (see page 279). For more information about coordinate systems, see “Coordinate Systems” on page 18.

## Set Bond Length

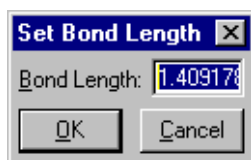
Set Bond Length displays the Set Bond Length dialog box for changing the length of a selected bond.

Select either a bond or the two atoms that form this bond, and then choose Set Bond Length. This bond must *not* be part of a ring structure. The bond length changes as soon as you L-click OK in the dialog box.

The Model Builder (see page 124) might remove this new bond length. To influence a bond length during model building, use Constrain Bond Length (see page 137).

You can also influence a bond length during a geometry optimization calculation (see “Restrains” on page 340).

## Set Bond Length Dialog Box



The dialog box first shows the existing bond length. Enter a new bond length, usually between 0.75 and 3 Ångstroms.

## Set Bond Angle

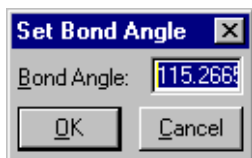
Set Bond Angle displays the Set Bond Angle dialog box for changing a selected bond angle.

Select either two bonds or the three atoms that include these bonds (see “Selecting” on page 32), and then choose Set Bond Angle. One but not both of the bonds can be part of a ring. The bond angle changes as soon as you L-click OK in the dialog box.

The Model Builder (see page 124) might remove this new bond angle. To influence a bond angle during model building, use Constrain Bond Angle (see page 138).

You can also influence a bond angle during a geometry optimization calculation (see “Restrains” on page 340).

## Set Bond Angle Dialog Box



The dialog box first shows the existing bond angle. Enter a number from 0 to 180 for the new bond angle.

## Set Bond Torsion

Set Bond Torsion displays the Set Bond Torsion dialog box for changing a torsion angle.

You can select either the four atoms or the three bonds forming the torsion angle, and then choose Set Bond Torsion. Only one end bond can be part of a ring structure. The bond torsion changes as soon as you L-click OK in the dialog box.

The Model Builder (see “Model Build/Add H & Model Build” on page 124) might remove this new torsion angle. You can also influence a torsion angle during model building, by using Constrain Bond Torsion (see page 139).

You can also influence a torsion during a geometry optimization calculation (see “Restrains” on page 340).

## Set Bond Torsion Dialog Box



The dialog box first shows the existing bond torsion angle. Enter a number from -360 to +360 for the torsion angle.

## Select Menu

The Select menu regulates the Selection tool. The Selection tool marks specific atoms, bonds, residues, or molecules for Hyper-

Chem operations. Selected atoms take on a different appearance (line width or color). If there are selected atoms in a molecular system, HyperChem acts usually only on these atoms.

The items on this menu instruct the Selection tool to work at different levels of molecular organization or to make specific types of selections:

<input checked="" type="checkbox"/> <b>Atoms<sup>a</sup></b>	The smallest unit of selection is an individual atom or bond
<input checked="" type="checkbox"/> <b>Residues<sup>a</sup></b>	The smallest unit of selection is an amino acid or nucleotide residue from a database or, if there are no residues in a molecular system, an atom or bond
<input checked="" type="checkbox"/> <b>Molecules<sup>a</sup></b>	The smallest unit of selection is a whole molecule
<input checked="" type="checkbox"/> <b>Multiple Selections<sup>b</sup></b>	Changes the Selection tool so that you can either accumulate selections ( <input checked="" type="checkbox"/> ) or replace the previous selection with each new selection (no check mark)
<input checked="" type="checkbox"/> <b>Select Sphere<sup>b</sup></b>	Changes the Selection tool so that you can select all atoms within a 3D sphere, plus atoms within a 2D rectangle ( <input checked="" type="checkbox"/> ) or only atoms within a 2D rectangle (no check mark)
<b>Select All</b>	Selects the whole molecular system
<b>Complement Selection</b>	Cancel the existing selections and selects all atoms previously not selected
<b>Select . . .<sup>c</sup></b>	Selects atoms using a variety of means. Selections can be via variables that are strings such as the symbol "C" or the named selection "MySelection" or via variables that are numeric such as atomic number, atomic charge, or atomic gradient. A numeric selection can specify a range.

<b>Name Selection . . .</b> <sup>c</sup>	Assigns a predefined name (POINT, LINE, PLANE, REACTANT, PRODUCT, PLOT1, PLOT2, GHOST ATOMS, NMR ATOMS, or QUANTUM ATOMS) or any name that you supply (up to 19 characters long) to selected atoms, and stores this selection for later use. Named selections apply only to this molecular system
<b>Name Molecule...</b> <sup>c</sup>	Assigns a name to an individual molecule of the workspace. Name is stored in HIN file. Can be particularly useful for database operations.
<b>Extend Ring</b> <sup>c</sup>	Selects all atoms in a ring if one bond or two bonded atoms in the ring are already selected
<b>Extend Side Chain</b>	Selects all atoms in a rotatable side chain if the first two atoms in the side chain are already selected
<b>Extend to sp<sup>3</sup></b>	Selects all atoms, starting from the current selection, until all bonds with only one atom selected are single bonds between sp <sup>3</sup> atoms. Use this to find the boundaries for a mixed mode quantum mechanics calculation
<b>Select Backbone</b>	Selects all atoms in the backbones of proteins or polynucleotides constructed from residues on the Databases menu or from a PDB file. For polypeptides, disulfide bridges are also selected

- a. Only one of these items can be active at a time, and only the active item has a check mark.  
 b. A check mark appears next to this item when it is on. Turning it on affects all further selecting, until you select it again to turn it off.  
 c. When this item is gray, it is temporarily unavailable. You must change some condition in HyperChem to make it available.

## Visible Atoms

The items on the Select menu act on all atoms in the molecular system, whether they are visible or not. The visibility of an atom depends on these factors:

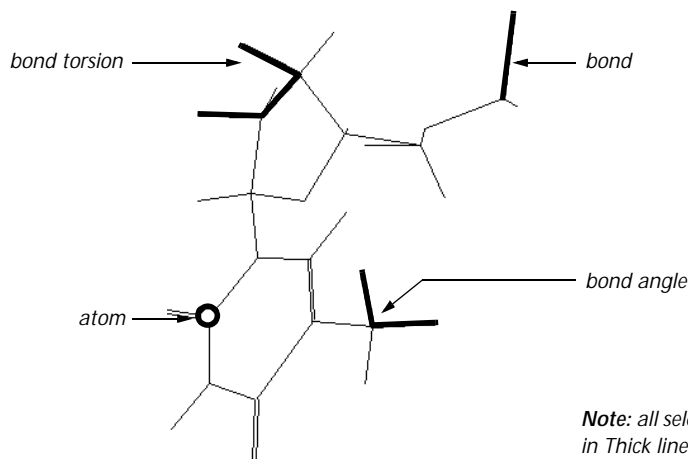
- The type of model you are using. For example, a spheres model hides atoms on the far side of a molecule.

- Settings on the Display menu. For example, Show Hydrogens can turn off the appearance of hydrogen atoms, but these atoms are still part of the molecular system.
- Location and size of molecules. For example, you can translate part of a molecular system off screen, but it is still part of the molecular system.
- Clipping slab settings. For example, the clipping slab might show only part of a molecular system.

## Atoms

When you choose Atoms (✓), the smallest unit of selection is an individual atom. With the Selection tool, you can then select individual atoms, bonds, and angles. This item remains active until you choose Residues or Molecules.

**Note:** Use Atoms for selection unless you specifically want to select residues or molecules.



To select atoms with the Selection tool:

- L-click on an atom or bond to select this component.
- R-click on an atom or bond to deselect it.
- Choose Multiple Selections on the Select menu to accumulate selections, without deselecting atoms.



- See “Selecting” on page 32 for methods of selecting bond angles and torsion angles.
- L-click on an empty area of the workspace to select all atoms in the workspace. R-click on any empty area of the workspace to deselect all atoms.

How the selection looks depends on your choice in the Preferences dialog box. A description of a single, selected atom appears on the status line with this information (from left to right):

<b>Atom number</b>	A number indicating the creation order of an atom within the molecule.
<b>Element name</b>	From the periodic table, for example, carbon.
<b>Atom name</b>	The atom name from the Brookhaven Data Bank or HIN file. By default, these names appear only for atoms found in proteins and nucleic acids.
<b>Force Field</b>	The force field that supplied the atom types for this molecular system (amber, mm+, bio+, or opl).
<b>Atom type</b>	The atom type for the force field.
<b>Charge</b>	The charge, if any, on this atom.
<b>Atomic coordinates</b>	The x, y, and z coordinates in the Molecular Coordinate System.

*Examples:*

These messages can appear on the status line when you select at the Atoms level.

For one atom:

Atom number 2 is Carbon C (mm+ \* CO) at -1.6188 0.6548 0

For two nonbonded atoms:

Distance from atom CA<sup>1</sup>97 to atom 1 HD<sup>1</sup> 108 is 3.3846Å

For two bonded atoms:

Bond distance from atom C86 to CA85 is 1.5364Å

For three bonded atoms:

Angle of atoms N96 - C86 - CA85 is 102.45°

1. Atom names in this and the following examples are from the Brookhaven Data Bank, for example, CA, C, HD.

**For three nonbonded atoms:**

Improper angle of atoms C98 - CD102 - C 85 is 88.703°

**For four bonded atoms:**

Torsion angle of atoms C98 - C 97 - N96 - CD102 is 116.38°

**For four nonbonded atoms:**

Improper torsion angle of atoms C98 - CA97 - C86 - CA85 is 126.95°

**For five or more atoms:**

Selected n atoms

where n= 5, 6, 7...

## Residues

When you choose Residues (✓), the smallest unit of selection is individual residues (amino acids or nucleotides). Residues must come from the Databases menu (see page 139) or from Protein Data Bank files (see page 503). HyperChem does not recognize as residues amino acids or nucleotides that you draw from individual atoms.

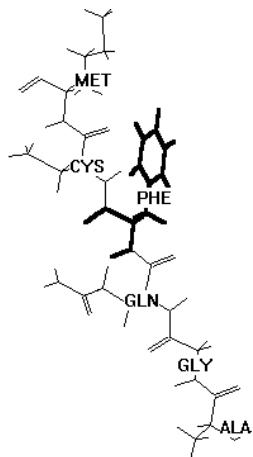
This item remains active until you choose Atoms or Molecules.

*Note:* If there are no residues in a molecule, this item works like Atoms.

**To select residues:**

- L-click on a residue to select it. If the Selection tool is on a molecule, it selects the nearest residue.
- R-click on a residue to deselect it.
- Choose Multiple Selections on the Select menu to accumulate selections, without deselecting any atoms.
- L-click on an empty area of the workspace to select all atoms in the molecular system. R-click on an empty area to deselect all atoms.

A description of your latest selection appears on the status line with this information:



*Note: selected residue appears in Thick line*

- |                                    |  |
|------------------------------------|--|
| <b>Number of residues selected</b> | This number can be from 0 to all residues in the molecular system.   |
| <b>Residue name and number</b>     | This is the name and sequence number for the first four residues you select, plus the last residue you select. |

**Example:**

This message appears on the status line when you select at the Residues level.

30 residues selected: THR 1 THR 2 CYX 3 CYX 4 ... ASN 46

## Molecules

When you choose Molecules (✓), the smallest unit of selection is an individual molecule. This item remains active until you choose Atoms or Residues.

To select molecules:

- L-click on a molecule to select it. If the Selection tool is close to two molecules, it selects the nearest molecule.
- R-click on a molecule to deselect it.
- Choose Multiple Selections on the Select menu to accumulate selections, without deselecting atoms.

- L-click on an empty area of the workspace to select all atoms in the workspace. R-click on an empty area to deselect all atoms.

The numbers of the selected molecules appear on the status line.

*Example:*

This is an example of the message that appears on the status line when you select at the Molecules level:

2 molecules selected

## Multiple Selections: Accumulating Selections

With Multiple Selections on (✓), you can accumulate selections. When this item is off (no check mark), each new selection replaces the previous selection.

*Note:* You can also select more than one structure at a time by using group selection (see the next section). Group selection does *not* require Multiple Selections to be on.

The type of structure selected depends on your choice of the Atoms, Residues, or Molecules on this menu. The number of selected atoms, residues, or molecules appears on the status line.

## Select Sphere and Group Selecting

Select Sphere controls group selection with the Selection tool.

*Group selection* means that you can select all atoms, residues, and molecules in part of a molecular system. Group selection is the easiest way to select a set of atoms.

To prepare for group selection:

1. Choose Atoms, Residues, or Molecules on the Select menu. This choice regulates the level of group selection. For example, if you choose Molecules, group selection works only for whole molecules.
2. L-click on the Selection tool icon to activate the Selection tool.

*Note:* Group selecting does *not* require Multiple Selections to be on.

## Selecting in Two Dimensions

This procedure selects all atoms that project onto a 2D selection rectangle. You can adjust the location and size of the selection rectangle. The Selection tool ignores information about the depth of the molecular system and treats all atoms as if they were in the plane of the screen.

To select in two dimensions:

1. Turn off Select Sphere on the Select menu. There should be no check mark to the left of this item.
2. Move the Selection tool to one corner of the region you want to select.

*Note:* You can also do this selection with Select Sphere on. At step 2, start your selection in an empty area of the workspace.

3. Hold down both mouse buttons, but press the *left* button first.
4. Move the Selection tool diagonally across the structures to select. A selection rectangle appears. Adjust the size and position of the rectangle by moving the mouse.
5. Release both mouse buttons. The number of structures selected appears on the status line.
6. To deselect part of a selection, repeat steps 1–5, but at step 3 hold down the *right* mouse button *before* holding down the left button.

---

group selection using a combination of mouse buttons

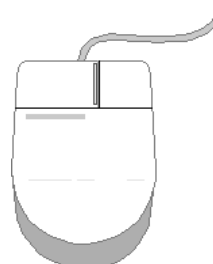
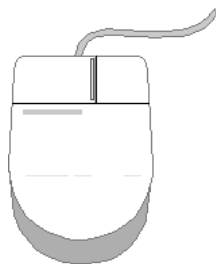
To select:  
LR-drag

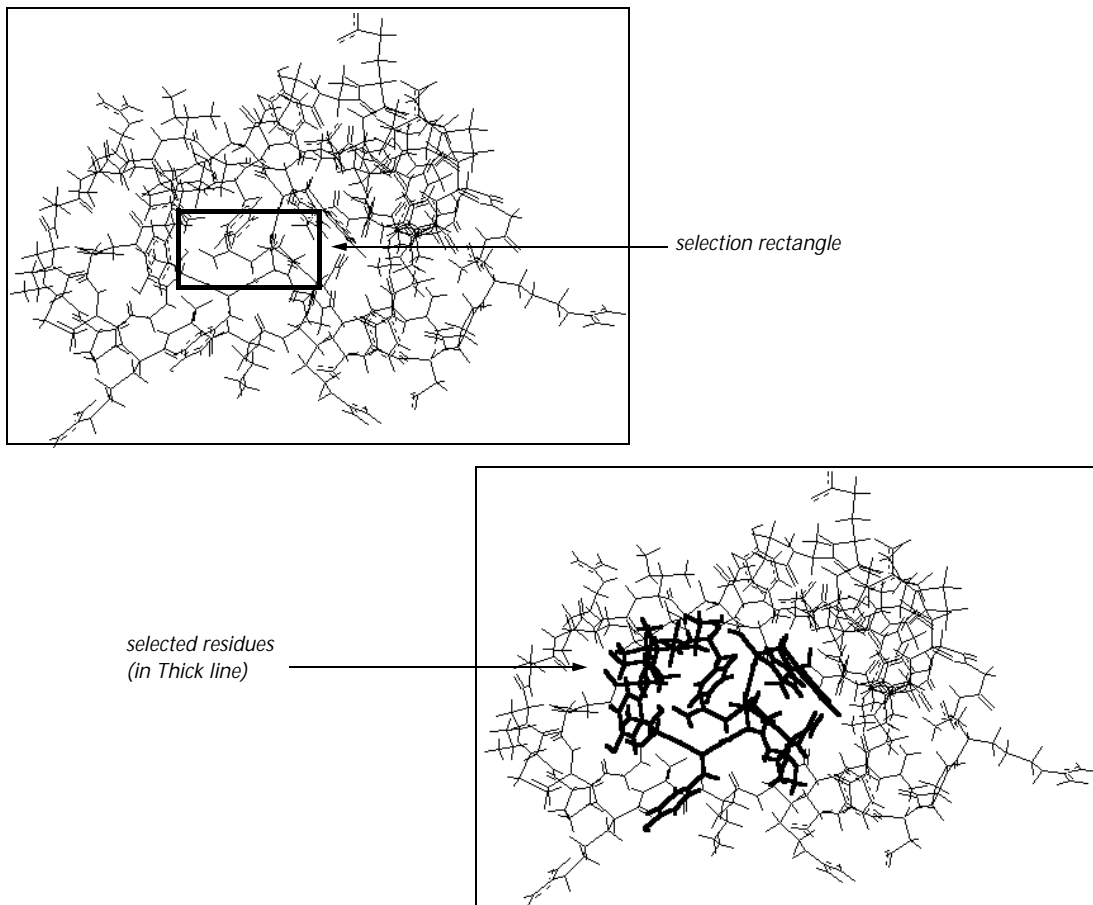
To deselect:  
RL-drag

Click:

L → R

L ← R





## Selecting in Three Dimensions

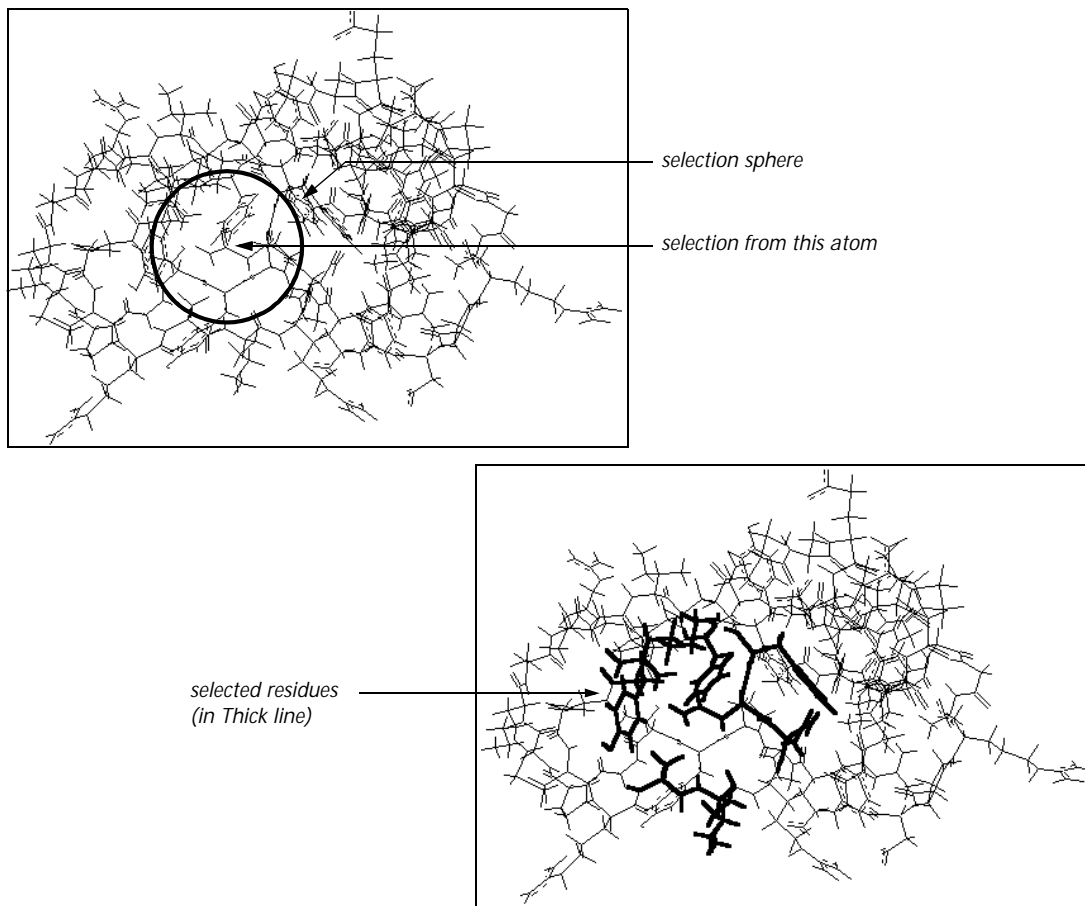
This procedure selects all structures within a 3D sphere. You select only the atoms that are within a specified distance and in all directions from an atom. You adjust the diameter of a circle that represents the selection sphere. You *must* start this selection at an atom.

To select in three dimensions:

1. Turn on Select Sphere (✓) on the Select menu.
2. Move the Selection tool to the atom at the center of the selection.

*Note:* If you begin group selecting away from an atom, the Selection tool uses a 2D rectangle for selection, even if Select Sphere is on.

3. Hold down both mouse buttons, but press the *left* button first.
4. Move the Selection tool diagonally across the structures to select. A circle, representing a sphere, appears. Adjust the size of the circle by moving the mouse in any direction. The number of the first atom selected and the radius of the sphere, in Ångstroms, appear on the status line.
5. Release both mouse buttons. The number of structures selected appears on the status line.
6. To deselect atoms within a sphere, repeat steps 1–5, but at step 3 hold down the *right* mouse button *before* holding down the left button.



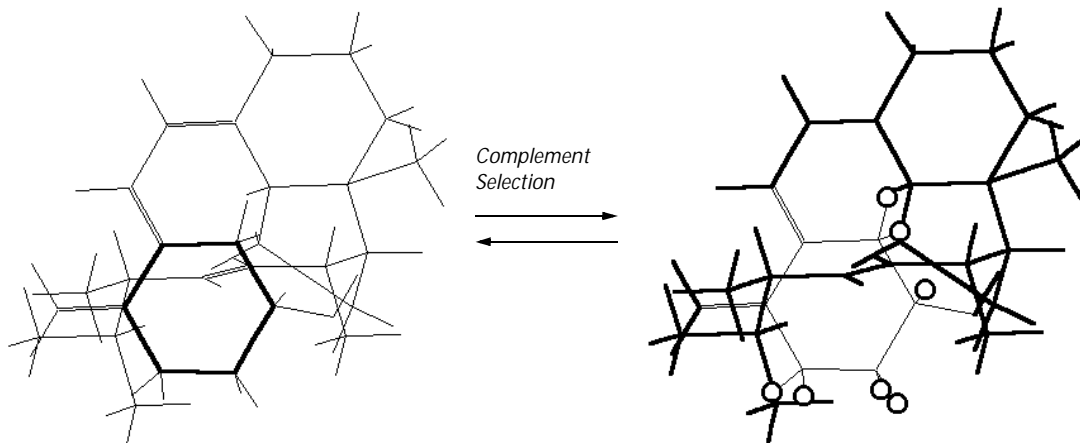
## Select All

Select All changes the selection to include all visible atoms (see “Visible Atoms” on page 222) in the molecular system. Choosing this item is equivalent to L-clicking in an empty part of the workspace.



## Complement Selection

Complement Selection deselects the existing selection and, instead, selects all atoms that were not selected before. When you have selected the only molecule in a molecular system, Complement Selection deselects this molecule.



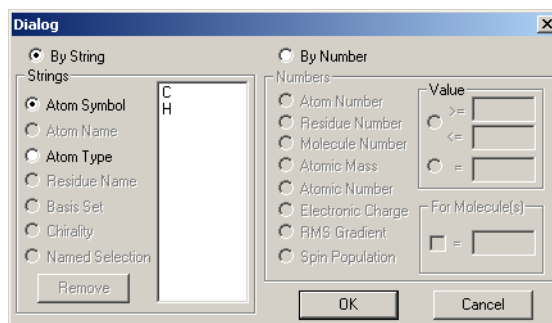
## Select

Using Select you can select atoms, residues, and molecules by string or by number. These selections occur immediately and either replace previous selections (with Multiple Selections off) or add to existing selections (with Multiple Selections on).

**Note:** Select is gray until there is a molecular system in the workspace.

When you choose Select, the Select dialog box (see page 233) appears listing the variety of possible selections. You may select via a string that assigned to a selection of atoms (see Name Selection in the following section) or a string that is the value of a string variable such as Chirality (R or S). You may also select via a numerical value assigned to atoms such as atomic number, atomic mass, atomic charge, etc. Numerical selections can specify a range. For example, you can select atoms that have a gradient greater than a minimum value.

## Select Dialog Box



**By string** L-click to select atoms with a specified value of a string variable. L-click on one of the radio buttons to see the appropriate string values in the panel to the right. The string variables describing collections of atoms can be chosen from the following strings:

**Atom Symbol** The conventional element symbol for an atom such as C, N, O, or F.

**Atom Name** The conventional name for an atom in an amino acid or nucleic acid. Applies only to the names defined in templates for these macro-molecule components. Examples are CA, 1HA, and 2HA for an alpha-carbon and the hydrogens attached to it.

**Atom Type** The molecular mechanics atom type such as CT, HC, etc.

**Residue Name** The name of an amino acid or nucleic acid residue such as GLY, ALA, etc.

**Basis Set** The *ab initio* or DFT basis set assigned to atoms. An example is 6-31G\*.

<b>Chirality</b>	The chirality, R or S, assigned to a chiral atom.
<b>Named Selection</b>	The list of named selections previously assigned using the Name Selection dialog box. The list can include predefined choices such as LINE, POINT, and PLANE) if they have been assigned, and other names you set up. conventional element symbol for an atom such a C, N, O, or F.
<b>Remove</b>	Permanently removes a named selection from the list of named selections kept by HyperChem. See Name Selection below.

**By Number** L-click to select atoms with a specified value of a numeric variable. L-click on one of the radio buttons to select the numeric variable and then specify its value or range of values.

If multiple molecules are in the workspace, you will need to specify, which molecules you are selecting from. The molecules can be specified in a list such as 1,2,4 or in a range such as 1-7, or combined such as 1-2,4 to make an atom selection in molecules 1,2, and 4. If For Molecule(s) is not checked, then the atom selection is made for all molecules.

Numeric variables can be chosen from the following values:

**Atom Number** Selects atoms. The number is the number of an atom within a molecule. Remember that each molecule has its own atom numbers starting at 1 (there may be many atoms numbered 1). A global atom numbering scheme requires specifying both the atom number *and* the molecule number.

<b>Residue Number</b>	Selects all atoms in specified residues. The number of a residue starts at 1 within each molecule. A global residue numbering scheme requires specifying both the residue number <i>and</i> the molecule number.
<b>Molecule Number</b>	Selects all atoms in specified molecules. The molecule number starts at 1.
<b>Atomic Mass</b>	Selects atoms based on their atomic mass. Provides a way to find atoms that have had their mass changed for isotope studies.
<b>Atomic Number</b>	Selects atoms based on their atomic number 1, 2, ... etc.
<b>Electronic Charge</b>	Selects atoms based on their atom charge calculated in quantum mechanical calculations, coming from amino acid and nucleic acid templates, etc. The atom charge is a positive or negative floating point number, e.g. -0.136, specifying excess number of electrons on that atom.
<b>RMS Gradient</b>	Selects atoms based on their root-mean-square gradient of the energy with respect to atom displacement in space. In an optimized structure these would be zero for all atoms.
<b>Spin Population</b>	Selects atoms based on their excess of alpha (spin up) electrons over beta (spin down) electrons. Only open-shell systems have non-zero spin populations. Unrestricted quantum mechanical calculations lead to non-zero spin populations for the associated atoms.
<b>Value</b>	Numeric values can be specified either as exact values, e.g. 3, or in a range, e.g.

greater than 0.0 and less than 1.5. Floating point values should be specified with a range because of precision issues.

For  
Molecule(s)

If checked this field specifies the molecules to be included in selecting atoms. If unchecked then all molecules are included. The molecules can be listed as 1,2,5, etc. or can be given as a range, e.g. 13-22, or both.

## Selecting by String

To select atoms or molecules by an assigned string:

1. Choose Select on the Select menu.
2. L-click on By String in the Select dialog box.
3. L-click on one of the radio buttons specifying a string variable.
4. L-click on one of the values presented in the panel to the immediate right.
5. L-click on OK.

*Note:* Strings that are inappropriate for use in the current context will be grayed.

## Selecting by Number

To select atoms, residues or molecules by number:

1. Choose Select on the Select menu.
2. L-click on By number in the Select dialog box.
3. L-click on one of the radio buttons specifying a number variable.
4. Enter the appropriate numeric range or exact value.
5. If appropriate, check For Molecule(s) and specify a set and/or range of molecules. If unchecked, all molecules is the default.
6. L-click OK.

## Numbering

HyperChem assigns a number to each atom, residue, and molecule, starting with 1 for each type of structure. This is the numbering order:

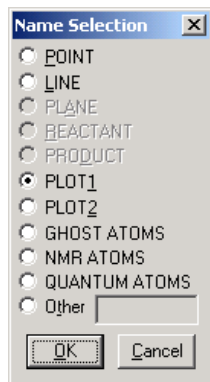
- **Atoms:** If you draw molecules in HyperChem, atoms are numbered in the order you draw them. Molecules from the Protein Data Bank have defined numbering. Atom numbering starts from 1 for each molecule.
- **Residues:** In polypeptides, residue 1 is at the N-terminus. In polynucleotides, residue 1 is at the 5' end, unless you constructed a polynucleotide starting from the 3' end (Backwards). Residue numbering starts from 1 for each molecule.
- **Molecules:** Molecules are numbered in the order you added them to the molecular system.

## Name Selection

Name Selection shows you the Name Selection dialog box for assigning a name to selected atoms. When atoms have a selection name, you can easily select them again. The name can be "special" (POINT, LINE, PLANE, REACTANT, PRODUCT, PLOT1, PLOT2, GHOST ATOMS, NMR ATOMS, QUANTUM ATOMS) or any name that you supply, up to 19 characters long (Other). "Special" selections have properties in addition to describing a set of atoms (see the next section).

*Note:* Name Selection is gray until you select at least one atom. Some special names are available only under some conditions.

## Name Selection Dialog Box



- |                 |  |
|-----------------|--|
| <b>POINT</b>    | The center of mass of the selected atom or atoms.  |
| <b>LINE</b>     | The best-fit straight line between two or more atoms, or the primary inertial axis. <sup>1</sup>   |
| <b>PLANE</b>    | The best-fit plane connecting three or more atoms, or the plane formed by the primary and secondary inertial axes.   |
| <b>REACTANT</b> | This selection contains the reactant atoms for setting up a Synchronous Transit transition-state search (see “Reaction Map . . .” on page 353). This line in the dialog box is active only if you have selected half of the atoms in the workspace.  |
| <b>PRODUCT</b>  | This selection contains the product atoms for setting up a Synchronous Transit transition-state search (see “Reaction Map . . .” on page 353). This line in the dialog box is active only if you have selected half of the atoms in the workspace.   |
| <b>PLOT1</b>    | This selection contains a set of 2, 3, or 4 atoms defining either a bond length, bond angle, or bond torsion that are being specified as the first variable in a possible 1D or 2D potential energy plot. If a potential energy plot is requested and there is no current selection, then PLOT1 will be the independent variable for the plot. If there is a current selection for a one-dimensional plot then PLOT1 is ignored. |

1. The primary inertial axis corresponds to the smallest eigenvalue of the moment of inertia matrix.

PLOT2	This selection contains a set of 2, 3, or 4 atoms defining either a bond length, bond angle, or bond torsion that are being specified as the second variable in a possible 2D potential energy plot. If a potential energy plot is requested and there is no current selection, then PLOT1 and PLOT2 will be the independent variables for the 2D plot. PLOT1 and PLOT2 both need to be defined to enable a two-dimensional potential energy plot.
GHOST ATOMS	This selection contains the set of atoms to be defined as ghost atoms for potential use in <i>ab initio</i> or DFT calculations. These atoms, depending on whether ghost atoms are invoked by the setup of <i>ab initio</i> or DFT calculations will have basis functions but their nuclei will not be used in the calculation. This selection is rigorously equivalent to you choosing “ghost-atoms” as a named selection.
NMR ATOMS	This selection contains the set of atoms to be given a nuclear spin and used in the HyperNMR module. See the documentation for HyperNMR in the Module manual for further information. This selection is rigorously equivalent to you choosing “nmr-atoms” as a named selection.
QUANTUM ATOMS	This selection contains the set of atoms to be defined as the quantum atoms in a mixed molecular mechanics - quantum mechanics (MM/QM) calculation. It is specifically used in the HyperNMR module and is not yet available for MM/QM calculations outside HyperNMR. Future releases of HyperChem would be expected to use this named selection for ordinary MM/QM calculations which currently use the current selection to distinguish the MM region from the QM region. This selection is rigorously equivalent to you choosing “quantum-atoms” as a named selection.
Other	A name that you supply, up to 19 characters long, with at least one lowercase character, and no spaces.

**Note:** GHOST ATOMS, NMR ATOMS, and QUANTUM ATOMS have rigorous equivalents ghost-atoms, nmr-atoms, and quantum-atoms unlike other pre-defined named selections. This is for com-



patibility with earlier use of these named selections prior to their appearing in the above Select dialog box.

## Naming Selections

To name a selection:

1. Using the Selection tool, select any set of atoms, residues, or molecules.
2. Choose Name Selection on the Select menu.
3. L-click on POINT, LINE, PLANE, REACTANT, PRODUCT, PLOT1, PLOT2, GHOST ATOMS, NMR ATOMS, QUANTUM ATOMS or Other. If you choose Other, enter a name, from one to 19 characters long, but don't use all uppercase letters. Names with all uppercase characters are reserved for HyperChem's special selections, like POINT.
4. L-click OK.

Each molecular system can have only one POINT, LINE, PLANE, REACTANT, PRODUCT, PLOT1, PLOT2, GHOST ATOMS, NMR ATOMS, or QUANTUM ATOMS selection, but any number of other selection names. For example, if you use POINT a second time, the new selection replaces the last POINT selection. HyperChem requests confirmation of this replacement.

To remove a named selection, use the Remove function of the Select Dialog Box (page 233).

## Using Named Selections

You can use these names any time during a HyperChem session to select a set of atoms. The names belong to a molecular system. You can store these names, with the molecular system, in a HIN file and use them for another HyperChem session.

HyperChem uses the special names POINT, LINE, PLANE, REACTANT, PRODUCT, PLOT1, PLOT2, GHOST ATOMS, NMR ATOMS, QUANTUM ATOMS to indicate particular properties of a selection. You can use these special names to recall a selection, and HyperChem can use the special properties of these selections for other operations. For example, you can translate molecules to a POINT (see

“Translate Dialog Box” on page 211) or invert a molecule through a POINT (see “Invert” on page 206).

HyperChem uses the special names REACTANT and PRODUCT for setting up a Synchronous Transit transition-state search (see “Reaction Map . . .” on page 353). These lines in the dialog box are active only if you have selected half of the atoms in the workspace.

*Note:* Selection names belong to individual molecular systems and appear in the Select dialog box only when that molecular system is in the workspace. If you save a molecular system in a HIN file, HyperChem also stores named selections.

You can also use named selections (Other) for setting restraints in a calculation (see the next section) and for defining lengths and angles that are averaged or plotted in a molecular dynamics simulation (see “Molecular Dynamics Averages Dialog Box” on page 377).

## Using Named Selections for Restraints

HyperChem can use 1, 2, 3, or 4 atom named selections to restrain a molecular system during a calculation (see “Setting Up Restraints” on page 342). These selections can be either predefined (POINT, LINE, and PLANE) or you can define them yourself (Other).

- A single atom can be tethered to a point in space
- Distances between any two atoms, including bond lengths
- Angles formed by any three atoms, including bond angles
- Torsion angles formed by any four atoms, including improper torsions

The named selections appear in the Restraint Forces dialog box with a prefix indicating the number of atoms in the selection. For example, a selection named C11/N78 that describes the distance between two nonbonded atoms appears as 2-C11/N78.

HyperChem stores these restraints, along with the named selections and the molecular system, in a HIN file.

## Using Named Selections in MD Calculations

HyperChem can use 2, 3, or 4 atom named selections to plot changing values or to calculate averages during a molecular

dynamics simulation (see “Molecular Dynamics Averages Dialog Box” on page 377). These selections can be either predefined (POINT, LINE, and PLANE) or definitions of your choice (Other). The named selections represent these values:

- Distances between any two atoms, including bond lengths
- Angles formed by any three atoms, including bond angles
- Torsion angles formed by any four atoms, including improper torsions

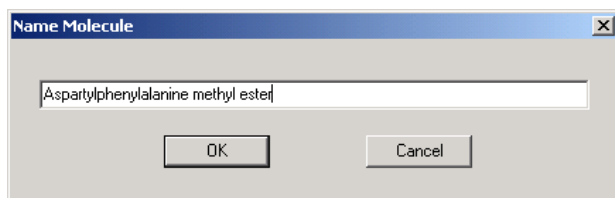
The named selections appear in the Molecular Dynamics Averages dialog box, described on page 377.

## Name Molecule

Name Molecule is used to give a name to a selected molecule. While not required, it may be useful to assign a relatively long name to specified molecules. For example, this name may become part of a database of molecules. The name is stored with a molecule in HIN files and is displayed on the status line when a molecule is selected.

The name may contain up to 64 characters and includes capitalization, numbers, symbols, blank spaces, etc.

### Name Molecule Dialog Box



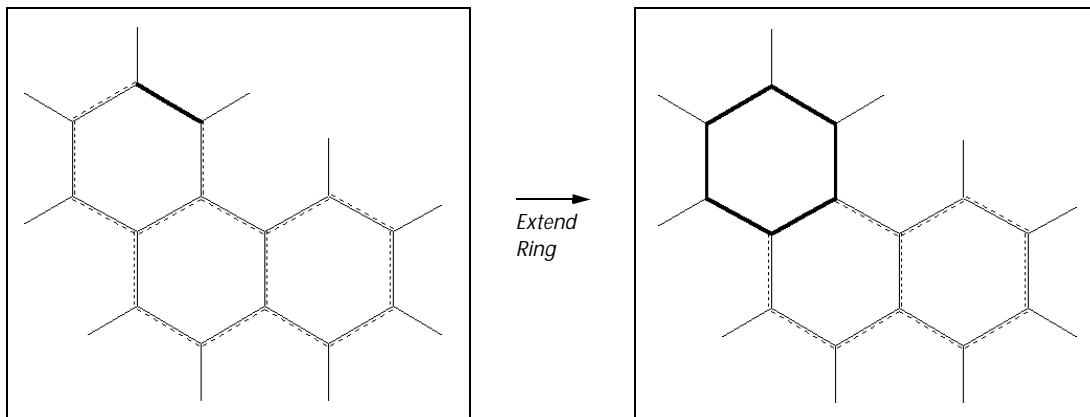
## Extend Ring

Extend Ring selects all atoms in a ring, starting with two selected, covalently bonded atoms.

To use Extend Ring:

1. Choose Atoms on the Select menu.

2. With the Selection tool, select one bond or two adjacent atoms in a ring. To select two atoms, first turn on Multiple Selections.
3. Choose Extend Ring.



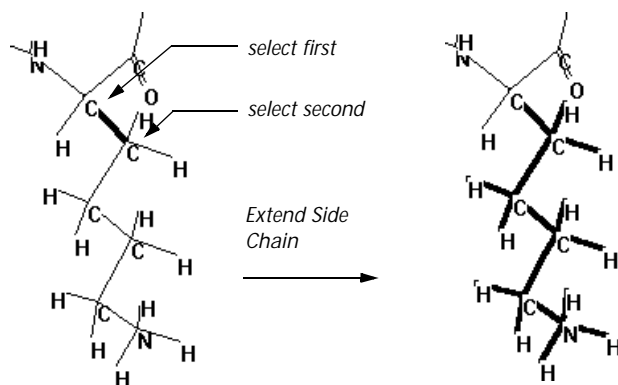
**Note:** You can also select a ring by double-clicking with the Selection tool on an atom or bond in the ring. You must first choose Atoms on the Select menu.

## Extend Side Chain

Extend Side Chain selects all atoms in a side chain that can rotate. For HyperChem, each bond that is not part of a ring defines two side chains, one to either side of the bond.

To use Extend Side Chain:

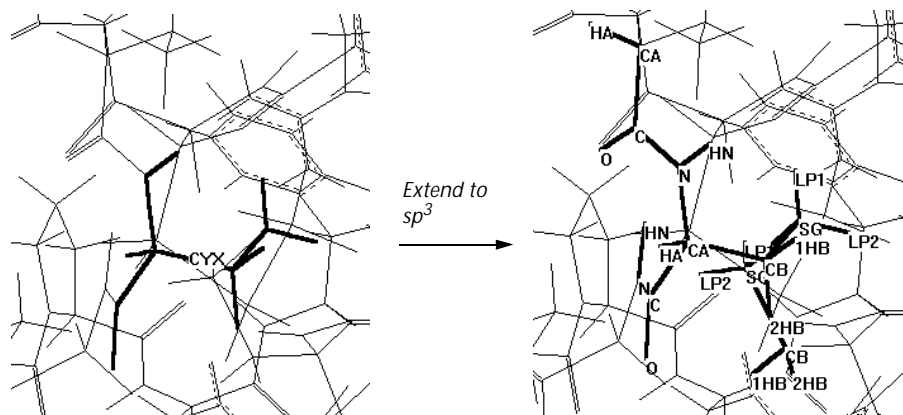
1. Choose Atoms and turn on Multiple Selections on the Select menu.
2. With the Selection tool, select two adjacent atoms that are not part of a ring. The order of selection determines the direction of the search for a side chain. HyperChem selects the side chain past the second selected atom.
3. Choose Extend Side Chain.



**Note:** You can also extend a side chain by double-clicking with the Selection tool on a bond. You must first choose Atoms on the Select menu. Then double-click on one end of a bond (from the midpoint of the bond to the bonded atom). HyperChem selects the side chain connected to the nearest atom.

## Extend to sp<sup>3</sup>

Extend to sp<sup>3</sup> starts at selected atoms and selects additional atoms, in all directions, until it meets sp<sup>3</sup>-sp<sup>3</sup> single bonds or singly bonded atoms. The extended selection includes the near half of the sp<sup>3</sup>-sp<sup>3</sup> bonds.



Use this item if you are going to apply a semi-empirical quantum mechanical calculation to part of a molecule. This HyperChem fea-

ture lets you use both molecular mechanical and quantum mechanical calculations on parts of the same molecule.

HyperChem considers these as  $sp^3$  atoms:

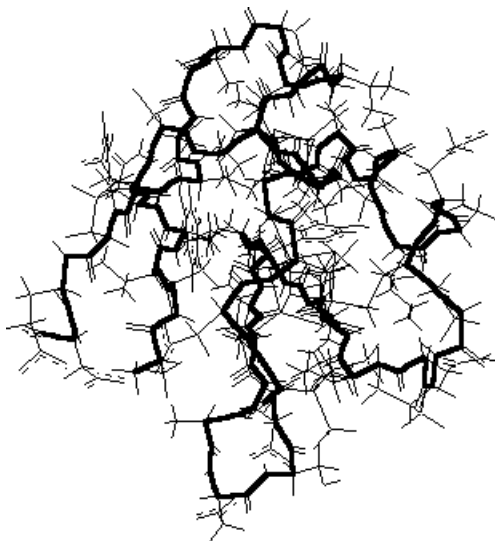
- A carbon bonded to four other atoms.
- A nitrogen bonded to three other atoms.
- An oxygen bonded to two other atoms.
- The nitrogen bonded to the alpha carbon in a peptide bond. This assignment for the peptide bond nitrogen allows boundaries between amino acid residues.

To extend a selection to  $sp^3$ - $sp^3$  single bonds, first select at least one atom, bond, or residue. Then choose Extend to  $sp^3$  on the Select menu.

*Note:* When you run a semi-empirical calculation on a selected part of a molecule, HyperChem replaces C, O, and N atoms on the far half of the  $sp^3$ - $sp^3$  bonds with parameterized fluorine atoms. These capping atoms have the appropriate electronegativity or electropositivity to represent the replaced atoms during quantum mechanical calculations (see *HyperChem Computational Chemistry*).

## Select Backbone

Select Backbone automatically selects all atoms that provide structural continuity in proteins and nucleic acids constructed with residues from the Databases menu or from PDB files. Select Backbone works on all polymers, constructed from residues, in the molecular system.



*ferredoxin*



*single strand of DNA*

For proteins, the backbone includes alpha carbons, peptide bonds, and disulfide bonds. For polynucleotides, the backbone includes all phosphodiester bonds, alternating with the C<sub>3'</sub>-C<sub>4'</sub> carbons in the sugar (ribose or deoxyribose).

This command does not work with polymeric molecules that you construct, atom by atom, with the Drawing tool. To select a backbone in those molecules, use the Selection tool to select the atom at one end of the backbone. Then L-drag to the atom at the other end of the backbone. Multiple Selections does not have to be on.





# Display Menu

The Display menu determines the appearance of molecules in the workspace. The Display menu has these commands:

<b>Scale to Fit</b>	Changes the magnification to fit all selected atoms into the workspace. If there is no selection, this item fits the whole molecular system into the workspace. Same as pressing [Spacebar]
<b>Overlay</b>	Places two selected molecules on top of one another. Requires two 3-atom selections.
<b>RMS Fit and Overlay</b>	Overlays molecules by minimizing the distance between corresponding atoms.
<b>Show All</b>	Displays all atoms in the workspace, canceling Show Selection Only or Hide Selection
<b>Show Selection Only</b>	Displays only the selected atoms that are presently visible in the workspace. Hides atoms that are not selected
<b>Hide Selection</b>	Turns off the display of selected atoms. Leaves visible only atoms that are not selected
<b>Rendering . . .</b>	Chooses options for representing all molecules in a molecular system. This model applies to the whole molecular system
<b>Last Rendering</b>	Restores the previous mode of rendering in the Workspace. Same as pressing [F2]. This option is not available (grayed out) while an isosurface is being displayed
<b>Raytrace...</b>	Creates high quality ray-traced molecular graphics.
<b>√ Show Isosurface<sup>a</sup></b>	If an isosurface has been calculated and is available for display, this controls whether it will be displayed or not. Same as pressing [F3]
<b>Isosurface . . .</b>	If an isosurface has been calculated and is available for display, this opens the Contour Isosurface Options dialog box or the Orbital Isosurface Options dialog box. Same as pressing [F4]

✓ Show Hydrogens <sup>a</sup>	Shows molecules with (✓) or without (no check mark) their hydrogen atoms
✓ Show Periodic Box <sup>a,b</sup>	Shows the outline of a periodic box (✓) or removes the outline (no check mark), leaving all molecules and the periodic boundary conditions. To place a periodic box in the workspace and to set up periodic boundary conditions, choose Periodic Box on the Setup menu
✓ Show Multiple Bonds <sup>a</sup>	Shows double, triple, and aromatic bonds with different types of lines (✓) or shows all bonds as single lines (no check mark)
✓ Show Aromatic Rings as Circles <sup>a</sup>	Shows aromatic rings with inner circle to indicate aromaticity (✓) or as the default with dotted lines to indicate aromatic bonds (no check mark)
✓ Show Hydrogen Bonds <sup>a</sup>	Shows hydrogen bonds as dotted lines (✓) or not at all (no check mark)
Recompute H Bonds	Calculates hydrogen bonding and shows hydrogen bonds if Show Hydrogen Bonds is on
✓ Show Inertial Axes <sup>a</sup>	Calculates and displays the inertial axes of selected atoms or of the molecular system
✓ Show Dipole Moment	Displays the dipole moment of the selected atoms or of the molecular system, if a semi-empirical or <i>ab initio</i> calculation has been performed
Labels . . .	Adds labels to identify atoms or residues in Sticks renderings of molecules. A Labels dialog box gives nine choices of labels for atoms, three choices for residues, or no labels at all
Color . . .	Changes the color applied to selected atoms, residues, or molecules. This color remains after you deselect the atoms
Element Color . . .	Changes the default color for each element

a. A check mark appears next to this item when it is on. Turning it on affects all further displays, until you choose it again to turn it off.

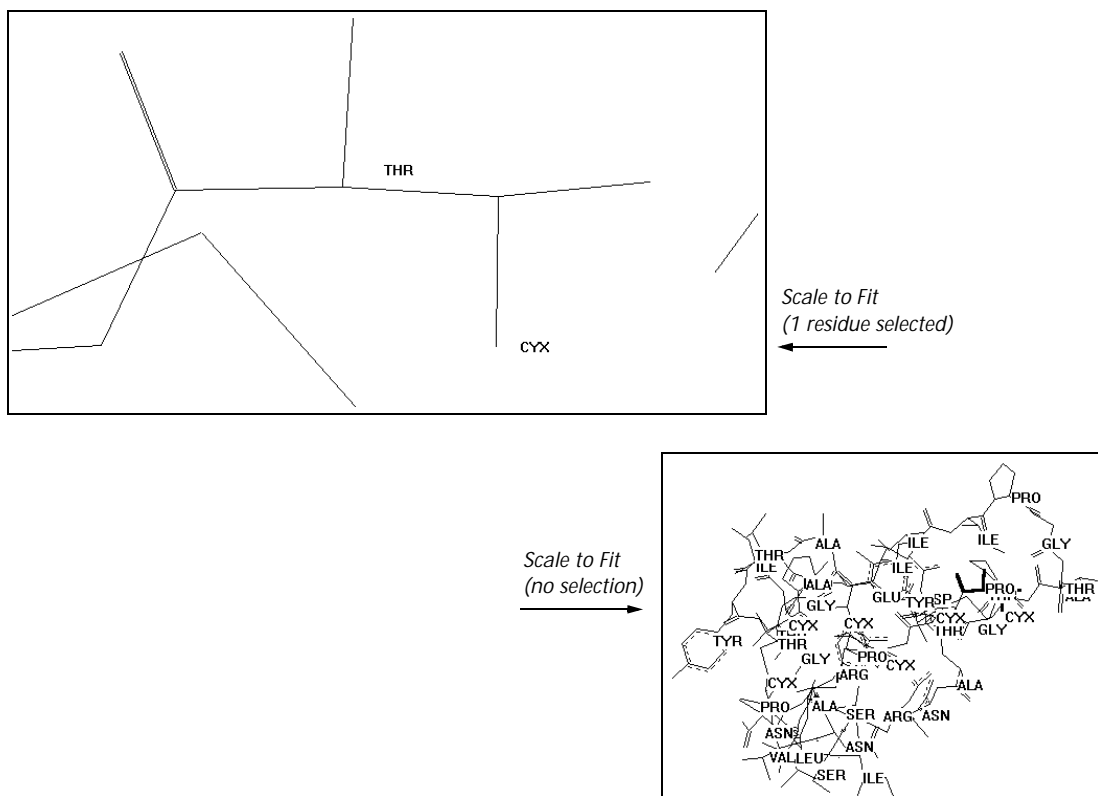
b. When this item is gray, it is temporarily unavailable. You must change some condition in HyperChem to make it available.

## Scale to Fit

Scale to Fit centers selected atoms or the whole molecular system in the workspace. HyperChem changes the magnification so that the molecules occupy about three quarters of the height of the workspace. This sets Zoom Factor to 1 (see “Zoom Dialog Box” on page 212).

Scale to Fit also resets the clipping planes to include the selection or the whole molecular system.

**Note:** Press [Spacebar] as a shortcut to choosing Scale to Fit.



You can use Scale to Fit for these adjustments:

- Focusing on selected atoms or molecules.

Group select (see “Select Sphere and Group Selecting” on page 227) atoms of interest, and then press [Spacebar]. The selected

structures fill the workspace. To reverse this enlargement, deselect the atoms and press [Spacebar].

- Locating molecules that are out of the workspace.

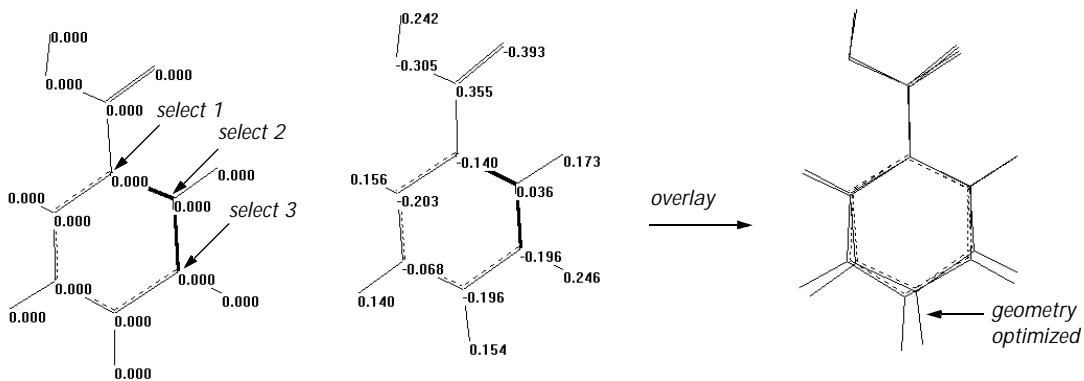
Deselect all atoms (R-click with the Selection tool in an empty part of the workspace) and press [Spacebar] to return the whole molecular system to the center of the workspace.

## Overlay

Overlay places one molecule on top of another so that you can make structural comparisons.

To use Overlay:

1. Choose Atoms and turn on Multiple Selections on the Select menu.
2. Select three atoms in one molecule. These can be, for example, the three atoms defining a bond angle. This becomes molecule number 1.
3. Select three atoms in a second molecule. These can be any three atoms, but your selection should correspond to selected atoms in molecule 1. This becomes molecule 2.
4. Choose Overlay. HyperChem moves molecule 1 to coincide with molecule 2.



after Model Building (left) then geometry optimization with AM1 (right)

HyperChem remembers the order in which you selected the atoms (1, 2, and 3). It first places atoms numbered 1 at the same coordinates. Then it superimposes the bonds between atoms 1 and 2. Finally, it rotates the first molecule so that atoms 1, 2, and 3 in both molecules are in the same plane.

## RMS Fit

### Overview

RMS Fit is a module for overlaying molecules by minimizing the distance between corresponding atoms. The RMS value of the residual distances (in Ångstroms) is reported on the status line as a measure of the similarity of the structures.

RMS Fit is fully compatible with HyperChem's select-and-operate way of working. If there are atoms selected, these atoms and only these atoms are overlaid. If no atoms are selected (and two molecules present) then all atoms are considered in the RMS fitting process.

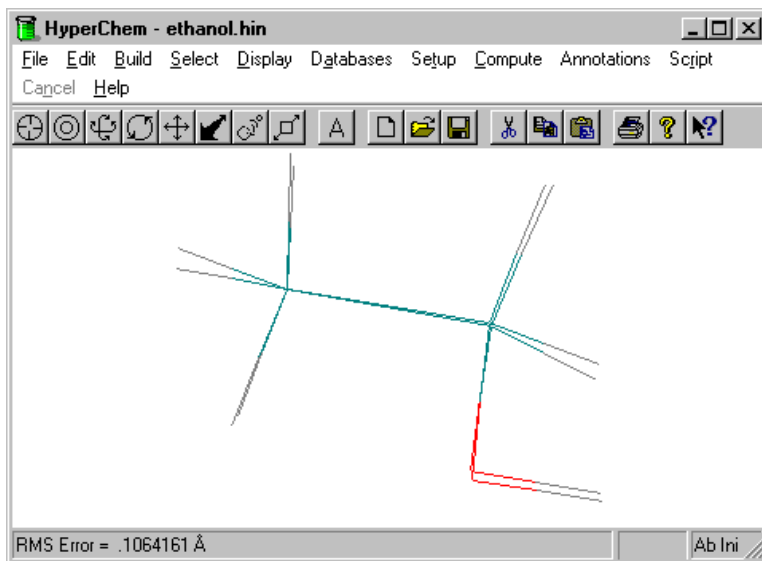
The executable file for RMS Fit is HCRMSFIT.EXE, which is in your Program directory.

### Tutorials

These tutorials illustrate how to use the RMS Fit module to overlay two molecules, using either all atoms or only selected atoms.

#### 1. Overlaying Two Structures Of The Same Molecule

Given two structures of the same molecule on the screen, such as two structures obtained from a single dynamics run, they can be overlaid in a very straightforward manner. Simply choose RMS Fit and Overlay from the Display menu, and RMS Fit moves molecule number two over molecule number one in such a way that the RMS distance between corresponding atoms is minimized. The residual RMS distance is reported on the HyperChem status line. The figure shows two slightly different structures of ethanol overlaid



## 2. Overlaying Selected Atoms Only

RMS Fit can overlay selected parts of a molecule, for instance if you wish to compare the similarity of the backbones of two peptides. To do this, select the atoms of interest in the two molecules and choose RMS Fit and Overlay from the Display menu. RMS Fit moves the second molecule selected so that the selected atoms fit as closely as possible.

RMS Fit offers two options for the way in which the overlay is carried out. You can overlay either by *molecule numbering* or by *selection order*. In general, overlay by molecule numbering is appropriate when overlaying two structures of the same molecule, or similar molecules in which the atoms to be overlaid are in the same sequence, while overlay by selection order is appropriate when you have specific atoms you want to overlay in distinct molecules.

To overlay by molecule numbering:

5. Choose the Databases/Amino Acids... menu item in HyperChem to bring up the amino acids dialog box.
6. Set the conformation to Beta sheet by clicking on the appropriate radio button.

7. Build a strand of six glycine residues, capped by acetyl and methyl amine groups, by clicking on the Ace Gly Gly Gly Gly Gly Gly NMe buttons.
8. Build a similar strand, but with one glycine replaced by a proline residue, by clicking on the Ace Gly Gly Gly Pro Gly Gly NMe buttons. The proline group cannot adopt a beta sheet conformation, and alters the structure of the backbone. You will hear a beep when you click on the proline, alerting you to this fact.
9. Choose the Select/Select Backbone menu item. Both backbones should be selected.
10. Launch the RMS Fit module by choosing RMS Fit and Overlay in HyperChem's Display menu.
11. When you see the Options dialog box, choose Molecule Numbering, and click on OK to continue. The structures are overlaid with an RMS error of 1.67 Å.

To overlay by selection numbering:

1. Draw 1,2-difluoroethene ( $\text{FCH}=\text{CHF}$ ) and 1,2-difluoroethane ( $\text{FCH}_2\text{CH}_2\text{F}$ ), and model build them. Both will adopt a trans configuration.
2. Select the fluorine and carbon atoms in each molecule. First click on the selection tool, and then drag the mouse from one fluorine atom to the other in each molecule.
3. Choose RMS Fit and Overlay from the Display menu.
4. When the dialog box appears, select the Selection Order option, and click on OK. The two molecules are overlaid, and the HyperChem status line shows an RMS Error of 0.84 Å.

## Reference

This section describes the RMS Fit Options dialog box and the HCRMSFIT.INI file.

## RMS Fit Options Dialog Box

If you invoke RMS Fit on two molecules (with the same number of atoms) with no selected atom, the overlay is automatically carried out by molecule numbering, and no dialog box appears. Atom 1 in molecule 1 is overlaid onto atom 1 in molecule 2, and so on.

If, however, you have selected some atoms in each molecule that you wish to overlay, then a dialog box appears. This dialog box offers a choice between overlay by molecule numbering or by selection order.

### **Overlay By Molecule Numbering**

If you choose to overlay by molecule numbering then atoms are ordered by number (as shown by choosing the Labels item in the HyperChem Display menu, and selecting Number) and the overlay is carried out in by matching the sequences. Thus, if atoms 1, 3, and 5 are selected in molecule 1, and atoms 1, 2, and 6 are selected in molecule 2, the overlay is carried out by minimizing the distance between the following pairs of atoms, regardless of the order in which the atoms were selected:

Atom 1 in molecule 1 with atom 1 in molecule 2

Atom 3 in molecule 1 with atom 2 in molecule 2.

Atom 5 in molecule 1 with atom 6 in molecule 2.

Overlay by molecule numbering is particularly useful when comparing different structures of the same molecule, or structures of closely related molecules. For instance, suppose you want to compare the structure of the ring in a cyclic peptide (molecule 1) with that in a similar peptide which differs in that one amino acid is mutated (molecule 2). You place both molecules on the screen, double click with the selection tool on a ring atom to select the ring in each molecule, and overlay by molecule numbering. The particular atom in the ring you clicked on does not matter in this case.

### **Overlay By Selection Order**

If you choose to overlay by selection order then pairs of atoms that are overlaid are determined by the order in which you select them. If you select atoms 1, 3, and 5 in molecule 1 (in that order) and atoms 1, 6, and 2 in molecule 2 (in that order) and overlay by selection order, then the overlay is carried out by minimizing the distance between the following pairs of atoms.

Atom 1 in molecule 1 with atom 1 in molecule 2

Atom 3 in molecule 1 with atom 6 in molecule 2

Atom 5 in molecule 1 with atom 2 in molecule 2



Note that the order-dependence is only within a particular molecule: it does not matter whether you select all atoms to be compared in molecule 1 first, and then those in molecule 2, or whether you select one atom in molecule 1, one in molecule 2, the second in molecule 1, and so forth. Overlay by selection order is particularly useful when comparing the structures of molecules that come from different sources, for which the atom numbering within the molecules may bear little or no resemblance to each other.

## Suppressing The Options Dialog Box

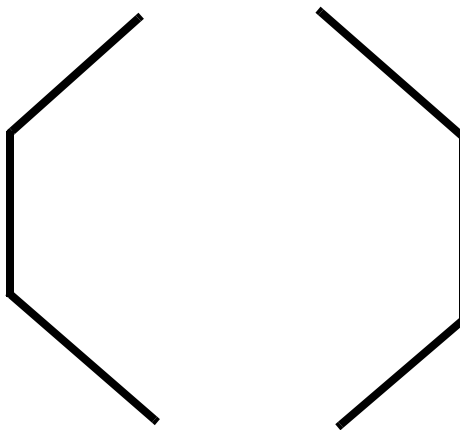
RMS Fit stores the most recent choice of option in a short initialization file, HCRMSFIT.INI in your Program directory. This file looks as follows:

```
ShowDialogBox=True  
OverlayBySelectionOrder=False
```

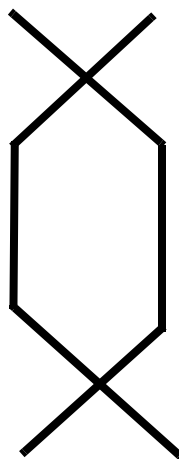
You can edit this file in a text editor to suppress the display of the options dialog box. Setting ShowDialogBox to False and OverlayBySelectionOrder to True (if you wish to overlay by selection order) or False (if you wish to overlay by molecule numbering) forces RMS Fit to not show the Options dialog box and to act as if the option you have just set is chosen, each time it is invoked. This can be useful if you routinely use one or other of the options, and do not wish to be prompted to choose each time. To go back to showing the dialog box, edit this file and set ShowDialogBox to True.

## Hints

There may be occasions when the RMS procedure does not find the best overlay between two molecules. The RMS procedure is an iterative one. It starts by superimposing the centers of geometry of the two structures, and then searches to improve the fit each iteration until no further significant improvement can be made. One special case occurs if you have two molecules with planes of symmetry, arranged as if reflecting in a mirror, as follows:



Here the overlay may end up with a structure that is not the best overlay, but instead looks as follows:



If you see this effect, rotate one or other of the molecules slightly out of the plane, so that they are no longer symmetrically arranged, and repeat the overlay.

## Show All

Show All cancels Show Selection Only and Hide Selection. You can then see all molecules in the molecular system.

*Note:* You might not see all atoms in a molecule or all molecules in the molecular system. This depends on the other display conditions (for example, Show hydrogens and Perspective View) and the viewpoint you use (for example, the zoom setting and the clipping slab settings).

## Show Selection Only

Show Selection Only displays only the selected atoms and molecules that are visible (see “Visible Atoms” on page 222). It hides all atoms that are not selected.

Use Show All to reverse Show Selection Only.

## Hide Selection

Hide Selection makes all selected atoms invisible. Use Show All to reverse Hide Selection.

## Rendering

Rendering displays the Rendering property sheet for choosing the type of molecular rendering model and options. Not all options are supported with all models and this is indicated by grayed items in the dialog box. The dialog box has several pages which can be accessed by clicking on the appropriate “tab control”.

### Rendering: Rendering Method

This page shows options for the rendering of atoms and bonds in the structure.



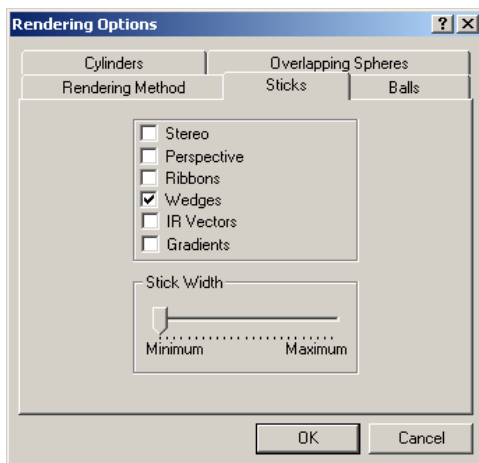
- Sticks** This shows a line for every bond. Each half bond has the color of the element at the end of the bond. This is the most efficient rendering and requires the least amount of computing time when you make changes in position or rotational angle.
- Balls** Atoms are drawn as spheres (if Shaded is on) or as flat disks (if Shaded is off). Atoms are drawn in strict order of appearance, back to front; they are not shown as overlapping. With shading, the quality of this rendering increases significantly when using a video display with 256 or more colors.
- Balls and Cylinders** Show bonds as solid objects. When Shaded is on, bonds are drawn as shaded cylinders; otherwise they are drawn without shading. Atoms are drawn as with the Balls option, above.
- Overlapping Spheres** Atoms are drawn as overlapping, with CPK radii, as unshaded disks (if Shaded is off) or as shaded spheres (if Shaded is on). With shading, the quality of this rendering increases significantly when using a video display with 256 or more colors.
- Dots** These are transparent, space-filling models with a surface constructed of dots. The surface shows the CPK radii of the atoms. Stereo is not available and the Wedges option is ignored.
- Sticks & Dots** These models are a combination of Sticks and Dots. They show bonding and give an impression of the

space-filling properties of a molecular system. Stereo is not available.

**Note:** You can apply labels (see “Labels” on page 281) only to Sticks and Sticks & Dots renderings.

## Rendering: Sticks

This page shows options for rendering structures as Sticks.



**Stereo** Turn this option on with Sticks to see duplicate images of the molecular system for stereo viewing (see “Using Stereo” on page 268). With Stereo on, you can use the Selection tool only with the image on the right to select atoms or residues. You cannot use the Drawing tool. With Stereo off you see the normal, single image of a molecular system.

**Perspective** Check this to have all molecules appear to be in perspective, with closer atoms appearing larger. Perspective must be on to see changes with the Z translation tool (see page 43). This option also shows the periodic box in perspective (see “Show Periodic Box” on page 275 and “Periodic Box” on page 335).

With Perspective View off (no check mark), all atoms appear projected onto the plane of the

screen; you have fewer visual clues to the depth of atoms within a molecule.

**Ribbons**

Use Ribbons to highlight the secondary and tertiary structure of a polypeptide chain or protein. Ribbons can be shown for polypeptides with residues that have originated from the amino acids database, or have been defined in a template file.

The ribbons follow the backbone of the polypeptide as a five-point spline curve, consisting of nine lines. The ribbons are displayed for the whole molecule, regardless of the selection or the segment of the molecule currently displayed. If you choose Hide Selections, this does not affect the visibility of the ribbons.

**Wedges**

This option toggles the display of bonds with stereochemistry constraints as either solid (wide end closer to you) or dashed (pointed end closer to you) wedges for Sticks and Stick & Dots. The wedges are used to depict stereochemistry constraints for the Model Builder, with the pointed end being connected to the constrained atom. Wedges can also be used for purely visual purposes as well. Wedges will appear with an exclamation point if the depicted stereochemistry constraint is not yet satisfied — use the Model Builder for it to take effect. See “Using Wedges to Display and Specify Stereobonds” on page 482.

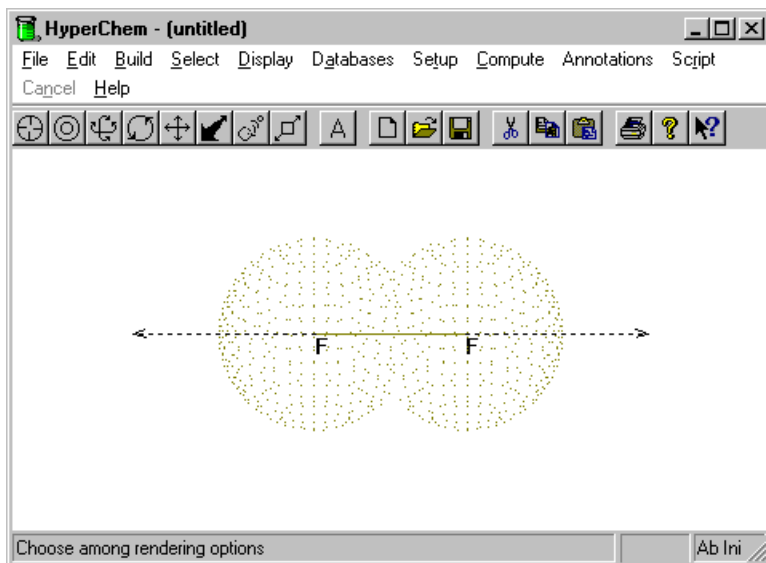
**IR Vectors**

Turn on this option with Sticks, Dots or Stick & Dots after performing a vibration calculation to display normal mode displacement vectors. See “Vibrational Spectrum” on page 406.

**Gradients**

Calculations of single points, optimized structures, vibrational spectra, molecular dynamics, etc. all involve calculating the energy and generally the first derivatives of the energy with respect to coordinate displacements [ab initio calculations of single points can request not to calculate these derivatives]. These derivatives (gradients or forces on an atom) can now be displayed as vectors with a direction parallel to the force and a length proportional to the magnitude of the force. The gradients can be

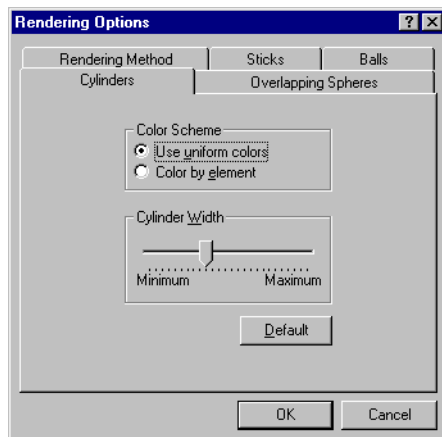
displayed only when rendering the molecule with a sticks rendering. For a diatomic molecule, where the bond length is less than the optimized bond length, the gradients vectors look as follows:



**Stick Width** The slider and number box indicate the width used to draw sticks, in units of pixels.

## Rendering: Cylinders

This page shows options for rendering bonds as cylinders.



**Color Scheme** In this section, you can specify whether bonds should all be drawn with the same color, or colored according to the elements they join.

**Use uniform colors** If this option is selected, all bonds will be drawn as shades of gray.

**Color by element** If this option is selected, each bond is colored according to the atoms that it joins. If the atoms are drawn in different colors, then the bond is drawn half in one color, and half in the other.

**Cylinder width** The slider and number box indicate the scaling factor which shows the cylinders' radius as a fraction of the maximum allowed value. This maximum value is the radius at which a boron atom is drawn; boron has the smallest drawing radius of all the atoms.

**Default** Restores the scale factors to values that will display atoms with their default radii, or cylinders to a standard value.

## Rendering: Balls

This page shows options for rendering atoms as balls.

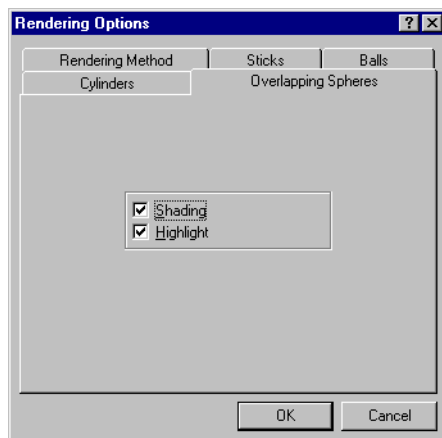




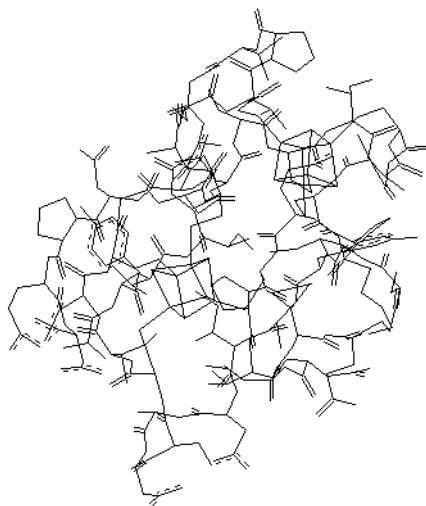
- Shading** This options toggles the display of atoms rendered as Balls, and bonds rendered as Cylinders, between a flat appearance and a shaded three-dimensional appearance. The flat display is much faster to draw, but the three-dimensional display looks much better.
- Highlight** This option adds bright highlights to a structure, enhancing its three-dimensional appearance. This option is available only when Shaded is turned on.
- Ball Radius** The slider and number box give the scaling factor which shows the spheres' radii as a fraction of the maximum allowed values.
- Default** Restores the scale factor to 0.5, which will display atoms with their default radii. For each atom, the default drawing radius is  $0.75 \times$  the van der Waals radius except when that value was not available, in which case the covalent radius is used. Setting the scale factor to 0.667 will therefore draw atoms with van der Waals radii.

## Rendering: Overlapping Spheres

This page shows options for rendering atoms as Overlapping Spheres.



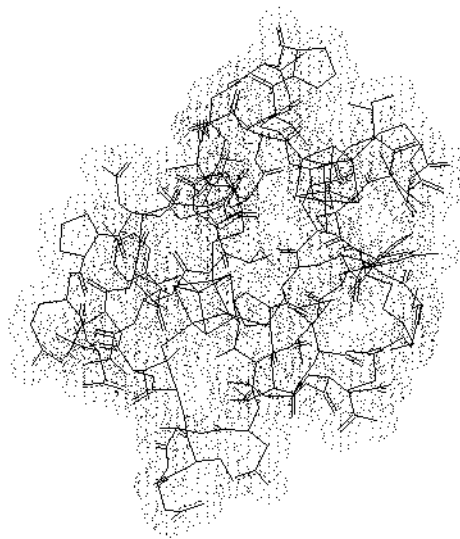
- Shading** This options toggles the display of atoms rendered as Overlapping Spheres between a flat appearance and a shaded three-dimensional appearance. The flat display is much faster to draw, but the three-dimensional display looks much better.
- Highlight** This option adds bright highlights to a structure, enhancing its three-dimensional appearance. This option is available only when Shaded is turned on.



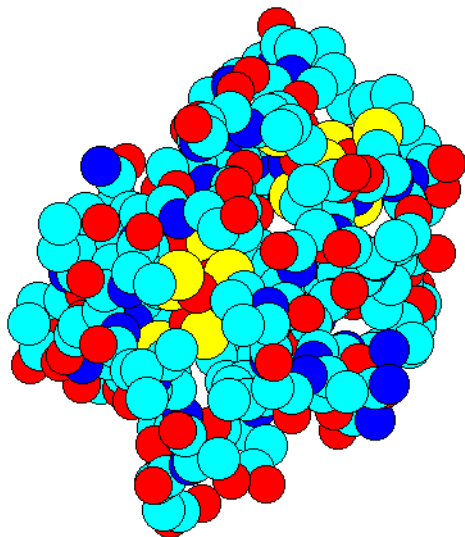
*Sticks*



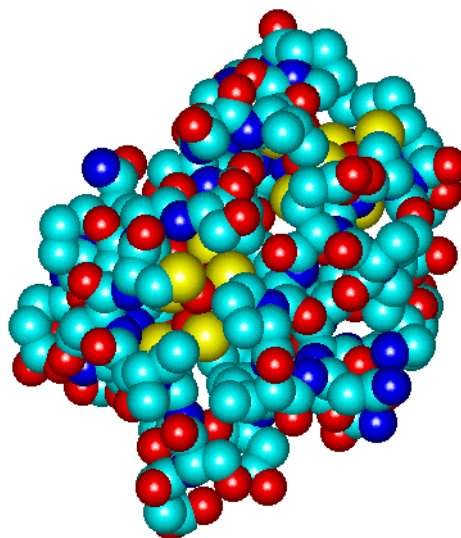
*Dots*



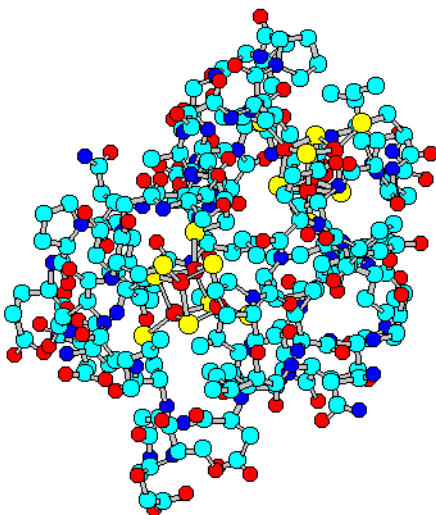
*Sticks and Dots*



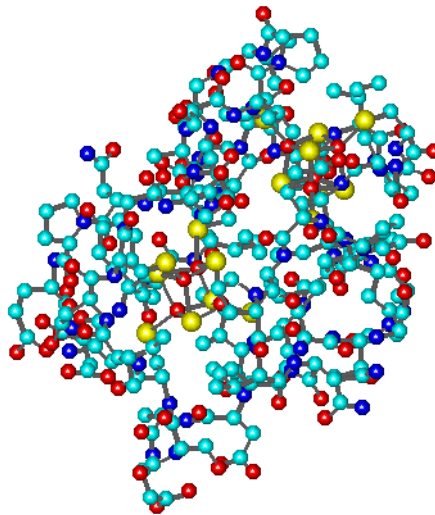
*Balls, Shading off*



*Overlapping Spheres, Shading and Highlights on*



*Balls and Cylinders, Shading off*



*Balls and Cylinders, Shading and Highlights on*

## Using Stereo

With the Stereo option on, you see duplicate images of the molecular system for stereo viewing. You can turn Show Stereo on only when you are using the Sticks rendering model (see page 258). If you see only part of a molecular system in the workspace, Show Stereo presents a stereo pair of the atoms in the workspace only.

*Note:* Turning Perspective View on might increase the stereo effect.

When the stereo view is on, you can use the Selection tool only with the image on the right to select atoms or residues. You cannot use the Drawing tool.

HyperChem uses two different methods for presenting stereo images: walleyed or cross-eyed. You can set HyperChem to use one or the other by changing this setting in the Registry or the chem.ini file (see “Ab Initio Settings” on page 747):

**WallEyedStereo=Yes (or No)**



*backbone of tRNA, with walleyed stereo and Perspective View on*

## Walleyed Stereo

For walleyed stereo viewing, the images for the left and right eyes are on the left and right sides of the workspace respectively. The advantage of walleyed viewing is that you see one, fused 3D image. It is not suitable for large molecules, because the images might overlap in the workspace.

*Note:* You might have to adjust the width of the HyperChem window to get the best stereo effect. This changes the separation of the

two images. The separation between images should be less than the distance between your eyes (about 8 centimeters).

To see a stereo effect, you must merge these images into one. To do this you can use a stereo viewer, or place a sheet of paper against the screen to block your view of each image from the opposite eye. Then change your focus as if you were looking at a distant object. You can also aid image fusion by squinting or removing your glasses (if you are short-sighted) to blur your vision.

## Cross-eyed Stereo

For cross-eyed stereo viewing, the images for the left and right eyes are reversed in the workspace: the image for the left eye is on the right side of the workspace. You must cross your eyes so that you see the left image with your right eye. The advantage of cross-eyed viewing is that you can see larger molecules in 3D.

This method requires more practice. You must learn to cross your eyes and to concentrate on the central, fused image in the presence of left and right images.

## Last Rendering

Last Rendering returns the rendering mode to the mode that was used before the most recent change. Two modes can therefore be alternated by using this option repeatedly. This option can be activated by pressing [F2]. This menu item is not available while an isosurface is being displayed; displaying an isosurface automatically switches the rendering mode to Sticks.

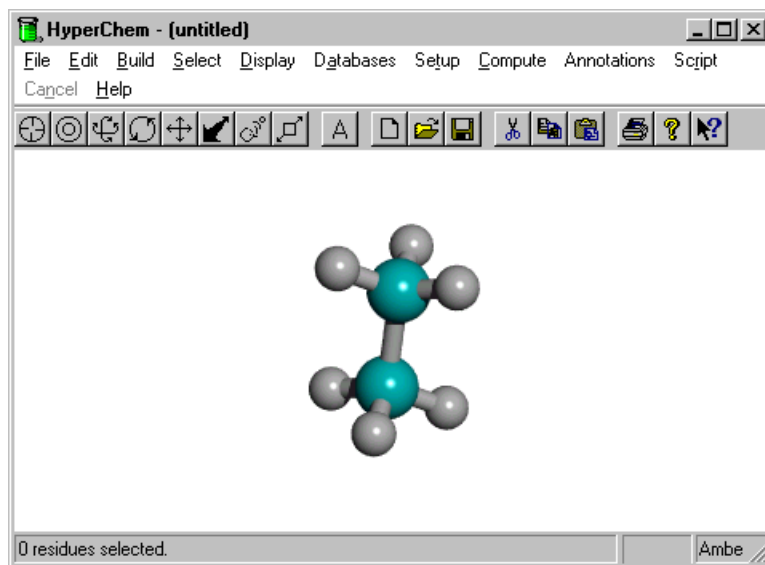
## Raytracing

### Overview

HyperChem Raytrace, together with POV-Ray (a third-party free raytracing program), provide facilities for producing high quality ray-traced molecular graphics from HyperChem. In order to use HyperChem Raytrace, you must have POV-Ray installed. Your normal HyperChem installation will install POV-Ray for you.

## Obtaining a Raytraced Image

Using the raytracing module is very simple. At any time you would like to see a raytraced image, you can do so by clicking on the menu item <Display/Raytrace...>. The operation may take some time but as soon as the operation is complete your workspace will show a raytraced image such as that shown below:



## Default Lighting Scheme

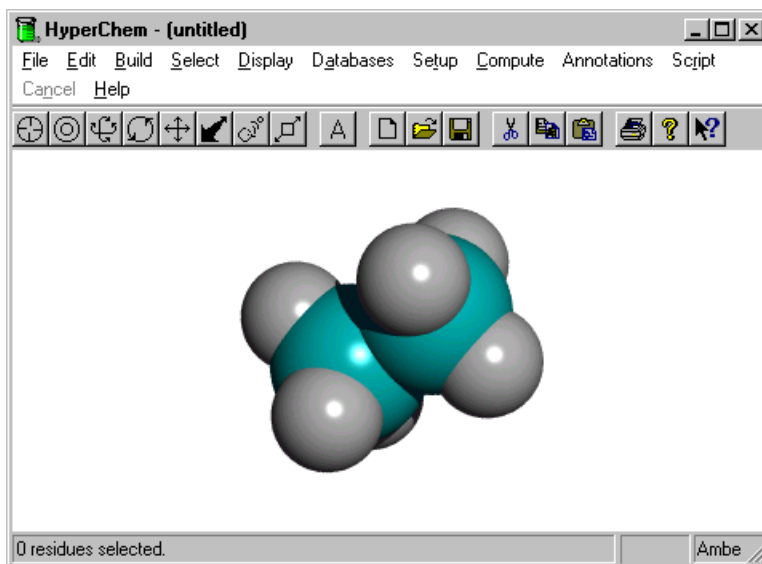
You will notice that the above is a raytraced image because of the shadows. The image above shows the default lighting scheme. This default lighting is easy to use and is highly recommended unless you need custom lighting (see below).

## Types of Images

Two types of raytraced images are available:

- Ball and cylinders
- CPK overlapping spheres

Which raytracing you get depends upon the current rendering mode. A sticks rendering mode (in addition to the ball and cylinder mode) gives a ball and cylinders raytraced image. An overlapping spheres mode gives a rendering like the following:



## Custom Lighting

If you wish not to use the default lighting scheme but add your own lights you may do so. The procedure is somewhat unusual but makes use of the existing tools in HyperChem for moving, positioning, and visualizing atoms.

The custom lighting model makes use of point sources of light that are defined as atoms but with a very specific named selection attached to them. Atoms associated with a named selection,

**light\***

i.e. a named selection, such as *light1* or *lightfromtheside*, etc., which begins with the lowercase letters l-i-g-h-t followed by any other letters, numbers, etc. By default such a light will be a white light.

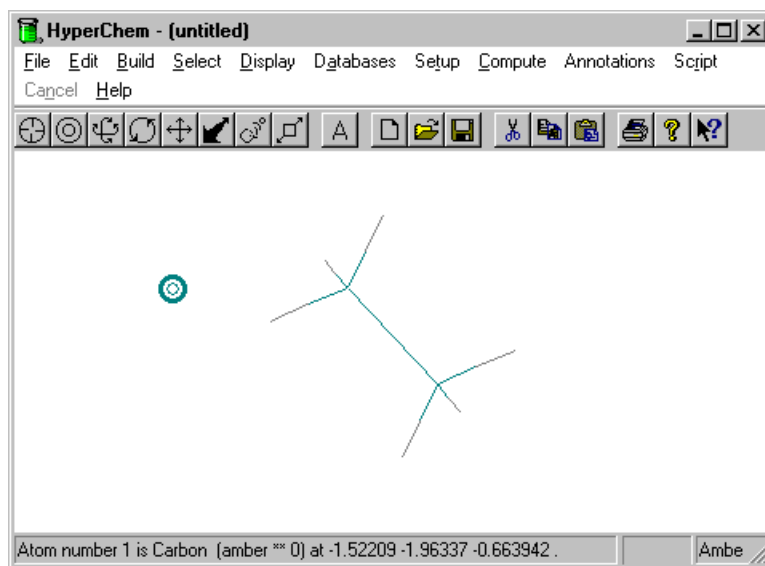


Colored lights may also be added. These are added by associating an atom with a named selection of three possible kinds:

light-red\*  
light-green\*  
light-blue\*

Atoms associated with such named selections will be colored lights, either red lights, green lights, or blue lights. All atoms that have been associated with a light this way are ignored as real atoms and function only as "lights".

For example, if we add a carbon atom at the position shown below and select it, we can turn the carbon atom into a light.



To turn it into a light (a red light for example) we just have to name it appropriately as shown below:



This lighting will leave most of the molecule in the shadows and it is likely you will want to add more lights than just this one. You can add as many lights of different colors (red, green, blue, white) as you like.

## Using POV-Ray

The first time you use POV-Ray it will ask you to agree to its license requirements. Thereafter it should appear quietly on the task bar when a raytraced image is being created for HyperChem. It will disappear after that image is passed to HyperChem. Basically what is happening is that HyperChem prepares a standard \*.POV file for input to the POV-Ray program. Advanced users may want to learn more about POV-Ray and customize its use.

Raytracing is a demanding computational task and depending on your computer it may take some time. While raytracing is going on, the task bar will show the percentage completion of the operation. After the operation is complete the raytraced image will be transferred to your HyperChem work space. Any attempt to move or otherwise modify the molecule in the workspace will cause the raytraced image to disappear. To get it to reappear you will need to raytrace an image again.

Although your HyperChem installation will install POV-Ray for you, advanced users may at some time wish to install their own version of POV-Ray. If you wish to explore other versions of POV-Ray, they may be obtained by anonymous ftp from <ftp.povray.org>, or by Web browser from <http://www.povray.org>. They can also be obtained from any of a constantly changing number of mirror sites.

## Show Isosurface

If an isosurface has been calculated after a semi-empirical single-point calculation, Show Isurface controls whether or not that surface will be displayed in the workspace. This option can be turned on and off by pressing [F3]. Displaying an isosurface automatically switches the rendering mode to Sticks.

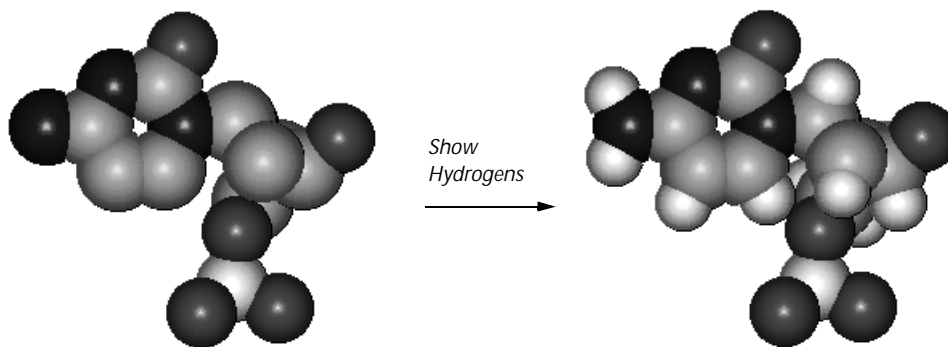
*Note:* While the Stereo or Perspective options of the Sticks rendering are turned on, you cannot display a contour plot or isosurface (see page 391). If the Wedges option is on, the wedges will not be shown while an isosurface is being displayed.

## Isosurface . . .

If an isosurface has been calculated after a semi-empirical single-point calculation, Isosurface opens the Plot Molecular Properties property sheet (see page 392) or the Orbital Options property sheet (see page 401), which can also be opened from the Compute menu or Orbitals dialog box, respectively. This feature can also be activated by pressing [F4]. The isosurface can be displayed only if the structure is rendered in Sticks mode. The property sheets displayed by this option allow you to change the way that the stored values are drawn but not to recalculate the isosurface values themselves. These reduced property sheets allow you to change the Isosurface Rendering options and the Transparency Level options; the reduced Plot Molecular Properties sheet also allows you to change the Mapped Function options.

## Show Hydrogens

Turn on Show Hydrogens (✓) to display all hydrogen atoms in the molecular system. The existence of hydrogens depends on your previous choices on the Build menu (Explicit Hydrogens, Add Hydrogens, and Add H & Model Build), plus the composition of the molecules.



Turn off Show Hydrogens (no check mark) to hide all hydrogens and simplify the rendering of molecules in the workspace. Only heavy atoms remain visible.

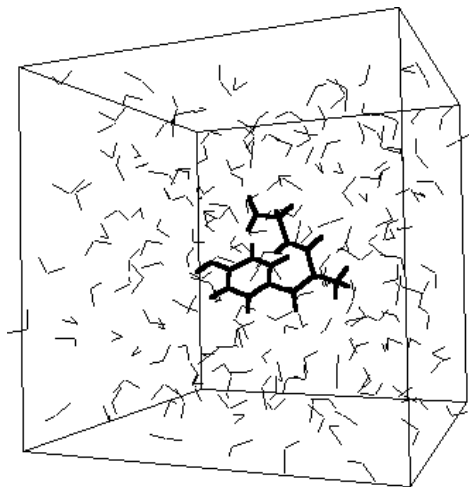
Hydrogens with a coordination number greater than 1 remain displayed when Show Hydrogens is unchecked.

## Show Periodic Box

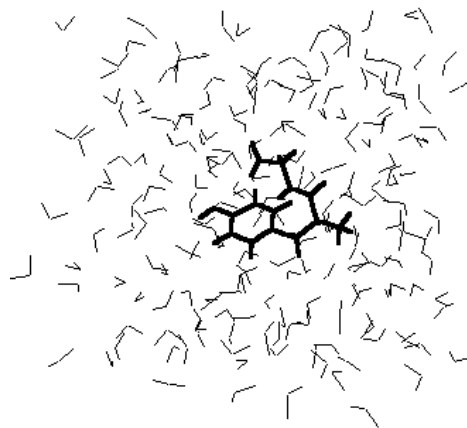
You create a periodic box, including water molecules, by choosing Periodic Box on the Setup menu (see page 335). Show Periodic Box turns off (no check mark) or on (✓) the display of the periodic box.

**Note:** Turn on Perspective View on the Display menu to see the depth of the periodic box.

Turning off Show Periodic Box hides the outline of the box but leaves water molecules and periodic boundary conditions in the molecular system. To permanently remove the periodic box, you must edit the HIN file (see “box” on page 716), removing the box record.



Show Periodic Box on

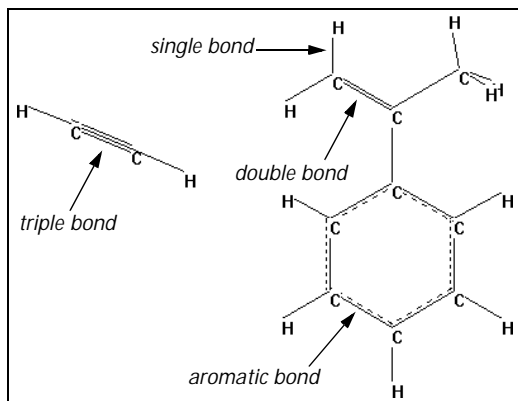


Show Periodic Box off

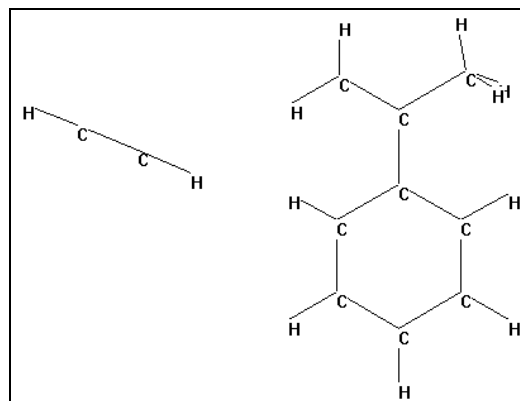
**Note:** Show Periodic Box is gray until you create a periodic box.

## Show Multiple Bonds

With Show Multiple Bonds on (✓), HyperChem uses different types of lines for single, double, triple, and aromatic bonds in Sticks or Sticks & Dots renderings.



Show Multiple Bonds on



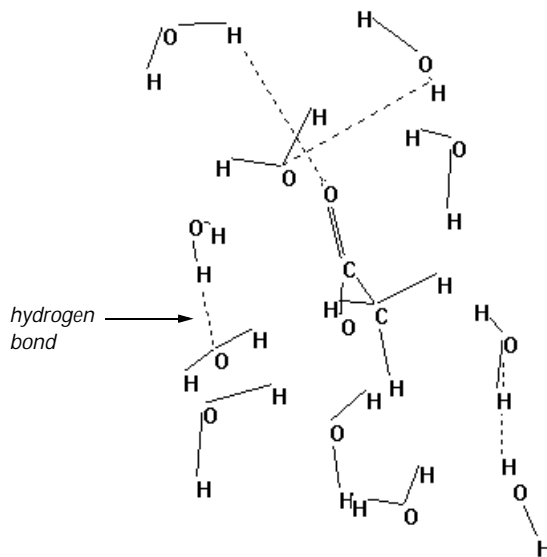
Show Multiple Bonds off

With Show Multiple Bonds off (no check mark), all bonds appear as single lines.

If Show Multiple Bonds is off and you change the bond order with the Drawing tool (see “Changing the Bond Order” on page 475), the order of the bond (single, double, triple, or aromatic) appears on the status line.

## Show Hydrogen Bonds

Show Hydrogen Bonds displays possible hydrogen bonds between selected atoms or, with no selection, for all molecules in the molecular system.



With Show Hydrogen Bonds on (✓), you see hydrogen bonds as white, dotted lines.

To show hydrogen bonds:

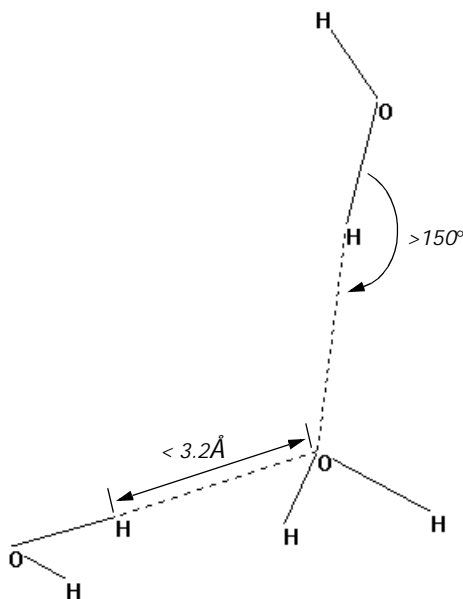
1. Select specific molecules or deselect all atoms.
2. Make sure Show Hydrogen Bonds is on.
3. Choose Recompute Hydrogen Bonds on the Display menu (see the following section).

**Caution:** If you change the arrangement of molecules in the workspace (for example, by XY translation) or if HyperChem changes the conformation during a calculation, choose Recompute Hydrogen Bonds to see up-to-date hydrogen bonds.

With Show Hydrogen Bonds off (no check mark), hydrogen bonds are hidden. You cannot choose Recompute Hydrogen Bonds.

## Recompute Hydrogen Bonds

Recompute Hydrogen Bonds determines hydrogen bonds between selected molecules or, if there is no selection, between all atoms in the molecular system. HyperChem creates a hydrogen bond if the distance between the donor hydrogen and acceptor atom is less than 3.2 Ångstroms and the angle made by covalent bonds to the donor and acceptor atoms is greater than 150 degrees. You can change these criteria in the Registry or the chem.ini file (see “Ab Initio Settings” on page 747).

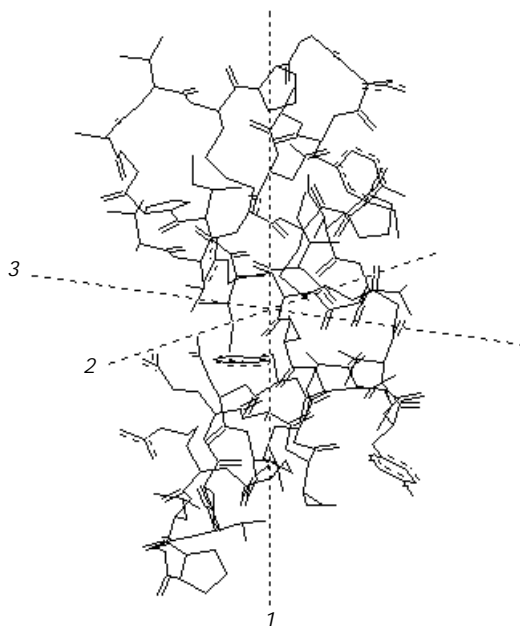


**Caution:** HyperChem stretches apparent hydrogen bonds as you move molecules or during a calculation (geometry optimization or molecular dynamics). You must use Recompute Hydrogen Bonds to determine correct hydrogen bonding.

To use this item, Show Hydrogen Bonds must be on.

## Show Inertial Axes

When you turn on Show Inertial Axes (✓), HyperChem computes and displays the inertial axes of selected atoms or, if there is no selection, the molecular system. The axes appear as three orthogonal (perpendicular) dotted lines intersecting at the center of mass of the atoms. The lines are numbered 1, 2, and 3 for primary, secondary, and tertiary axes. The status line shows the moments of inertia in units of a.m.u.-Ångstroms<sup>2</sup>.



**Note:** Inertial axes appear only with Sticks and Sticks & Dots renderings of a molecular system.

Inertial axes are fixed in the Molecular Coordinate System and depend only on the number, type (mass), and position of atoms. The inertial axes have their origin at the center of mass and are orthogonal. The primary inertial axis is often the longest distance from the center of mass to the edge of the molecular system. See “Coordinate Systems” on page 18 for more information about coordinate systems and axes.

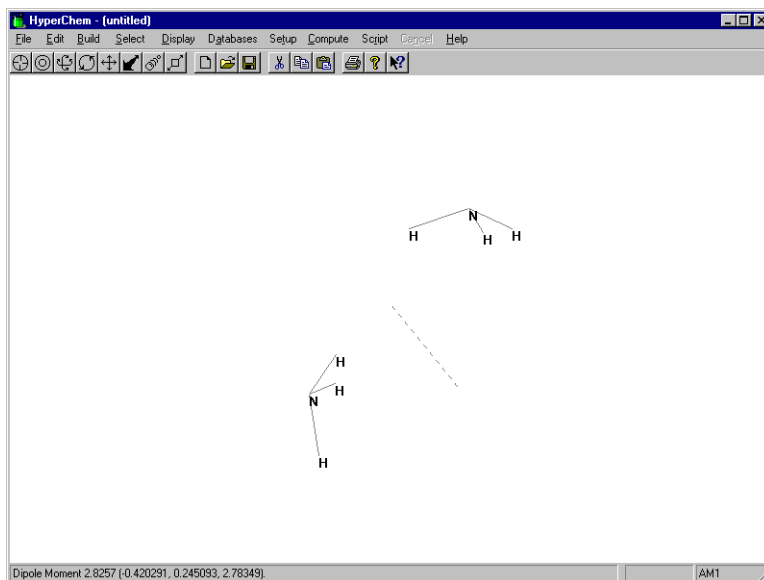
You can align a molecular system relative to its inertial axes (see “Align Molecules” on page 216).



Once Show Inertial Axes is on, it applies to any selection or molecular system in the workspace. For example, if you change the selected atoms or use Merge (on the File menu) to add other molecules, HyperChem automatically computes and displays new inertial axes.

## Show Dipole Moment

When you turn on Show Dipole Moment (✓), HyperChem displays the dipole moment that has been calculated for the selected atoms or for the molecular system. The menu item is gray and unavailable if a single-point calculation has not been performed to calculate the dipole. The dipole moment appears as a dotted line intersecting the center of mass of the atoms. The status line shows the total dipole moment, followed in brackets by its x, y, and z components, in units of Debyes.



Note that the sign convention used in the quantum mechanical calculation of dipoles is opposite to that used in molecular mechanics dipole calculations; this reflects the differing sign conventions of physics and chemistry.

## Labels

Labels lists names identifying individual atoms or residues in a molecular system. For atoms, you can use only one type of label at a time. For residues, you can show both atom and residue labels at the same time.

Labels applies to selected atoms or, if there is no selection, to the whole molecular system. By selecting atoms, applying different labels, and removing the selection, you can mix the types of labels that appear in the workspace.

**Note:** You can apply labels only to Sticks and Sticks & Dots renderings (see “Rendering” on page 258).

When you open the Labels dialog box, the default choice is always set to None. You can either accept this choice to remove all labels or choose another label. Each time you use the Labels dialog box, your choices replace the previous choices for selected atoms, or, if there is no selection, for all atoms.

### Labels Dialog Box



#### Atoms

None	Removes all labels. This is the default.
Symbol	Atomic symbol, for example, C, O, Na, or P.
Name	The names of atoms from the Brookhaven Protein database; for example, CA and CB for the alpha and beta carbons.

<b>Number</b>	Atom number.
<b>Type</b>	Atom type, according to the force field in effect; for example, CT for a tetrahedral carbon in the AMBER force field.
<b>Charge</b>	Net atomic charge, for example, -0.23.
<b>Spin Population</b>	Total net population (alpha-beta) of spins of an atom.
<b>Mass</b>	Atomic mass.
<b>Basis set</b>	Basis set applied to each atom (may be None).
<b>Chirality</b>	R or S.
<b>RMS Gradient</b>	Scalar value of the gradient vector describing the force on an atom.
<b>Custom</b>	Can be set to any string.
<b>Residues</b>	
<b>None</b>	Removes all labels. This is the default.
<b>Name</b>	Residue name, for example, TRP, or ILE.
<b>Sequence</b>	Sequence number from the N-terminus: 1, 2, 3 ... 25 ... .
<b>Name + Seq</b>	Both Name and Sequence.
<b>Bonds</b>	
<b>None</b>	Removes all labels. This is the default.
<b>Bond Length</b>	Length between bonded atoms.
<b>Bond Order</b>	Bond order computed in quantum mechanical calculations.

## Color

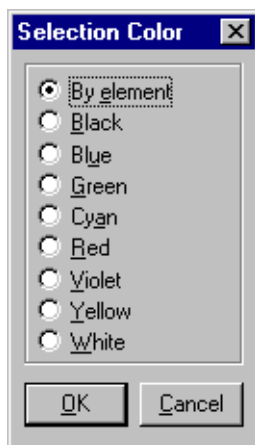
Color shows the Selection Color dialog box for marking selected atoms with a distinct, persistent color. Atoms keep this color after you deselect them. If there is no selection, the whole molecular system takes on this persistent color. Use this feature to keep track of a set of atoms in a larger molecular system.

When you make a selection, atoms first take the color for a selection set in the Preferences dialog box (see page 95). The default is Green. The color you choose from the Selection Color dialog box replaces a color chosen in the Preferences dialog box.

*Note:* You see the persistent selection color only after you deselect the atoms. However, if you are using Thick line to highlight selections (see “Preferences Property Sheet” on page 95), the persistent color appears immediately.

To remove the persistent color from a molecular system, open this dialog box again and choose the default, By element.

## Selection Color Dialog Box



To apply a color to a selection:

1. Select a set of atoms.
2. Choose Color on the Display menu.
3. Choose a color from the dialog box.
4. L-click OK.

To remove a persistent color:

1. Select a set of atoms or, for the whole molecular system, deselect all atoms.
2. Choose Color on the Display menu.
3. L-click OK in the dialog box to choose By element.

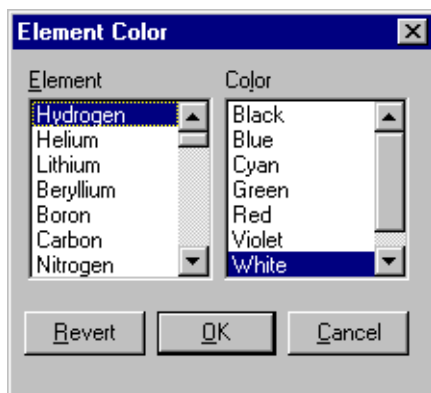
## Element Color

Element Color shows you the Element Color dialog box for changing the color assigned to each element. This establishes the By element choice you see in other dialog boxes:

Element	Default Color
Lone pair	broken white line
Carbon	cyan
Nitrogen	blue
Oxygen	red
Fluorine	yellow
Sodium	violet
Sulfur	yellow
Potassium	violet
Iron	red
Cobalt	blue
Copper	green
Bromine	yellow
Iodine	red
Gold	yellow
All other elements	white

You have a choice of six standard colors, plus black and white. HyperChem uses a default set of colors (for example, cyan for carbon and red for oxygen). If you change an element color, you can always return to the default set of colors by choosing Revert in this dialog box.

## Element Color Dialog Box



To set a new element color:

1. L-click on the element name in the Element list.
2. L-click on the new color assignment in the Color list.
3. L-click OK.

To return to the default colors:

1. L-click on Revert.

*Note:* This dialog box establishes the By element color scheme referred to in other dialog boxes (see “Preferences Property Sheet” on page 95 and “Selection Color Dialog Box” on page 283).



# Chapter 8

## Chemical Calculations

This chapter describes functions in the Setup and Compute menus for carrying out chemical calculations. The calculations always involve molecules in the workspace.

### Setup Menu

The Setup menu provides options for running energy calculations. After you choose Setup menu options, use the Compute menu to begin a calculation.

HyperChem performs molecular mechanics calculations using any of four methods (MM+, AMBER, BIO+, and OPLS), semi-empirical quantum mechanics using any of eleven methods (Extended Hückel, CNDO, INDO, MINDO3, MNDO, MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO), *ab initio* quantum mechanics calculations, and density functional calculations.

The Setup menu has these items:

---

Molecular  
Mechanics . . .<sup>a</sup>

Choose Molecular Mechanics to use a Newtonian method instead of a quantum mechanical (semi-empirical or *ab initio* or Density Functional) method for a calculation. Molecular mechanics remains on (✓) until you choose Semi-empirical or Ab Initio or Density Functional

---

Semi-  
empirical . . .<sup>a</sup>

Choose Semi-empirical to use one of these quantum mechanics methods instead of a molecular mechanics or *ab initio* or Density Functional-method for a calculation. Semi-empirical remains on (✓) until you choose Molecular Mechanics or Ab Initio or Density Functional

---



---

<b>✓ Ab Initio. . .<sup>a</sup></b>	Choose Ab Initio to use this quantum mechanics method instead of a molecular mechanics or semi-empirical or Density Functional method for a calculation. Ab Initio remains on (✓) until you choose Molecular Mechanics or Semi-empirical or Density Functional
<b>✓ Density Functional. . .<sup>a</sup></b>	Choose Density Functional to use this quantum mechanics method instead of a molecular mechanics or semi-empirical or Ab Initio method for a calculation. Density Functional remains on (✓) until you choose Molecular Mechanics or Semi-empirical or Ab Initio
<b>Periodic Box . . .</b>	Choose this item to enclose the molecular system in a periodic box containing water molecules
<b>Restraints . . .</b>	Choose this item to add restraining forces to 1, 2, 3 or 4 atom named selections for molecular mechanics and quantum mechanics. This item is gray if there are no named selections with 1 to 4 atoms
<b>Set Velocity . . .</b>	Choose this item to specify velocities for atoms. The velocities can be used in molecular dynamics calculations by using the Restart option. It is gray if there are no atoms
<b>Set Finite Field</b>	Choose this item to specify a finite electric or magnetic field to apply to the workspace. An electric field can be applied in conjunction with any quantum mechanical method. A magnetic field can only be applied in conjunction with the TND0 semi-empirical method.
<b>Network . . .</b>	Choose this item to specify a remote server to perform molecular mechanics, semi-empirical, <i>ab initio</i> , or Density Functional calculations

---

<b>Edit Parameters . . .</b>	Choose this to display and/or edit the current parameters (for the current computational method) for the current selection. This item applies only to molecular mechanics and semi-empirical methods. If a single atom is selected information is given on parameters for this atom. If two bonded atoms are selected, parameters are given for this pair of atoms and so on.
<b>Select Parameter Set . . .</b>	Choose this item for an alternative set of compiled parameters (see the following section) for a molecular mechanics calculation or TNDO calculation
<b>Compile Parameter File</b>	Choose this item to convert a new parameter set (see Select Parameter Set) from text or database files to a binary file. HyperChem uses a binary file for molecular mechanics calculations
<b>Reaction Map . . .</b>	Choose this item to map reactants to products for a Synchronous Transit transition state search. This item is gray and unavailable unless both REACTANT and PRODUCT named selections have been defined, with the same numbers of atoms in each

a. Only one of these items can be active at a time. A check mark appears next to the active item.

## Molecular Mechanics

Choose Molecular Mechanics to use a classical Newtonian calculation method instead of a quantum mechanical (semi-empirical or *ab initio* or density functional) method. This choice remains on (✓) until you choose one of the alternatives, Semi-empirical or Ab Initio or Density Functional. You can use any of these methods for Single Point, Geometry Optimization and Molecular Dynamics calculations on the Compute menu.

Molecular mechanics calculations treat atoms as Newtonian particles interacting through a potential energy function. The potential energies depend on bond lengths, bond angles, torsion angles, and nonbonded interactions (including van der Waals forces,

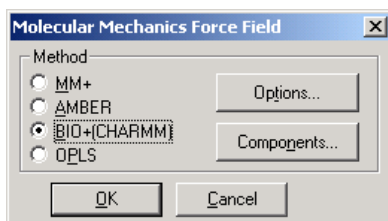
electrostatic interactions, and hydrogen bonds). In these calculations, the forces on atoms are functions of atomic position.

**Note:** If only part of the system in the workspace is selected, then only interactions which involve at least one selected atom will be included in the calculations. Interactions involving only unselected atoms will be omitted. For structural-optimization calculations, only selected atoms are allowed to move, while unselected atoms remain fixed in space.

The Molecular Mechanics Force Field dialog box is displayed to choose a force field (potential function) for a calculation.

From the Molecular Mechanics Force Field dialog box, you can go to either the MM+ Options dialog box (see page 291) or the Force Field Options dialog box (see page 293). Use these dialog boxes to change the conditions for a calculation.

## Molecular Mechanics Force Field Dialog Box



- |              |  |
|--------------|--|
| <b>MM+</b>   | Developed for organic molecules. This is an “all atom” force field. HyperChem supplements MM2 by providing additional parameters (force constants) using two alternative schemes (see <i>HyperChem Computational Chemistry, Theory and Methods</i> ). This extends the range of chemical compounds that MM+ can accommodate. MM+ also provides cutoffs for calculating nonbonded interactions (see “MM+ Options Dialog Box” on page 291), periodic boundary conditions, and a modified bond-stretch energy term that avoids repulsions at long bond lengths (see <i>HyperChem Computational Chemistry, Theory and Methods</i> ). |
| <b>AMBER</b> | Developed for proteins and nucleic acids. You can choose an all-atom or a united-atom simulation. The united-atom option treats certain atom groups  |

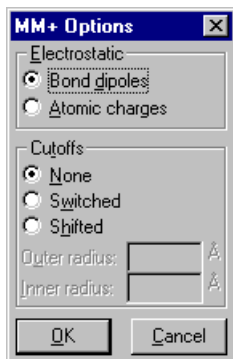
as one atom, with one atom type. You can also choose from the variety of parameters sets available for Amber. Choosing Amber94 and beyond gives results for a somewhat newer force field than the original.

- BIO+ (CHARMM)** Developed for biological macromolecules. You can choose an all-atom or a united-atom simulation. The united-atom option treats certain atom groups as one atom, with one atom type. You can choose from the variety of parameter sets available for BIO+(CHARMM). Choosing Charmm22 and beyond gives results for a somewhat newer force field than the original. BIO+ (CHARMM) is Hypercube, Inc.'s version of Chemistry at HARvard for Molecular Mechanics.
- OPLS** Developed for proteins and nucleic acids. It is similar to AMBER but treats more accurately non-bonded interactions. This force field provides only a united-atom simulation (see AMBER).
- Options . . .** L-click to go to the MM+ Options dialog box (see page 291) when MM+ is chosen or to the Force Field Options dialog box (see page 293) for AMBER, BIO+ or OPLS.
- Components . . .** L-click to go to the Components of Force Field Included dialog box (see page 295). This will allow you to choose which components of the force field are to be included in calculations.

Each type of molecular mechanics method (force field) has an extensive set of rules for determining atom types. See *HyperChem Computational Chemistry*, and appendix B of this book on page 653, for discussions of force fields, how to use them, and literature references.

## MM+ Options Dialog Box

Use the MM+ Options dialog box to set the conditions for the MM+ molecular mechanics force field. HyperChem stores these option settings, except for Cutoffs, in the Registry or in the chem.ini file and uses them for future HyperChem sessions.



**Electrostatics** Nonbonded electrostatic interactions are calculated using bond dipole interactions or partial atomic charges.

**Bond dipoles** This uses bond dipoles to calculate nonbonded electrostatic interactions. The bond dipole values come from the MM+ stretch parameter file (see page 666).

**Atomic charges** This uses partial atomic charges to calculate nonbonded electrostatic interactions. You can assign partial atomic charges via the Build/Set Charge menu item (see page 132) or you can do a semi-empirical or *ab initio* calculation first that calculates partial charges for each atom using a Mulliken population analysis.

**Cutoffs** These options determine the distance limits for calculating nonbonded interactions. Choosing Periodic Box on the Setup menu automatically sets Switched with default values. Cutoffs are necessary to prevent calculating interactions with neighboring periodic images.<sup>1</sup>

**None** Calculates all nonbonded interactions. This is the default for systems *in vacuo*. Do *not* use this option for periodic boundary conditions because it results in an implicit

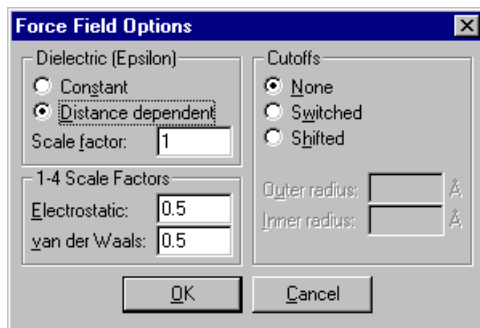
1. A periodic box implies a sharp, rectangular cutoff. An atom interacts only with one nearest image of all other atoms and never with an image of itself. This sharp cutoff can cause discontinuities in the potential surface that interfere with geometry optimization and molecular dynamics calculations. To avoid this, HyperChem uses Switched or Shifted cutoffs for a molecular system in a periodic box.

	undamped cutoff and causes discontinuities in the potential surface.
<b>Switched</b>	A smoothing function, applied from <i>the inner radius to the outer radius</i> , gradually reduces nonbonded interactions to zero. If the molecular system is in a periodic box, HyperChem chooses Switched and sets the Outer and Inner Radius. You can also use Shifted.
<b>Shifted</b>	A smoothing function, applied over the whole nonbonded distance, from <i>zero to outer radius</i> , gradually reduces nonbonded interactions to zero.
<b>Outer radius</b>	For Switched and Shifted cutoffs, this is the minimum distance at which nonbonded interactions are set to zero. This should normally be set to at least 4 Ångstroms larger than the Inner radius. With periodic boundary conditions, this value is half the smallest dimension of the periodic box.
<b>Inner radius</b>	For Switched cutoffs only, this is the maximum interatomic distance for full nonbonded interactions. With periodic boundary conditions, this value is 4 Ångstroms less than half the smallest dimension of the periodic box or a minimum of zero.

*Note:* The Cutoffs settings return to their default values if you place a new molecular system in the workspace.

## Force Field Options Dialog Box

Use the Force Field Options dialog box to set the conditions for the molecular mechanics force fields AMBER, BIO+, and OPLS. HyperChem stores these option settings, except for Cutoffs, in the Registry or in the chem.ini file and uses them for future HyperChem sessions.



**Dielectric permittivity (epsilon)** Constant or Distance dependent determine the method of calculating epsilon, a factor that modifies charge-charge interactions (and the electrostatic potential). Epsilon is the dielectric constant.

**Constant** This makes epsilon a constant. This is the form appropriate for systems in a gas phase or in an explicit solvent. Use this option for systems in a periodic box.

**Distance dependent** This makes epsilon proportional to the interatomic separation. This approximates solvent effects in the absence of an explicit solvent, and allows for faster calculations. Use Distance dependent with the OPLS force field. Since this option simulates the presence of a solvent, you normally choose Constant when solvent is present.

**Scale factor** For Constant,  $\epsilon = (\text{permittivity of free space}) \times (\text{Scale factor})$ . For Distance dependent,  $\epsilon = (\text{permittivity of free space}) \times (\text{Scale factor}) \times (\text{interatomic separation})$ . Scale factor must be  $\geq 1.0$ . The default of 1.0 is appropriate for most systems.

**1–4 Scale factor** Nonbonded interactions (van der Waals and electrostatic) between atoms separated by exactly three bonds are multiplied by this factor.

**Electrostatic** This modifies the magnitude of charge interactions between atoms separated by three bonds. The range is 0 to 1. For AMBER and OPLS, use 0.5. For BIO+, use

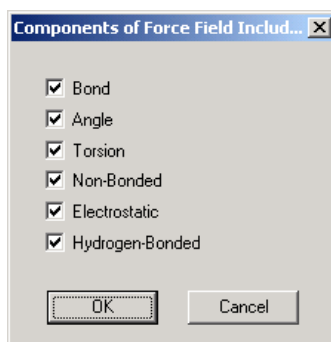
1.0, 0.5, or 0.4 depending on the parameter set.

**van der Waals** This modifies the magnitude of van der Waals interactions between atoms separated by three bonds. The range is 0 to 1. For AMBER, use 0.5; for OPLS, 0.125; for BIO+, the value is fixed at 1.0.

**Cutoffs** These options determine the distance limits for calculating nonbonded interactions and are identical to those for MM+ (see "Cutoffs" on page 292)

## Force Field Components Included Dialog Box

Use the Force Field Components Included dialog box to set which components of the MM+, AMBER, BIO+, and OPLS force fields are to be included in your calculations. HyperChem stores these components settings, which apply to all force fields, in the Registry and uses them for future HyperChem sessions. They are stored as bit flags in an integer referred to as Mechanics Component Flags. The flags are global and apply to all MM+, AMBER, BIO+, and OPLS force fields not individual force fields.



**Bond** These components include all force field terms associated with bond stretching. For MM+ this term includes the cubic stretch term as well as the normal stretch term.

**Angle** These components include all force field terms associated with bond angle bending. For MM+ this term includes the stretch-bend cross term and out-of-plane bending as well as the normal bending



	term. For Bio+ (Charmm) it includes the Urey-Braley term as well as the normal bending term.
Torsion	These components include all torsion terms, i.e. proper as well as improper torsions.
Non-Bonded	These components include all the 6-12 van der Waals type terms between non-bonded atoms.
Electrostatic	These components include all the coulomb and dipole (MM+) interactions.
Hydrogen-Bonded	These components include any specific hydrogen-bonding terms such as are included in the earlier versions of Amber and Charmm.

## Semi-empirical

Choose Semi-empirical to use a semi-empirical quantum-mechanical calculation method instead of a molecular mechanical or *ab initio* quantum-mechanical method. This choice remains on (✓) until you choose Molecular Mechanics or Ab Initio. You can use a semi-empirical method for all calculations on the Compute menu.

Semi-empirical calculations solve the Schrödinger equation, with certain approximations, to describe the electronic properties of atoms and molecules. To simplify and shorten these calculations, semi-empirical methods make many simplifications, including these: calculating only for valence electrons; neglecting the integrals for certain interactions; using standard, non-optimized, electron orbital basis functions; and using parameters derived from experiments. Experimental parameters eliminate the need to calculate certain quantities and to correct for errors resulting from approximations. See *HyperChem Computational Chemistry* for further information about semi-empirical quantum mechanics.

**Important:** The semi-empirical methods in HyperChem can treat all main group elements for which there are parameters in the parameter files, up to and including xenon. Extended Hückel, ZINDO/1, and ZINDO/S can treat the first two rows of transition metals as long as there are entries in the parameter files. See the periodic tables in “Available Elements” on page 300 for availability of parameters.

Most of the available semi-empirical methods include a scheme for eliminating time-consuming calculations of certain integrals for

overlapping orbitals. The Intermediate Neglect of Differential Overlap method (see later) does not calculate certain repulsion integrals that would have small values.

HyperChem also allows you to treat only part of a system quantum mechanically, using *mixed mode* calculations, so that you can study interesting portions of a molecular system at a better level of theory than would be feasible for the whole system. For instance you can analyze binding of a substrate at the active site of a protein quantum mechanically, with the rest of the protein and solvent modeled classically.

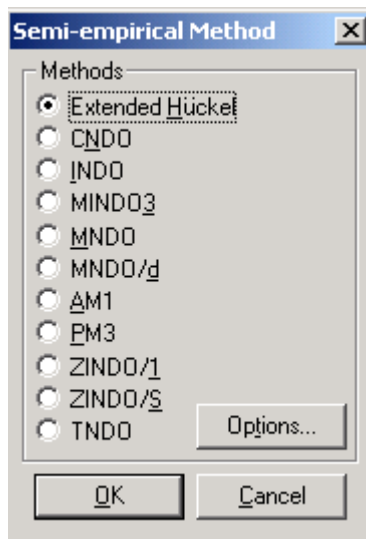
**Important:** Unless you wish to perform a mixed mode calculation, deselect all atoms before using the Semi-empirical Options dialog box or the Compute menu. If there is a selection, HyperChem only treats the selected atoms quantum mechanically. The remaining atoms are treated as a fixed potential field, and are not moved during structural optimizations.

**Important:** If you want to perform a mixed mode calculation, use Extend to sp3 on the Select menu before using the Semi-empirical Options dialog box or the Setup menu. See “Mixed Mode Calculations” on page 311.

**Note:** If you store a molecular system in a HIN file after a semi-empirical calculation, the file includes atomic charges determined in these calculations (see “atom” on page 717).

From the Semi-empirical Method dialog box, you can go to the Semi-empirical Options dialog box (see page 305) to change the conditions for a calculation.

## Semi-empirical Method Dialog Box



- |                        |   |
|------------------------|---|
| <b>Extended Hückel</b> | Useful for calculating molecular orbitals of molecules. Not available for geometry optimization or molecular dynamics calculations. Uses the “one-electron” simplification that assumes no explicit interactions between electrons. This is <i>not</i> a Self-Consistent Field (SCF) method (see <i>HyperChem Computational Chemistry</i> , Practical Guide). |
| <b>CNDO</b>            | Complete Neglect of Differential Overlap. This is the simplest of the SCF methods. It is useful for calculating ground-state electronic properties of open- and closed-shell systems, geometry optimization, and total energy.  |
| <b>INDO</b>            | Intermediate Neglect of Differential Overlap. Improves on CNDO by accounting for certain one-center repulsions between electrons in the same atom. Useful for calculating ground-state electronic properties of open- and closed-shell systems, geometry optimization, and total energy. This is an SCF method.   |
| <b>MINDO3</b>          | An extension of INDO that uses parameters, instead of calculations, for many interactions. Useful for large organic molecules, calculating  |

	electronic properties, geometry optimization, and total energy calculations. This is an SCF method.
<b>MNDO</b>	Useful for various organic molecules containing elements from long rows 1 and 2 of the periodic table, but not transition metals. Eliminates some errors in MINDO3. Calculates electronic properties, optimized geometries, total energy, and heat of formation. This is an SCF method.
<b>MNDO/d</b>	A version of MNDO with the addition of d-orbitals. Useful for main group elements where d-orbitals may be important (such as Sulphur). With better parameters than are currently available in HyperChem would be useful for transition metals. Calculates electronic properties, optimized geometries, total energy, and heat of formation. This is an SCF method.
<b>AM1</b>	An improvement of the MNDO method and one of the most accurate methods. Useful for organic molecules containing elements from long rows 1 and 2 of the periodic table, but not transition metals. Possibly better than MNDO for compounds containing both nitrogen and oxygen. Calculates electronic properties, optimized geometries, total energy, and heat of formation. This is an SCF method.
<b>PM3</b>	A reparameterization of AM1. PM3 differs from AM1 only in the values of the parameters. The parameters for PM3 were derived by comparing a much larger number and wider variety of experimental versus computed molecular properties. Typically, nonbonded interactions are less repulsive in PM3 than in AM1. PM3 is primarily used for organic molecules, but is also parameterized for many main group elements. This is an SCF method.
<b>ZINDO/1</b>	A variation of INDO extended to transition metals. Equivalent to the most recent version of the INDO/1 method which differs from the original by using constant orbital exponents. ZINDO/1 lets you calculate the energy states and geometries of molecules containing transition metals.

- ZINDO/S** An INDO method parameterized to reproduce UV/visible spectroscopic transitions when used with the CI singles method. Useful for predicting UV/visible spectra but not suitable for geometry optimization or molecular dynamics.
- TNDO** A Typed Neglect of Differential Overlap method that includes parameters not just for atomic numbers but for atom types. an attempt to realize a semi-empirical method that has much enhanced accuracy. Useful as a research tool for a limited range of molecules where parameters have been developed and are available. A quite general purpose method if parameters are available for your molecules or molecular property.
- Options . . .** L-click to display the Semi-empirical Options dialog box for setting the conditions for calculation. There are three versions of this dialog box, depending on the semi-empirical method that you choose (see the next two sections).

*See: HyperChem Computational Chemistry* for a discussion of these methods, how to use them, and literature references.

## Available Elements

The following tables show the elements that can be treated by the semi-empirical methods in HyperChem. Where the element symbol is shown it indicates that HyperChem includes the program code necessary to evaluate integrals for the principal and angular quantum numbers of the valence orbitals. In order to perform calculations for an element, HyperChem also needs to know parameters to use in evaluating the integrals. Parameters for some elements are not currently available in the literature. Once these parameters become available, you can add them to the appropriate parameter files (see "Semi-Empirical Files" on page 679). Elements for which parameters are available in the standard HyperChem distribution are shaded in the tables. Unless you have extended parameter files, use only the shaded elements with each method.

## Extended Hückel

The Extended Hückel method can be used with any element in the periodic table; the default parameter set includes all necessary parameters.

## CNDO, INDO

H																					He		
Li	Be																	B	C	N	O	F	Ne
Na	Mg																	Al	Si	P	S	Cl	Ar
K	Ca														Zn	Ga	Ge	As	Se	Br	Kr		
Rb	Sr														Cd	In	Sn	Sb	Te	I	Xe		

## MINDO3

H																							He	
Li	Be																							Ne
Na	Mg																							Ar
K	Ca														Zn	Ga	Ge	As	Se	Br	Kr			
Rb	Sr														Cd	In	Sn	Sb	Te	I	Xe			

**MNDO**

H																			He
Li	Be													B	C	N	O	F	Ne
Na	Mg													Al	Si	P	S	Cl	Ar
K	Ca												Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr												Cd	In	Sn	Sb	Te	I	Xe

**AM1**

H																			He
Li	Be													B	C	N	O	F	Ne
Na	Mg													Al	Si	P	S	Cl	Ar
K	Ca												Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr												Cd	In	Sn	Sb	Te	I	Xe

**PM3**

H																			He
Li	Be													B	C	N	O	F	Ne
Na	Mg													Al	Si	P	S	Cl	Ar
K	Ca												Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr												Cd	In	Sn	Sb	Te	I	Xe

**ZINDO/1**

H																			He
Li	Be												B	C	N	O	F	Ne	
Na	Mg												Al	Si	P	S	Cl	Ar	
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr		
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe		

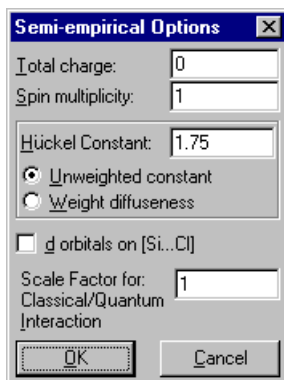
**ZINDO/S**

H																			He
Li	Be												B	C	N	O	F	Ne	
Na	Mg												Al	Si	P	S	Cl	Ar	
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr		
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe		



## Semi-empirical Options (Hückel) Dialog Box

This dialog box provides settings for the Extended Hückel semi-empirical method. All other semi-empirical methods use a different dialog box (see the next section).



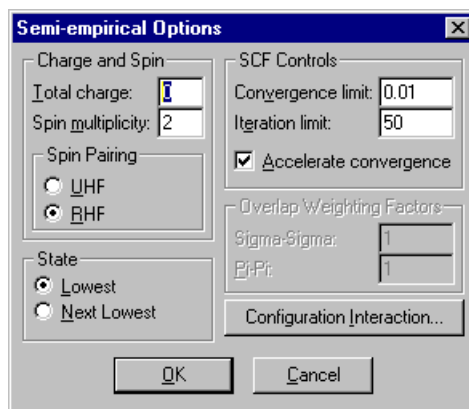
- |                            |  |
|----------------------------|--|
| <b>Total charge</b>        | This is the net excess of nuclear charge over electronic charge. It is zero for a neutral system, a positive integer for a cation, and a negative integer for an anion.  |
| <b>Spin multiplicity</b>   | This is the total spin multiplicity, $2S+1$ , where $S$ is the total electron spin of the system. Each unpaired electron counts for $1/2$ . A closed-shell system (singlet) has a multiplicity of 1. A doublet or triplet state has a multiplicity of 2 and 3, respectively. Enter an integer from 1 to 6.               |
| <b>Hückel Constant</b>     | A constant of proportionality between off-diagonal matrix elements and diagonal matrix elements. The usual value is 1.75. Larger values give more weight to the role of atomic orbital overlap in determining the energy, while smaller values give more weight to binding energies of electrons in the atomic orbitals. |
| <b>Unweighted constant</b> | Uses the Hückel constant without modification.   |
| <b>Weight diffuseness</b>  | Multiplies the Hückel constant by a factor that accounts for the diffuseness of atomic orbitals. This factor is only significant for molecular systems containing orbitals of  |

very different diffuseness. This is rarely the case for organic molecules and main group elements.

- d orbitals** Check this box (✓) to add d orbitals for Si, P, S, and Cl atoms.
- Scale Factor** This parameter scales the inclusion of the classical partial charges in a mixed mode calculation. See *HyperChem Computational Chemistry*, Theory and Methods, for details.

## Semi-empirical Options Dialog Box

This dialog box provides settings for any semi-empirical, quantum mechanical method that you chose, except for Extended Hückel. Extended Hückel has a separate Semi-empirical Options dialog box (see the previous section).



- Total charge** This is the net excess of nuclear charge over electronic charge. It is zero for a neutral system, a positive integer for a cation, and a negative integer for an anion.
- Spin multiplicity** This is the total spin multiplicity,  $2S+1$ , where  $S$  is the total electron spin of the system. Each unpaired electron counts for a spin of  $+1/2$ . A closed-shell system has a multiplicity of 1 (singlet). An open-shell system can have a multiplicity of 2, 3, 4 (doublet, triplet, quartet), or higher. Use integers from 1 to 6.

<b>State</b>	This describes the excitation state of the valence electrons in the system.
<b>Lowest</b>	The lowest electronic state of a given spin multiplicity.
<b>Next lowest</b>	This is the first electronically excited state of a given spin multiplicity (singlet, doublet, triplet, or quartet).
<b>Convergence limit</b>	An SCF calculation ends when the difference in energy after two consecutive iterations is less than this amount. This is the convergence limit for total electron energy during iterations of an SCF calculation. The default is 0.01 kcal/mol. The practical range is 1 to $10^{-3}$ . The calculation might not reach values less than $10^{-3}$ because of round-off errors. It is better to use small values for the convergence limit if you are doing a transition state search (see page 386).
<b>Iteration limit</b>	This sets the maximum number of iterations for an SCF calculation. The calculation ends after this iteration, even if it has not reached the Convergence limit (see above). The results might be incorrect if the calculation was far from convergence or the energy of the system is oscillating. The default of 50 should be correct for most cases, though larger values may be better for transition state searches (see page 386). The practical range is 50 to 200.
<b>Accelerate convergence</b>	Check this box (✓) to reach faster convergence for SCF iterations, using a procedure known as Direct Inversion of Iterative Subspace (DIIS), (see <i>HyperChem Computational Chemistry</i> ).
<b>Spin pairing</b>	This sets the method for calculating spin interactions, of which there are two:
<b>UHF</b>	The Unrestricted Hartree-Fock method provides separate spatial orbitals (alpha and beta) for electrons with each type of spin. This method is useful for open- and closed-shell systems. For closed-shell systems, it can accommodate dissociation

	reactions. Takes longer than an RHF calculation.
RHF	The Restricted Hartree-Fock method requires that a spin pair of electrons occupy the same spatial orbital. Single electrons can also occupy an orbital. This method is useful for open- and closed-shell systems.
Overlap Weighting Factors	Additional parameters for the two ZINDO methods which modify the relative contributions of sigma and pi bonding. See <i>HyperChem Computational Chemistry</i> , Theory and Methods, for further details.
Sigma-Sigma	Adjusts the weighting for sigma-sigma atomic orbital overlap. This should be 1.0 for ZINDO/1 and 1.267 for ZINDO/S. Adjusts the weighting for sigma-sigma atomic orbital overlap.
Pi-Pi	Adjusts the weighting for pi-pi atomic orbital overlap. This should be 1.0 for ZINDO/1. The default of 0.640 for ZINDO/S is appropriate for transition metal complexes, but 0.585 has been used for organic molecules in the literature.
TNDO Version	An option to choose which of two versions of TNDO you wish to use. This option replaces the Overlap Weighting Factors described above when TNDO is chosen. TNDO has its origin in CNDO and INDO and you can choose whether or not to add the extra INDO one-center terms that distinguish CNDO and INDO. In HyperNMR these are referred to as TNDO/1 and TNDO/2.
Configuration interaction	Use this option to activate and set options for a Configuration Interaction calculation in the Configuration Interaction dialog box. This is necessary if you wish to compute UV visible spectra. Choosing this option significantly increases calculation time.

## Configuration Interaction Dialog Box

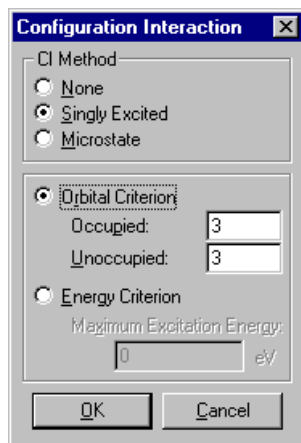
CI calculations can be used to improve the quality of the wavefunction and state energies. Self-consistent field (SCF) level calculations are based on the one-electron model, wherein each electron moves in the *average* field created by the other  $n-1$  electrons in the molecule. Actually, electrons interact instantaneously and therefore have a natural tendency to avoid each other beyond the requirements of the Exclusion Principle. This correlation results in a lower average interelectronic repulsion and thus a lower state energy. The difference between electronic energies calculated at the SCF level versus the exact nonrelativistic energies is the *correlation energy*.

There are two types of electron correlations: static and dynamic. Static correlation refers to a near degeneracy of a given state; a dynamic correlation refers to the instantaneous avoidance of electrons with each other.

CI calculations are perhaps the most widely used method of going computationally beyond an SCF description. An SCF computation yields a configuration describing the orbital occupancy of the electrons. Other configurations may be generated from this reference configuration by exciting electrons from the set of occupied orbitals to the set of virtual (*unoccupied*) orbitals. A CI calculation yields a set of improved states, each of which is represented by a linear combination of these configurations.

*Note:* Only single point calculations and geometry optimization can be performed with Configuration Interaction activated. It is not available for extended Hückel calculations.

Use the Configuration Interaction dialog box to control settings for a CI calculation. To access this dialog box, choose CI in the Semi-empirical Options dialog box.



- CI Method** Choose None, Singly Excited, or Microstate.
- None** Choose None if you do not want to use configuration interaction.
  - Singly Excited** Only singly excited configurations are involved in a CI calculation.
  - Microstate** Includes multiply excited states, as well as singly excited states.
- Orbital Criterion** Set the range of orbitals to and from which electrons are excited to generate a set of interacting configurations.
- Occupied** Range of occupied orbitals starting from HOMO from which electrons are excited.
  - Unoccupied** Range of virtual orbitals starting from LUMO to which electrons are excited.
- Energy Criterion** As an option to Orbital Criterion, this value sets the cutoff energy for generating a set of interacting configurations. This option is available only if you use Singly Excited as the CI method.
- Maximum Excitation** Highest orbital energy difference (eV) between the occupied and unoccupied orbitals to be included in the CI calculation. Generally configurations of high energy do not interact strongly with the reference calculation. The higher the

maximum energy, the greater the number of configurations included in the CI calculation.

## Practical Applications

You can use CI calculations to do the following:

- Calculate UV spectra
- Calculate the energy of excited states
- Study the making or breaking of bonds, and change of spin couplings (e.g. dissociation of H<sub>2</sub>)
- Capture the effects of London dispersion forces
- Describe a nearly degenerate state
- Study the geometry of an excited state
- Study singlet-triplet splittings more accurately

The Microstate CI Method lowers the energy of the uncorrelated ground state as well as excited states. The Singly Excited CI Method is particularly appropriate for calculating UV visible spectra, and does not affect the energy of the ground state (Brillouin's Theorem).

Be careful when you use the Orbital Criterion for symmetrical systems. To get correct results, you must include all or none of any set of degenerate orbitals in the CI, not just some of them. Carrying out an RHF calculation first and studying the Orbitals dialog box will help you to spot degenerate orbitals and avoid this pitfall.

You should also be careful when you use the Energy Criterion that the cutoff energy is large enough to include occupied and unoccupied orbitals.

In large systems there can be many orbitals in a small energy range, and the size of the CI matrix can be very sensitive to the value of the maximum excitation if you use Energy Criterion. Since calculation time depends heavily on the size of the CI matrix, you can end up with very long calculations, especially if you use the MNDO, AM1, or PM3 methods. Again, inspecting the results of an RHF (no CI) calculation will help you avoid these pitfalls.

For information on specific guidelines for calculating UV visible spectra, see "Electronic Spectrum" on page 411.

## Mixed Mode Calculations

In mixed mode calculations, HyperChem treats a selected part of a molecule quantum mechanically; the remainder is treated classically. You can do mixed mode calculations using any of the choices in the Semi-empirical Methods dialog box.

*Note:* If only some atoms in the workspace are selected, then only the selected atoms will move during a structural optimization. The unselected atoms are treated as producing a fixed potential field; the selected atoms interact with that field and with each other.

To perform a mixed mode calculation, you tell HyperChem which atoms to treat quantum mechanically by selecting them. If some molecules are partially selected, the boundary atoms must be  $sp^3$  centers to avoid an abrupt termination of conjugation effects. To ensure that the boundary is between  $sp^3$  atoms, you need to use Extend to  $sp^3$  on the Select menu (see page 244) before using the Semi-empirical Options dialog box or using the Compute menu. This extends the current selection in all directions until the selected region reaches the end of a molecule or finds an  $sp^3$ - $sp^3$  bond.

## Ab Initio and Density Functional

Choose Ab Initio to use an *ab initio* quantum-mechanical calculation method or choose Density Functional to use a Density Functional Theory (DFT) method instead of a molecular mechanical or semi-empirical method. These choices remain on (✓) until you choose Molecular Mechanics or Semi-empirical. You can use an *ab initio* or Density Functional method for all calculations on the Compute menu.

Ab initio calculations and Density Functional calculations have a great deal in common and are described together here since most of their options are very similar. To specify an *ab initio* calculation, you principally have to choose a basis set. For density functional calculations you also have to choose a basis set but must, in addition, choose a *functional*. The SCF parts of both types of calculations proceed essentially identically. In particular we still refer to RHF and UHF for restricted and unrestricted calculations although they apply just as well to Kohn-Sham DFT calculations as they do to Hartree-Fock *ab initio* calculations. We will not make a fine dis-



inction between whether a parameter like “UHF” actually refers to *ab initio* or DFT calculations.

Geometry optimization using an *ab initio* or DFT method may take much longer than using a semi-empirical method. You might first run a molecular mechanics optimization to get close to the optimized geometry, and refine with a semi-empirical method, before running an *ab initio* geometry optimization. For some inorganic structures in particular, molecular mechanics and/or semi-empirical optimizations may give poor results for lack of appropriate parameters, but running the Model Builder on your structure should provide at least a reasonable starting point.

## Choosing a Basis Set

Choosing a basis set is an important choice for *ab initio* and density functional calculations. Any set of one-electron functions can be a basis set in the LCAO approximation. However, a well-defined basis set will predict electronic properties using fewer terms than a poorly-defined basis set. So, choosing a proper basis set in *ab initio* or density functional calculations is critical to the reliability and accuracy of the calculated results<sup>1</sup>.

## DFT Specific Basis Sets

Basis sets in HyperChem can interchangeably be used either for *ab initio* or density functional calculations. The form of density functional theory used in HyperChem requires only one basis set, commonly referred to as the orbital basis set. While some basis sets, such as dzvp and dzvp2, may have been designed with DFT in mind, any basis set can be used in any *ab initio* or DFT context. In particular, a Pople-like approach to DFT as taken by HyperChem, suggests using the same basis sets for DFT as have been designed and used over many years for *ab initio* calculations. Other approaches to DFT use a basis set to expand the charge density and have more complex requirements for the choice of a basis set. For more details on Density Functional Theory see HyperChem's Computational Chemistry Manual.

## Generic Basis Sets

There may be as many basis sets defined for polyatomic calculations as there are quantum chemists! One would like to define, in advance, the standard basis sets that will be suitable to most users.

But, one also wants to allow sophisticated users the capability to modify existing basis sets or define their own basis sets. We have thus defined a *HyperChem basis set file format* and included with HyperChem a number of these \*.BAS files that define standard basis sets. Users, however, can define as many of their own basis sets as they like using this file format. The HyperChem basis set file format is described in Appendix C of this manual.

## Pople-like Basis Sets

Many conventional and commonly-used *ab initio* or DFT basis sets are supported in HyperChem. Many of these were developed by John Pople and collaborators. The nomenclature is such that a basis set like 6-311++G\*\* has an inner 1s orbital composed of 6 primitives (for heavy atoms only), three valence shells (sp shells for heavy atoms and s shells for Hydrogen) composed of three primitives, one primitive and one primitive, a diffuse(+) s-shell on Hydrogens, a diffuse(+) sp shell on heavy atoms, a d-polarization shell(\*) on heavy atoms, and a p-polarization shell on Hydrogens. The first + sign refers to heavy atoms and the first \* also refers to heavy atoms. The STO-3G basis sets is a minimal basis set of Slater orbitals fit to 3 Gaussian primitives. These basis sets include:

- STO-1G and STO-1G\* (H and He)<sup>1</sup>;
- STO-2G and STO-2G\* (H to Xe)<sup>1</sup>;
- STO-3G and STO-3G\* (H to Xe)<sup>1</sup>;
- STO-4G and STO-4G\* (H to Xe)<sup>1</sup>;
- STO-5G and STO-5G\* (H to Xe)<sup>1</sup>;
- STO-6G and STO-6G\* (H to Xe)<sup>1</sup>;
- 3-21G, 3-21G\*, and 3-21G\*\* (H to Xe)<sup>2</sup>;

1. W. J. Hehre, R. F. Stewart, and J. A. Pople, *J. Chem. Phys.*, **51**, 2657 (1969); J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, *J. Chem. Phys.*, **64**, 5142 (1976); R. F. Stewart, *J. Chem. Phys.*, **52**, 431 (1970).
2. J. S. Binkley, J. A. Pople, and W. J. Hehre, *J. Am. Chem. Soc.*, **102**, 939 (1980); M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre, *J. Am. Chem. Soc.*, **104**, 2797 (1982); W. J. Pietro, M. M. Francl, W. J. Hehre, D. J. Defrees, J. A. Pople, and J. S. Binkley, *J. Am. Chem. Soc.*, **104**, 5039 (1982); K.D. Dobbs, W.J. Hehre, *J. Comput. Chem.* **7**, 359 (1986); K.D. Dobbs, W.J. Hehre, *J. Comput. Chem.* **7**, 359 (1986); K.D. Dobbs, W.J. Hehre, *J. Comput. Chem.* **8**, 861 (1987); K.D. Dobbs, W.J. Hehre, *J. Comput. Chem.* **8**, 880 (1987).

- 3-21++G, 3-21++G\*, 3-21++G\*\* (H to Ar)<sup>2</sup>;
- 4-21G, 4-21G\*, and 4-21G\*\* (H to Ne)<sup>1</sup>;
- 6-21G, 6-21G\*, and 6-21G\*\* (H to Ar)<sup>2</sup>;
- 4-31G, 4-31G\*, and 4-31G\*\* (H to Ne)<sup>1</sup>;
- 5-31G, 5-31G\*, and 5-31G\*\* (H to F)<sup>1</sup>;
- 6-31G, 6-31G\*, and 6-31G\*\* (H to Zn)<sup>1</sup>;
- 6-31++G, 6-31+G\*, 6-31++G\*, and 6-31++G\*\* (H to Ca)<sup>1</sup>;
- 6-311G, 6-311G\*, and 6-311G\*\* (H to Kr)<sup>2</sup>;
- 6-311+G\*, 6-311++G\*\* and 6-311++(2d,2p) (H to Ne);

### Other Basis Sets

Other basis sets besides the Pople-like ones are included in HyperChem from among the many available. A very good source for basis sets is the web site established by the Pacific Northwest Laboratory at <http://www.emsl.pnl.gov:2080/docs/homepage.html>.

This site contains a database of basis sets in a HyperChem compatible \*.bas format ( you may have to eliminate some comments lines prior to placing the basis set into HyperChem).

Many of these basis sets are due to the groups of T. Dunning and R. Ahlrichs. Among the basis sets include in HyperChem are:

- D95, D95\* and D95\*\* (H to Cl)<sup>3</sup>;
- SV, DZ, DZP, and TZ (H to Cl)<sup>4</sup>;
- VDZ, pVDZ, VTZ, TZV (H to Kr)<sup>5</sup>;

1. W. J. Hehre, R. Ditchfield, and J. A. Pople, *J. Chem. Phys.*, **56**, 2257 (1972); P. C. Hariharan and J. A. Pople, *Theor. Chim. Acta*, **28**, 213 (1973); M. S. Gordon, *Chem. Phys. Lett.*, **76**, 163 (1980); M.M. Francl, W.J. Pietro, W.J. Hehre, J.S. Binkley, M.S. Gordon, D.J. DeFrees and J.A. Pople, *J. Chem. Phys.* **77**, 3654 (1982); V. Rassolov, J.A. Pople, M. Ratner and T.L. Windus, *J. Chem. Phys.* **109**, 1223 (1998) .
2. R. Krishnan, J. S. Kinkley, R. Seeger, and J. A. Pople, *J. Chem. Phys.*, **72**, 650 (1980); A. D. McLean and G. S. Chandler, *J. Chem. Phys.*, **72**, 5639 (1980); J-P. Blaudeau, M. P. McGrath, L.A. Curtiss and L. Radom, *J. Chem. Phys.* **107**, 5016 (1997); L. A. Curtiss, M. P. McGrath, J-P. Blandeau, N. E. Davis, ! R. C. Binning, Jr. L. Radom, *J. Chem. Phys.* **103**, 6104 (1995).
3. T.H. Dunning and P.J. Hay, in *Modern Theoretical Chemistry*, Plenum, New York, 1976.
4. T.H. Dunning, Jr., *J. Chem. Phys.* **53**, 2823 (1970). .
5. A. Schafer, C. Huber and R. Ahlrichs, *J. Chem. Phys.* **100**, 5829 (1994).

- cc-pcv5z, cc-pcvdz, cc-pcvqz, cc-pcvtz, cc-pv5z, cc-pvdz, cc-pvqz, cc-pvtz (H to Ne)<sup>1</sup>;
- aug-cc-pcv5z, aug-cc-pcvdz, aug-cc-pcvqz, aug-cc-pcvtz, aug-cc-pv5z, aug-cc-pvdz, aug-cc-pvqz, aug-cc-pvtz (H to Ne)<sup>1</sup>;
- DZVP, DZVP2, and TZVP (H to Xe)<sup>2</sup>;

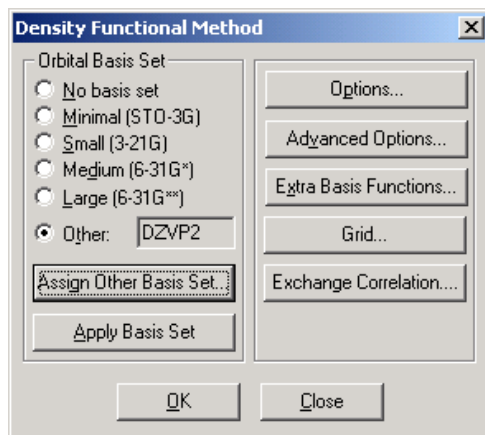
## Ab Initio and Density Functional Method Dialog Boxes

These two dialog boxes are very similar but differ in a few details.

You can use multiple basis sets in a single molecular system. The Apply Basis Set button in these dialog boxes applies the currently selected basis set to the selected atoms or to all the atoms in HyperChem if there is no current selection. For example, some heavy atoms might have a 6-31G basis set (s and p only) while other heavy atoms might use a 6-31G\* basis set (with d-orbitals). This is an unusual but flexible option for *ab initio* or Density Functional calculations.



1. T.H. Dunning, Jr. *J. Chem. Phys.* **90**, 1007 (1989); D.E. Woon and T.H. Dunning, Jr. *J. Chem. Phys.* **100**, 2975 (1994).
2. N. Godbout, D. R. Salahub, J. Andzelm, and E. Wimmer, *Can. J. Chem.* **70**, 560 (1992).



**Basis Set** Specify a basis set for the current selection (if any) or for the whole molecular system.

**No basis set** Do not use any basis set for the current selection (if any) or for the whole molecular system. This option may be useful only when applying an extra basis function to atoms.

**Minimal (STO-3G)** Selects the STO-3G basis set.

**Small (3-21G)** Selects the 3-21G basis set.

**Medium (6-31G\*)** Selects the 6-31G\* basis set.

**Large (6-31G\*\*)** Selects the 6-31G\*\* basis set.

**Other** Choose this option and invoke Assign Other Basis Set in order to choose another basis set.

**Assign Other Basis Set...** Click on this button to bring up the Assign Other Basis Set dialog box, which contains a full list of basis sets (except for the basis sets already listed in this dialog box).

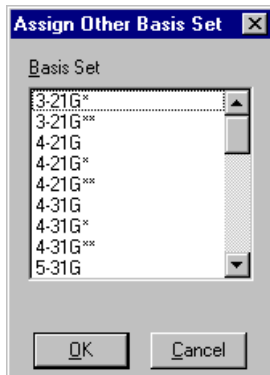
**Extra Basis Function...** Click on this button to bring up the Extra Basis Function dialog box, which lets you add an extra

	basis function to the selected atoms (or to all the atoms).
Options...	Click on this button to bring up the Ab Initio and DFT Options dialog box, which contains the basic options for executing an <i>ab initio</i> or Density Functional calculation.
Advanced Options...	Click on this button to bring up the Ab Initio and DFT Advanced Options dialog box, which contains options affecting the performance of <i>ab initio</i> or Density Functional calculations.
Apply Basis Set...	Click on this button to apply the current basis set selection to the current selection of atoms (if any) or to all the atoms in HyperChem.
Grid...	(DFT Only) Click on this button to bring up the DFT Integration Grids dialog box, which contains basic options for choosing a grid to be used for numerical integrations. The simplest option is to use the Standard Grid #1 defined by Pople and collaborators. Alternatively, you may explicitly specify the number of radial and angular points for a particular integration scheme. You may also specify a value of the density below which densities are considered too small to be considered. Larger values of this quantity will speed up execution at the expense of numerical accuracy.
Exchange Correlation...	(DFT Only) Click on this button to bring up the Exchange Correlation Potential dialog box to specify which exchange-correlation density functional to use in your calculations.
OK	Click on this button to save the current settings, apply the basis set selection, and close this dialog box.
Cancel	Click on this button to close this dialog box without saving the current setting or applying the current basis set selection.

You can apply different basis sets to different parts of a molecular system. HyperChem can use different basis sets for different atoms.

## Assign Other Basis Set Dialog Box

Use this dialog box to get a full list of basis sets, except for the basis sets explicitly specified in the Ab Initio and Density Function Method dialog boxes. You can click on any one of these basis sets to select one. HyperChem will bring your selection of basis set to the text box in the Ab Initio and Density Functional Method dialog box, when you click on the OK button.



HyperChem lists all the basis sets in this dialog box, except for the basis sets already listed in the Ab Initio Method dialog box. These basis sets can be one of the standard basis sets or your own basis set saved in a file ending with .BAS. To have your own basis set appear in this dialog box, an entry must be made for it in the [basisset] section of the Registry or the CHEM.INI file.

## Ab Initio and DFT Options Dialog Box

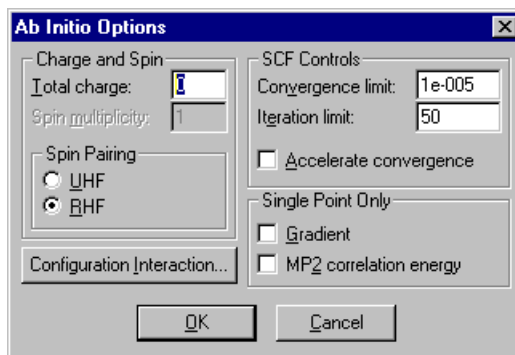
Use this dialog box to choose the basic options for executing *ab initio* or DFT calculations. The basic options are similar to the basic options in the Semi-empirical Option dialog box.

“Gradient” specifies the calculation of gradients (the first derivatives of the total energy with respect to the nuclear positions). The RMS gradient gives an indication of the deviation from an optimized structure. The computations of two-electron integrals and their derivatives are time-consuming, because of the huge number of the two-electron integrals even for a medium size of molecule. You may not be interested in gradients for single point calculations, so you can turn off (not check) gradient calculation to speed up your task. This option applies to single point calculations only.

HyperGauss always computes the gradients during geometry optimization, molecular dynamics, and vibration calculations.

“MP2 Correlation Energy” specifies the calculation of electron correlation energy using the Møller-Plesset second order perturbation theory (MP2). This option can only be applied to Single Point calculations. MP2 correlation energy calculations may increase the computational time because a two-electron integral transformation from atomic orbitals (AOs) to molecular orbitals (MOs) is required. HyperChem may also need additional main memory and/or extra disk space to store the two-electron integrals of the MOs.

**Note:** The MP2 energy for DFT calculations is not well specified and may be meaningless. Similarly, Configuration Interaction for DFT calculations is not well specified. In DFT calculations, Brillouin’s theorem is not satisfied as it is in *ab initio* Hartree-Fock calculations. Nor do the Kohn-Sham orbitals have the same meaning as Hartree-Fock orbitals. This leads to anomalies in post-SCF results, such as MP2 and CI, when applied to DFT rather than *ab initio* calculations.



This dialog box has the following components:

**Charge and spin** Use this area to specify the net charge on the current molecular system and whether the system is a singlet, doublet, etc.

**Total charge** This is the net excess of nuclear charge over the electronic charge. It is zero for a neutral system, a positive integer for a cation, and a negative integer for an anion.



<b>Spin multiplicity</b>	This lists the electronic spin multiplicity that will be used for the current molecular system. A closed-shell system has a spin multiplicity of 1 (singlet). An open-shell system can have a multiplicity of 2, 3, 4 (doublet, triplet, quartet), or higher. Use integers from 1 to 4.
<b>SCF controls</b>	This section is used to specify the desired accuracy for the SCF wave function, and the maximum number of iterations that are allowed for reaching that accuracy.
<b>Convergence limit</b>	An SCF calculation ends when the difference in energy between any two consecutive iterations is less than this amount, in kcal/mol. The default value is 0.00001. The practical range is 1 to $10^{-8}$ . The calculation might not reach values less than $10^{-10}$ because of round-off errors. It is better to use relatively small values for transition state searches (see page 386).
<b>Iteration limit</b>	This sets the maximum number of iterations for an SCF calculation. The calculation ends after this number of iterations, even if it has not reached the Convergence limit. The results may not be correct if the calculation was far from convergence or the energy of the system is oscillating. The default of 50 should be correct for most cases, though larger values may be appropriate for transition state searches (see page 386). The practical range is 50 to 200. If the calculation exceeds the default iteration limit before it reaches the convergence limit, then most likely there is a convergence failure; simply increasing this limit is unlikely to help. The convergence accelerator (below) may help in some cases.
<b>Accelerate convergence</b>	With this box checked (✓), HyperChem will apply the Direct Inversion of Iterative Subspace (DIIS) procedure to speed up the

SCF calculation at the expense of requiring much more memory. This option may increase the computational time for individual iterations because the Fock (or Kohn-Sham) matrix has to be calculated as a linear combination of the current Fock (K-S) matrix and Fock (K-S) matrices from previous iterations. It is likely to decrease the actual number of iterations required.

Single Point only	Specifies whether or not the calculation of gradient and MP2 correlation energy are required. These options are only used in Single Point calculations and ignored in any other type of computations.
Gradient	Check this box (✓) to compute the gradient (the first derivatives of the total energy with respect to the nuclear positions).
MP2 correlation energy	Check this box (✓) to compute the correlation energy by second order perturbation methods (MP2).
Spin pairing	This sets the method for calculating spin interactions, of which there are two:
UHF	The Unrestricted Hartree-Fock (or Kohn-Sham) method provides separate spatial orbitals (alpha and beta) for electrons with each type of spin. This method is useful for both open- and closed-shell systems. For closed-shell systems, it can be useful for studying dissociation of molecular systems. A UHF calculation takes longer than an RHF calculation. ROHF (spin Restricted Open-shell Hartree-Fock or Kohn-Sham) is not supported in the current version of HyperChem for <i>ab initio</i> or DFT calculations. Use UHF for open-shell doublets and triplets.
RHF	The spin-Restricted Hartree-Fock (or Kohn-Sham) method requires that spin-paired electrons reside in each occupied spatial orbital. One would normally use RHF for closed-shell singlets.

Cl...	Click on this button to bring up the Configuration Interaction dialog box.
OK	Accept any changes made and close the dialog box.
Cancel	Close the dialog box without accepting any changes made.

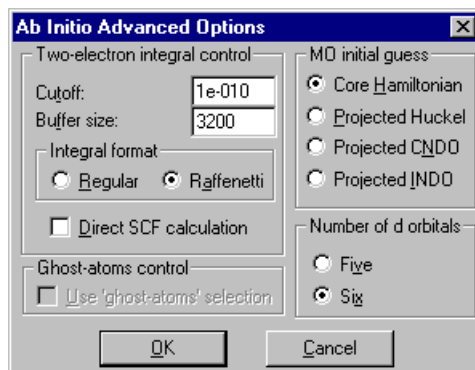
*Note:* HyperChem doesn't support the spin Restricted Open-shell Hartree-Fock (ROHF) methods when using *ab initio* or DFT methods.

## Ab Initio and DFT Advanced Options Dialog Box

You can use the advanced options dialog box to tune the performance of your *ab initio* calculations.

“Regular Integral Format” specifies the use of a “regular” format for saving the two-electron integrals. HyperChem uses 16 bytes to store every integral. The first 8 bytes store the four indices of an integral and the second 8 bytes store its value. HyperChem only stores an integral and its associated indices when the integral's absolute value is greater than or equal to the two-electron integral cutoff. The two-electron integral and its indices are stored without any modification when you choose this regular two-electron integral format. These two-electron integrals and their indices can be printed out to a log file by choosing a proper setting for QuantumPrintLevel in the Start Log dialog box.

“Raffenetti Integral Format” specifies the use of the Raffenetti two-electron integral format [R.C. Raffenetti, *Chem. Phys. Letters*, **20**, 335 (1973)]. HyperChem can calculate all two-electron integrals and store them in a special form that makes it easier to generate a Fock matrix during the SCF iterations. The Raffenetti two-electron integral format may take more main memory or disk space for the two-electron integrals, particularly in a UHF calculation, but may still be faster than the regular two-electron integral format for the same calculation. The Raffenetti format is not available when performing MP2 calculations. Similarly, the Raffenetti format is not available for DFT calculations and a DFT choice will automatically switch the integral format to “Regular Integral Format”.



This dialog box contains the following components:

**Two-electron integral control** Controls the two-electron integral calculation and storage.

**Cutoff** Keep only the two-electron integrals with an absolute value greater than or equal to this value. The default value is  $10^{-10}$  Hartree. This option controls the performance of the SCF iterations and the accuracy of the wave function and energies since it can decrease the number of calculated two-electron integrals.

**Buffer size** Gives the size of computer main memory in double-precision words (8 bytes for each double precision word) for storing the two-electron integrals prior to saving them to the hard drive. Once this buffer is full, these two-electron integrals are written to a temporary file on a disk (the selected disk can be set in the File/Preferences/Path property sheet). A large buffer may reduce the processing time through fewer disk accesses. If the two-electron integral buffer size is big enough to hold all the integrals, HyperChem does not use the disk.

**Note:** If an *ab initio* calculation is stopped before completion by a power failure or other abnormal termination, these temporary files will not be deleted automatically. In this case you must man-

ually delete the files to release the disk space. (The files are deleted automatically when a calculation ends, or is cancelled normally.) The files are stored in the directory specified in the Preferences/Paths dialog box.

<b>Integral format</b>	Specifies the format to store the two-electron integrals.
<b>Regular</b>	Store the two-electron integrals and their indices in their original format. Density functional calculations always use this format and automatically switch to this format, requiring you to manually switch back to Raffenetti for <i>ab initio</i> calculations, if so desired..
<b>Raffenetti</b>	Store the two-electron integrals and indices in the Raffenetti format to speed up the formation of Fock matrix.
<b>Direct SCF calculation</b>	Check this box (✓) to use a direct SCF calculation. The two-electron integrals are computed in each iteration, rather than computed and stored on a hard drive prior to the SCF calculation. This calculation will be considerably slower than a regular SCF calculation, but avoids using disk space or a large amount of main memory. This option may be practical for large molecular systems run on a computer which has little available space on the disk.
<b>Ghost-atoms control</b>	Controls the use of a “ghost-atoms” selection.
<b>Use “ghost-atoms” selection</b>	Check this box (✓) to use a “ghost-atoms” selection. This option is available only when a named “ghost-atoms” selection is present in HyperChem.
<b>MO initial guess</b>	Specifies an initial guess for the MO coefficients.

Core Hamiltonian	Choose this radio button to use the MO coefficients obtained from diagonalizing the core Hamiltonian.
Projected Hückel	Choose this radio button to use the projected MO coefficients generated by the Hückel method.
Projected CNDO	Choose this radio button to use the projected MO coefficients generated by a CNDO calculation.
Projected INDO	Choose this radio button to use the projected MO coefficients generated by an INDO calculation.
Number of d orbitals	Specifies using Cartesian Gaussians (6) or Hermite Gaussians (5).
Five	Choose this box to use Hermite Gaussians, which consist of five d orbitals, i.e., $d_0$ , $d_{+1}$ , $d_{-1}$ , $d_{+2}$ , and $d_{-2}$ .
Six	Choose this box to use Cartesian Gaussians, which consist of six d orbitals, i.e., $d_{xx}$ , $d_{yy}$ , $d_{zz}$ , $d_{xy}$ , $d_{xz}$ , and $d_{yz}$ .
OK	Accept the changes made and close the dialog box.
Cancel	Close the dialog box without accepting the changes made.

## Initial Guess of MO Coefficients

An initial guess at the molecular orbital coefficients is necessary for an SCF calculation. Usually, the initial guess is obtained by solving the Hartree-Fock-Roothaan equations with the replacement of the Fock matrix by the core Hamiltonian. This initial guess of the MO coefficients is usually acceptable. However, in some cases, the core Hamiltonian leads to incorrect occupied orbitals or the initial guess may be far from the final converged SCF MO coefficients. Hence, HyperChem supports a few alternative methods for generating an initial guess at the MO coefficients. These alternative methods are:

- Projected Hückel: the initial guess at the MO coefficients is obtained from an extended Hückel calculation;

- Projected CNDO: the initial guess at the MO coefficients is obtained from a CNDO calculation;
- Projected INDO: the initial guess at the MO coefficients is obtained from an INDO calculation.

However, these alternative methods can be only applied to certain elements. For example, the projected CNDO/INDO may be used only for molecular systems with atomic numbers less than or equal to 18 (Ar). Elements beyond 18 are not available in the projected CNDO/INDO initial guess.

## Number of d Orbitals

There are two different sets of d-type functions (d orbitals) used in *ab initio* calculations. One 3d set consists of five 3d functions —  $3d_0$ ,  $3d_{+1}$ ,  $3d_{-1}$ ,  $3d_{+2}$ , and  $3d_{-2}$ , and is normally used in STO-NG basis sets. The other is a set of six 3d functions —  $3d_{xx}$ ,  $3d_{yy}$ ,  $3d_{zz}$ ,  $3d_{xy}$ ,  $3d_{xz}$ , and  $3d_{yz}$ , and is used in the split-valence basis sets, such as, 3-21G, 4-31G, 6-31G, etc.

The contraction exponents and coefficients of the d-type functions were optimized using five d-primitives (the first set of d-type functions) for the STO-NG basis sets and six d-primitives (the second set of d-type functions) for the split-valence basis sets. Thus, five d orbitals are recommended for the STO-NG basis sets and six d orbitals for the split-valence basis sets.

## Ghost-Atoms

With HyperChem you can always name a selection of atoms using the Name Selection item of the Select menu. A named selection called “ghost-atoms” has a special meaning, however. In most situations any atom that is a “ghost” atom, (i.e., is a member of the ghost-atom set) is treated just like any other atom and the named selection “ghost-atoms” is treated like any other named selection. However, HyperChem may treat the ghost atoms differently from the regular atoms when performing certain *ab initio* and DFT calculations. If you have specified the use of “ghost-atoms” in the Ab Initio and DFT Advanced Options dialog box, then a ghost atom is represented in *ab initio* and DFT calculations by its basis set only and does not have any nucleus or electrons associated with it. Since there are atomic orbitals (basis functions) belonging to a ghost atom, there will be Mulliken charges attached to the ghost atoms after performing an *ab initio* or DFT calculation.

The concept of “ghost-atoms” only applies to Single Point calculations in the current version of HyperChem.

A major use of ghost atoms is to be able to offset the effect of basis set superposition effects (BSSE). For example, as two monomers come together the basis set of one monomer extends the basis set of the other, lowering the dimer energy more than just from the physical monomer-monomer interaction. A way to offset this in computing the dimerization energy is to use the same basis set in both monomer and dimer calculations. That is, the monomer energy is the energy of the dimer with ghost-atoms for the atoms of one of the monomers and “real” atoms for the other monomer.

## Configuration Interaction

Use Configuration Interaction to predict the electronic spectra of molecules. The Configuration Interaction wave function computes a ground state plus low lying excited states. You can obtain electronic absorption frequencies from the differences between the energy of the ground state and the excited states.

A configuration interaction calculation is available only for single points when the reference ground state is obtained from an RHF calculation.

The calculation mixes all single determinant wavefunctions that can be obtained from the ground state by exciting electrons from a subset of the occupied orbitals (of the ground state) to a subset of the unoccupied orbitals. The subsets are specified as a fixed number (highest occupied or lowest unoccupied) or by an energy criterion associated with the energy difference between the occupied orbital and the unoccupied orbital.

For density functional calculations, the difference in orbital energies can be used to predict spectra unlike *ab initio* calculations. The difference corresponds to the average of singlet-triplet excitation energies. The use of configuration interaction for DFT calculations is questionable.

## Configuration Interaction Dialog Box

This dialog box can be used to invoke CI calculations in order to obtain an electronic spectrum. For details, please see the information about the same box as used in semi-empirical calculations, on page 308. You cannot use an *ab initio* method with the CI option turned on for geometry optimizations, molecular dynamics simu-



lations, or vibrational calculations, in the current version of HyperChem.

## Extra Basis Function Dialog Box

This dialog box offers an easy way to modify an existing basis set by adding extra basis functions to atoms. You may want to add a polarization function to hydrogen or diffuse functions to heavy atoms. An alternative way to add extra basis functions to the existing basis sets is to edit an existing basis set file, make your own changes, and create a new basis set.

One and only one extra shell (of type S, P, D, SP, or SPD) can be applied to any atom in the current version of HyperChem. Different extra shells can be applied to different atoms but any atom can receive only one extra shell. If you would like to add more than one extra shell to any atom, you will need to modify a basis set file and create a new basis set. To use this new basis set you will need to include it in the [basisset] section of the Registry.

A basis set for an atom thus consists of a standard basis set (3-21G, for example) and an optional extra shell.



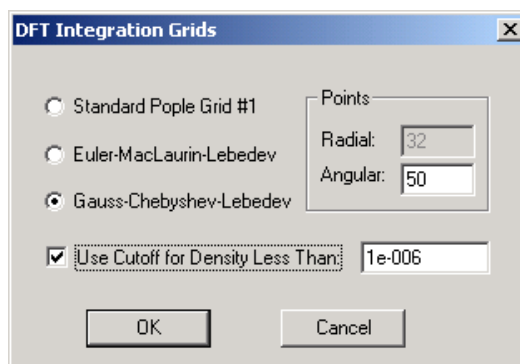
This dialog box contains the following items:

Shell type	Specifies the type of shell to be added to the selected atoms (if any) or to the whole molecular system.
None	Choose None if you do not want to add extra basis function(s) to the atoms.
S	Add an extra basis function associated with an S-type shell to the selected atoms.

P	Add extra basis functions associated with a P-type shell to the selected atoms.
D	Add extra basis functions associated with a D-type shell to the selected atoms.
SP	Add extra basis functions associated with an SP-type shell to the selected atoms.
SPD	Add extra basis functions associated with an SPD-type shell to the selected atoms.
Exponent	Give the exponent for the selected type of Gaussian function (shell).
OK	Accept the changes made and close the dialog box.
Cancel	Close the dialog box without accepting the changes made.

### DFT Integration Grids Dialog Box

This dialog box offers an easy way to specify the integration grid to be used in Density Functional Theory (DFT) calculations. In DFT, the functional is too mathematically complicated to allow analytical integration schemes, as used in *ab initio* calculations. Numerical integration is thus required and you must specify a numerical integration scheme. The more numerical points used the more accurate the integration but the longer it takes.



This dialog box contains the following items:

**Grid type** Specifies the basis numerical integration scheme to be used. See the HyperChem Computational Chemistry Manual for further details. You must choose from among the following types:

**Standard Pople Grid #1** Choose this option for a standard grid that need not be specified further. The grid is a basic Euler-MacLaurin-Lebedev where the points have been optimized (fewer angular points at smaller radii) and is known to work effectively for basis sets like 6-31G\*.

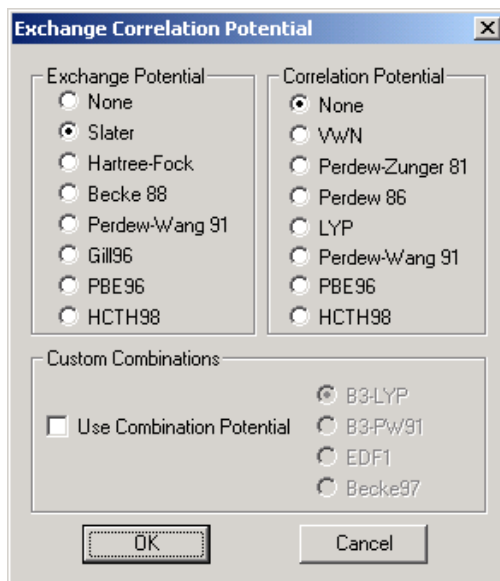
**Euler-MacLaurin-Lebedev** Choose this option to use a Euler-MacLaurin scheme for an arbitrary number of radial points (50 is a reasonable default) and a Lebedev scheme for the angular integration. The number of Lebedev angular points must be one of 6, 26, 38, 50, 86, 110, 194.

**Gauss-Chebyshev-Lebedev** Choose this option to use a Gauss-Chebyshev radial integration using a fixed number of 32 points mapped to the range 0 to infinity. The angular integration will use Lebedev scheme as do each of the integration options. The number of Lebedev angular points must be one of 6, 26, 38, 50, 86, 110, 194.

**Use Cutoff for Density** Allows you to specify a value of the density below which points in space having this density are ignored in the numerical integration. This speeds up calculations. If no cutoff is specified, all integration points are used in the integrations involving the exchange-correlation functional. A value which gives an accuracy of less than a tenth of a Kcal/mole may be  $1.0 \times 10^{-6}$ .

## Exchange Correlation Potential Dialog Box

This dialog box provides the means to specify the exchange-correlation functional to be used in your DFT calculations.



There are many choices and/o choice combinations. You may need to be conversant in density functional theory to make an optimum choice. Many of the options provide ways of exploring the difference between functionals for research purposes rather than that they are expected to be the optimal functional for particular situations. Density functional theory is still an active area of research and many researchers are involved in trying to find better functionals.

The functionals may be chosen by specifying one of the 8 exchange functionals in combination with one of the 8 correlation functions (64 combinations). Alternatively, a specific exchange-correlation combination that has been developed as an optimum combination is possible. Unless you are explicitly exploring the difference between functionals you may wish to use a combination potential.

The dialog box contains the following items:

<b>Exchange Potential</b>	Specifies how to treat exchange. You can choose from among the following exchange options:
---------------------------	--

None	Choose this option if you want to ignore exchange.
Slater	Choose this option if you want the simplest local density approximation to exchange.
Hartree-Fock	Choose this option if you want the usual exchange as per a Hartree-Fock calculation.
Becke 88e	Choose this option if you want to use a popular gradient-corrected potential developed by Becke in 1988.
Perdew-Wang 91	Choose this PW91 option if you want to use an exchange potential developed in conjunction with the popular PW91 correlation potential.
Gill96	Choose this option if you want a simple gradient-corrected exchange potential.
PBE96	Choose this option if you want a simpler version of PW91 developed by Perdew, Burke, and Ernzerhof.
HCTH98	Choose this option if you want a recent flexible exchange term developed by Hamprecht, Cohen, Tozer, and Handy. Closely related to Becke97 but is pure functional with no HF component.  If you use the script <code>read-hcth-coefficients</code> or Registry setting <code>ReadHCTHCoefficients</code> , you can read the 15 coefficients defining the exchange-correlation potential from a file <code>hcth.coef</code> .
Correlation Potential	Specifies how to treat correlation. You can choose from among the following exchange options:
None	Choose this option if you want to ignore correlation. In conjunction with the Hartree-Fock choice for exchange produces a Hartree-Fock result.
VWN	Choose this option if you want a local density approximation to correlation as developed by Vosko, Wilk, and Nusair.

Perdew-Zunger 81	Choose this option if you want a local density approximation to correlation. Often used as precursor to other Perdew-like correlation potentials.
Perdew 86	Choose this option if you want to use a gradient-corrected correlation potential. Often used in conjunction with Becke 88.
LYP	Choose this option if you want to use the Lee-Yang-Parr gradient-corrected correlation potential. Commonly combined with Becke88 as the B-LYP potential.
Perdew-Wang 91	Choose this PW91 option if you want to use a later gradient-corrected exchange potential. Commonly combined with the corresponding PW91 exchange potential.
PBE96	Choose this option if you want a simpler version of PW91 by Perdew, Burke, and Ernzerhof. Commonly used with corresponding PBE96 exchange potential.
HCTH98	Choose this option if you want a recent flexible exchange term developed by Hamprecht, Cohen, Tozer, and Handy. Closely related to Becke97.
Custom Combinations	Choose this option if you want to use a combined exchange-correlation potential rather than specifying each component. These potentials are sometimes referred to as hybrid potentials. They are not easily separated into their exchange and correlation terms. You can choose from among the following option:
B3-LYP	A popular combination developed by Becke. Includes a component of HF exchange
B3-PW91	A popular combination developed by Becke. Includes a component of HF exchange. Similar to B3-LYP but replaces the LYP correlation potential with the Perdew-Wang 91 correlation potential.

EDF1	An empirical Density Functional #1 combination developed by Pople and co-workers for 6-31G*-like basis sets.
Becke97	A recent proposal by Becke for both exchange and correlation. Includes a component of HF exchange.

## Mixed Mode Calculations

You can perform quantum mechanical calculations on a part of a molecular system, such as a solute, while using molecular mechanics for the rest of the system, such as the solvent surrounding the solute. This boundary technique is available in HyperChem for all quantum mechanical methods. It is somewhat less complete with *ab initio* or DFT calculations than with semi-empirical calculations, however. With *ab initio* or DFT calculations the boundary must occur between molecules rather than inside a molecule.

Choose the region (single or multiple molecules) of interest for an *ab initio* or DFT calculation from the total molecular system. HyperChem performs the *ab initio* or DFT calculation for the selected region using the perturbation of an electrostatic potential arising from the net charges on the atoms of the unselected part. (For further details of this electrostatic potential perturbation implemented in HyperChem, please see the *Computational Chemistry* manual.)

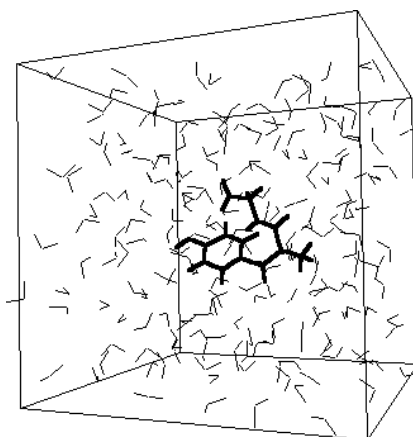
The algorithms of the mixed classical-quantum model used in HyperChem are different for semi-empirical and *ab initio* or DFT methods. The semi-empirical methods in HyperChem treat boundary atoms (atoms that are used to terminate a subset quantum mechanical region inside a single molecule) as specially parameterized pseudo-fluorine atoms. However, HyperChem will not carry on mixed model calculations, using *ab initio* or DFT quantum mechanical methods, if there are any boundary atoms in the molecular system. Thus, if you would like to compute a wavefunction for only a portion of a molecular system using *ab initio* or DFT methods, you must select single or multiple isolated molecules as your selected quantum mechanical region, without any boundary atoms.

Semi-empirical methods could thus treat the receptor portion of a single protein molecule as a quantum mechanical region but *ab initio* and DFT methods cannot. However, both semi-

empirical and *ab initio* or DFT methods could treat solvents as a perturbation on a quantum mechanical solute. In the future, HyperChem may have an algorithm for correctly treating the boundary between a classical region and an *ab initio* or DFT quantum mechanical region in the same molecule. For the time being it does not.

## Periodic Box

This item places the molecular system in a periodic box containing water molecules.



A periodic box imposes periodic boundary conditions on calculations. Periodic boundary conditions provide identical molecular systems (virtual images), identical to the one in the periodic box, surrounding the periodic box. For the purposes of calculations, molecules can move in a constant-density environment, similar to being in a liquid. The box also prevents solvent molecules from diffusing away or evaporating during a simulation (for example, during a molecular dynamics calculation).

**Note:** Periodic boundary conditions apply only for molecular mechanics calculations. The quantum mechanics calculation methods disregard the periodic effects.

Periodic Box adds to a molecular system TIP3P models of water molecules (Jorgenson *et al.*, 1983, see page 621), equilibrated at 300 K (25°C) and one atmosphere. Without a periodic box, HyperChem assumes that molecules are in a vacuum.



**Caution:** Do not use the Model Builder (see “Converting 2D Drawings into 3D Structures” on page 487) for the water molecules in a periodic box. The Model Builder removes the conformational information that comes with these molecules.

Selecting a periodic box makes these changes to the molecular system:

- First, HyperChem changes the orientation of the molecular system to align the inertial axes with a box that has one face parallel to the screen. HyperChem chooses the size and shape of the box to minimize the number of water molecules required. The standard size box is a cube 18.70 Ångstroms on a side. Boxes with dimensions 1, 2, or 3 times the standard box minimize bad contacts. HyperChem does not accept box dimensions larger than three times the standard box (56.10 Ångstroms).
- HyperChem fills the box with water molecules. The number of water molecules depends on the size of the box that you choose. At the edges of the box, HyperChem places water molecules only if their oxygens fall within the box, and excludes water molecules from a specified distance of solute atoms.
- The solute molecules remain in their original conformation. The geometry of solvent molecules come from the database files of equilibrated TIP3P water molecules.
- HyperChem displays the new molecular system in a box. The periodic box appears 3D if you turn on Perspective View on the Display menu. You can remove or replace the appearance of this box, but not the boundary conditions, by choosing Show Periodic Box on the Display menu.

When you choose Periodic Box, HyperChem shows you the Periodic Box Options dialog box for setting the solvation conditions. You can choose the size of the box and the smallest distance between the solute and water molecules.

## Periodic Boundary Conditions

Periodic boundary conditions and solvation are separate features of HyperChem. After you set up a periodic box, you can select and remove the water molecules, but keep the boundary conditions and the periodic box. You can turn off established periodic boundary conditions during a molecular dynamics calculation by choos-

ing In vacuo (see the next section). The water molecules remain in the molecular system.

HyperChem's periodic boundary conditions provide a sharp, rectangular cutoff of the distance for calculating interactions between nonbonded atoms. An atom interacts only with one nearest image of other atoms and never with an image of itself. This sharp cutoff can cause discontinuities in the potential surface that interfere with geometry optimization and molecular dynamics calculations. To avoid this, HyperChem automatically changes the cutoff condition to Switched (see "Force Field Options Dialog Box" on page 293) for a molecular system in a periodic box. You can also choose Shifted cutoffs. These options gradually decrease nonbonded interactions over a specified distance.

### Periodic Box Options Dialog Box

Smallest Box Enclosing Solute (Å)		Periodic Box Size (Å)	
X:	4.7571726	X:	18.70136
Y:	0.0000000	Y:	18.70136
Z:	4.8217773	Z:	18.70136

Maximum number of water molecules: 216

Minimum distance between solvent and solute atoms: 2.3 Å

OK Cancel

**Smallest box enclosing solute** For information only. X, Y, and Z give the smallest box size, in Ångstroms, that can completely enclose the molecular system (solute). This minimal-sized box cannot accommodate the appropriate number of water molecules and is for reference only.

**Periodic box size** These are the recommended dimensions, in Ångstroms, of the periodic box. HyperChem recommends dimensions (X, Y, and Z) for a cube at least twice the largest dimension for Smallest box enclosing solute, or the standard size box, whichever is greater. A standard size box is a cube 18.70 Ång-

stroms on a side. Boxes with dimensions 1, 2, or 3 times the standard box minimize bad contacts. Larger boxes prevent the solute from interacting with its images, but require more computing time. Do not use box dimensions larger than three times the standard box (56.10 Å). HyperChem accepts noncubic dimensions.

<b>Maximum number of water molecules</b>	For information only. This reports the maximum number of water molecules that HyperChem can add. This number depends on the periodic box size. The actual number added is usually less than the maximum, because of the volume that the solute occupies.
<b>Minimum distance between solvent and solute atoms</b>	This is the minimum separation between water atoms and atoms already in the system to avoid bad contacts. The default is 2.3 Ångstroms. A distance larger than half the largest box dimension (shown for Smallest box enclosing solute) excludes all solvent molecules. HyperChem does not draw a periodic box but turns on periodic boundary conditions. Practical range: 1–5 Ångstroms.

If you need a specific number of water molecules in the box, you can select water molecules and use Cut, Copy, and Paste on the Edit menu to change the number of molecules.

## Changing periodic box size

HyperChem considers the entire system when the Setup/Periodic Box option is used. If you wish to change the size of an existing periodic box it is usually best to remove any solvent water molecules first.

To Remove solvent:

1. Choose molecule selection via Select/Molecules.
2. Select the solute molecule(s).
3. Use Select/Complement Selection to select the solvent.
4. Use Edit/Clear to delete the selected solvent.

You can hide or display the periodic box by choosing Show Periodic Box on the Display menu. The boundary conditions remain as part

of the molecular system when you turn off the appearance of the box.

## Removing the periodic box

To remove an existing periodic box from a molecular system, the easiest method is to open the Periodic Box dialog box, and then click on the Cancel button. Alternately, you can edit the HIN file containing the information for this molecular system (see “box” on page 716), deleting the record beginning with box. This will remove the box but not the solvent molecules; to delete the solvent molecules, see the previous section.

## Solvents other than water

Pre-defined files for periodic boxes of solvents other than water are not distributed with HyperChem, but creating your own solvent boxes is relatively straightforward. To find the number of solvent molecules appropriate for your box, you will need the box volume (length  $\times$  width  $\times$  height) and the molecular weight and density of your solvent. Then:

$$\# \text{ molecules} = \frac{\text{volume} \times \text{density} \times \text{Avogadro's number}}{\text{molecular weight}}$$

For example, for a periodic box 18.62 Å in each dimension, filled with water at 25°C, the density is 1.00 g/cm<sup>3</sup> and the molecular weight is 18.02 g/mol. Substituting into the above, and including the conversion from Å to cm:

$$\begin{aligned} \# \text{ molecules} &= \frac{(18.62 \text{ \AA})^3 \times \left(\frac{1 \text{ cm}}{10^8 \text{ \AA}}\right)^3 \times 1.00 \text{ g/cm}^3 \times 6.023 \times 10^{23} \text{ mol}^{-1}}{18.02 \text{ g/mol}} \\ &= 215.8 \end{aligned}$$

which matches the 216 water molecules found in the standard periodic box.

The solvent molecules should be oriented randomly in the box and the entire system equilibrated. You can do this by running Molecular Dynamics on the system. You can then add a solute

molecule to the system; you will probably want to delete one or more solvent molecules to make space for it, as HyperChem does with its standard water boxes. An equilibrated box of solvent molecules can be saved as a file, and re-used as you wish.

## Restraints

Use this menu item to add harmonic restoring forces to atoms, interatomic distances, angles and torsion angles during a molecular mechanics or quantum-mechanical calculation. You must first set up the atoms that you want to restrain as named selections (see “Name Selection” on page 237). A named selection of one atom can be tethered to a point. Named selections of two, three or four atoms can be restrained to specific atomic separations, angles or torsions respectively.

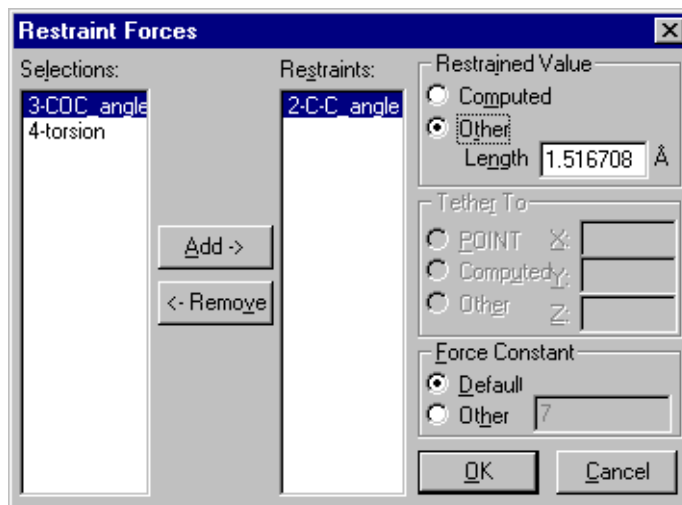
### Restraint Forces Dialog Box

Use this dialog box to restrain selected atomic geometries during a geometry optimization. These restraints apply to tethering (one atom), atomic distances (two atoms), angles (three atoms), and torsion angles (four atoms).

In this dialog box, you must give the Restrained Value (length, angle, or torsion angle) for a restraint and a harmonic Force Constant. To tether an atom, you must specify the value that you want the atom restrained to (Tether to).

HyperChem stores the restraints along with the molecular system in a HIN file.

**Caution:** If you apply a Restraint to your system, then any energy calculations that you perform will include contributions from it that are not part of the force field. To get an accurate calculation of the energy of your system, you should remove all applied restraints and calculate a single-point energy. You can do this with a structure that was optimized with restraints, to determine the energy of the system in its current configuration.



- Selections** Lists the named selections for the molecular system in the workspace. The selections that appear here include only 1, 2, 3, or 4 atom selections. The selection names have a prefix that shows the number of atoms in the selection. For example, 4-torsion means that the named selection, torsion, contains four atoms.
- Restrains** Lists the named selections that you chose from the Selections list. These provide restraints during a molecular mechanics or semi-empirical (except Extended Hückel) or *ab initio* or DFT calculation.
- Add** Moves highlighted name from the Selections list to the Restrains list.
- Remove** Removes highlighted name from the Restrains list to the Selections list.
- Restrained Value** Shows the restrained value for the named distance, angle or torsion selection highlighted in the Restrains list.
- Computed** Uses the current value for the selected atoms and shows this value in the text box below.

Other	Enter a value for Length (in Ångstroms), Angle, or Torsion (in degrees). The value should be greater than zero for Length.
Tether to	Shows the location that the named selection (one atom) highlighted in the Restraints list will be tethered to.
POINT	Active only if you have defined a named selection called POINT (see page 237). Select this to tether the named atom to the Cartesian coordinates of the center of mass of the atoms used to define POINT. When selected, the Cartesian coordinates of POINT are shown in the X, Y and Z text boxes.
Computed	Uses the current Cartesian coordinates for the named atom and shows the coordinates in the X, Y and Z text boxes.
Other	Enter the Cartesian coordinates for the location that you want to tether the named atom to.
Force Constant	This constant regulates the strength of the restraint. Larger values give a stronger restraint. The Default values are appropriate for molecular dynamics calculations. Much larger values (for example, $10^5$ ) are appropriate for geometry optimization, but these values might require more optimization cycles.
Default	The defaults are Length, $7 \text{ kcal mol}^{-1}\text{Å}^{-2}$ ; Angle, $125 \text{ kcal mol}^{-1}\text{degree}^{-2}$ ; and Torsion, $16 \text{ kcal mol}^{-1}\text{degree}^{-2}$ .
Other	Enter a number for the force constant.

## Setting Up Restraints

To set up restraints for a molecular mechanics calculation:

1. Turn on Multiple Selections on the Select menu.
2. With the Selection tool, select 1, 2, 3, or 4 atoms that define an atom, length, angle, or torsion angle that you want to restrain

3. Choose Name Selection on the Select menu. Enter a name for the selected atom(s).
4. Repeat steps 2 and 3 as needed.
5. Choose Restraints on the Setup menu. The Restraint Forces dialog box appears.
6. Choose your restraints (see the next section).
7. L-click on OK.

## Applying Restraints

To use selections for restraints:

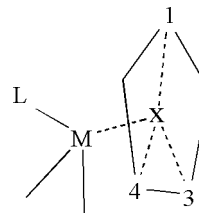
- **Choosing a name:** L-click on a name (for example, 2-Distance1). HyperChem highlights the name, indicating an active choice.
- **Canceling a choice:** L-click on a highlighted name.
- **Moving a choice to the Restraints list:** L-click on Add. Active choices (highlighted) in the Selections list move to the Restraints list.
- **Moving a choice to the Selections list:** L-click on Remove. Active choices in the Restraints list (highlighted) move to the Selections list.

## “Dummy atoms” and Conformational Restraint

You can apply Restraints to atoms that are not connected to a structure. If an atom does not interact with a structure apart from applied restraint forces, you can use it as a “dummy” atom or pseudo-atom to force a structure to maintain a particular conformation. For example, an uncharged unbonded atom whose van der Waals  $\epsilon$  parameter has been set to 0 has no interaction with other atoms in a molecular mechanics calculation; a “lone pair” atom does not interact in a semi-empirical or *ab initio* or DFT calculation. You could, for example, add a “lone pair” near the middle of a benzene ring, and restrain several C–lone-pair–C angles to be 120°. This forces the lone pair to stay at the center of the ring. Then you can add restraints involving a ring atom, the lone pair, and other atoms, to control the orientation of the benzene ring.



For a five-membered ring (see right), one might restrain angles 1-x-3 and 1-x-4 to  $144^\circ$  and angle 3-x-4 to  $72^\circ$ . When the dummy atom x is in the proper location, there are no net forces acting on the ring atoms, so the ring is not distorted. One could then restrain the distance x-M, or one of the angles 1-x-M or x-M-L, or the torsion angle 1-x-M-L.



## Set Velocity . . .

Use Set Velocity to interrogate current values of velocity and specify the velocities of the selected atoms (or all atoms, if none are selected).

### Get and Set Velocity Dialog Box

Property	Value
Average Velocity Magnitude (Å/ps)	8.91043
Average Velocity Direction Cosines (X, Y, Z)	0.765, -0.210, -0.609
Velocity of Center of Mass Magnitude (Å/ps)	5.117259
Velocity of Center of Mass Direction Cosines (X, Y, Z)	0.815, -0.236, -0.529
Angular Velocity about Center of Mass Magnitude (rad/ps)	16.80705
Angular Velocity about Center of Mass Direction Cosines (X, Y, Z)	-0.289, 0.950, -0.120
Trans. Energy (Kcal)	0.9409932
Rot. Energy (Kcal)	3.081889
Quench Translation of Center of Mass	<input type="checkbox"/>
Quench Rotation about Center of Mass	<input type="checkbox"/>
Set New Values for Selection	<input type="checkbox"/>
New Values Magnitude (Å/ps)	10
New Values Direction (Relative components)	X: 1.000, Y: 0.000, Z: 0.000

The value in this dialog box that show you current conditions are:

**Average Velocity** Shows the current average vector (magnitude and direction cosines) of the selected atoms

**Velocity of Center of Mass** Shows the current center-of-mass velocity vector (magnitude and direction cosines) of the selected atoms

**Angular Velocity about Center of Mass** Shows the current angular velocity about the center of mass vector (magnitude and direction cosines) of the selected atoms

**Translational Energy** Shows the current total translational energy of the selected atoms

**Rot. Energy** Shows the current total rotational energy of the selected atoms

The values in this dialog box that allow you to change velocities of selected atoms are:

**Direction** Input the desired direction to be set for the atoms representing the current selection

**At POINT** Sets the direction of motion towards the selection named POINT

**Along LINE** Sets the direction of motion along the vector of the selection named LINE

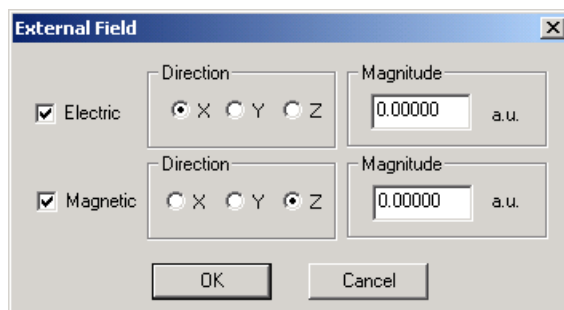
**Relative Components** Sets the direction of motion as specified by the relative x, y, and z components

**Quench Translation of Center of Mass** Modify velocities of selected atoms so as to quench all translation of the center of mass of the selected atoms.

**Quench Rotation of Center of Mass** Modify velocities of selected atoms so as to quench all rotation about the center of mass of the selected atoms.

## Applying an Electric or Magnetic Field

To apply an electric or magnetic field to the workspace, one simply clicks on the menu item <Setup/Set Finite Field...> to bring up the following dialog box,



The applied electric or magnetic field is a uniform or homogeneous one that can be applied along one of the three molecular axes, either the X axis, the Y axis or the Z axis. Prior to applying the field it may be desirable to orient the molecular system along one of these axes using the <Edit/Align Molecules...> menu item. To be clear that the screen coordinates correspond to the the molecular coordinates you might wish to align the viewer along the Z axis prior to aligning the molecule.

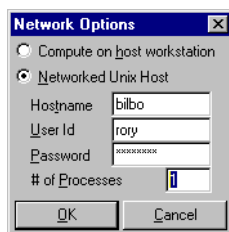
To return to normal operation, the check boxes should be unchecked. Not doing so will keep applying the electric or magnetic field with no obvious visual indication to the user.

The electric field can be applied to the workspace for any quantum mechanical engine, i.e. semi-empirical, *ab initio*, or DFT. However, magnetic fields can only be applied to a molecule using the TNDO semi-empirical method. Magnetic fields for *ab initio* or DFT computations (with gauge-invariant orbitals) would be useful but require considerably more effort. TNDO, however, is potentially the most accurate semi-empirical method and a good vehicle for exploring new technologies.

## Network . . .

Use Network to specify a remote server to perform molecular mechanics and quantum mechanical calculations.

## Network Options Dialog Box



<b>Compute on host workstation</b>	Calculations are performed on the local workstation.
<b>Networked Unix Host</b>	Calculations are performed on a remote server.
<b>Host name</b>	Enter the server name.
<b>Userid</b>	Enter your login id on the remote server.
<b>Password</b>	Enter your login password on the remote server. (What you type will not be displayed.)
<b>Number of Processes</b>	Number of parallel processes to run. Ignored in the current version of HyperChem, which will use '1' regardless of the value entered.

## Edit Parameters

Use Edit Parameters to interactively see the parameters being used in your calculations and to potentially set new values for (i.e. edit) a parameter. This menu item is enabled only for molecular mechanics and semi-empirical quantum mechanical methods but not for ab initio or DFT.

Depending upon whether one, two, three, or four atoms are selected, a dialog box will show the parameters for the selected atom (type or atomic number), bond (two types), angle (three types), or torsion (four types).

If you edit a parameter associated with certain types or atomic numbers, the new value will apply to all types or atomic numbers with the same value. thus, for example, you might select a specific carbon atom and change its parameters. These changes apply to all carbon atoms. You cannot change the parameters for only a single

atom but rather the changes apply to all identical atomic numbers (semi-empirical quantum mechanics) or types (molecular mechanics plus TNDO).

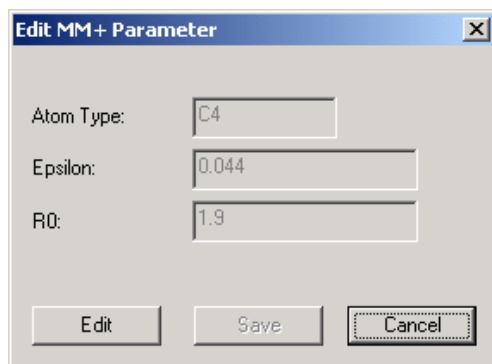
The information you see will be determined by the current computational method and by whether it is atom, bond, angle, or torsion information.

**Note:** Prior to editing parameters, it is important to copy your parameter set so as to retain an original unedited copy. If you lose your original parameters, you may have to reinstall HyperChem to get them back. The Copy button in the Select *Force Field* Parameter Set dialog box described below assists in this process.

**Caution:** If you edit parameters, HyperChem automatically creates a new \*.PAR file for you. Since HyperChem only uses PAR files in actual calculations, be sure your parameters files have been compiled into PAR files if you edit parameter files directly.

## Atom Parameters

Selecting an atom and molecular mechanics will tell you about the non-bonded parameters:



You must click on the Edit button in order to edit values. Click on Save to adopt the new value or Cancel to leave the values alone.

Selecting an atom and semi-empirical quantum mechanics will tell you about the semi-empirical parameters for that atomic number:

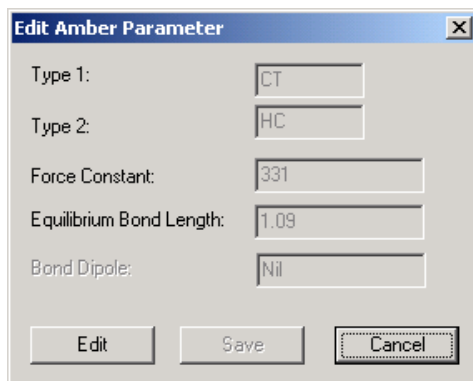
Edit CNDO/INDO Parameter (Atomic Number = 6)			
S exponent:	1.625	Beta:	21
P Exponent:	1.625	F2:	0.17372
D Exponent:	1.625	G1:	0.267708
-1/2 (Is + As):	-14.051	Atom Energy (CNDO):	-167.7561
-1/2 (Ip + Ap):	-5.572	Atom Energy (INDO):	-161.4964
-1/2 (Id + Ad):	0	Heat of Formation:	170.89
Edit		Save	Cancel

The information you obtain will differ depending upon the particular semi-empirical method chosen:

Edit PM3 Parameter (Atomic Number = 6)					
S exponent:	1.565085	a (core-core):	2.707807	Gss:	11.20071
P Exponent:	1.842345	D1:	0.8332396	Gsp:	10.26503
D1 Exponent:	0	D2:	0.664775	Gpp:	10.79629
Uss:	-47.27032	Beta-S:	-11.91002	Gp2:	9.042566
Upp:	-36.26692	Beta-P:	-9.802755	Hsp:	2.29098
Udd:	0	Beta-D:	0	AM(monopole):	0.4116394
Energy Of Atom:	-111.23	Heat of Formation:	170.89	AD(dipole):	0.5885862
				AQ(quadupole):	0.7647667
Edit		Save	Cancel		

## Bond Parameters

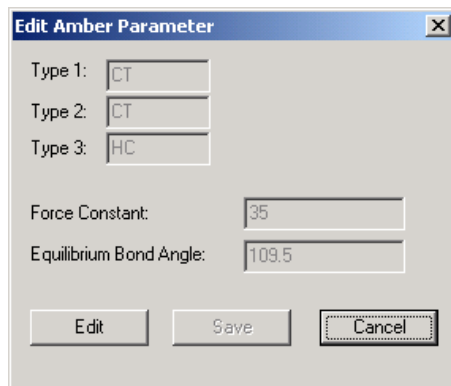
If you select a bond, then the parameters shown are the bonding parameters for the particular molecular mechanics method:



In this case, Amber has no bond dipole parameters, unlike MM+, so that this field is grayed.

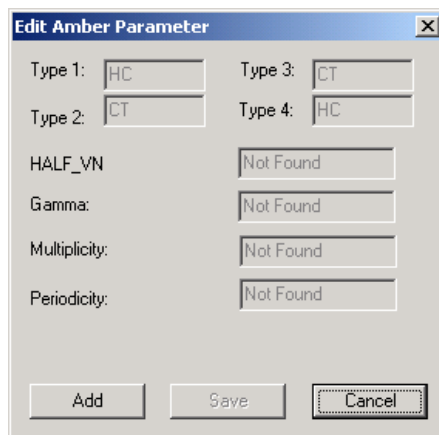
## Angle Parameters

The display of angle parameters (ethane) is as follows:



## Torsion Parameters

The display of torsion parameters is as follows:



In this case, Amber has no explicit HC-CT-CT-HC parameter for ethane (It does, however, have a more generic \*-CT-CT-\* parameter that will be used. These are not currently reported.

## Select Parameter Set

Use Select Parameter Set to choose a set of constants (parameter set) for a chosen force field or for TNDO. You see a Select *Force Field* Parameter Set dialog box with a list of choices. *Force Field* can be MM+, AMBER, BIO+, OPLS, or TNDO depending on your choice in the Molecular Mechanics Force Field or Semi-empirical Method dialog box (see page 290).

A “parameter set” is a set of files that includes information about bond lengths and angles, torsional angles, and electrostatic interactions. The exact contents are different for each force field (see page 661). A force field can have more than one parameter set.

You receive HyperChem with several standard sets of parameter files. These are text or database files with *.txt* or *.dbf* extensions (see “Parameter Files” on page 660). HyperChem uses compiled, binary versions of these standard parameter files (see page 353), with the extension *.par*. Each PAR file, for example *amber2.par*, includes all information from the parameter files *amber\*.txt* or *amber\*.dbf*.

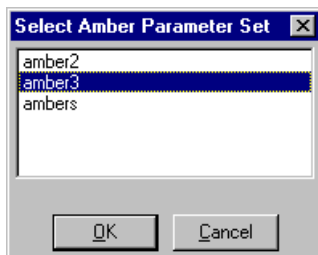
Before you use a new or modified parameter set for a molecular mechanics calculation, choose Compile Parameter File on the Setup menu (see “Compile Parameter File” later). This converts the parameter files to one binary PAR file.



**Note:** The Registry lists files for each force field that make up a parameter set. The parameter files are in the directory specified by the ChemParmPath setting in the File/Preferences/Path property sheet.

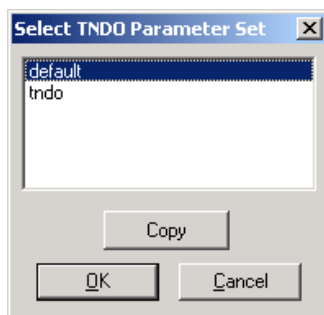
To modify existing parameter files or to set up alternate parameter files, see “Force Field Parameters” on page 659.

### **Force field Parameters Dialog Box**



Left-click on the parameter set that you want to use for a force field.

With Release 7 and the TNDO method, it is possible to have a variety of parameter sets for a semi-empirical method and the above dialog box has been extended to the TNDO method.



For TNDO, it is possible to copy a complete parameter set in a single operation by just pushing the Copy button. This makes it easy to back up your original parameters prior to editing them with the Edit Parameters menu item.

## Compile Parameter File

Compile Parameter File combines and converts a set of parameter files (chosen with Select Parameter Set, described in the previous section) to one binary file (with the extension *.par*) that HyperChem needs for molecular mechanics calculations. The Registry (see page 743) lists the parameter sets available to compile into PAR files and specifies the use of text or dBase<sup>®</sup>-type files.

Each force field uses a different PAR file. For each force field, you can have more than one PAR file to choose from. Use Select Parameter Set (see the previous section) to choose a PAR file. The uncompiled parameter files used for each PAR file (or parameter set) are defined by settings in the Registry.

Uncompiled parameter files can be either text files (*.txt*) or dBase type files (*.dbf*). You can create or modify these files with either a text editor (like Windows Notepad, for TXT files) or a database program (like dBase, for DBF files). For more information about uncompiled parameter files, see “Force Field Parameters” on page 659.

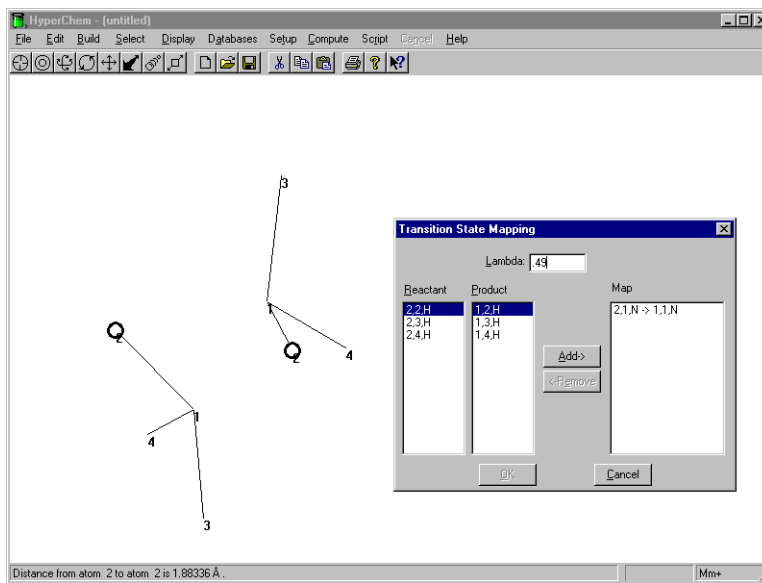
*Note:* With Release 7 and beyond, DBF files are no longer being updated nor considered current. For the latest parameters, always uses TXT files for parameters. DBF files were redundant and may be eliminated in future releases. If you need DBF files for some reason, Microsoft Excel can convert parameter TXT files to DBF.

## Reaction Map . . .

After you have defined the named selections REACTANT and PRODUCT (see “Name Selection” on page 237), you can use this dialog box to set up a starting point for a Synchronous Transit transition state search (see “Transition State” on page 386). The menu item is gray and unavailable unless both REACTANT and PRODUCT have been defined and contain the same numbers of atoms. If REACTANT and PRODUCT do not contain the same numbers of the same kinds of atoms, you will not be able to map all the atoms, and you will have to exit this dialog box with Cancel.

*Important:* For good results in a Synchronous Transit transition state search, both the reactants and products should be minimized (separately) to give local minima before the reaction mapping is done. One good procedure for doing this is to set up the reactant state, minimize it, and save it as a file; then rearrange the atoms to

the product state, and minimize again; then use Merge (see page 82) to add the reactant state back to the workspace.



### Lambda

This value describes the position of the initial guess of the transition state along the reaction coordinate between the reactant configuration and the product configuration. Lambda must be between 0 (the reactant state) and 1 (the product state).

*Note:* For symmetrical transition states, a lambda value of 0.5 may give poor results because the search algorithm looks for an optimal saddle point in the potential energy surface, but the perfectly symmetrical starting point for the search is already close to a saddle point. For such symmetrical transition states, a lambda value of 0.49 or 0.51 will probably give better results.

### Reactant

This list box shows the atoms of the Reactant named selection that have not yet been matched with corresponding Product atoms. When you select one of the Reactant atoms from this list, it is selected in the workspace as a visual aid. Atoms are listed as (molecule number, atom number, element).

<b>Product</b>	This list box shows the atoms of the Product named selection that have not yet been matched with corresponding Reactant atoms. When you select one of the Product atoms from this list, it is selected in the workspace as a visual aid. Atoms are listed as (molecule number, atom number, element).
<b>Map</b>	This list box shows the currently-defined pairs of reactant and product atoms. In principle, as the configuration of atoms changes from the Reactant state to the Product state, each reactant atom “becomes” the corresponding product atom.
<b>Add</b>	This button becomes active when an atom has been selected from the Reactant list box and an atom has been selected from the Product list box, and they are both the same element. It allows you to move the selected atoms as a matched set to the Map list box.
<b>Remove</b>	This button becomes active when you select a reactant/product atom pair in the Map list box. It allows you to unpair the atoms, and returns them to the Reactant and Product list boxes.
<b>OK</b>	This button becomes active only when all of the reactant and product atoms have been mapped — that is, when the Reactant and Product list boxes are empty and all of the atoms are in the Map list box. It sets up the atom pairs for use in the Synchronous Transit transition state search. The Product atoms are deleted (along with any information attached to them, such as named selections and charges) and the Reactant atoms are repositioned according to the coordinates from the interpolation between the two sets.
<b>Cancel</b>	This aborts the mapping procedure and returns you to the HyperChem window. All mapping information will be lost; if you open the Reaction Map dialog box again, all of the atoms will have been returned to the Reactant and Product list boxes.



# Compute Menu

The Compute menu directs chemical calculations using the molecular system in the workspace and the method you choose on the Setup menu. You can use the MM+, AMBER, BIO+ (CHARMM), and OPLS force fields and the various semi-empirical and *ab initio* or DFT quantum mechanical methods for calculations.

Using your choice of calculation method in the Setup menu, HyperChem automatically carries out a calculation.

The Compute menu has these items:

Single Point	Calculates the total energy and the root-mean-square (RMS) gradient for selected atoms or, if there is no selection, for all atoms. For a semi-empirical calculation, Single Point also calculates wave functions for the selected atoms
Geometry Optimization . . . <sup>a</sup>	Calculates an optimum molecular structure (lowest energy and smallest RMS gradient) for selected atoms or, if there is no selection, for all atoms. Use either a molecular mechanics or semi-empirical or <i>ab initio</i> or DFT method
Molecular Dynamics . . . <sup>a</sup>	Calculates the motion of selected atoms or all atoms in a molecular system, over picosecond time intervals. Demonstrates stable conformations, transition states, and thermodynamic properties. Use either a molecular mechanics or semi-empirical or <i>ab initio</i> or DFT method. Can use periodic boundaries with explicit solvent molecules
Langevin Dynamics . . .	Calculates the motion of selected atoms or all atoms in a molecular system, over picosecond time intervals. Demonstrates stable conformations, transition states, and thermodynamic properties. Use either a molecular mechanics or semi-empirical or <i>ab initio</i> or DFT method. Uses frictional effects to simulate the presence of a solvent

<b>Monte Carlo . . .</b>	Calculates ensemble averages for selected atoms or all atoms in a molecular system. Use either a molecular mechanics or semi-empirical or <i>ab initio</i> or DFT method
<b>Vibrations</b>	Calculates the vibrational motions of selected atoms or all atoms in a molecular system. Use an <i>ab initio</i> or DFT or semi-empirical method other than Extended Hückel
<b>Transition State . . .</b>	Searches for transition states for the current system, or for a set of reactant and product atoms that have been defined with <b>Setup/Reaction Map</b> . Use an <i>ab initio</i> or DFT or semi-empirical method other than Extended Hückel
<b>Invoke NMR...</b>	Invokes the HyperNMR module for calculating shielding and coupling constants and possibly the spectra
<b>Properties</b>	Displays the properties of selected atoms or bonds or molecules derived from computations on the molecule
<b>Plot Molecular Graphs . . .</b>	Use this to choose to display the electrostatic potential, total spin density, or total charge density after a semi-empirical or <i>ab initio</i> or DFT calculation <sup>b</sup>
<b>Orbitals . . .</b>	Use this to choose to display orbitals and their energy levels after a semi-empirical or <i>ab initio</i> or DFT calculation <sup>b</sup>
<b>Vibrational Spectrum . . .</b>	Use this to analyze and display the vibrational frequencies and normal modes calculated by a Vibrations calculation <sup>b</sup>
<b>Electronic Spectrum . . .</b>	Use this to analyze and display the ultraviolet-visible spectra calculated from a Single Point calculation using singly-excited configuration interaction <sup>b</sup>
<b>Potential . . .<sup>c</sup></b>	Plots 1D and 2D potential energy curves

---

Conformational Search...	Finds low energy conformations of molecular systems by varying user-specified dihedral angles.
QSAR Properties...	Allows several properties generally used in QSAR studies to be calculated.

---

- These items are gray and unavailable when you choose the Extended Hückel method for a calculation or when CI is turned on.
- This item is gray and unavailable until after you have performed a semi-empirical or *ab initio* or DFT calculation on the current system.
- This item is gray until you have made the proper selection or defined the proper PLOT1, PLOT2 selections.

Many chemical calculations are time-consuming. Once you start a calculation, you can stop it before completion by L-clicking on Cancel on the menu bar or by pressing [Esc].

**Important:** You can store atomic charge information from a calculation by saving the molecular system in a HIN file. Use Save or Save As on the File menu. To store the energy or gradient values from a calculation, use the log file (see “Start Log . . .” on page 90). The QuantumPrintLevel and MechanicsPrintLevel settings in the Start Log dialog box affect the amount of information stored in the log file. To save results from quantum calculations, use File/Export. You can also record an image of the molecular system by choosing Print on the File menu or Setup Image in File/Preferences, and then choosing Copy Image on the Edit menu.

## Warning Messages

When HyperChem begins a calculation using a molecular mechanics force field and cannot find appropriate parameters (see “Parameter Files” on page 660) for the molecular system, it may show warning messages in a dialog box that tell you the type of force constant that is missing and the atoms involved. You have these choices:

Abort	Stops the calculation.
Continue	Accepts the warning and proceeds to the next warning message or begins the calculation. With molecular mechanics force fields, HyperChem attempts to supply force default constants. Warning messages may show you the progress of this process. These warning messages go to a log file, if active.



**Ignore** HyperChem begins the calculation without showing you additional warning messages. A calculation assigns default force constants. Warning messages go to a log file, if active.

You can eliminate these warnings or send them only to a log file by setting `WarningType` in the Registry (see page 759).

## Single Point

A single point calculation determines the total energy and RMS gradient of a molecular system or of selected atoms. With a semi-empirical or *ab initio* or DFT method, Single Point also calculates the electron and charge distribution in the system. Single point means the calculation is for the present molecular conformation, a single point on the potential energy surface for the molecular system. Certain results of single point calculations are available for display via the Properties... menu item. At the end of a calculation, HyperChem reports values for Energy and Gradient on the status line:

<b>Energy</b>	The total energy (in kcal/mol).
<b>Gradient</b>	The RMS value of all forces on all atoms. The gradient, or force, is the rate of change (first derivative) of the total energy with respect to displacement in the x, y, or z direction. Since the forces are a derivative of energy with respect to atomic position, the units are kcal/mol/Ångstrom. A value close to zero indicates a molecular system close to a conformation with a minimal energy. This minimum is usually one of several possible low energy conformations (local minima) and not necessarily the lowest possible state (global minimum).

**Caution:** Since each calculation method uses different assumptions, you should compare numerical results from the same method only.

**Caution:** If you apply a Restraint to your system, then any energy calculations that you perform will include contributions from it that are not part of the force field. To get an accurate calculation of the energy of your system, you should remove all applied restraints and calculate a single-point energy.

## Using a Molecular Mechanics Method

If you are doing a molecular mechanics calculation on a selected set of atoms in a molecular system, HyperChem calculates the interactions between the selected atoms, between selected atoms and the set of unselected atoms, but not among the unselected atoms.

To use a molecular mechanics method:

1. Select the atoms you want to include in the calculation, or deselect all atoms to compute the energy of the whole molecular system.
2. Optionally you can choose Types in the Labels dialog box (see page 281) to be certain that each atom has an atom type.
3. Choose Molecular Mechanics on the Setup menu.
4. From the Molecular Mechanics Force Field dialog box, choose a force field. L-click on Options to change the way the force field works. See page 290.
5. Choose Single Point on the Compute menu.

## Using a Semi-empirical or *Ab Initio* or DFT Method

If you are using a quantum-mechanical method for the single point calculation, HyperChem determines the distribution of electrons along with the energy and gradient values. You can plot these results on top of the molecular system in the workspace using the Plot Molecular Graphs menu item. You can inspect the individual molecular orbitals using the Orbitals menu item, including display of orbital isosurfaces.

If you want a single point calculation for a selected part of a molecule and you are using a semi-empirical method, first use Extend to  $sp^3$  on the Select menu. This option selects additional atoms, in all directions, until it reaches  $sp^3$ - $sp^3$  single bonds or singly bonded atoms (see page 244). HyperChem incorporates the effects of all unselected atoms (their charges) into an electrostatic potential and uses this (as an electric field effect) in the calculation (see "Mixed Mode Calculations" on page 311). *Ab initio* or DFT calculations can work only on complete molecules, though you can select one or more molecules from the workspace and omit others from the calculations.

To perform a single point calculation:

1. Select atoms for the computation, or deselect all atoms for the whole molecular system.
2. Choose Semi-empirical or Ab Initio or Density Functional on the Setup menu.
3. From the Semi-empirical Method or Ab Initio Method or Density Functional Method dialog box, choose a method. L-click on Options or other buttons to change the way the calculation method works.
4. Choose Single Point on the Compute menu.

## Geometry Optimization

Geometry Optimization calculates and displays a structure with a minimum energy and minimal atomic forces (Gradient). This is usually a local minimum. You can do this calculation for selected atoms or, if there is no selection, for all atoms in a molecular system. Use either a molecular mechanics or quantum-mechanical method.

If you select atoms from the molecular system, unselected atoms do not move (no geometry optimization), but the calculation uses the forces that unselected atoms exert on selected atoms. In a semi-empirical calculation with selected atoms, HyperChem treats only selected atoms quantum mechanically (see “Extend to sp<sup>3</sup>” on page 244).

**Note:** To save the results of these calculations, choose Start Log on the File menu. You can also record an image of the molecular system by choosing Print on the File menu or Setup Image from File/Preferences, then choosing Copy Image on the Edit menu.

For more information about this calculation and its applications, see *HyperChem Computational Chemistry*, Practical Guide.

To perform a geometry optimization calculation:

1. Select atoms for the computation, or deselect all atoms to optimize the whole molecular system.
2. Choose Molecular Mechanics or Semi-empirical or Ab Initio or DFT on the Setup menu. Choose a calculation method or basis set, etc. from the dialog box(s) that appears next Select OK.

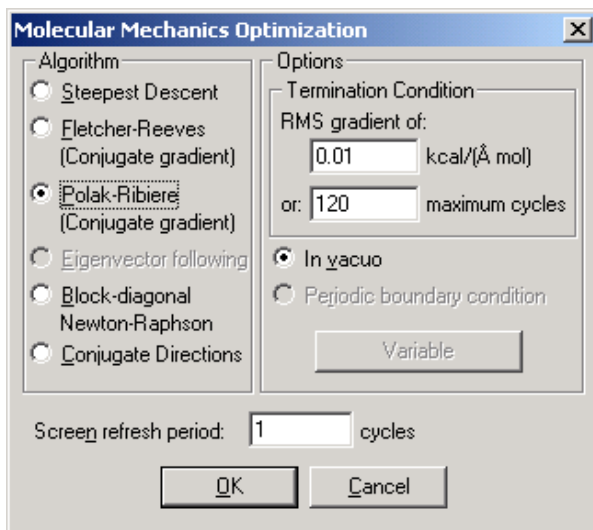
3. Choose Geometry Optimization on the Compute menu.
4. Choose an algorithm in the Optimization dialog box. You can also choose Options for the calculation or the variable to be optimized for the Conjugate Direction algorithm. L-click OK.

During a geometry optimization calculation, HyperChem shows you small adjustments to the molecules in the workspace. The status line reports this information:

Energy	The total energy (in kcal/mol) relative to the same atoms that are not interacting.
Gradient	The RMS value of forces on atoms. The gradient, or force, is the rate of change (first derivative) of the total energy with respect to displacement in the x, y, or z direction. Since the forces are a derivative of energy with respect to atomic position, the units are kcal/(mol·Ångstrom). A value close to zero indicates a molecular system close to a conformation with a minimal energy. This minimum is usually one of several possible low energy conformations (local minima) and not necessarily the lowest possible state (global minimum).
Converged	HyperChem reports YES when the calculation reaches the gradient criterion you set in the Optimization dialog box (see the next section) for reaching a minimum.
cycles	The number of search directions (see the next section) in reaching a minimum.
points	The number of single point calculations (energy and gradient) performed while calculating a minimum structure.
iter	For a semi-empirical or <i>ab initio</i> or DFT calculation: The iteration number of the SCF calculation.
diff	For a semi-empirical or <i>ab initio</i> or DFT calculation: The difference in energy (kcal/mol) between the results of the last two SCF cycles.

## Molecular Mechanics/Semi-empirical/Ab Initio/DFT Optimization Dialog Box

You see this dialog box if you choose Geometry Optimization on the Compute menu. The title bar contains Molecular Mechanics Optimization or Semi-empirical Optimization or Ab Initio Optimization, depending on the calculation method you are using (see page 289, page 296, and page 311).



**Algorithm** HyperChem uses these algorithms to calculate, for the molecular system or for selected atoms, a geometry with a minimum potential energy. These methods attempt to lower the energy of a molecular system by adjusting its geometry.

**Steepest Descent** Moves directly down the steepest slope (of interatomic forces) on the potential energy surface. If the energy of the molecular system decreases, it continues in the same direction, but by a larger step (by an increment of 1.2 times larger). If the energy increases after a step, the rate of adjusting the geometry decreases: the next step size is half of the last step.

This method makes limited changes to the molecular structure. It is useful for quickly correcting bad starting geometries or for removing bad contacts; it is most effective when the molecular system is far from a minimum. The calculation does not readily converge and can oscillate.

**Fletcher-Reeves**

This is a conjugate gradient method, which means that it chooses a descent direction (to lower energy) by considering the current gradient, its conjugate (initially the negative of the current gradient), and the gradient for the previous step. For each descent direction, a one-dimensional search attempts to locate a minimum. There are at least two evaluations (points) of the energy and gradient for each direction (cycle). To reduce cumulative errors in the calculation, when the number of directions is three times the number of atoms, this procedure resets the conjugate direction to the negative of the gradient. This method has super-linear convergence; it converges better than the Steepest descent method.

**Polak-Ribiere**

This is a conjugate gradient method using one-dimensional searches (see above). It improves on the Fletcher-Reeves method by also considering the previous conjugate direction. This method does not require resetting the conjugate direction. It requires slightly more memory but tends to converge more quickly than the Fletcher-Reeves method.

**Eigenvector-Follow**

Available for semi-empirical or *ab initio* or DFT optimizations; not available for molecular mechanics optimizations. It uses eigenvectors from a diagonalized Hessian matrix (second derivatives of total energy with respect to atomic displacements) to determine the optimal directions

for atomic motion. The initial guess is computed empirically.

**Block  
Diagonal  
Newton-  
Raphson**

Available only for the MM+ force field. This is the only method offered that directly calculates second derivatives, or force constants. Second derivatives indicate the curvature of the potential energy surface and can assist in locating minima. The method computes the second derivative matrix for one atom at a time, avoiding second derivatives with respect to two atoms. It cycles through each atom until the RMS gradient for all atoms is near zero. Where atoms are in an area of negative curvature, the calculation uses a steepest-descent step. This method is efficient in locating a minimum but requires a reasonable starting geometry. The result sometimes oscillates because the calculation neglects coupling between atoms.

**Conjugate  
Directions**

Available for semi-empirical or *ab initio* or DFT optimizations; not available for molecular mechanics optimizations. It uses only the energy and not any derivatives of the energy. It can thus be used more broadly although it is not efficient compared to derivative methods. Can be used to optimize MP2 geometries or excited state CI geometries.

**Termination  
condition**

Use these text boxes to set conditions for ending a calculation. When the calculation reaches either of these criteria, the calculation ends.

**RMS gradient**

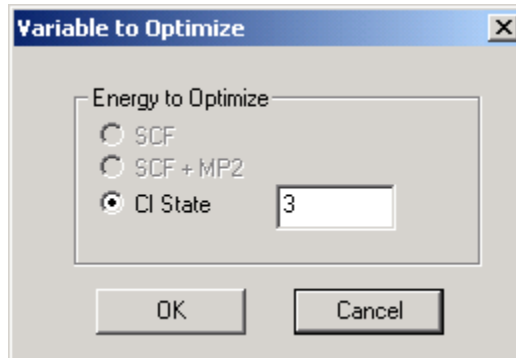
When the RMS gradient of the energy is less than this value, the calculation ends. The practical range is  $10^{-3}$  to 0.1 (the default). Values less than  $10^{-3}$  might not be possible due to the numerical round-off error. You can use values greater than 0.1 for quick, approximate calculations.

**max.  
Cycles**

If the calculation uses this number of search directions before converging, the

calculation ends. The default is 15 times the number of atoms. The practical range is 100 to 1000.

- In vacuo** This removes periodic boundary conditions from the calculation. In vacuo is the only choice if the molecular system is in a periodic box (either visible or hidden).
- Periodic boundary conditions** This uses the periodic boundary conditions already established in the molecular system (see “Periodic Box” on page 335) for the calculation. This is only available if the molecular system is in a periodic box (either visible or hidden).
- Variable** This brings up the Variable to Optimize dialog box which allows you to choose whether to optimize the SCF energy ( for comparison with equivalent result using another optimization algorithm) or the SCF + MP2 energy (*ab initio* only) or the energy of an excited state (provided Configuration Interaction has been chosen).



- Screen refresh period** This is the frequency, in optimization cycles, of showing the results of a calculation on screen. Screen refresh period can be any number of data collection periods (an integer) from 1 to 32,767.

**Note:** Changing the Screen refresh period also changes the Data collection period in the Molecular Dynamics dialog box.



## Molecular Dynamics

Molecular dynamics calculations simulate the movement of molecules. These calculations have various uses, including studying equilibrium properties (motion around a minimum on the energy surface) and kinetic behavior (motion involving change from one state to another). Equilibrium simulations (at one temperature) can predict thermodynamic properties of a molecular system. Kinetic simulations (with controlled heating and cooling) help to find an energy minimum (simulated annealing). By setting two molecules to collide, it is possible to model chemical reactions at the molecular level. (Molecular Dynamics simulations using *ab initio* methods take much longer than with molecular mechanics or semi-empirical methods, but they may model bond breaking and bond formation more accurately. See comments in *Computational Chemistry* about bond breaking and formation.)

HyperChem provides records of molecular dynamics calculations. You can take snapshots of the molecular system during a simulation and store these in a file for later analysis. You can also calculate and graph values for structural or energy variables.

Use this calculation on selected atoms or on all atoms in a molecular system. If you select atoms from the molecular system, unselected atoms do not move (no geometry optimization), but the calculation uses the forces that unselected atoms exert on selected atoms. In a semi-empirical calculation with selected atoms, HyperChem treats only selected atoms quantum mechanically (see “Extend to sp<sup>3</sup>” on page 244).

To perform a molecular dynamics calculation:

1. Select the atoms for the computation or deselect all atoms to run the dynamics calculation on the whole molecular system.
2. Choose Molecular Mechanics, Semi-empirical, or Ab Initio on the Setup menu. From the next dialog box that appears, choose the calculation method. Set up the calculation options that you wish. (In order to conserve the total energy in molecular dynamics calculations, using an *ab initio* quantum mechanical method, the gradient needs to be very accurate. Although the gradient is calculated analytically, it is a function of the SCF accuracy of a wave function. Tests for H<sub>2</sub>O show that the convergence limit needs to be at least at 10<sup>-5</sup>.)
3. Choose Molecular Dynamics on the Compute menu.

4. In the Molecular Dynamics Options dialog box, you can choose times and temperatures for heating, equilibration, and cooling. You can also choose data analysis options.
5. L-click on Proceed.

This information appears on the status line during a calculation:

Time	The total elapsed time of the molecular dynamics simulation, in picoseconds.
Total Energy	The total energy (potential and kinetic) of this molecular system.
T	The temperature of the molecular system in Kelvin.
iter	For a semi-empirical or <i>ab initio</i> calculation: The iteration number of the SCF calculation.
diff	For a semi-empirical or <i>ab initio</i> calculation: The difference in energy (kcal/mol) between the results of the last two SCF cycles.

For more information about this calculation and its applications, see *HyperChem Computational Chemistry*, Practical Guide.

## Changing the Initial Velocity for Calculations

If you want to perform a molecular dynamics simulation using a different set of initial atom velocities, you can change the value of the HyperChem variable, called dynamics-seed, that is used for the random number generator. For details see “Scripts and DDE” on page 509.

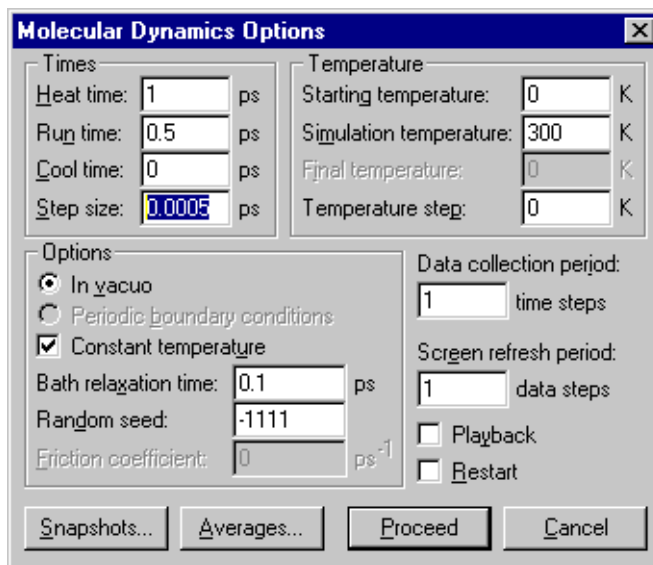
## Molecular Dynamics Options Dialog Box

Essentially the same dialog box is used for Molecular Dynamics, Langevin Dynamics, and Monte Carlo. Each of these calculations can have up to three phases: heating, running, and cooling. The running phase of a dynamics calculation may involve constant temperature control (coupling to a “water bath”) or no temperature control (“free” dynamics).

The options you choose for a molecular or Langevin dynamics calculation depend on the type of simulation. If you want to determine the thermodynamic properties of a system, the system must reach equilibrium, indicated by a stable temperature, without a

temperature bath. Reaching equilibrium might require long simulation times. For most equilibrium simulations, you should have a long simulation period to reach equilibrium, then start the simulation again (see Restart, below) to collect data.

If you want to find a minimum energy conformation, equilibrium is not required. The simulation can use high temperatures to aid in crossing barriers on the potential surface. For example, you can heat a system, maintain it at a high temperature, and then cool it.



**Times** For Molecular Dynamics and Langevin Dynamics, the lengths of time periods and the size of the time steps in the simulation. For Monte Carlo, the number of steps in each phase, and the maximum atomic movement at each step. Each time period starts and ends at a Temperature.

For Molecular Dynamics and Langevin Dynamics, the options are:

**Heat time** The time in picoseconds (ps) for changing the temperature from Starting to Simulation. Suggested range: 0 to 10 ps.

**Run time** The time in picoseconds for holding the system at the Simulation temperature. Suggested range: 0 to  $10^3$  ps.

**Cool time** The time in picoseconds for changing the temperature from Simulation to Final. Suggested range: 0 to 10 ps.

**Step size** This is the time interval, in picoseconds, between evaluations of the total energy and temperature of a molecular system. Suggested range:  $10^{-4}$  to  $10^{-3}$  ps, and greater than zero. For molecular systems with explicit hydrogens (All Atom), try  $5 \times 10^{-4}$ .

For Monte Carlo, the options are:

**Heat** The number of steps for changing the temperature from Starting to Simulation.

**Run steps** The number of steps for doing the calculation at the Simulation temperature.

**Cool** The number of steps for changing the temperature from Simulation to Final.

**Max delta** This is the maximum distance (in Å) that any atom in the system will be moved at each step in the calculation. Suggested range: 0.01 to 0.1 Å; the default is 0.05 Å.

**Note:** For dynamics calculations, step size is critical. If the simulation is unstable (atoms move erratically or bonds increase in length beyond reasonable limits), try a smaller step size. In general, the step size should be an order of magnitude smaller than the shortest period of a characteristic system oscillation. For systems including individual hydrogen atoms, use a step size of about  $5 \times 10^{-4}$  ps, or less.

**Temperature** The temperature levels for the simulation. During Heat time and Cool time, HyperChem sets the temperature by periodically adjusting velocities (see *HyperChem Computational Chemistry*, Theory and Methods). During Run time with Constant temperature on, HyperChem adjusts velocities at each time step. If Constant temperature is off, adjustment occurs once at the beginning of the simulation.

**Starting temperature** If Heat time is greater than zero, initial atomic velocities are adjusted to give this temperature. Suggested range: 0 to 300 K.

<b>Simulation temperature</b>	At the start of Run time, atomic velocities are adjusted to give this temperature. If you want initial heating in the simulation, this temperature should be larger than Starting temperature. Suggested range: 0 to 400 K.
<b>Final temperature</b>	If Cool time is greater than zero, atomic velocities are adjusted to approach this temperature at the end of Cool time. If you want cooling to occur in this simulation, this temperature should be less than Simulation temperature. Suggested range: 0 to 300 K.
<b>Temperature step</b>	This sets the size of the steps, in Kelvin, for heating or cooling. Suggested range: 0 to 100 K.
<b>Options</b>	
<b>In vacuo</b>	This removes periodic boundary conditions from the calculation. In vacuo is the only choice if the molecular system is not in a periodic box (either visible or hidden).
<b>Periodic boundary conditions</b>	This includes periodic boundary conditions in the calculation. The system must be in a defined Periodic Box (see page 335).
<b>Constant temperature</b>	Tends to regulate the temperature of the molecular system about the Simulation temperature, only during Run time. This creates a non-equilibrium condition; the simulation adjusts atomic velocities to control the temperature. The amount (severity) of temperature regulation depends on the Bath relaxation setting (below). This option is inactive for Monte Carlo.
<b>Bath relaxation</b>	A time constant (in picoseconds) for Constant temperature regulation during Run time (see above). The Bath relaxation constant affects the amount of regulation. Constants similar to the Step size (see above) give better regulation, but take the

system further from equilibrium. Larger time constants give less regulation, but are often useful for counteracting round-off errors that accumulate during the long simulations. Suggested range: Step size to 0.1 ps (the default).

Random seed	This number is the starting point for the random-number generator used for the simulations. It is used to determine the starting velocities for atoms in a Molecular Dynamics calculation if Restart is not checked, and the random motions and interactions in Monte Carlo and Langevin Dynamics. The value is saved in a HIN file. Acceptable values: integers from -32768 to 32767.
Friction coefficient	This option is used for Langevin Dynamics, and is gray and inactive for Molecular Dynamics and Monte Carlo. It describes the coefficient of friction which is used to simulate solvent interactions. Acceptable values: from 0 to infinity.
Data collection period	This is the frequency, in time steps, that the molecular dynamics simulation sends new calculated values to HyperChem. Step size, also in this dialog box, sets the length of time steps. Data collection period can be any number of time steps (an integer) from 1 to 32,767.  <i>Note:</i> Changing the Data collection period also changes the Screen refresh period in the Molecular Mechanics, Semi-empirical, and Ab Initio Optimization dialog boxes.
Screen refresh period	This is the frequency, in data steps, of showing the results of a calculation on screen. Data collection period, above, sets the length of a data step. Screen refresh period can be any number of data collection periods (an integer) from 1 to 32,767.
Playback	Check (✓) this option to run a dynamics or Monte Carlo simulation stored in a HIN and a snapshot (SNP) file. To use Playback, the molecular system

from the HIN file must be in the workspace. The molecular system is in the workspace if you just completed a dynamics or Monte Carlo calculation, or if you use Open on the File menu to display a system stored with snapshots.

With Playback on (✓), only Data collection period, Screen refresh period, Snapshots, and Averages are available in this dialog box. L-click on Snapshots to see the Molecular Dynamics Playback dialog box (see page 375). L-click on Averages to calculate averages or plot values during the playback (see “Molecular Dynamics Playback Dialog Box” on page 375).

- |                  |  |
|------------------|--|
| <b>Restart</b>   | With Restart on (✓), the calculation sets the initial velocities of atoms to previously determined values. The initial velocities are either from the last molecular dynamics calculation with the current molecular system, from a HIN file saved with velocities (see “Save File Dialog Box” on page 83), or set in the Set Velocity dialog box (see page 344). With Restart off, the calculation creates initial velocities, using a random number generator, which provide a Gaussian distribution associated with the starting temperature. |
| <b>Snapshots</b> | With Playback off, this option shows the Molecular Dynamics Snapshots dialog box (see the next section). With Playback on, Snapshots shows the Molecular Dynamics Playback dialog box (see page 375).  |
| <b>Averages</b>  | Displays the Molecular Dynamics Averages dialog box (see page 377), so that you can calculate averages and graph results of a dynamics calculation. Use this option after a calculation or playing back a snapshots file (see Playback, earlier). Averages are available only in the Molecular Dynamics Averages dialog box. Graphs appear in the Molecular Dynamics Results dialog box.   |
| <b>Proceed</b>   | Starts the dynamics or Monte Carlo calculation or starts the playback of a snapshots file.   |

## Molecular Dynamics Snapshots Dialog Box

Use this dialog box to store data into two files, a HIN file that stores the system information and a binary snapshot file (with the extension *.snp*) that cumulatively stores the system coordinates and velocities at regular intervals during a molecular dynamics calculation. You specify the name for the HIN file in a slightly modified version of the standard Save File dialog box (see “Save File Dialog Box” on page 83), the only difference being an additional Snapshot Period text box (described below) for specifying the frequency of saving snapshots. The snapshot binary file will have the same name as the HIN file that you specify, except the extension will be SNP instead of HIN. You can later open the HIN file and use Playback in the Molecular Dynamics Options dialog box (see page 369) to review the molecular dynamics simulation.

<b>Snapshot period</b>	This is the interval, in data steps, between snapshots. A data step occurs every time a calculation reports new data (see “Time Periods in a Molecular Dynamics Simulation” on page 382).
------------------------	---

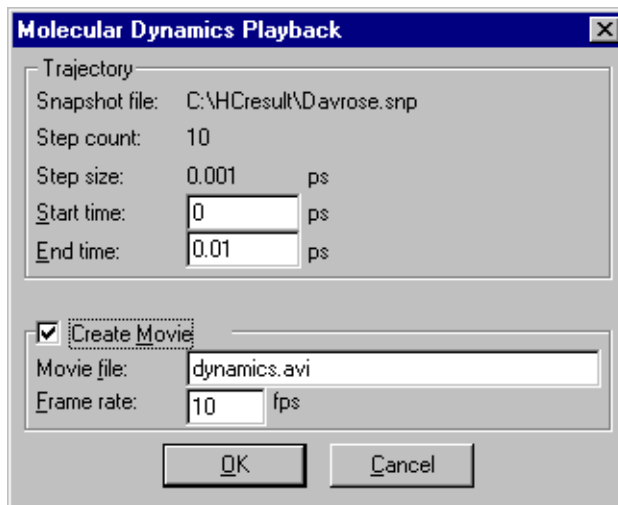
## Molecular Dynamics Playback Dialog Box

This dialog box is available only when the molecular system in the workspace includes snapshots of a previous molecular dynamics simulation. This is true if you just completed a molecular dynamics calculation and collected snapshots, or if you use Open on the File menu (see “Open” on page 78) to display a system stored with snapshots. The snapshot file contains information about the kind of calculation that created it: Molecular Dynamics, Langevin Dynamics, or Monte Carlo. It can be played back only from the playback menu of the kind of calculation that created it.

To display the dialog box:

1. L-click on Playback in the Molecular Dynamics Options dialog box.
2. L-click on Snapshots in the same dialog box.





**Trajectory** This section gives you information about the snapshot file, and allows you to specify that all or part of it should be replayed.

**Snapshot file** For display only. The path and name of the SNP file containing molecular dynamics snapshots.

**Step count** For display only. The number of snapshots stored during a simulation.

**Step size** For display only. The time interval, in picoseconds, between snapshots.

**Start time** The time in the simulation to begin the playback. Enter a number from 0.000 (the default) to End time.

**End time** The time in the simulation to end the playback. Enter a number from Start time to the default (total simulation time).

**AVI Movie** This section contains options for generating an AVI movie from a snapshot file. The AVI movie can be played back using the Multimedia Player or comparable application (different versions are available as shareware).

**Warning:** If you change the size of the HyperChem window while you are generating an AVI movie, HyperChem may crash.

Create Movie	If this option is turned on (✓), an AVI movie will be generated.
Movie file	Enter the name of the AVI file to be created in this area.
Frame rate	Enter the frames-per-second rate of the movie in this text box.

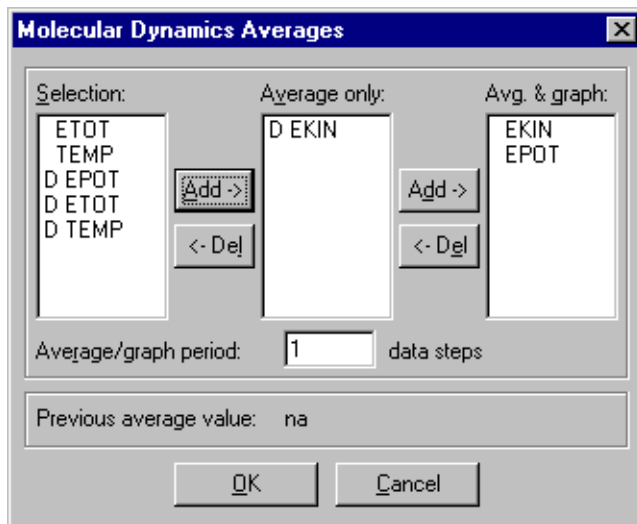
## Molecular Dynamics Averages Dialog Box

Use this dialog box to average and plot data from a molecular dynamics calculation or from the playback of a recorded calculation (see “Molecular Dynamics Playback Dialog Box” on page 375). Choose from these variables:

- Four standard measurements of energy and temperature.
- Bond lengths, bond angles, and torsion angles for named selections of 2, 3, or 4 atoms (see “Name Selection” on page 237).
- Acceptance ratio (available only for Monte Carlo simulations).
- Deviations from the mean for each of these values (with a prefix “D”).

HyperChem plots the instantaneous value of each quantity during the calculation and calculates the average of a value over the whole simulation.

You can reach this dialog box by choosing Averages in the Molecular Dynamics Options dialog box.



**Selection** This lists ten predefined choices, plus named selections, which HyperChem can plot and average:

<b>Ekin</b>	Kinetic energy (kcal/mol).
<b>EPOT</b>	Potential energy (kcal/mol).
<b>ETOT</b>	Total energy (kcal/mol).
<b>TEMP</b>	Temperature (Kelvin).
<b>ACCR</b>	Acceptance ratio (available only for Monte Carlo simulations)

**Named selections** These are interatomic distances, angles, and torsion angles for named selections of *any* 2, 3, or 4 atoms in a molecular system. These also appear in the Select dialog box (see page 233). HyperChem reports these values:

- Distance in Ångstroms between any two atoms.
- Bond angle in degrees for any three atoms.
- Torsion angle (proper or improper) in degrees for any four atoms.

D EKIN	The RMS deviation from the mean for the kinetic energy. <sup>1</sup>
D EPOT	The RMS deviation from the mean for the potential energy. <sup>1</sup>
D ETOT	The RMS deviation from the mean for the total energy. <sup>1</sup>
D TEMP	The RMS deviation from the mean for the temperature. <sup>1</sup>
ACCR	The RMS deviation from the mean for the acceptance ratio (available only for Monte Carlo simulations). <sup>1</sup>
D <i>Named selections</i>	The RMS deviation from the mean for the named selections. <sup>1</sup>

**Average only** This lists the values that HyperChem averages at the end of a molecular dynamics calculation. HyperChem saves the instantaneous values in a file with the same name as the HIN file, but with the extension .csv (see “Plot and Averages Files” on page 629). If the molecular system is not yet associated with a HIN file, HyperChem stores the values in chem.csv.

**Avg. & Graph** This lists up to four values that HyperChem averages and graphs during a molecular dynamics simulation. Graphs appear in the Molecular Dynamics Results dialog box (see page 380). HyperChem saves the instantaneous values and averages in a file with the same name as the HIN file, but with the extension .csv (see “Plot and Averages Files” on page 629). If the molecular system does not yet have a file name, HyperChem stores the values in chem.csv. The .csv file is stored in the current working directory, which you must be able to write to (e.g., not on a CD-ROM.)

1.  $Dx = \sqrt{\langle x^2 \rangle - \langle x \rangle^2} = \sqrt{\langle (x - \langle x \rangle)^2 \rangle}$ , where  $x$  = instantaneous value of EKIN, ETOT, and so on.

$$\langle x^2 \rangle = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right), \text{ and } \langle x \rangle^2 = \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2.$$

**Value** The average that appears here is for the last value that you highlighted in one of the lists (Average only or Avg. & Graph). na means no average yet available. For a value  $x$ , the average over time is

$$\langle x \rangle = \frac{1}{n} \sum_{i=1}^n x_i.$$

For the RMS deviation of  $x$ , the reported average is  $\sqrt{\langle x^2 \rangle - \langle x \rangle^2}$ .

**Average/graph period** This sets the interval for averaging data or plotting it. Enter a number from 1 to the maximum number of Data steps. A data step occurs whenever the calculation reports new data (see “Time Periods in a Molecular Dynamics Simulation” on page 382).

To choose values for averages and plots:

- **Choosing a value:** L-click on a value (for example, TEMP). A highlight appears, indicating an active choice. L-click on additional values to add them to the active choices.
- **Canceling a choice:** L-click on a value that has a highlight.
- **Moving a choice to the right:** L-click on Add. Active choices (highlighted) move to the list on the right.
- **Moving a choice to the left:** L-click on Del. Active choices (highlighted) move to the list on the left.

## Molecular Dynamics Results Dialog Box

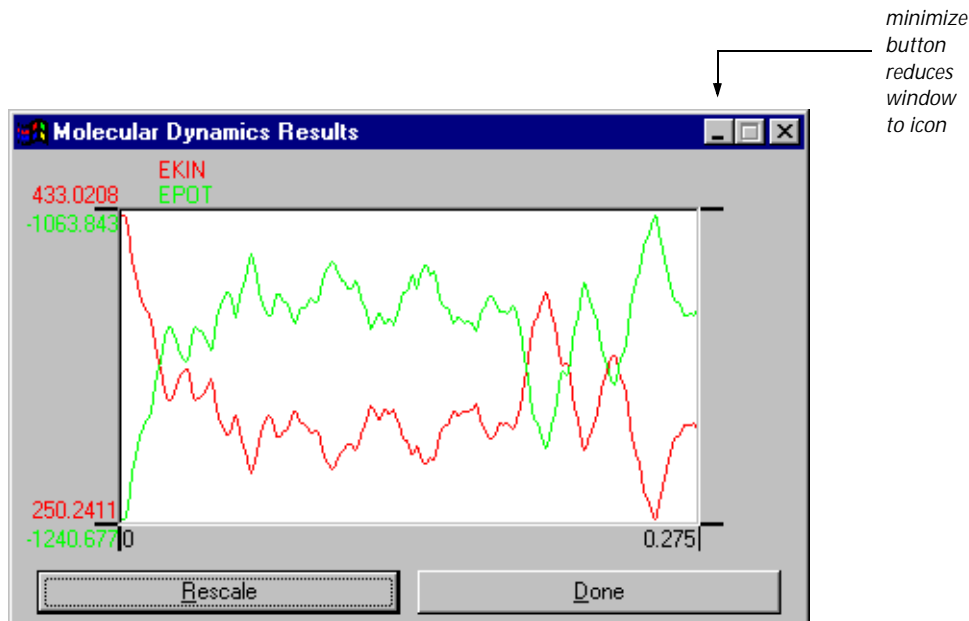
This dialog box graphs instantaneous values (either  $x$  or  $D x$ ) from a molecular dynamics simulation. A plot of  $x$  shows instantaneous values of  $x$  with time. A plot of  $D x$  shows the current value of the RMS deviation for  $x$ , with the computed value of  $D x$  converging to its final value of  $\sqrt{\langle x^2 \rangle - \langle x \rangle^2}$ . HyperChem computes the value of  $D x$  from running averages of  $x$  and  $x^2$ . A CSV file records the exact values of  $x$  and  $D x$  used for the graphs (see “Plot and Averages Files” on page 629).

The dialog box is available when you are doing a molecular dynamics simulation or are playing back a previous simulation. To see the averages of values from a simulation, you must return to the Molecular Dynamics Averages dialog box.

To activate this dialog box:

1. In the Molecular Dynamics Options dialog box (see “Molecular Dynamics Options Dialog Box” on page 369), L-click on Averages.
2. In the Molecular Dynamics Averages dialog box (see “Molecular Dynamics Averages Dialog Box” on page 377), choose the values to average and graph. You can graph up to four quantities or their deviations.
3. L-click OK to return to the Molecular Dynamics Options dialog box.
4. L-click Proceed to start the simulation or a playback of snapshots from a previous simulation.

You see the Molecular Dynamics Results dialog box on top of a HyperChem window. To temporarily remove this dialog box, L-click on the Minimize button in the upper-right corner. This turns the dialog box into an icon. Double-click on the icon to see this dialog box again.

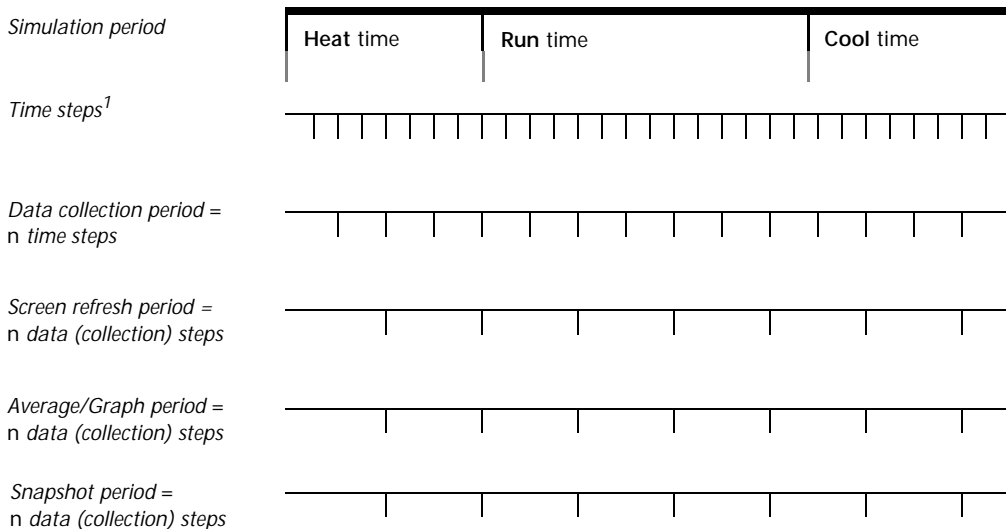


<b>Rescale</b>	L-click to change the y axis to show all data. Use this option if a plot goes off scale. Pressing the [Spacebar] is equivalent to L-clicking on Rescale.
<b>Done</b>	Ends plotting and removes this dialog box. Double-clicking on the Control-menu box is equivalent.
<b>Minimize button</b>	Reduces the dialog box to an icon.

## Time Periods in a Molecular Dynamics Simulation

Because of the frequencies of molecular motions, molecular dynamics simulations run over a picosecond time scale. For example, it takes about 4 ps at 300 K to break a hydrogen bond and make a new bond.

The total length of a simulation is the sum of Heat, Run, and Cool times, though some simulations might use only one or two of these simulation periods. HyperChem uses four timing periods during a molecular dynamics simulation:



1. To adjust the length of Time steps, set Step size in the Molecular Dynamics Options dialog box.

A molecular dynamics simulation calculates the energy and gradient of a molecular system after each time step (see *HyperChem Computational Chemistry*, Practical Guide). Step size determines the interval (period) between time steps. Timing of all other events in the simulation is related to time steps. The simulation reports information at the end of each Data collection period. HyperChem stores this data in a log file, a snapshots file, a plot (CSV) file (see “Plot and Averages Files” on page 629), or uses it for a molecular dynamics graph, if you choose those options.

## Running a Script During Molecular Dynamics

You can run a script (in the HyperChem Command Language, i.e. \*.scr) at regular intervals during a molecular dynamics run. This allows you, for example, to plot a molecular orbital as a function of time during a chemical reaction. It allows you to easily calculate correlation functions or other aspects of a trajectory because you can obtain access to all properties of the system at any time step in a trajectory. Because a Hcl script can include the command *read-tcl-script*, it is possible to run elaborate Tcl scripts at any time step of a trajectory.

A script is run by including the file *chem\_dynamics.scr* in the Program folder where chem.exe resides. It is similar in concept to chem.scr which is run at the installation of HyperChem. In the present case, chem\_dynamics.scr is run if and when you play back a molecular dynamics trajectory and the file exists. The script will not (can not!) be run during the creation of a trajectory but only during the playback of a trajectory. Thus, the script cannot alter the path of the trajectory but is rather used to evaluate properties of a given trajectory. This capability has no graphical user interface but depends upon the existence of a specific file name in a specific folder. After use, you should be sure to change the script file name or move the script file to turn off the capability.

## Langevin Dynamics

Langevin dynamics is similar in principle to regular molecular dynamics. It includes frictional effects to simulate the presence of a solvent implicitly, where regular molecular dynamics requires inclusion of explicit solvent molecules. Please see the description in *HyperChem Computational Chemistry*.



You perform Langevin Dynamics calculations with HyperChem in the same way as you do Molecular Dynamics calculations. All of the dialog boxes for Langevin Dynamics are the same as for Molecular Dynamics except that a few of the available options are different. The Langevin Dynamics Options dialog box allows you to specify a Friction coefficient which describes the effects of the simulated solvent, and a Random seed which is the starting point for the random number generator. Please see “Molecular Dynamics Options Dialog Box” on page 369, and the subsequent sections on its dialog boxes, for information about setting up Langevin Dynamics calculations.

## Monte Carlo

The Monte Carlo method samples phase space by generating random configurations from a Boltzmann distribution at a given temperature. Averages computed from a properly equilibrated Monte Carlo simulation correspond to thermodynamic ensemble averages. Thus, the Monte Carlo method can be used to find average energies and equilibrium structural properties of complex interacting systems.

A sequence of successive configurations from a Monte Carlo simulation constitutes a trajectory in phase space; with HyperChem, this trajectory may be saved and played back in the same way as a dynamics trajectory. With appropriate choices of setup parameters, the Monte Carlo method may achieve equilibration more rapidly than molecular dynamics. For some systems, then, Monte Carlo provides a more direct route to equilibrium structural and thermodynamic properties. However, these calculations can be quite long, depending upon the system studied.

You perform Monte Carlo calculations with HyperChem in the same way as you do Molecular Dynamics calculations. All of the dialog boxes for Monte Carlo are the same as for Molecular Dynamics except that a few of the available options are different. The Monte Carlo Options dialog box allows you to specify numbers of steps for running instead of time periods, and a Random seed which is the starting point for the random number generator. The Monte Carlo Averaging dialog box allows you to average (and optionally plot) the acceptance ratio and its RMS deviation, along with the other averaging values. Please see “Molecular Dynamics Options Dialog Box” on page 369, and the subsequent sections on

its dialog boxes, for information about setting up Monte Carlo calculations.

## Vibrations

Use Vibrations to compute the vibrational motions of the nuclei, and use Vibrational Spectrum to display graphically the normal modes associated with individual vibrations, as well as to display a graphical representation of an infrared (IR) vibrational spectrum. You can calculate vibrations using any of the *ab initio* or DFT or semi-empirical methods except Extended Hückel.

Use Vibrational Spectrum on the Compute menu to view the results of the computation.

Use vibrational analysis to perform the following tasks:

- Provide insight into the rigidity of the molecular framework.
- Visualize normal modes corresponding to lines in the IR spectrum.
- Help identify unknown compounds by correlating predicted versus experimental vibrational frequencies.
- Differentiate minima from saddle points on a potential energy surface.

To calculate vibrations:

1. Create or load the system that you wish to calculate vibrations for.
2. Choose a semi-empirical method other than Extended Hückel. Make sure that None is selected as the CI Method and that the Convergence limit is 0.01 or smaller.
3. Choose Geometry Optimization on the Compute menu, set the termination condition to 0.01 and choose OK to start the optimization.

**Important:** For a valid vibrational analysis the system must be optimized. Extra tight convergence criteria should be used for geometry optimization ( $<0.01$  kcal/mol Ångstrom) because the harmonic analysis is based upon a zero gradient. You must use the same quantum mechanical method for both the vibrational analysis and the geometry optimization.

4. After the calculation finishes, choose Vibrations on the Compute menu.

HyperChem computes the SCF wavefunction and evaluates the gradient analytically at the optimized geometry. The second derivatives of the energy with respect to the atomic Cartesian coordinates are computed using a finite differencing of the analytical gradients.

The evaluation of second derivatives, by finite difference of analytic gradients, is the most time consuming step. The result is a matrix of mixed partial second derivatives (force constants), which is diagonalized to yield normal modes of vibration and their corresponding energies. The status bar shows the extent to which the matrix is completed.

You may then choose Vibrational Spectrum from the Compute menu to analyze the results.

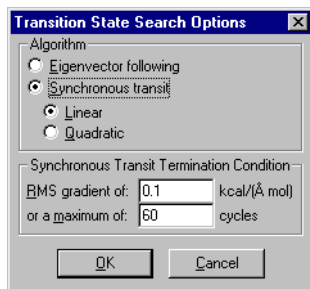
To store vibrational spectra:

If you wish to save vibrational results, see “Export . . .” on page 93. Alternately, you can run the sample script `saveir.scr` (see “Open Script” on page 106 for information on running scripts). This will create another script file that you can later open to restore the saved vibrational results into HyperChem.

## Transition State

This command allows you to find transition states of chemical systems with any kind of semi-empirical (except Extended Hückel) or *ab initio* or DFT method. Several different search algorithms are available.

The semi-empirical methods were optimized for stable structures, not for transition states. You are likely to get better results for a transition state search from an *ab initio* or DFT calculation than from a semi-empirical calculation. When you set up the options for the SCF calculation, you should specify a relatively good convergence (SCF convergence around 0.001) and a larger number of cycles than usual (perhaps 150 cycles). It is a good idea to have the convergence acceleration turned on for transition state searching.



### Eigenvector following

This method follows an eigenvector of a vibrational mode to find a transition state. If you select this option and click on “OK”, HyperChem will perform a vibrational calculation and open the Transition State Search dialog box (below) to allow you to specify which vibrational mode to follow. This method is especially useful if one of the natural vibrational modes of the system leads to a transition state (for example, the “breathing” mode of a pyramidal structure leading to its inversion).

### Synchronous transit

This method is used to search for a transition state after you have defined a “Reactant” state and a “Product” state, and have used Reaction Map (see page 353) to define a starting point for the search. From that information, HyperChem is able to determine the best eigenvector to search along without further user intervention.

### Linear

HyperChem uses a linear interpolation between the reactant and product states.

### Quadratic

HyperChem uses a quadratic interpolation between the reactant and product states.

### Synchronous Transit Termination Condition

Use these text boxes to set conditions for ending the transition state search. When the calculation reaches either of these criteria, the search ends.

### RMS gradient

When the RMS gradient of the energy is less than this value, the calculation ends. The practical range is  $10^{-2}$  to 0.1 (the default). Values less than  $10^{-2}$  might not be possible due to the numerical round-off error. You can use values greater than 0.1

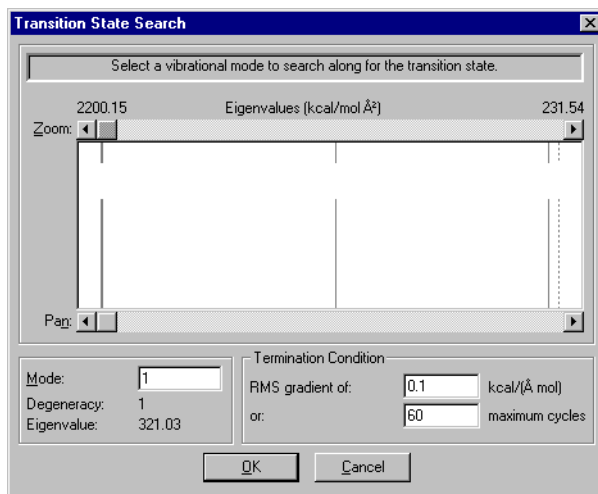
for quick, approximate calculations. If you want a very small RMS gradient criterion, you should change the options for the SCF calculation to a smaller convergence limit and a larger number of SCF cycles.

max.  
Cycles

If the calculation uses this number of search directions before converging, the calculation ends. The default is 15 times the number of atoms. The practical range is 100 to 1000.

## Transition State Search Dialog Box

This dialog box opens after you Click on OK after specifying Eigenvector Following in the Transition State Options dialog box. HyperChem performs a vibrational calculation and then displays this box.



Most of the features of this dialog box are the same as in the Vibrational Spectrum dialog box (see page 406). You can select a vibrational mode by clicking on its line, or by entering the mode number in the Mode text box. You can use the Zoom and Pan slider bars to focus on part of the spectrum at a time.

Usually, the vibrational mode with the lowest frequency (Mode #1, at the right side of the box) is the one that will lead to a transition state.

The options for Termination Condition are as described in the Transition State Options dialog box, above.

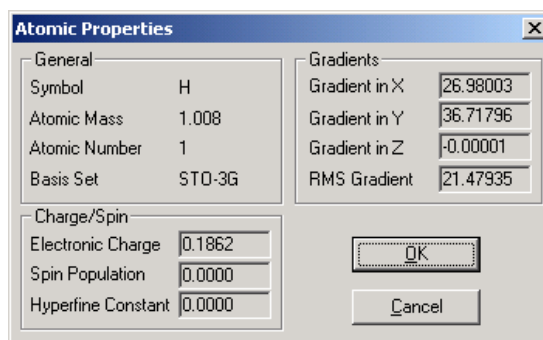
## Properties Display

HyperChem will display certain computed properties such as the Energy and Gradient on the status line. However, this display is transient. The Properties... menu item allows you to investigate certain recently-computed properties. To see other computed properties you might need to create a log file.

The Properties display depends on what is selected at the time you click on the menu item. If a single atom is selected, then the display shows the properties of that atom. If two atoms are selected, then the display shows the properties of "bond". Otherwise, the properties of the total molecular system are displayed.

## Atom Properties

This dialog box shows the properties of an atom. The dialog box appears if a single atom is selected when the Properties menu item is invoked. Leaving out the obvious ones, the properties include:



**Electronic Charge** This is the atom charge from molecular mechanics templates or the atom charge computed in quantum mechanical calculations (Mullikan charge).

**Spin Population** This is excess of alpha (spin up) over beta (spin down) electrons on the atom. It is analogous to electronic charge above but weights the spin of the

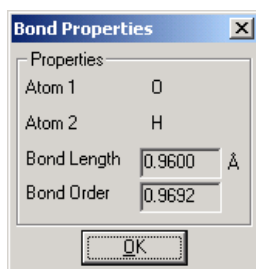
electron (positive for spin up, negative for spin down). Will be zero for closed-shell systems.

**Hyperfine Constant** This is the ESR Hyperfine constant for an open-shell system computed from the net spin at the nucleus of the atom in question. Will be zero for closed-shell systems.

**Gradient** This is the force on the atom computed as the gradient of energy with respect to atomic displacement in the X, Y, or Z direction.

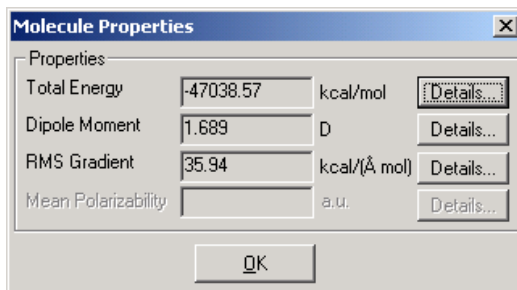
## Bond Properties

The properties of a bond include its length and the computed bond order. For quantum mechanical calculations, two atoms need not be “bonded” to have a bond order between them.



## Molecular System Properties

The following dialog box shows the displayed properties of a molecular system.



These properties include:

**Total Energy** This energy is relative to a sum of atomic energies for semi-empirical computations. The Details button shows further information about the nuclear and electronic components of this energy as well as the binding energy (semi-empirical only) and heat of formation (semi-empirical only). If an MP2 correlation energy has been computed it will appear here.

Energetic properties		
Energies		
Total Energy	-47038.57	kcal/mol
Binding Energy	-47038.57	kcal/mol
Heat of Formation		kcal/mol
Electronic Energy	-52784.79	kcal/mol
Nuclear Energy	5746.216	kcal/mol
MP2 Energy		kcal/mol

**Dipole Moment** The dipole moment will be computed for quantum mechanical calculations (from the wavefunction) as well as molecular mechanics (where it is computed from the atom charges). The Details button shows the Cartesian components of the dipole moment.

**RMS Gradient** This shows the total gradient on the system which for an optimized structure would be zero.

**Mean Polarizability** This appears only if the polarizability has been computed in a quantum mechanical calculation. The Details button shows the whole tensor.

## Plot Molecular Graphs

Use this command if you want to display electrostatic potential, total spin density, or total charge density results of a semi-empirical or *ab initio* or DFT calculation. It displays the Plot Molecular Properties property sheet. This command is unavailable unless a

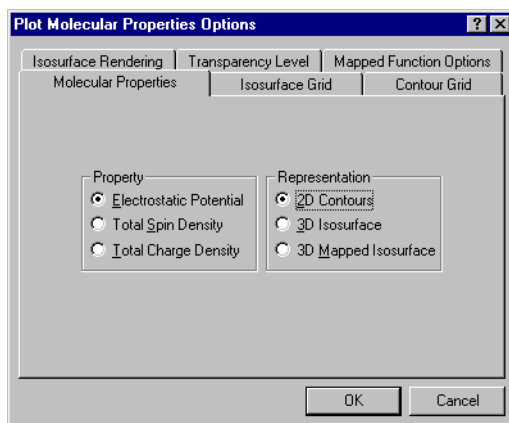


quantum-mechanical wavefunction has been calculated, via Single Point, Geometry Optimization, Molecular Dynamics or Vibrations.

## Plot Molecular Properties Property Sheet

Use the Plot Molecular Properties property sheet to display electrostatic potential, total spin density, and total charge density. Plot Molecular Properties is activated on the Compute menu only after you complete a semi-empirical or *ab initio* or DFT calculation. The property sheet has several “pages” which you can select with the tab controls.

## Plot Molecular Properties



<b>Property</b>	This lets you select the property that you are displaying.
<b>Electrostatic potential</b>	This plots the electrostatic potential field due to the electronic charge distribution and nuclear charges, in units of $e/a_0$ . Not available for Extended Hückel. Selecting Electrostatic Potential makes the 3D Mapped Isosurface option available under Representation; otherwise that option is grayed out.
<b>Total spin density</b>	This plots the probability of finding more spin-up electrons than spin-down electrons at any point in space, in units of $e/a_0^3$ .

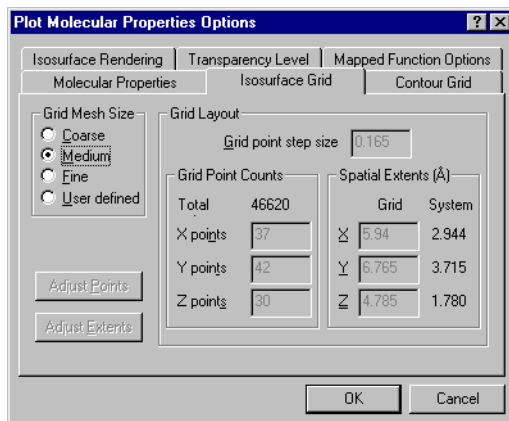
<b>Total charge density</b>	This plots the electron density function for the molecular valence electrons, in units of $e/a_0^3$ . It is often associated with the surface of a molecule. It describes the probability of finding an electron at a point in space. The value is the sum for each electron of $\psi_i^2$ , where $\psi_i$ is the molecular orbital occupied by the $i$ th electron. For a closed-shell system this is $2\psi_i^2$ , summed over the occupied orbitals.
<b>Representation</b>	In this section you specify how you wish the property to be displayed.
<b>2D Contours</b>	Use this option to draw a two-dimensional contour plot of the selected property.
<b>3D Isosurface</b>	Use this option to draw a three-dimensional surface which maps out a specified value of the selected property.
<b>3D Mapped Isosurface</b>	Use this option to draw a three-dimensional surface which is colored according to the values of the selected property. This option is available only when Electrostatic Potential has been chosen as the property to be displayed.

If you have a monochrome monitor or you chose black bonds on a white workspace (see “Using Black-and-white Output” on page 105), positive contour lines are solid and negative contour lines are dashed.

*Note:* You cannot display a contour plot or isosurface if Stereo or Perspective of the Sticks rendering are turned on. If you display an isosurface and the Wedges option of the Sticks rendering is on, the wedges will not be shown while the isosurface is displayed.

### Isosurface Grid Dialog Box

Use these options to specify values for the 3D grid used to render an isosurface.



**Grid Mesh Size** Select a Coarse, Medium, or Fine grid mesh, or select User defined and then specify the grid details that you wish. Selecting User defined makes the other sections of this page active.

**Grid Layout** Use these options to specify the options for the 3D grid.

**Grid point step size** The distance between data points in the 3D grid, in units of Å. The same spacing is used for all three coordinate axes.

**Grid point counts** The size of the 3D grid in each of the coordinate directions, in units of the number of grid points. At the top is the total number of grid points, which is the product of the three values below it. If you change any of these values, you must click on Adjust Extents in order to have the change accepted.

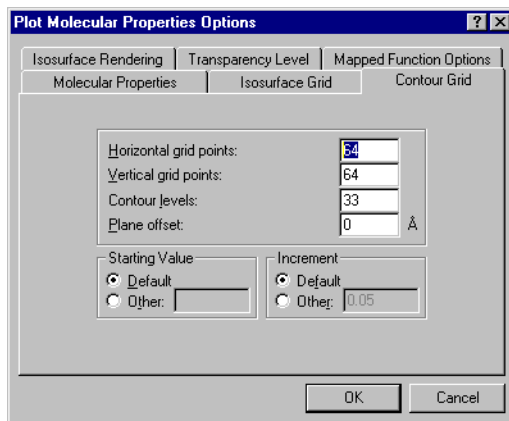
**Spatial extents** The size of the 3D grid in each of the coordinate directions, in units of Å. The numbers to the right of these values give the spatial extent of the nuclear framework. If you change any of these values, you must click on Adjust Points in order to have the change accepted.

**Adjust points** Use the Grid point step size and the given Spatial extents values to recalculate the values of the Grid point counts

**Adjust Extents** Use the Grid point step size and the given Grid point counts values to recalculate the values of the Spatial extents

## Contour Grid

Use these options to specify values for the 2D grid used to render a contour plot. If you have specified black-and-white output settings the positive contours will be drawn as solid lines and the negative contour lines will be drawn as dashed lines.



**Horizontal grid points** Enter a number for the horizontal resolution of the contour lines. The default of 45 means that HyperChem calculates 45 values in each horizontal line and interpolates contour lines between grid points. Range: 2 to 8,172.

**Vertical grid points** Enter a number for the vertical resolution of the contour lines. The default of 45 means that HyperChem calculates 45 values in each vertical line and interpolates contour lines between grid points. Range: 2 to 8,172.

**Contour levels** Enter a value for the number of contour lines to plot. The default is 15 lines. Range: 1 to 32,767.

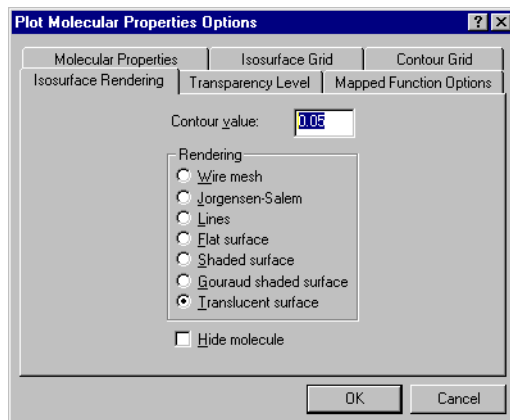
**Plane offset** The default plane of a set of contour lines is perpendicular to the viewer's z axis (in the plane of

the screen) and through the center of mass of selected atoms or, if there is no selection, the whole molecular system. This offset moves the plane of contour lines, in Ångstroms, along the viewer's z axis: a positive offset is outward from the screen, a negative offset is inward.

<b>Starting value</b>	The value of the lowest contour line.
<b>Default</b>	The starting value that HyperChem calculates from the grid points and Contour levels settings, or the value you entered for the last contour plot.
<b>Other</b>	Enter a new value. When you L-click Other, HyperChem shows the present Default value.
<b>Increment</b>	The interval between two adjacent contour lines.
<b>Default</b>	The starting value that HyperChem calculates from the grid points and Contour levels settings, or the value you entered for the last contour plot.
<b>Other</b>	Enter a new value. When you L-click Other, HyperChem shows the present Default value.

### Isosurface Rendering

These options specify how the isosurface should be drawn.



**Contour Level** The isosurface shows where, in 3D space, the property that you have specified has this value.

**Rendering** In this section, you specify which rendering mode you wish the isosurface drawn with. In general, the modes at the beginning of the list are simpler and more quickly drawn than the modes at the end of the list.

**Wire mesh** The isosurface is drawn as a transparent pattern of crossed lines. This mode is the fastest to calculate and display.

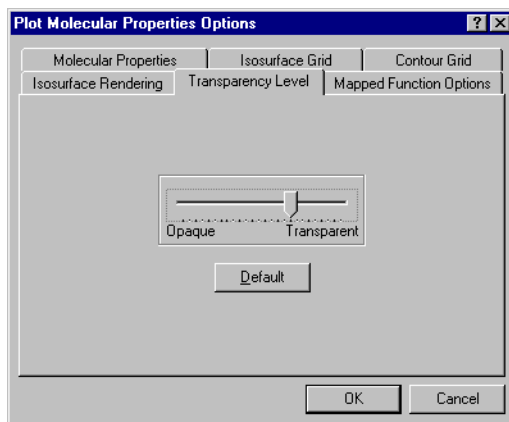
**Jorgensen-Salem** The isosurface is drawn as Lines, below, except that while parts of the isosurface can hide other parts of the isosurface, the molecular structure is *not* hidden. (See *The Organic Chemist's Book of Orbitals*, by William L. Jorgensen and Lionel Salem, New York, Academic Press, 1973.) If you have set the bond and window colors for black and white (see "Using Black-and-white Output" on page 105), the positive parts of the isosurface are drawn with solid lines and the negative parts are drawn with dotted lines when you print the structure.

**Lines** The isosurface is drawn as a pattern of crossed lines; objects close to the user's point of view will hide objects farther away.

<b>Flat surface</b>	The isosurface is drawn as a solid surface, without shading.
<b>Shaded surface</b>	The isosurface is drawn as a solid surface with shading to enhance the three-dimensional appearance. The surface is divided into small segments for coloring; this gives a reasonably good appearance at moderate speed.
<b>Gouraud shaded surface</b>	The isosurface is drawn as a smooth shaded surface for optimal three-dimensional appearance. This mode is the slowest to calculate and display of the simple opaque surfaces.
<b>Translucent surface</b>	The isosurface is drawn as a smooth, semi-transparent surface. The molecular structure, and far-away parts of the isosurface, can be seen showing through the nearer parts of the surface. The degree of transparency can be specified in Transparency Level (below). The extra calculations required for this display make it slower than Gouraud shaded surface.

**Hide molecule** If this option is selected (✓), the molecule will be hidden when the isosurface is displayed.

## Transparency Level

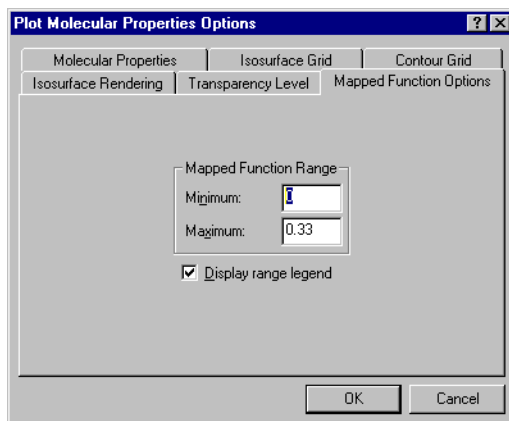


Use this slider to control the transparency level of an isosurface rendered as a Translucent Surface. The setting can be anywhere from completely transparent (isosurface will be invisible) to completely opaque. If you want a completely opaque surface, you should use a Gouraud Shaded Surface instead; the appearance will be the same but the Gouraud surface is drawn more quickly. If you want a completely transparent isosurface, you can simulate it quickly by turning the isosurface display off.

The Default button restores the default transparency level.

## Mapped Function Options

These options control the appearance of an isosurface drawn as a 3D Mapped Isosurface.



**Mapped Function Range**

These boxes show the minimum and maximum values used to render the color range of the isosurface.

**Display Range Legend**

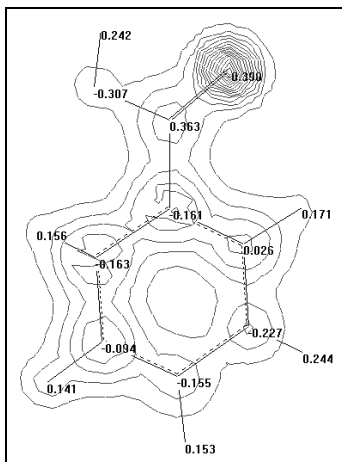
If this option is selected (✓), a legend bar will be drawn on the workspace, showing the range of colors used and the function values that they represent.

## Plotting Results

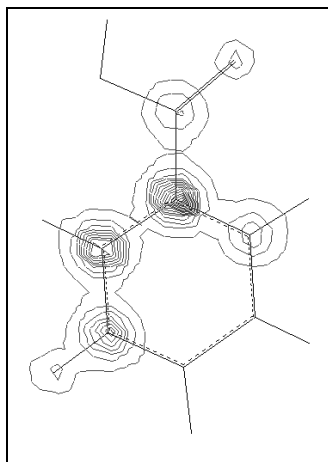
HyperChem can plot the results of semi-empirical or *ab initio* or DFT calculations as 2D contour plots or 3D isosurfaces and show the contour lines or isosurfaces on top of the molecular system in the workspace. If you have a color monitor, the contour colors are



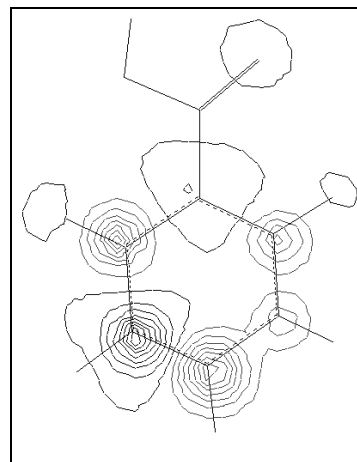
as specified in the Preferences dialog box (page 95). If you have a monochrome monitor or you chose black bonds on a white workspace (see "Using Black-and-white Output" on page 105), positive contour lines are solid and negative contour lines are dashed.



*total charge density  
with charge Labels*



*electrostatic potential*



*total spin density*

Contour plots are always created parallel to the plane of the screen. To see different aspects of the value in the plot, you can change the view (move molecules in the Viewer's Coordinate system) and calculate a plot again. If the atomic coordinates change (because of movement in the Molecular Coordinate system), you must repeat the calculation with Single Point, Geometry Optimization, Molecular Dynamics or Vibrations.

Contour plots are temporary and disappear whenever HyperChem revises the display of a molecular system. For example, when you use a HyperChem tool, such as the Selection tool, contour lines disappear. Isosurfaces can persist if the workspace is not actually altered; one can, for example, rotate a structure and isosurface.

To produce a printed record of the plotted wave functions, use Print (see "Print" on page 94) to send the image directly to a printer. Use Copy Image (see "Copy Image" on page 206) to send the image to the Clipboard or a file (or both). You can then use another program to print this image.

## Orbitals

Once you have performed a semi-empirical or *ab initio* or DFT calculation you can choose Orbitals to display the energy levels of all the orbitals, or the contours or isosurfaces for an orbital you specify. Use the Orbitals dialog box to see degeneracies and near degeneracies, HOMO-LUMO gaps, symmetries, orbital occupation scheme, alpha and beta spin manifolds separately (for UHF calculations of open shell systems), or d-d splittings (for transition metals).

### Orbitals Dialog Box

After you complete a single point semi-empirical or *ab initio* or DFT calculation, use this dialog box to display the contours or isosurfaces of a user-specified orbital or the energy levels of all the orbitals. You can display the relative energy levels of both the occupied and virtual set of orbitals. You can label each orbital in the energy-level diagram according to its occupation (empty, singly occupied, doubly occupied). This dialog box allows you to see the following:

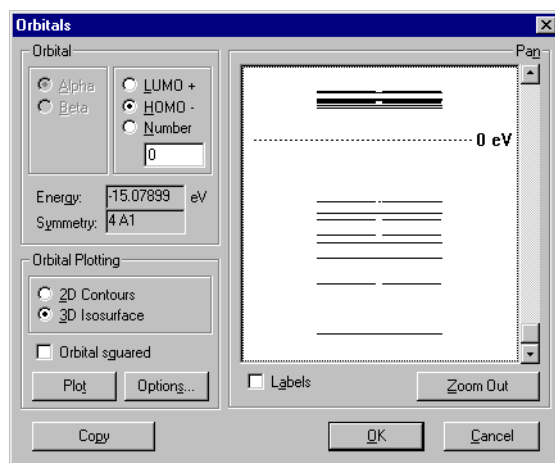
- degeneracies and near degeneracies
- symmetries
- HOMO-LUMO gaps
- orbital occupation scheme
- alpha and beta spin manifolds separately (for UHF calculations of open shell systems)
- d-d splittings (for transition metals)

To display this dialog box, choose Orbitals on the Compute menu. Orbitals is activated on the Compute menu only after you complete a single point calculation using a quantum-mechanical method.

**Note:** To zoom in on the orbital energy diagram, L-drag a rectangle around the vertical region that you want to zoom in on. When you release the mouse button the energy levels will be redrawn. Use the Zoom Out button to see all the energy levels again.

The orbital energy levels in an *ab initio* or DFT calculation may look different from those in semi-empirical calculations. The *ab initio* and DFT quantum mechanical methods are all-electron

methods, as compared to the valence-electron only methods of the semi-empirical quantum mechanical calculations. For heavy-atom containing molecular systems, the molecular orbitals consist of two sets. The first set of molecular orbitals are core orbitals and the second set of molecular orbitals are valence orbitals. The first set of orbitals have much lower energies than the second set of orbitals. Thus, the core orbitals are shown at the bottom of this dialog box and the valence orbitals are displayed at the top of this dialog box. If you would like to examine the valence-orbital splitting closely, you can use the mouse to Zoom and Pan to get a proper picture.



**Pan** Scroll up or down in orbital energy.

**Orbital** Reports information on the selected orbital. Each of the options in this area may also be used to specify which orbital is to be selected. Orbital selections are nominally done by L-clicking with the mouse on the desired orbital in the display window. An alternative is to select the orbital by number by entering a value into the text box adjacent to the LUMO+, HOMO-, or Number options.

**Alpha** Indicates whether the selected orbital has alpha spin. If the selected orbital has beta spin, L-clicking on Alpha changes the selection to an alpha orbital. This option is active only after a UHF semi-empirical calculation has been done.

<b>Beta</b>	Indicates whether the selected orbital has beta spin. If the selected orbital has alpha spin, L-clicking on Beta changes the selection to a beta orbital. This option is active only after a UHF semi-empirical calculation has been done.
<b>LUMO+</b>	Reports the location of the selected orbital relative to the LUMO. For example, when you select the orbital that is just above the LUMO in energy, +1 appears in the text box, whereas if you selected the orbital just below the LUMO (i.e., HOMO), -1 is reported in the text box.
<b>HOMO-</b>	Reports the location of the selected orbital relative to the HOMO. For example, the orbital just above (in energy) the HOMO reports -1 in the text box, whereas the orbital just below the HOMO is reported as 1.
<b>Number</b>	Reports the absolute number of the selected orbital starting from the lowest energy orbital (bottom). For UHF calculations where both alpha and beta columns of orbitals are displayed, the numbering is done separately within each group. This is also true for the HOMO- and LUMO+ options.
<b>Energy</b>	Reports the energy of the selected orbital in units of electron volts (eV).
<b>Symmetry</b>	Reports the irreducible representation of the selected orbital.
<b>Labels</b>	Displays the electron occupation of each orbital in the orbital display window. By convention, up arrows represent electrons with alpha spin, and down arrows represent electrons with beta spin. For RHF calculations the maximum orbital occupancy is 2, whereas for UHF calculations the maximum occupancy is one. This is a consequence of the Pauli Exclusion Principle. Labels also displays the energy of each orbital.

**Zoom Out** After you zoom in (by L-dragging the mouse over a selected group of orbitals), this option displays the entire range of orbitals in one screen.

**Orbital Display** Provides you with options for setting the display options and displaying contour plots or isosurfaces.

**2D Contours** Use this option to draw a two-dimensional contour plot of the selected orbital.

**3D Isosurface** Use this option to draw a three-dimensional surface which maps out a specified value of the selected orbital.

**Orbital squared** When selected, plot displays the square of the orbital value in a specified plane. This corresponds to a probability density rather than a probability amplitude.

**Options . . .** This displays the Orbital Isosurface Options property sheet. It is the same as the "Plot Molecular Properties Property Sheet" on page 392 except that the "Molecular Properties" and "Mapped Function" pages are not available.

**Plot** Displays a contour plot or isosurface of the selected orbital with the specified options.

**Copy** Copy the orbital energy diagram to the Windows clipboard. To get solid and dashed black lines on a white background for the energy levels, set Window to White and Bond to Black in the Preferences dialog box (see page 95).

*Note:* You cannot display a contour plot or isosurface if Stereo or Perspective of the Sticks rendering are turned on. If you display an isosurface and the Wedges option of the Sticks rendering is on, the wedges will not be shown while the isosurface is displayed.

## Describing Orbitals

Molecular orbitals are numbered from 1 to M, where 1 is the lowest energy orbital and M is the total number of orbitals involved in the calculation for the current system. In semi-empirical quantum mechanics methods, the contribution to M is 1 for hydrogen and

helium atoms ( $1s$ ), 4 for all main-group atoms ( $ns, np$ ), and 9 for all the transition metals. For the Extended Hückel method, which can calculate d orbitals for Si, P, S, and Cl, use 9. The number of occupied orbitals,  $N$ , is half the number of valence electrons for closed-shell molecules.

In addition to using a number from 1 to  $M$ , HyperChem can identify orbitals by reference to the HOMO or the LUMO. You can describe an orbital by HOMO- $d$  or LUMO+ $d$ , where  $d = 0$  to  $max$ . For HOMO,  $max$  is the number of occupied orbitals minus 1 ( $N-1$ ). For LUMO,  $max$  is  $M-N-1$ .

## Practical Guidelines for Orbitals

For transition metals the splitting of the d orbitals in a ligand field is most readily done using EHT. In all other semi-empirical methods, the orbital energies depend on the electron occupation.

HyperChem's molecular orbital calculations give orbital energy spacings that differ from simple crystal field theory predictions.

The total molecular wavefunction is an antisymmetrized product of the occupied molecular orbitals.

The virtual set of orbitals are the residue of SCF calculations, in that they are deemed least suitable to describe the molecular wavefunction.

Normally, you would expect all 2p orbitals in a given first row atom to be identical, regardless of their occupancy. This is only true when you perform calculations using Extended Hückel. The orbitals derived from SCF calculations depend sensitively on their occupation. For example, the 2px, 2py, and 2pz orbitals are not degenerate for a CNDO calculation of atomic oxygen. This is especially important when you look at d orbital splittings in transition metals. To see a clear delineation between  $t_{2u}$  and  $e_g$  levels you must use EHT, rather than other SE methods.

The orbital phases are denoted by the colors of the orbital lobes. Two lobes with the same color are *in phase*, and those of different color are said to be *out of phase*. It is important to note the colors have nothing to do with charge.

## Vibrational Spectrum

Use Vibrational Spectrum to display graphically the normal modes associated with individual vibrations, as well as to display a graphical representation of an infrared (IR) vibrational spectrum. Vibrational Spectrum is only available after you have used Vibrations to compute the vibrational motions of the nuclei (see page 385).

In HyperChem, you can calculate IR vibrational spectra using any of the quantum mechanical methods except Extended Hückel.

Vibrations represent the intrinsic rhythmic motions of the atoms within a molecule. Whereas translations and rotations correspond to motions of atoms relative to an external frame of reference, vibrations correspond to motions of atoms relative to each other.

The seemingly complex and chaotic trajectories of the nuclei undergoing vibratory motion can be conceived of as a superposition of simpler motions called *normal modes of vibration*, which generally fall into one of the following classes: stretching, bending, rocking, or wagging.

Even at 0 Kelvin vibrational motions exist as a consequence of the Heisenberg Uncertainty Principle. Every molecule contains  $3N-6$  vibrational modes (or  $3N-5$  if it is linear), where  $N$  is the number of atoms. Each mode can be IR active, Raman active, IR and Raman active, or inactive.

Each mode is quantized in energy and collectively (all modes) result in a spectrum of discrete frequencies. There is a one-to-one relationship between a normal mode of vibration and a fundamental vibrational frequency.

Once you have calculated Vibrations, you can use the Vibrational Spectrum dialog box to do the following:

- Provide insight into the rigidity of the molecular framework
- Visualize normal modes corresponding to lines in the IR spectrum
- Help identify unknown compounds by correlating predicted versus experimental vibrational frequencies
- Differentiate minima from saddle points on a potential energy surface.

### To analyze vibrations:

1. Perform a Vibrations calculation (see “Vibrations” on page 385) for your optimized system.

You may wish to rearrange your screen so that you will be able to see the system in the workspace and the Orbitals dialog box at the same time. You cannot resize or move the main window when the Vibrational Spectrum dialog box is active.

2. Choose Vibrational Spectrum from the Compute menu.

This opens the Vibrational Spectrum dialog box, which shows the spectrum of frequencies corresponding to each normal mode. The vertical lines at the top represent all the vibrational fundamental frequencies. The spectrum at the bottom corresponds to IR-active vibrations. The frequency increases from the right side to the left side of the dialog box. The height of the bottom row of lines corresponds to their IR intensities.

3. L-click on a line.

The highlighted line is now violet. Information (normal mode sequence number, degeneracy, frequency, and intensity) on this line is shown beneath the spectrum at the bottom of the dialog box.

With large molecules, the spectrum will have many lines, and you may not be able to easily select a given line of interest. To focus in on a line you can use the Zoom and Pan slider bars. Zoom compresses and expands the displayed range of frequencies. Pan translates or shifts a given range of frequencies.

If you zoom in on the spectrum, the lowest frequency converges toward the upper frequency, which remains constant.

4. Move the dialog box to try and see the system in the workspace.

This allows you to see the structure and the dialog box simultaneously.

5. To visualize the vibrational motions (normal mode) for the selected line, turn on Animate vibrations, and choose Apply.

HyperChem animates the normal mode that corresponds to the frequency that you have selected.

6. L-click on a different line or use the [1] and [2] keys to select a different line and choose Apply.



A different mode will be animated.

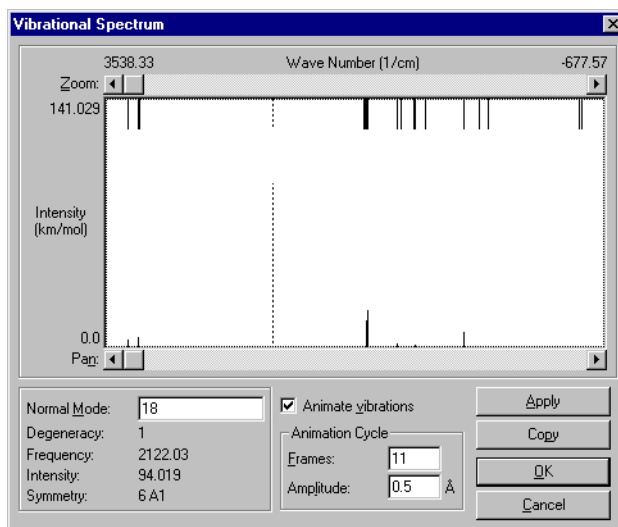
To store vibrational spectra:

If you wish to save vibrational results after closing the Vibrational Spectrum dialog box, see “Export . . .” on page 93. Alternately, you can run the sample script saveir.scr (see “Open Script” on page 106 for information on running scripts). This will create another script file that you can later open to restore the saved vibrational results into HyperChem.

## Vibrational Spectrum Dialog Box

To perform vibrational analysis, choose Vibrations on the Compute menu to invoke a vibrational analysis calculation, and then choose Vibrational Spectrum to visualize the results.

*Suggestion:* You can capture an image of the Vibrational Spectrum dialog box by using Setup Image on the Tools/Preferences menu to set Window to Top level (see page 99), activating the Vibrational Spectrum dialog box and then pressing [F9] to capture the image. To get black lines on white background for the spectrum, set Window to White and Bond to Black in the Preferences dialog box (see page 95). You can also use the Copy button in this dialog box; see below.



<b>Normal Mode</b>	Sequence number of normal mode in order of ascending frequency. In the case of degeneracy, the number begins with the last identical mode.
<b>Degeneracy</b>	The number of modes that have the same frequency.
<b>Frequency</b>	The number of oscillations per unit time, expressed in units of reciprocal centimeters. Frequency relates to energy by $E=h\nu$ , where $\nu$ is the frequency. Imaginary frequencies are shown as negative numbers; their presence indicates that the structure is not at a local minimum.
<b>Intensity</b>	The height of the IR absorption line. Related to the dipole derivative with respect to normal coordinate displacement. Low intensities correspond to forbidden frequencies.
<b>Animate vibrations</b>	Controls the display of animated vibrations.
<b>Animation Cycle</b>	Sets the distance in Ångstroms to move the fastest atom during vibrational animations.
<b>Frames</b>	Specifies the number of steps to use in each cycle while animating vibrations.
<b>Amplitude</b>	Sets the distance in Ångstroms to move the fastest atom during vibrational animations.
<b>Apply</b>	The specified vibrational mode is animated in the workspace, leaving this dialog box active
<b>Copy</b>	This copies a bitmap picture of the Vibrational Spectrum dialog box to the Clipboard. See also the "Suggestion" just before the picture of the dialog box.

If you want to vary the speed and the range of motion of the animation, try changing the values for Frames and Amplitude. The speed of the animation slows down as the number of frames increases.

## Practical Guidelines

The quality of the vibrational frequencies varies widely with the semi-empirical method that is used. Generally, AM1, and PM3 are

in closer agreement with experiment than methods based on CNDO or INDO.

The vibrational frequencies are derived from the harmonic approximation, which assumes that the potential surface has a quadratic form.

Large amplitude (floppy) vibrational modes often exhibit significant anharmonicity that may increase errors in computed frequencies. In addition to anharmonicity, usually there is coupling between vibrational modes in reality which is not accounted for in the computational model.

If there are negative frequencies in an IR spectrum, it is a sign that you are not at a minimum energy structure. A valid minimum energy structure possesses only positive frequencies.

Results of a recent literature study indicate that frequencies computed using semi-empirical PM3, AM1, and MNDO methods compare well to values obtained at the *ab initio* or DFT level using medium size basis sets. Of these three methods, PM3 showed the closest correspondence to experimental values, which is generally about 10 percent too high in value from stretches (Seeger, D.M.; Korzeniewski, C.; Kowalchuk, W., *J. Phys.Chem.* 95:68-71, 1991).

The following table shows the accuracy of computed fundamental frequencies for CO<sub>2</sub> (cm<sup>-1</sup>):

Normal Mode	CNDO	INDO	AM1	PM3	Experiment
bend	571	553	525	521	667
asymmetric stretch	1888	2080	1480	1406	1340
symmetric stretch	6427	5737	2565	2386	2349

## Comparing Computed Results with Experiment

After you compute an IR spectrum with HyperChem, you can refer to the table in *HyperChem Computational Chemistry, Practical Guide*, to assign computed IR lines and qualitatively assess the accuracy of the computation.

## Electronic Spectrum

Use this command to calculate an ultraviolet-visible (UV-visible) or “electronic” spectrum following a semi-empirical Single Point calculation with singly excited configuration interaction (CI). ZINDO/S is specifically parameterized to reproduce ultraviolet-visible spectra. However, you can use any of the *ab initio* or semi-empirical methods, except Extended Hückel. To generate a UV-visible spectrum, you must first perform a singly excited CI calculation with the quantum mechanical method you choose. This menu item is unavailable unless a CI calculation has been performed for the current system.

*Caution:* You can request Configuration Interaction calculations using Density Functional methods but the results are suspect. The computed Kohn-Sham orbitals from a density functional calculation have a different interpretation from the Hartree-Fock orbitals of an *ab initio* calculation and the meaning of Configuration Interaction calculation is not obvious. The difference in energy between occupied and unoccupied Kohn-Sham orbitals is expected to represent an average of the appropriate singlet-triplet excitation energy.

Applications of UV-visible spectroscopy include diverse topics in photochemistry, photobiology, photophysics, and analytical spectroscopy. Examples include:

- Studying changes in long wave absorption of dye molecules upon modification of substituent groups
- Studying metal-ligand charge-transfer spectra
- Assigning UV-visible absorption transitions in terms of the orbitals involved in the transition
- Predicting triplet-triplet absorption spectra
- Calculating  $S_1 - T_1$  energy gap to assess rate of intersystem crossing.

To compute UV visible spectra:

1. Choose a semi-empirical or *ab initio* method and select Singly Excited as the option for CI.

You will need to specify at least one occupied and at least one unoccupied orbital, or an energy criterion that is at least as

large as the HOMO-LUMO gap, in the CI Options dialog box. Otherwise you will get no spectrum. In general, as you include more orbitals or increase the energy criterion, the calculation will take longer but the spectrum will be more accurate.

2. Choose Single Point on the Compute menu.

HyperChem performs an SCF calculation to obtain the reference electronic configuration associated with the singlet ground state of your molecule. Next, HyperChem generates a series of singly excited configurations, computes the Hamiltonian matrix elements between them, and then diagonalizes the matrix to get the spectrum of electronic states.

3. When the calculation finishes, choose Electronic Spectrum on the Compute menu.

This opens the Electronic Spectrum dialog box. Two sets of lines (transitions) appear in the dialog box. The top set shows all the excited electronic states (both singlet and triplet); the bottom set shows only states that are spectroscopically active and their relative intensities.

4. L-click on a different line or use the [1] and [2] keys to cycle through the transitions.

To store UV visible spectra:

If you wish to save UV-visible results after closing the Electronic Spectrum dialog box, see “Export . . .” on page 93. Alternately, you can run the sample script saveuv.scr (see “Open Script” on page 106 for information on running scripts). This will create another script file that you can later open to restore the saved UV-visible results into HyperChem.

*Note:* Some guidelines for CI singlet calculations are provided in “Configuration Interaction Dialog Box” on page 308 and in *HyperChem Computational Chemistry*. We suggest that you review those guidelines before computing UV visible spectra.

Once the spectrum has been calculated it will be shown in the Electronic Spectrum dialog box.

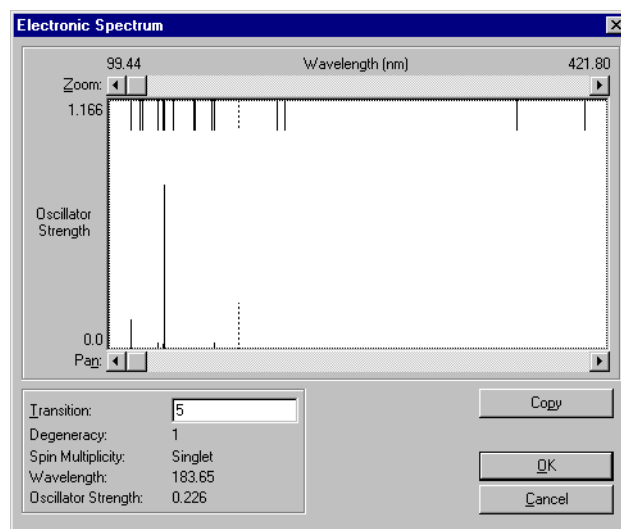
## Electronic Spectrum Dialog Box

Use the Electronic Spectrum dialog box to display and analyze the UV-visible spectrum produced by a singly excited CI calculation. This dialog box is available only after you do a single point CI

semi-empirical or *ab initio* calculation. Electronic Spectrum is then activated on the Compute menu.

Transitions in the spectrum are represented as vertical lines contained in a *spectral display*, with the wavelengths marked along the top of the display. The top set of lines represents all the excited electronic states (both singlet and triplet), and the bottom set represents only states that are spectroscopically active.

**Suggestion:** You can capture an image of the Electronic Spectrum dialog box by using Setup Image on the Tool/Preferences menu to set Window to Top level (see page 99), activating the Electronic Spectrum dialog box and then pressing [F9] to capture the image. To get black lines on white background for the spectrum, set Window to White and Bond to Black in the Preferences dialog box (see page 95). You can also use the “Copy” button (described below) in this dialog box.



**Wavelength** Numbers in the upper left- and right-hand corners of the display show the limits of the currently displayed spectrum.

**Zoom** Moving the Zoom scroll button to the right narrows the wavelength range of the display. The value at the left-hand edge of the display remains fixed regardless of the Zoom position.

<b>Oscillator Strength</b>	The scale of the oscillator strength (intensity) is indicated at the upper and lower left-hand corners of the spectral box.
<b>Pan</b>	Scrolls the display left or right. You can pan the display by L-clicking on the Pan scroll button, or by L-clicking on the scroll bar and pressing the left or right arrow keys.
<b>Transition</b>	Shows the transition number of the currently selected transition. Transitions are ordered from longest wavelength to shortest wavelength (left to right). You can select a transition by entering the number of the transition in this box.
<b>Degeneracy</b>	Specifies the degeneracy of the excited state corresponding to the selected transition.
<b>Spin multiplicity</b>	This is the total spin multiplicity, $2S+1$ , where $S$ is the total electron spin of the system. Each unpaired electron counts for $1/2$ . A closed-shell system (singlet) has a multiplicity of 1. A doublet or triplet state has a multiplicity of 2 and 3, respectively.
<b>Wavelength</b>	The wavelength (nm) of the selected transition.
<b>Oscillator Strength</b>	A dimensionless quantity that is proportional to the intensity of the selected transition.
<b>Copy</b>	This copies a bitmap picture of the Electronic Spectrum dialog box to the Clipboard. See also the "Suggestion" just before the picture of the dialog box.

To show various characteristics of individual transitions, L-click inside the display and the spectral box is highlighted with a violet line. Then select individual transitions by L-clicking on a line or by using the arrow keys on the keyboard. The options in the dialog box show the characteristics of the selected line.

## Practical Guidelines

If you are studying the electronic spectra of conjugated organic molecules, such as dyes, it is often sufficient to include only a few orbitals in the CI. For instance, if you are calculating the UV visible spectrum of p-amino-benzoic acid, the wavelength of the two

intense lines in the UV spectroscopic range (approximately 200 nm to 350 nm) is calculated just as accurately using two occupied and two unoccupied orbitals as using 10 of each. The extra orbitals yield more lines in the high-energy region (beyond the range of UV spectrometers) but do not affect the lines of interest, which are those at longer wavelengths. As with many computational methods, trends and differences are given more accurately than absolute values. ZINDO/S calculations on several substituted benzenes underestimate the wavelength of the major UV absorption by about 20 nm, but the relative values are very well reproduced.

Absorption lines for transitions that are electronically forbidden—but vibrationally allowed—do not appear in the Electronic Spectra dialog box. For example, d-d transitions in metals do not show intensity in HyperChem's UV visible calculations.

A CI singlet calculation using a singlet state as a reference yields both singlet and triplet eigenstates as output. If a doublet reference state is used, then both doublet and quartet eigenstates are output.

The accuracy of the computed spectrum depends on the size (i.e., orbital active space) of the configuration interaction singlet calculation.

Since CI calculations are longer than SCF level calculations, the size of molecules may be a limiting factor.

## **Saving Information in a Log File**

When you compute UV-visible spectra, you can obtain additional information about the excited state properties by using Start log on the File menu. When `QuantumPrintLevel=0` in the Start Log dialog box, the log file saves the following information for UV-visible spectra calculations:

- Absolute energy in eV for the reference configuration
- Excitation energy in  $\text{cm}^{-1}$  for the excited states
- Dipole moments in Debyes
- Transition dipole for each possible excitation
- Intensities for each possible excitation
- Energies and gradient
- Eigenvalues of the reference configuration (the orbital energies)



- Atomic orbital electron populations
- Net charges and coordinates

## Potential Energy Plotting

Given the currently chosen engine for computing energies, i.e. Amber94, PM3, etc., HyperChem allows you to directly and immediately obtain,

- One-dimensional potential energy plots
- Two-dimensional potential energy plots

A one-dimensional potential energy plot shows the energy as a function of one variable. A two-dimensional potential energy plot shows the energy as a function of two variables. These variables can be either a bond length, a bond angle, or a bond torsion. Sometimes these 1D and 2D potential energy plots are referred to as 2D and 3D graphs. For example, a 2D potential energy plot involves three variables in the graph — the energy and the two independent variables.

### One Dimensional Potentials

The creation of a 1D potential energy plot is very easy. One first chooses the functional form of the energy that is to be used (as one would prior to any single point calculation). This choice might be AMBER, PM3, STO-3G, etc. Assuming there is a molecule in the workspace, then one simply selects the 2, 3, or 4 atoms involved in the bond, angle, or torsion that is to be used as the independent variable.) For example, one might select the two carbon atoms in ethane to plot the energy as a function of the C-C bond length.

#### Choosing the Independent Variable

The independent variable may be one of three types:

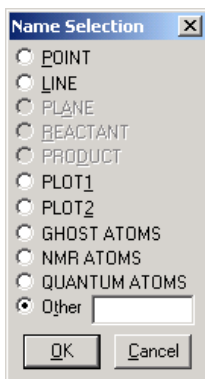
- bond length (two adjacent atoms selected)
- bond angle (three adjacent atoms selected)
- bond torsion (four adjacent atoms selected)

## Current Selection

The independent variable can be chosen via the *current selection*. If the current selection is appropriate (one of the above) then the menu item Potential... will be active and a potential energy plot can be initiated by clicking on this menu item. If the current selection is inappropriate, e.g. six atoms are selected or two atoms are selected but they are not bonded, then the menu item will be inactive (grayed).

## Named Selection PLOT1

The independent variable can alternatively be specified by using a *named selection* called PLOT1. At any time, independent of specifically requesting a potential energy plot via the Potential... menu item, you can define a 2-, 3-, or 4-atom selection and name it PLOT1,

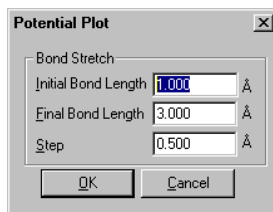


This option for named selections will only be available if the current selection is an appropriate one for a potential energy plot. If it is, however, then the named selection will be remembered and can be used to request a potential energy plot at some later time. The named selection PLOT1 is used, however, *only if no current selection has been made*. A current selection always overrides any PLOT1 named selection.

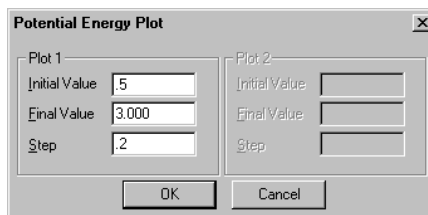
## Requesting a Plot

If the current selection is appropriate or there is no current selection but PLOT1 is defined, then the menu item Potential... will be

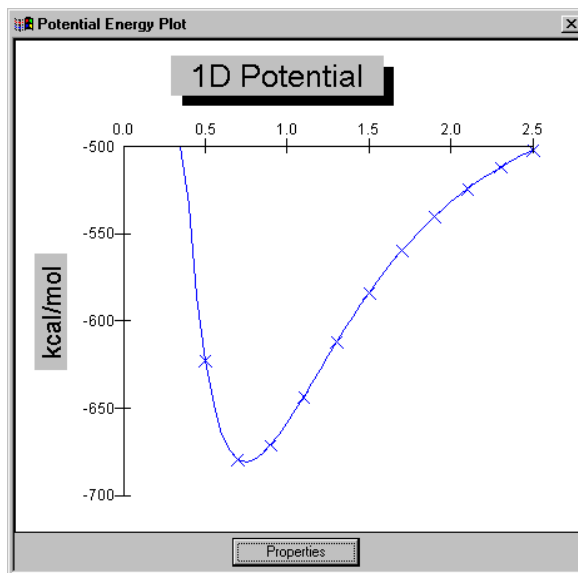
active. Clicking on this menu item will bring up a dialog box for selecting the values of the independent variable. If you are using the current selection, you will see the following dialog box:



If, however, there is no current selection but PLOT1 is named, then the following dialog box will appear:

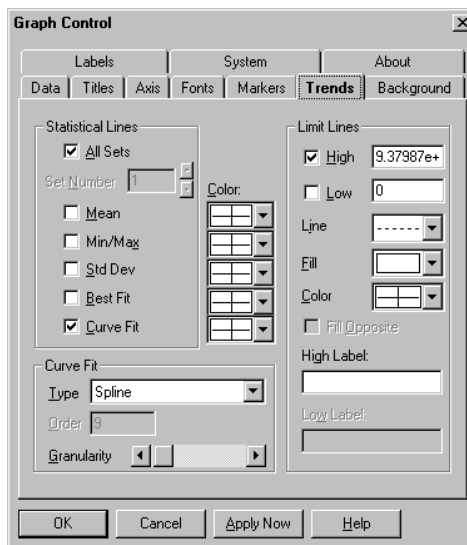


In either case, you are now ready to choose the equally-spaced points to be plotted. The values above were used to produce the following potential energy curve for the  $H_2$  molecule. This plot resides within one of HyperChem's dialog boxes and has a button (Properties...) for choosing the options for the plot.



## Plotting Options

Pushing the Properties button brings up the following dialog box:



These plotting capabilities in HyperChem are derived from a third-party graphing control and contain extensive options. Indeed, you could plot your own data in addition to data that is prepared automatically for potential energy plots, as described here. Not all of these options are described fully here and you may wish to explore this elaborate dialog box (shown above) if you require quite custom plots.

## Help

On-line help for the graphing control is available via the help button shown above. HyperChem does not use all the controls capability and some of the tabs (for example, the 2D and 3D Gallery tabs) shown in association with HELP are not available in HyperChem. For example, only 2D line plots and 3D surface plots are used for plotting potential energy curves and no other choice is available via HyperChem and the tabs shown above.

## Types of Fitting

The first thing that may be required in obtaining a good plot is to choose an appropriate fitting method. By default splines (as shown above) are drawn through the points. This is often a poor choice when the data is not very smooth. A better choice is often

a Variable-Order Polynomial but other choices are also available by clicking on the Trends tab. Only experimenting with your particular plot will give the best plot. The on-line help button, for example, describes each of the options in addition to HyperChem's spline default.

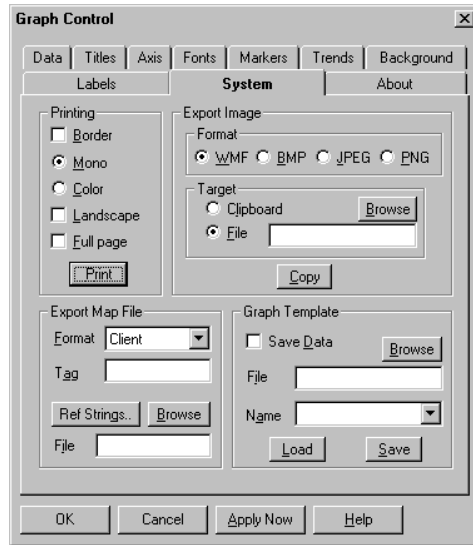
<b>Curve Type (default is Variable-Order Polynomial)</b>	
Choose the type of curve you want to plot.	
<b>Curve type</b>	<b>Description/formula</b>
Variable-Order Polynomial	Polynomial curve of variable order
Logarithmic	$y = a + b * \log(x)$
Exponential 1	$y = a * \exp(b * x)$
Exponential 2	$y = a * x * \exp(b * x)$
Power	$y = a * (x ^ b)$
Inverse 1	$y = a + b / x$
Inverse 2	$y = a / (b + x)$
Inverse 3	$y = 1 / (a + b * x)$
Inverse 4	$y = x / (a * x + b)$
Inverse 5	$y = 1 / (a + b * x) ^ 2$
Spline	Spline fit through all points
Moving Average Mid	Moving average plotted at midpoint of averaged group
Moving Average End	Moving average plotted at end point of averaged group

## Data

The data values computed by HyperChem and plotted by HyperChem can be displayed by clicking on the Data tab of the Graph Control dialog box obtained by clicking on the Properties... button of the graph. These values can be modified so that you can request your own plotting of data if so desired.

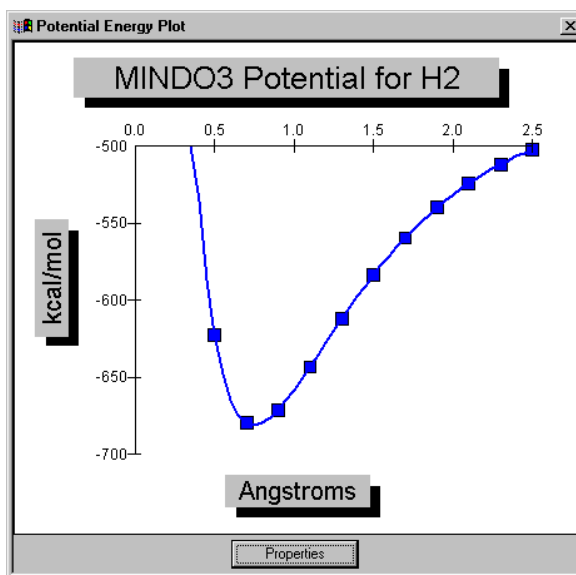
## Exporting and Printing Plots

By clicking on the System tab, you can request the printing of a plot or the transfer of a plot to the clipboard for use in another program such as for placing a plot on a web site:



## **Titles, Labels, Etc.**

Clicking on the tabs for Titles, Labels, Axis, Fonts, Markers, and Background will allow you to set the esthetics of your plot. A wide variety of options is available to make your plot look the way you would like it to look. For example, the plot above might look better if we labeled the X-axis, changed the title, made the drop shadow consistent, used a different marker for each point, and widened the line drawn:



Note that this is a screen shot of the dialog box not what the system tag would allow you to print or copy to the clipboard. Those pictures would not have the Properties button, the dialog frame and title bar, etc.

## Two-Dimensional Potentials

Two-dimensional potential energy plots operate essentially identical to the one-dimensional potential energy plots except that you cannot use the current selection to describe the independent variables. You must use named selections, PLOT1 and PLOT2. A current selection can describe one variable but not two! Thus, you must first name PLOT1 and PLOT2 as any bond length, angle or torsion. The two variables need not be of the same kind. Clicking one the menu item Potentials... will then initiate the creation of a 3D graph describing the energy as a function of PLOT1 and PLOT2.

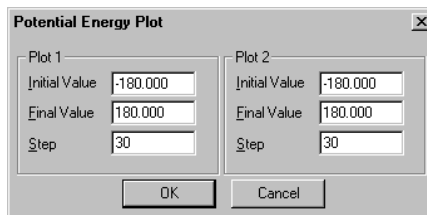
### Example

A simple example of a 2D potential energy plot is a plot of the energy as a function of the  $C_1-C_2$  and  $C_2-C_3$  torsional angles in n-Pentane. These two torsions describe the carbon backbone and the energy is at a nominal *minimum* when these angles describe a

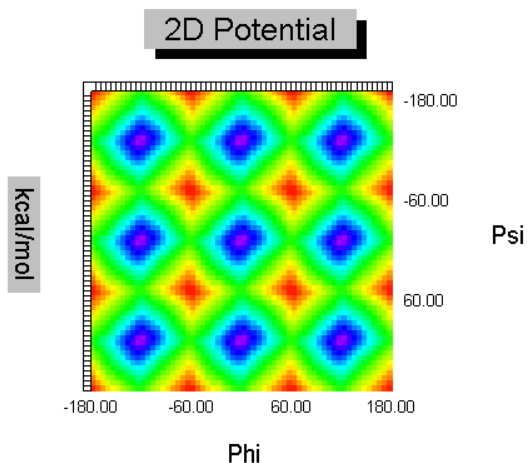


gauche orientation (60 degrees, 180 degrees, etc). The energy is at a nominal *maximum*, on the other hand, for eclipsed orientations (0 degrees, 120 degrees, etc.).

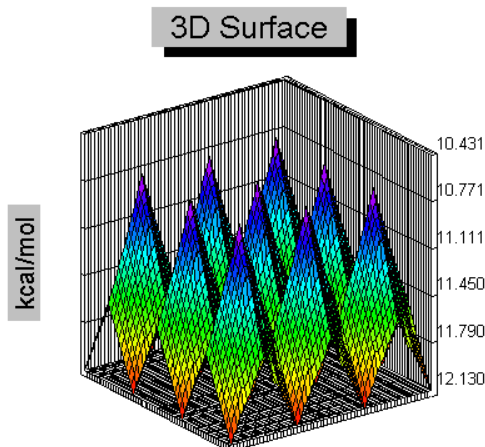
The setup for this plotting is:



An MM+ calculation of this potential energy is shown below looking directly down on the surface. The minima (red) occur at values of -180, -60, 60, and 180 degrees (the range of the plot is from -180 to +180 degrees). The maxima occur for values of -120, 0, and +120 degrees.



The same data but with a different view of the 3D surface is shown below:



## Cautions

Two-dimensional potential energy plots may require time and patience to obtain. For large numbers of points, the delay in displaying the data can be significant. HyperChem may also add points to obtain a smoother final plot.

## Conformational Search

### Overview

The Conformational Search module is a program for finding low energy conformations of molecular systems in HyperChem by varying user-specified dihedral angles. The method involves random variation of dihedral angles to generate new structures and then energy minimizing each of these angles. Low-energy unique conformations are stored while high-energy or duplicate structures are discarded.

The generation of new starting conformations for the energy minimization uses random variation of dihedral angles. Rotation is used for acyclic bond dihedral angles. For dihedral angles in a ring, dihedral angles are rotated by the "torsional flexing" motion of Kolossvary and Guida [I. Kolossvary and W. C. Guida, *J. Comput. Chem.*, **14**, 691 (1993)], which effectively leads to new ring confor-

mations while avoiding large atomic displacements that can decrease the efficiency of optimization.

Features of Conformational Search include:

- It can handle systems containing rings, there is effectively no software limit for the size of system or number of torsions to vary.
- The search employs random variation of dihedral angles with options for choosing initial structures, determining duplicate conformations and energetic tests for accepting new conformations.
- It uses any selection that you have set while minimizing, so part of a system can be kept fixed or treated with semi-empirical quantum mechanics in HyperChem.
- It can automatically determine equivalent atom orders for RMS comparison of conformations.
- Results are presented in a spreadsheet and can be copied via the Windows clipboard to other applications.

The Conformational Search executable file is HCCONF.EXE, installed in your Program directory. The initialization file, HCCONF.INI, is stored in your WINDOWS directory and will be created automatically by Conformational Search.

## Tutorial

The following steps should be used to run Conformation Search:

1. Conformational Search may be started by clicking on the Conformational Search menu item of the Compute menu in HyperChem. HyperChem should have a molecular system in the workspace and the desired calculation options set up (via the Setup menu). It is a good idea to run a single point calculation (via Compute/Single Point) first to check for required parameters.
2. The dihedral angles to be varied must be set up as named selections in HyperChem. If you wish to choose the names for the selections or only wish to vary a few dihedral angles, use the Select/Name Selection menu in HyperChem to name each of the dihedral angles that you wish to be varied in the search.

3. Select the dihedral angles to be varied by using the Options/Choose Torsions menu item and adding items from the Named Torsion Angles list to the Torsions To Vary list in the Setup Torsions dialog box. You may wish to use Shift-Enter or Control-Enter to select multiple dihedral angles in the list before clicking on the Add button.
4. If the molecular system includes rings, use the Options/Define Rings menu item. For each ring, select it in HyperChem and then press the Add button.
5. Choose the rest of the search options under the Options menu. These include the choice for initial structures, pre-optimization tests for non-bonded distance or duplication, the optimization parameters, and the geometric and energetic criteria for accepting conformations. See the Background section for details.
6. Use the Run/Start menu (or press the F2 key) to start the search. If you have selected the Auto Save option, the Auto Save File Name dialog appears and you can specify the file name to use for automatically saving results. Updated results of the search are displayed as the search progresses, and if the Auto Save option is on, the updated results and conformation atomic coordinates are written out to file.

The search may be suspended by clicking on the Cancel menu, or by pressing Alt+C on the keyboard, while HyperChem is optimizing. It may be resumed via the Run/Resume menu or by pressing F3 on the keyboard.

If a search isn't running, you can put tabular results in the Windows clipboard, or use the clipboard to put a conformation into HyperChem. Select cells in either of the output tables and use the Edit/Copy menu to copy the selected cells, plus row and column headings for the Conformations Found table, to the Windows clipboard. These may then be copied into another application, such as a spreadsheet, plotting program or word processor. If you select a cell or row in the Conformations Found table and use the Edit/Put Molecule menu item, the selected conformation will be put into HyperChem, with the previously found potential energy given on the status line.

## Background

The goal of a conformational search is usually to determine conformations of a molecular system that are low in potential energy. As well as determining the global minimum of the potential energy surface, it is important to determine all minima that will be thermally populated and will therefore influence macroscopically observable properties of the system.

For small systems with few conformational degrees of freedom, a search by systematic variation of all conformational parameters is possible. However the number of combinations to be systematically explored grows exponentially with the number of degrees of freedom. The straightforward systematic search quickly becomes prohibitive for systems of more than few conformational parameters.

Several approaches have been applied to the problem of determining low-energy conformations of molecules. The field has been reviewed recently [A. E. Howard and P. A. Kollman, *J. Med. Chem.*, **31**, 1669 (1988)]. A common theme is to limit the exploration of conformational space by varying only those parameters that affect the potential energy the least. Since molecular flexibility is usually due to rotation of unhindered bond dihedral or torsion angles, with little change in bond lengths or bond angles, a frequent choice (used here) is to only consider the variation of bond torsion angles. Another strategy, to reduce computation, is to discard structures without the cost of energy evaluation or minimization by using quick tests for unfavorably close non-bonded atom separations and for ring closure bond lengths that are too long or short. Many of the approaches share a common overall design, also employed here.

The repeated steps are:

1. Select an initial structure.
2. Modify the initial structure by varying geometric parameters.
3. Geometry optimize the modified structure to a energy-minimized conformation.
4. Compare the conformation with those found previously and accept if unique and if the energy satisfies a criterion.

The methods differ in the details of these basic steps.

## 1. Selection Of An Initial Structure

For step 1, the initial structure may be one that is unchanged for the whole search, is the most recently accepted conformation or is chosen from a set of previously accepted conformations. If one is applying systematic variations in step 2, keeping the initial structure unchanged will avoid repeating the modified structures that are minimized. The choice of using the last accepted conformation, often referred to as a *random walk* scheme in Monte Carlo searches, is based on the observation that low-energy conformations tend to be similar, so starting from an accepted conformation should tend to keep the search in a low-energy region of the potential surface. As well as preferring low-energy regions, the search should not get trapped in a single low-energy region. One scheme, introduced in the Monte Carlo Multiple Minimum, or MCMM, method [G. Chang, W. C. Guida and W. C. Still, *J. Am. Chem. Soc.*, **111**, 4379 (1989)] is called *usage directed*. It seeks to uniformly sample low-energy regions by cycling through all previously accepted conformations (in order from lowest energy to highest) when selecting each initial structure. Several comparisons [G. Chang, W. C. Guida and W. C. Still, *J. Am. Chem. Soc.*, **111**, 4379 (1989), M. Saunders, K. N. Houk, Y.-D. Wu, W. C. Still, M. Lipton, G. Chang, and W. C. Guida, *J. Am. Chem. Soc.*, **112**, 1419 (1990), C. E. Peishoff and J. S. Dixon, *J. Comput. Chem.*, **13**, 565 (1992)] find the usage-directed scheme preferable to simply using the last accepted structure. In this conformational search module either using the last accepted conformation (*random walk*) or the usage directed scheme may be used.

## 2. Variation Of Geometric Parameters

For step 2, the variations are either systematic or random. As pointed out above, systematic variations applied to an unchanging initial structure can avoid duplication and can also exhaustively search for low energy conformations, but the number of variations becomes prohibitive for all but the simplest systems. One approach to reduce the dimensionality of systematic variations, that is used in a development of MCMM called SUMM [J. M. Goodman and W. C. Still, *J. Comput. Chem.*, **12**, 1110 (1991)], is to first exhaust variations at a low resolution, then exhaust the new variations allowed by successively doubling the resolution. Random approaches may randomly choose a new value for one or more geometric parameters from a continuous range or from sets of discrete values. The main disadvantage with

random variations is that as the search progresses, more and more of the variations lead to recurring conformations. To reduce repetition several methods have some sort of comparison, preferably quick, with the sets of previous structures resulting from step 2, from step 4 or from both. In this module, the pre-optimization torsion test compares torsion angles with those of previous structures from step 4, and goes back to step 1 if they are within a maximum deviation.

The structures resulting from step 2 may be tested for close interatomic distances. If close contacts are detected, then the search goes back to step 1.

In searches that vary torsion angles, cyclic systems are treated as special cases since endocyclic torsion angles cannot be varied independently of other internal coordinates. Two types of approaches are common.

The first type of endocyclic torsion treatment involves the user selecting a bond which can be broken (or an atom that can be "split") to allow torsion angles to change and then reform the bond (or atom) to close the ring again. Since most of the resulting structures will have energetically unfavorable ring closures, such as very long bonds, it is usual to discard variations unless the resulting ring closure is within a user-specified tolerance. An undesirable consequence of this approach is that considerable strain can exist in the ring closure bond, so that the conformation is far from a local minimum and the optimization in step 3 either takes a long time or does not find a low-energy minimum. The results can also be very sensitive to the bond chosen for breaking the ring.

The second type of treatment for varying endocyclic torsions is to vary the angles in a concerted fashion that hopefully minimizes the formation of long bonds. The main advantage of this technique is that the resulting structures have low strain and thus tend to be optimized more quickly in step 3. There is also the advantage that the effort of generating and checking structures with poor ring closures is avoided. A disadvantage can be that some low-energy conformations cannot be reached. Some of the methods reported are *corner flapping* [H. Goto and E. Osawa, *J. Am. Chem. Soc.*, **111**, 8950 (1989)], *edge flipping* [H. Goto and E. Osawa, *J. Chem. Soc. Perkin Trans. 2*, **1993**, 187], and *torsional flexing* (TFLEX) [I. Kolossváry and W. C. Guida, *J. Comput. Chem.*, **14**, 691 (1993)]. Corner flapping has been shown to miss conformations, but this has not been shown for the latter two methods of varying endocy-

clic torsions. Torsional flexing compares favorably with other methods in finding almost all known minima for well-studied systems while requiring less computational effort, and is the method employed for cyclic torsions in this module.

### 3. Geometry Optimization

In step 3 geometry optimization is used to find a local minimum on the potential surface starting from a structure resulting from step 2. The choice of optimizer has minor effect on the conformational search, though it is preferable to employ an optimizer that converges quickly to a local minimum without crossing barriers on the potential surface. The latter point is important for not missing local minima, but usually a characteristic of an efficient optimizers.

### 4. Comparison With Previously Accepted Conformations And Energy Criterion

Two types of criteria are used to decide when to accept structures, geometric and energetic.

Geometric comparisons are made with previously accepted conformations to avoid duplication. Conformations are often compared by the maximum or r.m.s. deviation for internal coordinates (usually dihedral angles), interatomic distances or least-squares superposition of conformers. The comparisons may only use a subset of atoms (e.g. ignoring hydrogens) to speed comparison. If the comparison parameter falls within a specified tolerance then the duplicate is discarded.

The comparisons can be complicated by topological symmetry, particularly for cyclic systems. In these cases several mappings or re-orderings may need to be considered to avoid accepting duplicate structures.

Another geometric test that may be performed is to check for inversion of chiral centers, and discard any conformations where chiral centers have been inverted. Although this test could be performed before step 3, it is best performed afterwards since geometry optimization can invert chiral centers.

The energetic test for accepting a new conformer (if it is not a duplicate) may be of two types, a simple cutoff relative to the best energy found so far, or a Metropolis criterion where higher energy



structures are accepted with a probability determined by the energy difference and a temperature. In Monte Carlo searches that use the random walk option for step 1, it is often necessary to raise the temperature used in the acceptance test to increase the probability of accepting a high energy conformation to cross a potential barrier. Temperature adjustment is typically done after repeatedly finding the same duplicate conformation or repeatedly rejecting new conformations based on the Metropolis criterion [B. von Freyberg and W. Braun, *J. Comput. Chem.*, **12**, 1065 (1991)].

## Options Supported In Conformational Search

Conformational Search offers the following options for steps 1 to 4 discussed above:

1. For step 1, either the last accepted structure or the usage directed scheme may be used to choose the initial structure. As noted above, comparative studies have found the usage directed scheme to be superior for quickly finding low-energy conformations. This may be set by the Options/Search Method menu.
2. In step 2 random variations are applied to the initial conformation chosen in step 1. Random combinations of one or more dihedral angle variations are applied. In order for Conformational Search to treat ring systems, you must use the Options/Define Rings menu item to tell Conformational Search where rings are in your system. If the dihedral angle is in a ring the variation is a torsional flexing, otherwise a simple rotation is used. The variations may be restricted to a specific range or a specific set of values (using the Options/Setup Ranges menu). A non-bonded distance test and torsion duplication test may optionally be performed before proceeding to step 3 (using the Options/Tests menu).
3. The optimization step may use any of the optimizers available for the particular type of calculation in HyperChem. Since the structures from step 2 may be quite strained, it is preferable to use an optimizer that is globally convergent, such as Polak-Ribiere or Fletcher-Reeves. Please see the HyperChem documentation for further details.
4. Both geometric and energetic tests are used as criteria in accepting new conformations. Conformations with chiral centers that have inverted may optionally be discarded. Up to

three tests may be used to compare conformations in the post-optimization check for duplication: energy difference, maximum deviation of torsions being varied, and r.m.s. deviation from a least squares overlay of specified atoms. More than one order for comparing atoms may be specified where symmetry considerations warrant it. The tests to be used for checking chirality and determining duplication are specified by the Options/Tests menu. The energetic test may be either a cutoff relative to the best energy or a Metropolis criterion. Temperature adjustment may be used with the Metropolis criterion. The energetic tests for acceptance are set by the Options/Search Method menu.

The options available in Conformational Search may be chosen to duplicate published search methods. Choosing the usage directed scheme, with restricted ranges for torsional flexing and a non-Metropolis cutoff, gives the successful scheme of Kolossvary and Guida. Using random walk choice of initial structure with a Metropolis energy criterion gives a conventional Monte Carlo search, similar to that of von Freyberg and Braun (which uses temperature adjustment) or that of Li and Scheraga [Z. Li and H. A. Scheraga, *Proc. Natl. Acad. Sci. USA*, **84**, 6611 (1987)].

## Reference

This section describes the main window, menus and dialog boxes of Conformational Search.

## Main Window

The main window for Conformational Search has two tables for displaying search results.

<b>Search Progress</b>	This table summarizes the progress of the conformational search.
<b>Conformations</b>	This indicates the number of unique conformations that have been kept.
<b>Lowest found</b>	This shows the lowest energy of the conformations kept so far.

<b>Highest kept</b>	This shows the highest energy of the conformations kept so far.
<b>Accept rate</b>	This shows the ratio of the number of conformations kept over the number of conformations subjected to the acceptance test. It does not include duplicates or conformations rejected for reasons other than high energy. If the Metropolis criterion is being used, this shows the rate that it is being passed.
<b>Iterations</b>	This shows the number of times that initial structures have been chosen and variations applied.
<b>Optimizations</b>	This shows the number of times that HyperChem has performed optimizations.
<b>Energies</b>	This shows the number of energy and gradient evaluations that HyperChem has performed while optimizing.
<b>Unconverged</b>	This shows the number of optimizations that did not converge within the RMS gradient threshold specified in the Test Options For Search dialog box.
<b>Close contacts</b>	This shows the number of trial structures that were rejected before optimization due to non-bonded atoms being closer than the threshold specified in the Test Options For Search dialog box. The option that this reflects is set in the Test Options For Search dialog box

<b>Torsion tests</b>	This shows the number of trial structures that were rejected before optimization due to torsion angles differing from a previously accepted conformation by less than the threshold specified in the Test Options For Search dialog box. The option that this reflects is set in the Test Options For Search dialog box
<b>Wrong chirality</b>	This shows the number of optimization results that were rejected because one or more chiral centers changed chirality. The option that this reflects is set in the Test Options For Search dialog box
<b>High energy</b>	This shows the number of conformations that have been rejected by the energetic acceptance criterion.
<b>Conformations Found</b>	This table details information about each of the conformations that have been kept. The conformations are sorted in order of increasing energy. The information for each conformation corresponds to the lowest energy duplicate conformation.
<b>Index</b>	The index of the conformation.
<b>Energy</b>	The energy of this conformation.
<b>Gradient</b>	The RMS gradient of this conformation.
<b>Found</b>	The number of times this conformation has been found.
<b>Used</b>	The number of times this conformation has been used as an initial structure.
<b>Torsion Angle Values</b>	The last columns hold the values of the torsion angles that are being varied. The columns are labeled with the selection names of the torsions.

## **File Menu**

The file menu is used to read and write files, and it contains:

<b><u>N</u>ew</b>	Use <b>New</b> to clear the workspace. If changes have been made, you will be offered the chance to save the system to a file.
<b><u>O</u>pen...</b>	Use <b>Open...</b> to read in results from a previously saved search. You will be presented with the <b>File</b> dialog box for selection of the file. Conformation Search uses Hypercube Conformational Search (HCS) files for saving results. If the file holds a molecular system it will be loaded into HyperChem, and the search will be set up to resume (via <b>Run/Resume</b> ) if possible. All the options under the <b>Options</b> menu are restored from the file.
<b><u>S</u>ave</b>	Use the <b>Save</b> menu item to save a copy of the current search into the current file. If there is no current file, then this item is the same as the <b>Save As...</b> menu item displaying the File Save dialog for selection of a file name.
<b><u>S</u>ave <u>A</u>s...</b>	Use <b>Save As...</b> to save the current system into a named file. This item invokes the File Save dialog for you to use in selecting a file name.
<b><u>A</u>uto Save...</b>	Use this menu to display the Auto Save Option dialog box, which can set the option to automatically create and update a results file as the search proceeds.
<b><u>E</u>xit</b>	Use <b>Exit</b> to exit Conformational Search. If search results have changed since the last time the file was saved, then you will be offered the chance to save changes.
<b><u>P</u>revious file names</b>	As you read and write any files, the last four files you accessed will be available as items at the end of the file menu. Initially, these menu items are absent (you haven't referred to any files yet), but they will appear as you use files with Conformational Search. If you click on one of these file names, it will open that file just as if you had selected it with <b>File/Open</b> .

## **Edit Menu**

The Edit menu is used to copy the tables from the main window to the clipboard or put conformation into HyperChem.

<b><u>C</u>opy</b>	This will copy any selected cells from the tables into the clipboard. Only cells from one table at a time can be copied. If the selected cells are in the Conformations Found table, the row and column headings will be copied to the clipboard as well. The clipboard format is text, with tab characters separating the columns.
<b><u>P</u>ut Molecule</b>	If a cell in the Conformations Found table has the focus (indicated by a dotted line around the cell), this will load that conformation into HyperChem and display the previously found potential energy in the status line.

## **Options Menu**

This menu sets up options for performing the conformational search. It contains the following menu items:

<b><u>C</u>hoose <u>T</u>orsions...</b>	This menu item causes Conformational Search to query HyperChem for its named selections and determines which of these are torsion angles. It then displays the <b>Choose Torsions</b> dialog box to allow you to specify which named torsions you wish to vary in a search.
<b><u>D</u>efine Rings...</b>	This brings up the <b>Define Rings</b> dialog box, which you must use to identify rings containing torsions that you wish to vary. Conformational Search will not be able to vary torsions in a ring if you do not use this menu.
<b><u>S</u>etup Ranges...</b>	This brings up the <b>Setup Ranges</b> dialog box, which lets you specify ranges for how many torsion to vary simultaneously and how much to vary them by.

<b>S</b> earch <b>M</b> ethod...	This brings up the <b>Search Method</b> dialog box, which lets you specify the method for choosing initial structures and the method for deciding to accept new conformations.
<b>T</b> ests...	This brings up the <b>Test Options For Search</b> dialog box, which lets you specify pre-optimization, chirality and duplication tests.
<b>O</b> ptimization...	This brings up the <b>Optimization Options</b> dialog box, which lets you specify the termination conditions for optimizations and the frequency of HyperChem screen updates during optimization.
<b>R</b> andom <b>S</b> eed...	This brings up the <b>Random Number Seed</b> dialog box, which lets you specify the seed to be used by the random number generate when new (rather than resumed) searches are run.
<b>L</b> imits...	This brings up the <b>Limits For Search</b> dialog box, which lets you specify how long to let the search run or a maximum number of conformations to keep.

The options should be set up after you have read in or created your system in HyperChem and have selected and named the torsion angles that you wish to vary.

## **Run Menu**

This menu is used to start a new search or resume a previous search. It contains the two items:

<b>S</b> tart	This will clear any previous search results, prompt you to save previous results if you have not already done so, prompt you for a file name to save new results to if the <b>Auto Save</b> option is turned on, and then start the new search.
<b>R</b> esume	If active, this menu can be used to resume a previous search, either read in from a file or previously suspended from the current session of Conformational Search.

## **Cancel Menu**

This menu is active when a search is running and HyperChem is optimizing. You may click on it, or use Alt+C on the keyboard to suspend or stop the search.

## **Help Menu**

This menu can be used to view information about the Conformational Search module. It contains:

---

<b><u>A</u>bout...</b>	Displays the About dialog box, which gives version and copyright information.
------------------------	---

---

## **A**uto Save Option Dialog Box

Check the check box in this dialog if you wish to have the search results saved to a file that is automatically updated as the search proceeds. This can be useful for long, unattended searches where output might otherwise be lost from power outages or other abnormal program termination. If this option is activated, you will always be asked to specify a file name when starting or resuming a search. The Auto Save File Name dialog box, which differs only in title from the standard File Save dialog box, is used for specifying the name of the file.

## **C**hoose Torsions Dialog Box

Use the Choose Torsions dialog box to specify which torsion angles should be varied in Conformational Search. You must first use HyperChem to select and name the torsion angles, and then use the Options/Choose Torsions menu item to invoke this dialog. The options in this dialog box are:

<b><u>N</u>amed Torsion Angles</b>	Lists the named selections which specify bond torsion angles. You may select one or more items in this list. To select a contiguous range, click on the first and shift-click on the last that you desire. You may use control-clicks to independently select or deselect members of the list. Double-clicking on an item will add it to the Torsions To Vary list.
------------------------------------	---



<b>Torsions To Vary</b>	Lists the named bond torsion angles which are to be varied in the search. You may select one or more items in this list, as in the above list. Double-clicking on an item will remove it from the list.
<b>Add -&gt;</b>	Adds the selected names in the Named Torsion Angles list to the Torsions To Vary list.
<b>&lt;- Del</b>	Removes the selected names in the Torsions To Vary list.
<b>Add All</b>	Adds all names in the Named Torsion Angles list to the Torsions To Vary list.
<b>Del All</b>	Removes all names from the Torsions To Vary list.
<b>OK</b>	Close the dialog box and accept the changes made.
<b>Cancel</b>	Close the dialog box without accepting the changes made.
<b>Define Rings...</b>	Display the Define Rings dialog box.
<b>RMS Atoms...</b>	Display the Atoms To RMS Compare dialog box.
<b>Set Ranges...</b>	Display the Setup Ranges dialog box.

## Define Rings Dialog Box

Use this dialog box to define atoms whose bonds form rings. This is necessary for Conformational Search to distinguish acyclic bond torsion angles from cyclic bond torsion angles. Any selection that you had is temporarily cleared until the dialog box is closed, when it will be restored. Once the dialog box is shown, you switch to HyperChem and select a ring (e.g. by double-clicking on a bond in the ring). You then switch back to this dialog box and click on Add to define the ring. Repeat the process for each ring that contains torsions to be varied. If you make a mistake, you can select the incorrect member of the list and use Delete to remove it.

*Note:* If you are using united atoms in HyperChem, the invisible united hydrogens are selected when the carbon atoms that they are united with are selected. Thus united hydrogens attached to ring carbons will show up in the Rings list but they will be ignored by Conformational Search. If you find this behavior annoying, you can avoid it by using Build/All Atoms in HyperChem when you are defining rings and then you can switch back to united atoms for the search after all rings are defined.

<b>R</b> ings	This lists the atoms that you have indicated as forming rings. The atoms are listed by atom number within a molecule followed by a comma and the molecule number and a space.
<b>A</b> dd	Use this button to add the currently selected atoms in HyperChem to the list.
<b>D</b> elete	Use this button to delete the selected set of atoms from the Rings list. You only need to do this if atoms have been added (other than extended hydrogens) that are not in rings.
<b>O</b> K	Close the dialog box and accept the changes made.
<b>C</b> ancel	Close the dialog box without accepting the changes made.

## Setup Ranges Dialog Box

Use this dialog box to specify the ranges for how many torsion angles to vary simultaneously and for the variation of the torsions.

<b>R</b> anges for <b>v</b> aria- <b>t</b> ions	This section is used to set the ranges for variation of bond torsion angles.
<b>N</b> umber of <b>s</b> imul- <b>t</b> aneous <b>v</b> ariations	Use this to set the range of how many torsions will be varied simultaneously. If the range includes values larger than the total number of torsions to vary they will not be used. For acyclic systems this number can be as large as the actual number of torsions to vary. For cyclic systems larger values can lead to severely distorted structures after variation, resulting in reduced efficiency of the search.
<b>R</b> ange for <b>a</b> cylic <b>t</b> orsion <b>v</b> ariation	Use this to set the range for changing bond torsion angles that are not in rings.

<b>Range for ring torsion flexing</b>	Use this to set the range for changing bond torsion angles that are in rings. The torsion angle will actually vary by twice the given range, since each end is rotated by the same amount but in opposite directions.
<b>OK</b>	Close the dialog box, accepting the changes made.
<b>Cancel</b>	Close the dialog box without accepting the changes made.

### **Search Method Dialog Box**

Use this dialog box to specify which type of conformational search to perform.

<b>Initial Conformations</b>	This specifies the method to use for choosing initial conformations.
<b>Usage Directed</b>	Uses the lowest energy previously found conformation that has not already been used. Once all previously found conformations have been used, it cycles through them again.
<b>Random Walk</b>	Uses the most recently accepted conformation.
<b>Acceptance Energy Criterion</b>	This section can be use to specify the energetic criteria for deciding whether to accept a new conformation or not.
<b>Maximum</b>	Choose this option to set an upper limit, relative to the lowest energy conformation found, for accepting new conformations. Type a number for the relative energy cut-off that you desire in the text box.

<b>Metropolis Criterion</b>	Choose this option to use a Metropolis criterion for accepting new conformations, where the probability of accepting a higher energy conformation than the initial structure increases with a higher temperature. (A lower energy structure will always be kept.) You should set a normal temperature and a much higher temperature to switch to after a specified number of consecutive times finding the same conformation or rejecting new conformations.
<b>OK</b>	Close the dialog box, accepting the changes made.
<b>Cancel</b>	Close the dialog box without accepting the changes made.

For a traditional Metropolis Monte Carlo search, albeit with torsional flexing for cyclic torsions, use the Random Walk choice for initial structures and the Metropolis Criterion for acceptance.

For an MCMM search, or more specifically the TFLEX variant of MCMM, use the Usage Directed choice for initial structures and a Maximum energy range for acceptance.

### Test Options For Search Dialog Box

Use this dialog box to specify pre-optimization tests, chirality checking, and the tests to use to determine duplicate conformations.

<b>Pre-Optimization Checks</b>	These tests are performed before optimization. If they fail the optimization and subsequent steps are skipped.
<b>Skip if atoms closer than</b>	Check this box and enter a distance to specify a minimum allowed separation between non-bonded atoms.

- Skip if torsions within** Check this box and enter an angle to try to avoid finding a duplicate conformation by starting with torsion angles close to those of a conformation already accepted.
- Post-Optimization Check** Check this box to discard conformations where chiral centers differ from those in the starting structure. Atoms with only three bonded neighbors are not considered in this test.
- Duplication Tests** This sets the tests for determining duplicate conformations. All checked tests must be passed for a conformation to be considered a duplicate. For efficiency the testing is aborted as soon as any test fails. The tests are performed in the order listed, with the quickest tests coming first for efficiency.
- Energy within** Check this box and enter an energy difference to compare energies as a quick test for duplicates. An appropriate value depends on the optimization convergence criterion. This option is inappropriate if there are regions of the system where you are not interested the structure but structural changes affect the energy. In such cases you may want energetically different structures to be considered duplicates based on only a subset of the structure.
- Varied torsions within** Check this box and enter an angle to compare values for the torsions to varied. This will only compare dihedral angles of the same selection name, so it should not be used if there are equivalent dihedral angles.

- RMS error within** Check this box and enter a distance to calculate the residual root mean square error from a least squares overlay of the atoms in the conformations. This option is relatively expensive to calculate, so it is advisable to use one or both of the above duplication tests when possible to reduce the number of RMS error calculations. The following three options affect the calculation of the RMS error.
- Ignore hydrogens in RMS fit** Check this box to ignore all hydrogen atoms during the calculation of the RMS error. (United hydrogens are unaffected by this option, since they are always ignored.) This option can significantly reduce the time for calculating RMS error, and since hydrogen positions are rarely of interest, it should almost always be used.
- Use equivalent atom orders** Check this box to have Conformational Search scan the system to determine equivalent orders for the atoms being compared to use for calculating the RMS error. If checked, the RMS error test will only fail if the error for *all* of the orders is less than the threshold value. This option can dramatically increase the startup and running times for a search, so it should be used only when the system is known to have exactly equivalent atoms among those that are being compared. For efficiency only the atoms to compare and those atoms that connect them are checked for equivalency, so incorrect matching orders can be generated if too few atoms are chosen with the next option.

**Specify Atoms To Compare...** This button brings up the Atoms To RMS Compare dialog box which can be used to choose a subset of atoms for the RMS error test. Using a smaller number of atoms decreases the time taken to calculate the RMS error, but if too few are used spurious matches will be found. If no atoms are specified all atoms are used, but the option to ignore hydrogens takes precedence over this option.

**OK** Close the dialog box, accepting the changes made.

**Cancel** Close the dialog box without accepting the changes made.

### Atoms To RMS Compare Dialog Box

Use this dialog box to specify a subset of atoms to use in the RMS error calculation. Any selection that you had is temporarily cleared until the dialog box is closed, when it will be restored. Once the dialog box is shown, you switch to HyperChem and select the atoms you wish to include in the RMS error calculation. You then switch back to this dialog box and click on Accept. If you make a mistake, you can select the incorrect atoms and use Delete to remove them.

**Note:** If you are using united atoms in HyperChem, the invisible united hydrogens are selected when the carbon atoms that they are united with are selected. Thus united hydrogens attached to selected atoms will show up in the Atoms to compare list but they will be ignored by Conformational Search. If you find this behavior annoying, you can avoid it by using Build/All Atoms in HyperChem when you are choosing atoms to compare and then you can switch back to united atoms for the search.

**Atoms to compare** This lists the atoms that you have chosen to include in RMS error calculations. The atoms are listed by atom number within a molecule followed by a comma and the molecule number and a space.

**Accept** Use this button to add the currently selected atoms in HyperChem to the list.

<b>D</b> elete	Use this button to delete the selected set of atoms from the list.
<b>O</b> K	Close the dialog box, accepting the changes made.
<b>C</b> ancel	Close the dialog box without accepting the changes made.

## Optimization Options Dialog Box

Use this dialog box to set optimization options for HyperChem during the search.

<b>Optimization Termination</b>	The optimization will terminate successfully if the RMS gradient is sufficiently small or will terminate unsuccessfully if the maximum number of cycles is reached.
<b>RMS Gradient</b>	Use this to set the gradient threshold for considering an optimization to be converged. The default HyperChem value of 0.1 is often too large to find a true local minimum, a value of 0.01 or less will produce fewer spurious duplicates and can allow tighter duplication tests but can take much longer to reach convergence.
<b>Maximum Cycles</b>	Use this to set the maximum number of optimization cycles allowed to reach convergence. Because many of the starting structures are highly strained, a value larger than the HyperChem default is often appropriate. The value will also depend on the above setting, since it requires more cycles to reach smaller gradients. If this number is reached without meeting the gradient threshold, the conformation is discarded and the Unconverged count is increased.



<b>Screen update period</b>	Increase this value to speed up optimization by updating the HyperChem workspace less often. Since HyperChem takes some time to update its screen during optimizations, setting this to values greater than one can often speed up optimization significantly, particularly for relatively small molecules with molecular mechanics.
<b><u>O</u>K</b>	Close the dialog box, accepting the changes made.
<b><u>C</u>ancel</b>	Close the dialog box without accepting the changes made.

### **Random Number Seed Dialog Box**

Use this dialog box to specify a seed for the pseudo-random number generator for new searches. Numbers from the pseudo-random number generator are used to determine how many torsions to vary, how much to vary each of those torsions and for acceptance testing with the Metropolis criterion. For resumed searches, the previous pseudo random number is used.

<b>Random Number Seed</b>	This is used to specify the seed for the pseudo random number generator for new searches.
<b>From the computer's clock</b>	Check this to use a number from the computer's clock. This will effectively use a different seed every time.
<b>The number</b>	Check this and type the number that you wish to use if you want to be able to repeat a search.
<b><u>O</u>K</b>	Close the dialog box, accepting the changes made.
<b><u>C</u>ancel</b>	Close the dialog box without accepting the changes made.

## Limits For Search Dialog Box

Use this dialog box to specify termination conditions for the search or an upper limit on the number of conformations to keep.

<b>Termination</b>	This is used to specify when the search should terminate on its own.
<b>iterations</b>	The value that you type here specifies the maximum number of times that initial structures can be chosen and varied.
<b>optimizations</b>	The value that you type here specifies the maximum number of times that optimization will be run.
<b>Conformations</b>	This is used to specify the maximum number of conformations to keep
<b>conformations</b>	The value that you type here specifies the maximum number of conformations to keep. When they are found, lower energy conformations will replace higher energy conformations.
<b><u>O</u>K</b>	Close the dialog box, accepting the changes made.
<b><u>C</u>ancel</b>	Close the dialog box without accepting the changes made.

## Files

Conformational Search reads and writes two types of file, its initialization file and Hypercube Conformational Search (HCS) files.

### HCCONF.INI File

Conformational Search saves all option settings, previous files and window size and position in the file HCCONF.INI in your Windows directory when it is closed. This file is read and the settings are restored when Conformational Search is started. If the file can-

not be found or if there is an error reading it, default settings are used.

## HCS Files

The Hypercube Conformational Search (HCS) file format is a text file that contains all search options (in the same format as in HCCONF.INI), a HIN description of the system, the chosen torsions, the defined rings and the results of the search including Cartesian coordinates for all conformations that are kept.

This file format is HIN compatible in that HyperChem can read the HIN sections of these files and ignore the rest of the information without error. However if HyperChem is used to read such file and then it is written out again with the same name, all search information will be lost.

## Limitations

A search can only be stopped reliably while HyperChem is performing an optimization. Choosing Cancel in Conformational Search does not stop the optimization in HyperChem.

Conformational search sets up continuous DDE links to HyperChem to send commands, receive data, and monitor for changes in the molecular system. On slower computers, the monitoring of the HyperChem system can interfere with drawing atoms and bonds. If you find this to be a problem, create your system before starting Conformational Search and close Conformational Search when you are no longer using it.

## QSAR Properties

### Overview

Quantitative Structure-Activity Relationships (QSAR) are attempts to correlate molecular structure, or properties derived from molecular structure, with a particular kind of chemical or biochemical activity. The kind of activity is a function of the interest of the user: QSAR is widely used in pharmaceutical, environmental, and agricultural chemistry in the search for particular properties. The molecular properties used in the correlations relate as directly as possible to the key physical or chemical processes taking place in the target activity.

QSAR Properties is a module that, together with HyperChem, allows several properties commonly used in QSAR studies to be calculated. The calculations are empirical, and so, generally, are fast.

You can calculate the following properties using QSAR Properties:

- Atomic partial charges, using the Gasteiger-Marsili method.
- Van der Waals and solvent-accessible surface areas.
- Hydration energy (for peptides).
- Van der Waals-surface-bounded molecular volume.
- Solvent-accessible surface-bounded molecular volume.
- Log P (the log of the octanol-water partition coefficient), a measure of hydrophobicity.
- Molar refractivity.
- Polarizability.
- Mass.

The QSAR Properties executable file is HCQSAR.EXE, installed in your Program directory. Several parameter files are stored in the LIBRARY subdirectory, namely SASAPAR.TXT, VDWGRID.TXT, LOGP.TXT, REFRACT.TXT, POLARIZE.TXT, and ATOMMASS.TXT.

## Tutorials

The following tutorials demonstrate some ways that you can use QSAR Properties.

### 1. Calculating Partial Charges

This lesson shows you how to carry out simple calculations using QSAR Properties.

1. Draw ethanol ( $C_2H_5OH$ ) on the HyperChem screen, and use Display/Labels to turn on atomic charge labels.
2. Set QSAR Properties running by selecting the Compute/QSAR Properties menu in HyperChem, and click on the Partial Charges option.
3. Ensure that the Base on current charges check box is unchecked.

4. Press the Compute button, or press the Enter key, to set the calculation going. The status line gives information concerning the progress of the calculation, and the charges are updated on the screen at the end of the calculation. Note that oxygen is assigned a substantial negative charge (-0.39) and that this charge has been drawn almost equally from the hydroxyl hydrogen atom and the ethyl group.

*Note:* You do not have to display the charges in order to do this calculation; the display was included in this exercise for pedagogical purposes only.

## 2. Calculating Log P

This lesson shows how you can choose how to see the results of your calculations, and save them for later reference.

1. Draw butan-1-ol ( $\text{CH}_3\text{CH}_2\text{CH}_2\text{CH}_2\text{OH}$ ) on the HyperChem screen, and launch QSAR Properties, if you have not already done so.
2. Press the Log P button to select that property.
3. Choose Output Options... under the Output menu, and check the Results Window and Atomic Contributions check boxes. Then press OK to save these options.
4. Press the Compute button in the QSAR Properties main window to set the calculation going.
5. The net result appears on the QSAR Properties status line ( $\log P = 0.94$ ). The Results Window appears and contains the contribution to this value from each atom in the system, in addition to the net result. If you wish to save this information for later use, you can do so using the menus on the Results Window. The experimental  $\log P$  for butan-1-ol is 0.88.

## 3. Calculating Surface Areas

This lesson introduces you to the Options dialog box and to the two different methods of calculating surface areas.

1. Draw benzene on the HyperChem screen.
2. In the QSAR window, select the Surface Area (Approx.) property.

3. Select Calculation Options under the Options menu. In the Calculations Options dialog box, select the Van der Waals Surface Area option, and click on OK to save this choice.
4. Press the Compute button of the QSAR Properties main window to set the calculation going. The surface area of benzene is calculated.
5. Compare the result to the Surface Area (Grid) option, which is slower but more accurate. Note that the approximate surface area calculation does not evaluate hydrogen atoms attached to carbon atoms explicitly.

## Background

This section provides background on the methods used to calculate properties in QSAR Properties.

### Partial Charges

QSAR Properties uses an empirical model based on the Partial Equalization of Orbital Electronegativity (PEOE) approach of Gasteiger and Marsili [J. Gasteiger and M. Marsili, *Tetrahedron*, **36**, 3219-3288 (1980)]. It computes charges on the system in the HyperChem workspace, or selected part thereof, for use with molecular mechanics force fields or in QSAR studies. The calculation is very much faster than a semi-empirical computation.

### Limitations

The PEOE scheme of Gasteiger and Marsili is implemented, but their modifications for conjugated systems and small rings have not been incorporated into the module. All unparameterized atoms act as "walls": no charge is transferred to or from them during the course of the calculation.

Electronegativity is an empirical concept that has been closely related to atomic charges and bond polarity in a qualitative manner over many years. According to the electronegativity model, electronic (negative) charge flows from less electronegative atoms to more electronegative atoms. The electronegativity of an atom decreases as it accumulates negative charge, and thus the electronegativity of each atom will equalize, stopping charge flow at a certain point, which corresponds to the observed charge distribution. Gasteiger and Marsili incorporated these basic ideas into a

quantitative model for predicting "atomic charges" or "partial charges" of atoms in molecules.

If the electronegativity model is implemented in a complete manner, a clearly non-physical result is obtained. All electronegativities of atoms in the system would be identical, and so all atoms of a given type would have equal charges. A hydrogen atom bonded to oxygen will have the same charge as a hydrogen atom bonded to carbon. Gasteiger and Marsili limited the charge flow by constructing a "Partial Equalization of Orbital Electronegativity" (PEOE) scheme. The term "Orbital Electronegativity" is used to indicate that atoms are parameterized not only according to element, but also, in important cases, according to hybridization as inferred from the number of neighbors.

The computation of partial charges by the PEOE method is iterative. One starts with an initial set of charges, generally all zero for a neutral system (N.B., the results depend on the starting distribution of charge). Each atom has an electronegativity that depends quadratically on the charge:

$$\chi_i = A_i + B_i \cdot Q_i + C_i \cdot Q_i^2$$

The initial charge is zero, so the initial electronegativity is  $A_i$ . The values of  $A_i$ ,  $B_i$ , and  $C_i$  for each element (in its hybridization state) are the required parameters for the method.

The next step is to shift the charge from less electronegative atoms to more electronegative atoms. The general formula for the transfer of charge to atom  $i$  in the  $n^{\text{th}}$  iteration is

$$q_n^i = \frac{1}{2^n} \cdot \left[ \sum_j \frac{1}{\chi_j^+} (\chi_j^n - \chi_i^n) + \sum_k \frac{1}{\chi_k^+} (\chi_k^n - \chi_i^n) \right]$$

where  $j$  runs over atoms bonded to  $i$  that are more electronegative than  $i$ , and  $k$  runs over atoms bonded to  $i$  that are less electroneg-

ative than i.  $\chi^+$  refers to the electronegativity of the atom with a charge of +1. This slightly unsymmetrical expression ensures that charge transfer is balanced.

The electronegativity is prevented from completely equalizing by the damping factor  $(\frac{1}{2})^n$ , and this imposes the result that atoms are more sensitive to their immediate neighbors than to the effects of more distant neighbors; which only exert effects during later iterations.

One feature of the method is that it depends only on the molecular graph: elements and their connectivity. Geometry is irrelevant. This is useful in some circumstances, in that it avoids building a bias into atomic charges that may come from semi-empirical calculations at a particular geometry. It does, however, make the method insensitive to conformational changes.

### Surface Area (Approx.)

Computation of van der Waals and solvent-accessible surface areas are carried out by an approximate method due to Still and coworkers [W. Hasel, T. F. Hendrickson, W. C. Still, *Tet. Comput. Meth.*, **1**, 103-116 (1988), W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, *J. Am. Chem. Soc.*, **112**, 6127-6129 (1990)]. The solvent-accessible surface, in particular, is useful in QSAR studies.

The approximate method is fast, and generally accurate to within 10 per cent for a given set of atomic radii. Very compact structures may have greater errors than this.

The calculation employs atomic radii as given in the paper developing the method, and stored in the parameter file SASAPAR.TXT in the LIBRARY subdirectory. Hydrogens attached to carbon atoms are not considered explicitly, but are included implicitly with their carbon atom.

### Surface Area (Grid)

The grid calculation of (solvent-accessible or Van der Waals) surface area is much slower than the approximate calculation, but is more accurate for a given set of atomic radii. It is recommended that this method be used as a benchmark for the approximate surface area calculation. The grid method used is that described by Bodor *et al.* [N. Bodor, Z. Gabanyi, C. Wong, *J. Am. Chem. Soc.*, **111**, 3783 (1989)], using the atomic radii of Gavezotti [A. Gavezotti, *J. Am. Chem. Soc.*, **10**, 5220 (1983)]. These radii are



stored in the VDWGRID.TXT parameter file in the LIBRARY directory.

## Volume

This calculation is very similar to the Surface Area (Grid) calculation. It employs a grid method described by Bodor *et al.* The speed is similar to that of the Surface Area (Grid) calculation. The volume calculation employs the same VDWGRID.TXT parameter file as the Grid Surface Area calculation.

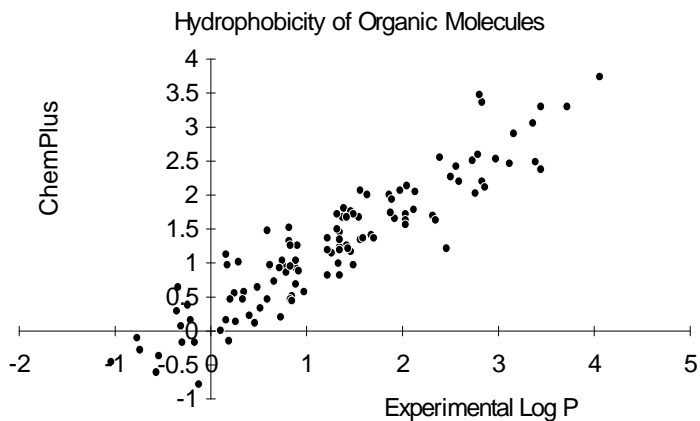
## Hydration Energy

The hydration energy is a key factor determining the stability of different molecular conformations. The method employed here is designed for peptides and proteins, and is that described by the research group of Scheraga [T. Ooi, M. Oobatake, G. Nemethy, H. A. Scheraga, Proc. Natl. Acad. Sci. USA, **84**, 3086 (1987)]. The calculation is based on exposed surface area, and employs the surface area as computed by the approximate method (above), weighted by atom type. Parameters here are limited.

## Log P

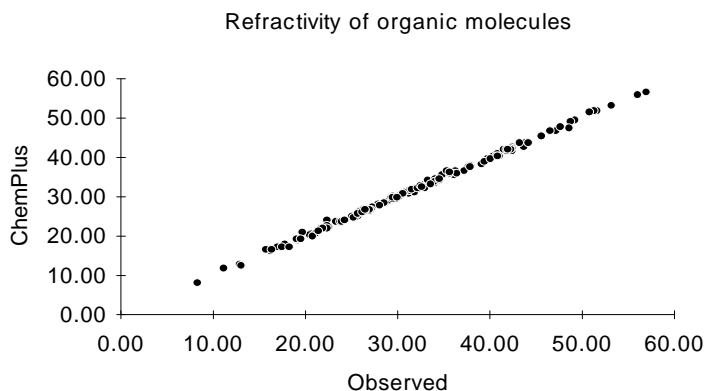
Calculation of Log P is carried out using atomic parameters derived by Ghose, Pritchett and Crippen [*J. Comp. Chem.*, **9**, 80 (1988)] and later extended by Ghose and coworkers [V. N. Viswanadhan, A. K. Ghose, G. N. Revankar, R. K. Robins, *J. Chem. Inf. Comput. Sci.*, **29**, 163 (1989)].

The parameters are stored in the file LOGP.TXT in your LIBRARY directory. The following graph gives an indication of the accuracy of the method.



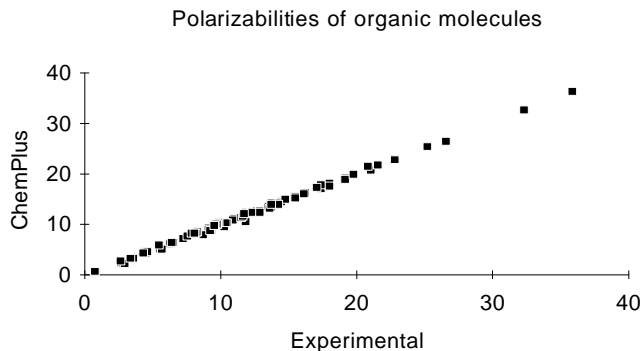
## Refractivity

The refractivity is estimated by the same method as Log P (above). Ghose and Crippen presented atomic contributions to the refractivity in exactly the same manner as to the hydrophobicity. [A. K. Ghose and G. M. Crippen, *J. Chem. Inf. Comput. Sci.*, **27**, 21 (1987), V. N. Viswanadhan, A. K. Ghose, G. N. Revankar, R. K. Robins, *J. Chem. Inf. Comput. Sci.*, **29**, 163 (1989)]. The parameters are contained in the file REFRACT.TXT in your LIBRARY directory.



## Polarizability

The polarizability is estimated from an additivity scheme presented by Miller [K. J. Miller, *J. Am. Chem. Soc.*, **112**, 8533 (1990)]. Different increments are associated with different atom types. For a variety of organic molecules, the estimates are accurate to within 1 to 3 %. However, the method does not distinguish isomers of any type. The parameters are held in the file POLARIZE.TXT in your LIBRARY directory.



## Reference

This section describes the main window, menus and dialog boxes in QSAR Properties. The DDE commands to which QSAR Properties can respond are described in a separate section.

## Main Window

The QSAR Properties main window contains a set of properties that can be calculated by the program. To fully specify some of the calculations you also need to set options in the Calculation Options

dialog box, displayed by selecting the Options/Calculation options menu item.

More information about the methods used to compute properties is provided in the Background section. Each calculation is carried out on the selected atoms only, if there is a current selection, or on the whole system if no atoms are selected. The following properties are available, and can be selected by checking the appropriate box in the main window:

**Partial Charges**      Computes the atomic charges on the atoms in the system using the Gasteiger-Marsili Partial Equalization of Orbital Electronegativity method. This is parameterized for the elements C, H, N, O, S and the halogens, although less reliable parameters are provided for other elements.

**Base on current charges**      Use this option for calculating charges for ions or for influencing the charge calculation. The starting charges for the calculations will be those currently in the system, rather than zero.

**Surface Area (Approx.)**      Use this for a rapid calculation of the surface area of a system using an analytical method. The type of surface (van der Waals or solvent-accessible) is specified in the Calculation Options dialog box. The method is parameterized for organic molecules.

**Surface Area (Grid)**      Use this for a slower but more accurate calculation of the surface area of a system using a grid method. The type of surface (van der Waals or solvent-accessible) and the density of grid points is specified in the Calculation Options dialog box. This method is of general applicability.

**Volume**      Use this to calculate the volume of a system using a grid method. The bounding surface (van der Waals or solvent-accessible) and the density of grid points is specified in the Calculation Options dialog box. This method is of general applicability.

- Hydration Energy** Use this to calculate the hydration energy of peptides. The calculation is based on the approximate surface area method. Use on systems other than peptides will not give meaningful results.
- Log P** Use this to calculate the hydrophobicity of a system, as measured by the log of its octanol-water partition coefficient. The method is parameterized for organic systems with a variety of heteroatoms.
- Refractivity** Use this to calculate the refractivity of a system. The method is parameterized for organic systems with a variety of heteroatoms.
- Polarizability** Use this to calculate the polarizability of a system. The method is parameterized for organic molecules with a variety of heteroatoms.
- Mass** Use this to calculate the mass of a system. The method is of general applicability.

## **File Menu**

Use this menu to read HIN files into QSAR Properties or to get a molecular system from HyperChem.

<b><u>N</u>ew</b>	Use <b>New</b> to clear the data in QSAR Properties.
<b><u>O</u>pen...</b>	Use <b>Open</b> to read a HIN file into QSAR Properties.
<b><u>G</u>et HyperChem</b>	Use <b>Get HyperChem</b> to read the current HyperChem system into the module, and sets the <b>Compute HyperChem System</b> flag (set manually under the <b>Options</b> menu) to Off. Subsequent calculations will be carried out on the system you have read in until you read in a new system or set the <b>Compute HyperChem System</b> flag to Off..
<b><u>E</u>xit</b>	Exit closes the QSAR Properties module

## **Options Menu**

Use this menu to specify options for calculating properties.

<b><u>C</u>alculation Options...</b>	Use <b>Calculation Options</b> to display the Calculation Options dialog box for specifying details of the calculation to be carried out.
<b>Compute <u>H</u>yperChem System</b>	If Compute HyperChem System is checked, then when the <b>Compute</b> button on the main window is pressed to start a calculation, QSAR Properties reads in the current system from HyperChem and carries out the calculation on that system. If <b>Compute HyperChem System</b> is unchecked, then QSAR Properties' current system is the one for which the calculation is carried out.
<b><u>O</u>utput To...</b>	Use <b>Output To...</b> to specify the level of detail and destination of the computed results via the Output Options dialog box.

## **Cancel Menu**

Press Cancel to stop a calculation. No results are recorded for that calculation.

## **Help Menu**

Use this menu to access information about this module.

---

<b><u>A</u>bout QSAR Properties...</b>	Brings up the About QSAR Properties dialog box.
--	---

---

## **Calculation Options Dialog Box**

Use this dialog box to specify particulars of the calculation you are about to perform. Only options that are appropriate for the current property type are active.

<b>Surface Options</b>	These options apply to both the Approximate and Grid-based surface calculations.
------------------------	--

<b>V</b> an der Waals Surface	Choose this option to calculate the exposed Van der Waals surface area of the molecule.
<b>S</b> olvent Accessible Surface	Choose this option to calculate the solvent-accessible surface area of the molecule, with a solvent radius as specified below.
<b>S</b> olvent Probe Radius	For Solvent-Accessible Surface Area calculations, specifies the effective radius of the solvent molecule (in Ångstroms). Generally, a value of 1.4 is used for water.
<b>T</b> hree Body Correction	Applies to the Approximate Surface Area and Hydration Energy calculations. The Three body correction makes the calculations more sensitive to conformational changes, at the cost of some increase in computational time.
<b>P</b> oints On Cube Side	Applies to the Grid calculations of Surface Areas and Volumes. Specifies how many points to use on each side of a cube for the grid. Generally, 20 is adequate for an accuracy of 2 or 3 per cent.
<b>O</b> K	Pressing <b>OK</b> saves the selected options and dismisses the dialog box.
<b>C</b> ancel	Pressing <b>Cancel</b> dismisses the dialog box without saving the altered options.

## Output Options Dialog Box

Use this dialog box to specify the destination and level of detail of the results of QSAR Properties calculations. The Output Options dialog box contains the following options:

<b>H</b> yperChem <b>S</b> tatus Line	Select this option to display the system result on the HyperChem status line.
---------------------------------------	---

<b>M</b> essage Box	Select this option to display a message box with the system result in it.
<b>A</b> tom Charge <b>L</b> abels	Select this option to replace the current atom charges with the atomic contributions to the system property.

**Note:** Use the above option with care. If you choose this option you will lose the current charges in your HyperChem system and replace them with numbers that are not atomic charges.

<b>R</b> esults Window	Select this option to place results in a results window, from where they can be saved to a file if desired.
<b>A</b> tomic Contribu- <b>t</b> ions	Sends more detailed results to the Results Window, where applicable.
<b>S</b> how <b>W</b> arning <b>M</b> essages	Checking this option to display a warning message when QSAR Properties lacks a parameter for an atom in a calculation. You may wish to uncheck this option when carrying out batch calculations, to avoid stopping the sequence of calculations.
<b>O</b> K	Press <b>OK</b> to save the selected options and dismiss the dialog box.
<b>C</b> ancel	Press <b>Cancel</b> to dismiss the dialog box without saving any changes to the options.

## Results Window Dialog Box

The Results Window dialog box collects the results of your calculations in a text window. The maximum amount of data that can be stored in this Window is about 32,000 characters: once this limit is reached the beginning of the contents "scrolls off" the Window. The Results Window has some menu items of its own to allow you to save the data to a file for later reference or to cut and paste to or from the clipboard. These are common Windows menu commands, and operate in the same manner as the described in the Introduction.

## About QSAR Properties Dialog Box

Provides version and copyright information about QSAR Properties.



## Batch Calculations Using DDE

QSAR studies are typically carried out on a set of molecules, and you can carry out batch calculations on a set of molecules stored in HIN files using QSAR Properties. These batch calculations are controlled by an external program or macro.

Like HyperChem, QSAR Properties does not provide a programming language for you to learn, but instead responds to a set of commands through the Microsoft Windows "Dynamic Data Exchange" mechanism. This means that you can write your batch control program from any other Windows application that has an appropriate language, whether that be a full program or a macro from within another application. Instead of having a single control language you can use any macro language or programming language that supports DDE.

A typical application would be to run a batch QSAR Properties calculation from a spreadsheet. The results are then fed straight back into the spreadsheet for analysis, without manual intervention.

### QSAR Properties As DDE Server

Like certain other modules of HyperChem, QSAR Properties interacts with core HyperChem as a Dynamic Data Exchange (DDE) client: it sends commands to HyperChem and makes requests of it, and HyperChem (the "server") follows the instructions.

QSAR Properties can also act as a DDE server, which means that you can send instructions to it from another program or from, say, a spreadsheet macro, and QSAR Properties will follow those instructions. This ability has been built into QSAR Properties to allow calculations on sets of molecules to be carried out in a "batch" model. The specifics of how to send DDE messages to a server depend on the particular program you are using.

Note that QSAR Properties cannot act as a DDE client (interacting with HyperChem) and a DDE server simultaneously.

### Starting A DDE Conversation

Before any DDE commands and requests can be sent to an application, a conversation must be started. This step may also be called "opening a channel to the application" or "opening a link with the application." The particular command that you send to open a DDE conversation depends on the macro or programming

language you are using, but necessarily requires an *application name*, and a *topic*.

Each DDE server has an *application name*, and supports one or more *topics*. For QSAR Properties, all DDE channels must be opened to "HCQSAR", and the topic of the conversation must be "System", which is a common name in the case that an application supports a single topic.

If opening a DDE conversation from a Microsoft Excel macro, the appropriate command is

```
=Initiate("HCQSAR", "System")
```

while to open a DDE conversation from a Microsoft Visual Basic program between a control on a Visual Basic form ("Text1", say) and QSAR Properties the command is:

```
Text1.LinkTopic = "HCQSAR|System"
```

## DDE Message Syntax

QSAR properties can interpret a small number of DDE messages. These messages may be commands, telling QSAR Properties to carry out an action, or a request, asking QSAR Properties to provide some information.

Each message must be sent as a single string to HyperChem, and the string must be enclosed in square brackets. The following table details the specific messages understood by QSAR Properties, and examples follow.

Message	Type	R/W	Use
qsar-option three-body	(boolean)	W	Include the three body terms in approximate surface area calculations.
qsar-option current-charges	(boolean)	W	Use the current HyperChem charges as a starting point for partial charges calculations.
qsar-option solvent-radius	(real)	W	Set the solvent radius to a real number (default is 1.4 Å).

<b>qsar-option van-der-waals-surface</b>	(boolean)	W	Choose the van der Waals surface for surface area and volume calculations. Setting to false uses a solvent-accessible radius.
<b>qsar-option points-on-side</b>	(even integer)	W	Set the density of points in the grid for grid-based surface area and volume calculations.
<b>qsar-property</b>	(partial-charges, approximate-surface-area, grid-surface-area, volume, hydration-energy, logp, refractivity, polarizability, mass)	W	Set the current property to be calculated to the argument.
<b>do-calculation</b>			Carries out a calculation using the current property and options.
<b>open-file</b>	filename		opens a HIN file for carrying out calculations, and sets the "Compute HyperChem System" flag to off.
<b>QSARResult</b>		R	Requesting this variable gives the result of the most recent calculation.

Boolean options can be indicated by Yes/No, or True/False (case insensitive).

### Example

To see how these commands can be put together to make a simple macro that carries out a set of calculations on named HIN files, we present an example. The following is an Excel macro.

Line number	Names	Excel command
1	Ctrl-R	<b>Compute.LogP</b>
2	Channel	=INITIATE("HCQSAR","System")
3		=EXECUTE(Channel,"[qsar-property logp]")

4		=WHILE(NOT(ISBLANK(SELECTION())))
5		=EXECUTE(Channel,"[open-file &SELECTION() &".hin]")
6	vlResult	=EXECUTE(Channel,"[do-calculation]")
7		vlResult=REQUEST(Channel,"QSARResult")
8		=FORMULA(vlResult,"rc[1]")
9		=SELECT("r[1]c")
10		=NEXT()
11		=TERMINATE(Channel)
12		=RETURN()

The "Line Numbers" column is not part of the macro, but is included to make the discussion more straightforward.

In order to run this macro you need two things. First, HCQSAR must be running. Second, you need a worksheet with a set of HIN filenames in a column, and the cell containing the first file name selected.

Line 1 presents the name of the macro and a shortcut key for invoking it; items specific to Excel.

The macro starts off by opening a link to HCQSAR (line 2), with topic "System" as described above. From here to the TERMINATE statement on line 11, the macro is talking to QSAR Properties.

Line 3 tells HCQSAR to set the property to be calculated to log P. Note the square brackets surrounding the message, and the double quotes, which delimit a string in Excel's macro language.

Lines 4 and 10 provide a loop wherein the active cell in the spreadsheet moves down a column one at a time until a blank line is reached, whereupon the loop is finished. The move down is done explicitly in Line 9.

Lines 5 to 8, inclusive, are the instructions sent to QSAR Properties for each HIN file named in the column. Line 5 opens the file. SELECTION() contains the text in the currently selected cell. The "&" characters concatenate the string, so if the text in the Excel cell is "Sample" then the string sent to QSAR Properties is "[open-file Sample.hin]". You should note that in this line, either the Excel current directory should be where the files are, or you may need to add a path description to the file name so that QSAR can find it.

Line 6 tells QSAR Properties to carry out a calculation (i.e., it presses the Compute button). Line 7 obtains the result of the cal-

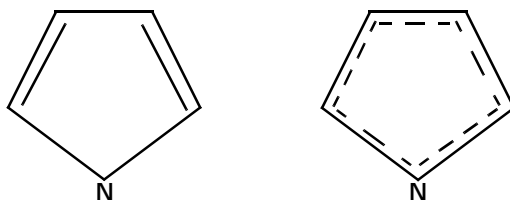
calculation from QSAR Properties, and stores it as a variable (vIResult), which is placed in the cell to the right (one column over) of that containing the file name by Line 8.

## Hints

### Drawing Heterocyclic Rings For Log P Calculations

Several of the methods included in QSAR Properties are based on the idea of *type* of atoms: different types are distinguished by their coordination and neighbors. In particular, the Log P and Refractivity methods include a wide range of atom types, to provide flexibility in different environments.

The type assigned to an atom by QSAR Properties depends on the way the bonding in a molecule is drawn in HyperChem. In most cases there is but a single way of drawing bond orders that is consistent with the rules of valence, but in some cases there are alternatives, and drawing molecules different ways can lead to significantly different answers. For instance, many heterocyclic rings can be depicted in alternative ways, using a localized or a delocalized depiction of the bonding. For instance, pyrrole can be depicted in either of the following two ways:



*For Log P and refractivity calculations, the general rule is that such structures should be drawn in a delocalized manner. Pyrrole, for instance, should be drawn with an aromatic ring as on the right rather than two localized double bonds as on the left.*

## Cancel

You can choose Cancel on the menu bar or press [Esc] when HyperChem is directing a chemical calculation or when it is reading in a structure from a file. When you L-click on Cancel, there might be a short delay before the process ends and HyperChem is fully active again.

During a calculation or reading a file, all menu items are gray and inactive except for Cancel. You can use only the HyperChem tool icons to translate or rotate molecules, to zoom, or to adjust the clipping slab.

# Chapter 9

## Constructing Molecules

HyperChem provides a number of methods for constructing new molecules:

- The Drawing tool sketches molecules from individual atoms. Use this tool to position atoms, choose elements, and connect atoms by covalent bonds. The HyperChem Model Builder can then convert these drawings into 3D molecular structures.
- HyperChem provides databases of residues for constructing proteins, polynucleotides (including RNA and DNA), and equilibrated water environments.
- A crystal builder will help you construct crystalline molecular systems.
- A sugar builder will help you construct polysaccharides.
- A polymer builder will help you construct polymeric systems from monomers you specify along with rules for joining them together.

Chapter 6 described the last three specific builders. Here we focus on the first two methods. You can also use molecular structures from other sources, stored in files. For example, HyperChem interprets and displays PDB files (see page 503) that contain detailed structures of proteins and nucleic acids.

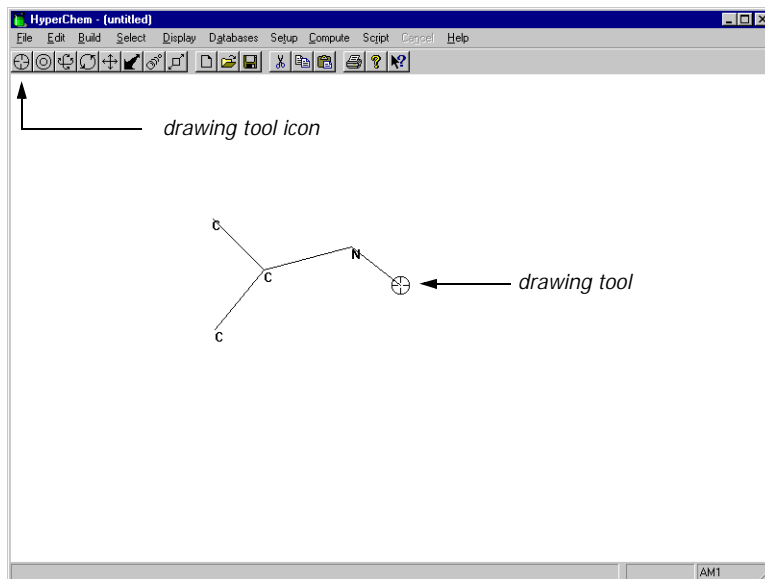
### Drawing Molecules

Using the HyperChem Drawing tool (the first tool on the toolbar), you can sketch the atoms in a molecule, connecting them with covalent bonds. You can then use HyperChem's Model Builder to

convert a 2D drawing into a 3D model, with realistic bond lengths, bond angles, and torsion angles.

To draw a new molecule, you can either start with an empty workspace or add the new molecules to a molecular system already in the workspace.

## Adding Atoms and Bonds



To draw a molecule:

1. Choose Default Element on the Build menu. In the Element Table dialog box (see page 121), L-click on the element for drawing. For example, you can start to draw with carbon atoms, then change to another element when needed.
2. From the Element Table dialog box, turn on Allow ions (3) if you need to add more than the normal number of bonds to an atom. You can add up to twelve bonds per atom. Turn on Explicit hydrogens (3) if you want to add hydrogen atoms while drawing. If not, HyperChem can automatically add hydrogens later, when you use the Model Builder.
3. L-click on the Drawing icon and move the Drawing tool into the workspace.



4. To create individual atoms, L-click anywhere in the workspace.
5. To remove an atom, R-click near the atom.
6. To create two singly bonded atoms, L-drag, then release the mouse button.
7. To add another bond and atom, L-click near any atom, L-drag, and release the mouse button.
8. While adding atoms, change the default element as needed. You can also change the element after adding an atom (see “After Drawing” on page 475).
9. To specify the chirality or local conformation of an atom, [Shift]+L-click on one of its bonds (see “Using Wedges to Display and Specify Stereobonds” on page 482). You can also [Shift]+L-click on an atom to change the chirality of that atom after model building (see “Changing Chirality” on page 485).

*Note:* The maximum number of atoms that you can add is 32,767 less the number of molecules and residues., structural restraints, etc.

## Removing Atoms

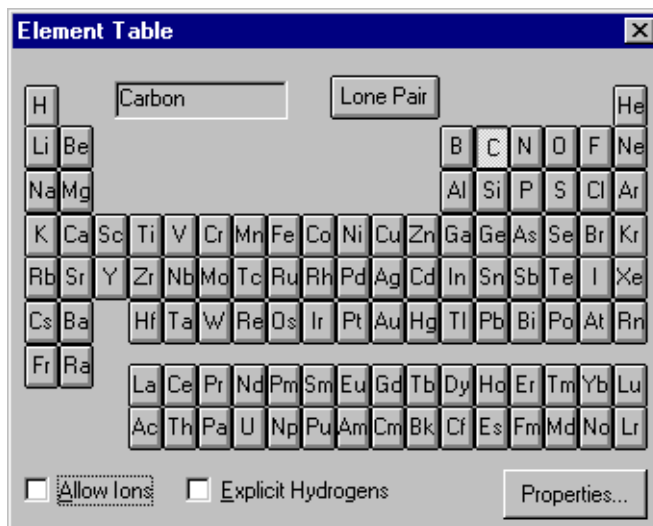
To remove atoms and bonds:

1. L-click on the Drawing tool icon to activate the tool.
2. Position the Drawing tool on a bond or atom that you want to remove.
3. R-click on an atom. This removes the atom and its bonds to other atoms. If you R-click on a single bond, HyperChem removes the bond and any atoms connected *only* through this bond.

## Changing the Drawing Element

Normally, the Drawing tool places only heavy (nonhydrogen) atoms in a molecule. You choose the heavy atom (from helium through lawrencium) from the Element Table dialog box. Your choice is the default element. To see the Element Table dialog box, double-click on the Drawing icon or choose Default Element on the Build menu.

**Caution:** You can use any elements from the periodic table for drawing. However, the Model Builder (see page 124) might not provide appropriate bond lengths or angles for all atoms.



## While Drawing

You can change the default element any time *while* drawing by changing the choice in the Element Table dialog box:

To change the default element:

1. Double-click on the Drawing icon, choose a new element from the Element Table dialog box.
2. Move the Drawing tool into the workspace and continue drawing. The next atoms you add are the new default element.

**Note:** Single clicking on an element in the Element Table dialog box will change the default element and leave the dialog box open. Double clicking on an element will change the default element and dismiss the dialog box. Shift-clicking on an element will bring up the Element Properties display for that element. Clicking on the Properties button will display the properties of the default element.

## After Drawing

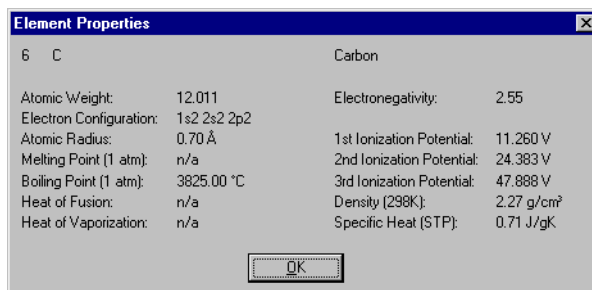
To change an element after drawing a molecule:

1. From the Element Table dialog box, choose a different element. For example, if you drew your molecule with carbon as the default element and you want to change a carbon to an oxygen, click on oxygen in the dialog box.
2. With the Drawing tool, L-click on the atom to change. The atom changes to the new default element.

*Note:* You can change atoms in 2D drawings or, after using the Model Builder, in 3D structures.

## Element Properties

If you shift-click on an element button in the Element Table dialog box, the Element Properties box will appear. This shows physical properties of the element that you selected. Alternately, if you click on the Properties button of the Default Element dialog box, the properties of the default element will be displayed.



This does *not* change the default drawing element.

## Changing the Bond Order

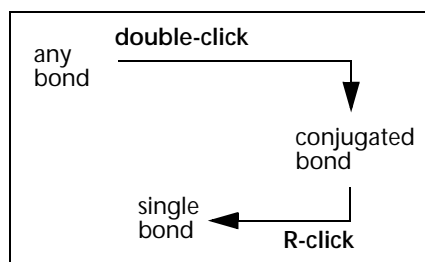
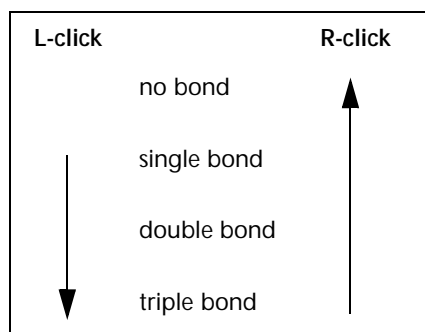
HyperChem first draws single, covalent bonds. Using the Drawing tool, you can change these bonds to double, triple, or aromatic bonds.

To change the order of individual bonds:

- To create a double bond, L-click on any single bond with the Drawing tool.

- To create a triple bond, L-click on a double bond.
- To create a conjugated bond, double-click on any bond.
- To create an aromatic ring, double-click on any bond in the ring.
- To decrease the order of any bond (conjugated to single; triple to double; double to single), R-click on it.
- To remove a single bond and any atom connected only through that bond, R-click on the bond.

To change bonds by using your mouse:

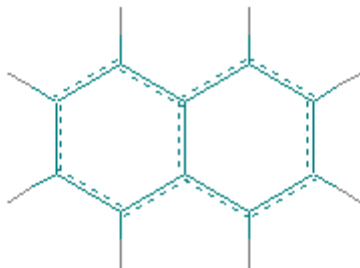


**Note:** You can change the order of bonds in 2D drawings or, after using the Model Builder, in 3D structures.

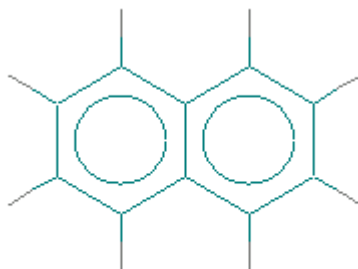
If Show Multiple Bonds on the Display menu is on (3), you see the order of the bonds in the molecular structure as single, double, triple, or dotted lines (possibly aromatic circles). If Show Multiple Bonds is off, all bonds appear as single lines. If you change the bond order, the new bond order appears on the status line.

## Aromatic Rings

An aromatic ring is created by double clicking on the bond of a ring. The default is to display this ring with aromatic bonds that are rendered with a dotted line as shown below.



The alternative is to use the menu item, Show Aromatic Rings as Circles, to display aromatic rings in a way that may be more pleasing for printing or other situations. Either rendering can be used interchangeably.

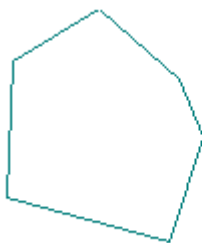


## Alternative Drawing Mode

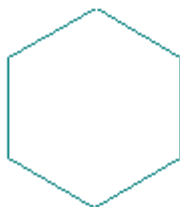
The menu item, Constrain Lengths and Angles, when checked will allow you to draw regular structures with equal bond lengths and angles.

In most situations, a 2D molecule drawing is simply a way to indicate connections to the model builder which will then give you a

3D structure. In this case, it is easier to draw without this constraint. The default is to not have this menu item checked and to draw without regard to the “aesthetic beauty” of the drawing. In this case a six-membered ring might be drawn and look as follows:



In situations where you want to have a 2D drawing for printing or other purposes, you can set the constraints on drawing and create a regular hexagon,



Submitting either of these to the model builder will create exactly the same 3D structure. You can create 2D structures mixing the two modes if you prefer.

## Adding Hydrogen Atoms

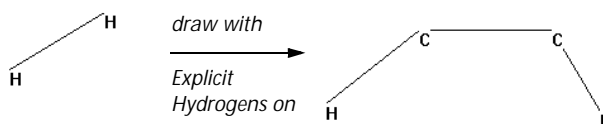
You can draw hydrogen atoms or let HyperChem do this for you.

To add hydrogens during drawing, turn on Explicit Hydrogens in the Element Table dialog box (3) or on the Build menu (3). As you use the Drawing tool, the first two atoms appear, as hydrogens. When you add a third atom, the second atom changes to the default element. The Drawing tool, with Explicit Hydrogens on, follows this simple rule: singly bonded atoms appear as hydrogens; multiply bonded atoms appear as the default element.

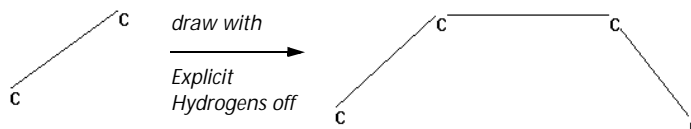
With Explicit Hydrogens on, Add H & Model Build (on the Build menu) changes to Model Build. The Model Builder does not automatically add more hydrogens.

**Note:** To draw a hydrogen with more than one bond, choose hydrogen as the default element and turn on Allow Ions.

If you draw atoms with Explicit Hydrogens on, you can later choose Add Hydrogens on the Build menu to fill in the hydrogens you did not add manually.



To draw only heavy atoms, use the Drawing tool with Explicit Hydrogens off. HyperChem automatically adds hydrogens and then creates a 3D chemical structure when you use the Model Builder (Add H & Model Build on the Build menu).

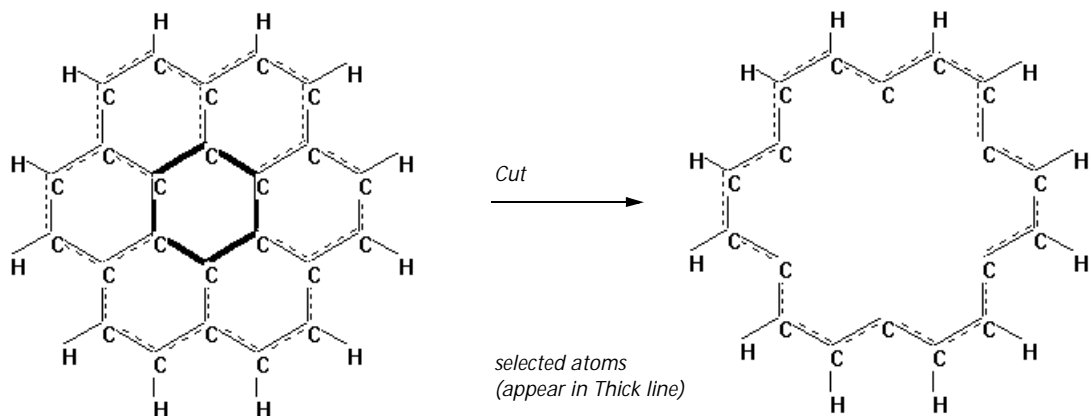


## Cutting and Pasting Atoms

Use Cut, Paste, and Copy on the Edit menu to move or duplicate atoms (see "Build Menu" on page 117). You can move atoms within a HyperChem window or between different HyperChem windows.

### Cut

Cut moves the selected atoms from the workspace to the Clipboard, replacing previous contents of the Clipboard. You can then use Paste to copy the atoms back to the workspace.



**Note:** Cut removes selected atoms and all bonds connecting these atoms to each other and to the rest of the molecular system.

## Copy

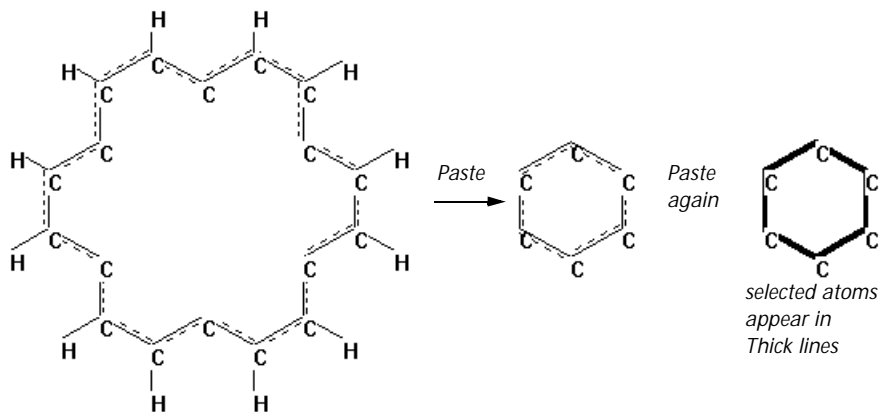
Copy places a duplicate of selected atoms onto the Clipboard, replacing the previous contents of the Clipboard. You can use Paste to copy the atoms back to the workspace.

**Note:** Copy duplicates selected atoms and the bonds connecting these atoms. It does not remember the bonds between this selection and the rest of the molecular system.

## Paste

Paste copies the contents of the Clipboard to the workspace. The atoms from the Clipboard appear a few Ångströms away from the center of mass of the molecular system previously in the workspace.





**Note:** Paste is not the exact reverse of Cut. Cut removes selected atoms and all bonds connecting these atoms to each other and to the molecular system. Paste provides only the atoms and the bonds connecting these atoms to each other.

## Allowing Arbitrary Valence

The menu item, Allow Arbitrary Valence, is particularly useful for beginning chemistry students. When this menu item is unchecked, HyperChem will complain if drawings are made that give Oxygen a valence greater than 2. Nitrogen a valence greater than 3, Carbon a valence greater than 4, etc. An exception to this is made if the atom has a formal charge.

If the menu item is checked, then the valence can be anything up to a limit of 12 bonds per atom. This is necessary when drawing certain systems with dative bonds or other unusual bonding connections.

## Changing Molecular Conformation

HyperChem provides several ways of changing the conformation of a molecule, either before or after using the Model Builder:

- The Set items on the Edit menu (see "Build Menu" on page 117) change a selected bond length, bond angle, or torsion angle (see "Selecting" on page 32). These changes are immedi-

ate when you L-click on OK. The Model Builder or a geometry optimization calculation might change these adjustments.

- The Constraints items on the Build menu (see “Build Menu” on page 117) provide instructions to the Model Builder to produce these configurations. The Model Builder adds these constraints to its list of rules as it constructs a 3D structure.
- The Restraints options for molecular mechanics calculations (see “Restraint Forces Dialog Box” on page 340) limit the conformation of selected atoms during a molecular mechanics calculation.
- The Drawing Tool can be used to turn bonds into dashed or solid wedges (for “bond down” or “bond up”) by [Shift]+L-clicking on them. The Model Builder uses these wedges to help it determine positions for atoms (see “Using Wedges to Display and Specify Stereobonds” on page 482).
- The Drawing tool can also be used to change chirality of atoms after model building by [Shift]+L-clicking on them.
- Invert and Reflect on the Edit menu can transform a molecule or a set of atoms.

## Using Wedges to Display and Specify Stereobonds

The stereochemistry feature of HyperChem lets you display bonds as wedges. Wedges always correspond to stereochemistry builder constraints used to enforce a specific arrangement of neighbors about a center when the Model Builder is invoked. However, wedges can be used for purely visual purposes as well. Wedges serve two functions:

- To provide you with 3D depth perspective information about bonding arrangements.
- To allow you to assign builder constraints so that you can build the structure with a specified stereochemistry or conformation.

In the first case you can simply display the wedges to show the current stereochemical bonding relationships. You do not modify the structure.

In the second case you use the wedges to assign a constraint, and you invoke the Model Builder to build (or rebuild) the structure

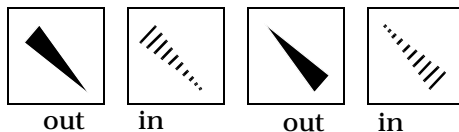
with the assigned stereochemistry or conformation. An exclamation mark is shown on a wedge when the current conformation is different than the restraint that you have specified with the wedge.

To use wedges:

1. Choose Rendering on the Display menu to open the Rendering dialog box. Choose Sticks and check Wedges, then L-click on OK.
2. L-click on the Drawing tool icon to get into Drawing mode.
3. Place the Drawing Tool over a bond between two atoms each with at least three neighbors, hold down the [Shift] key, and L-click.

If one of the atoms has two or less neighbors only two types of wedges are possible. If both bonded atoms have two or less neighbors, wedges cannot be used on that bond.

4. Continue [Shift]+L-clicking to cycle through the four possible wedges.



As you cycle through the wedges, notice that at least two have exclamation marks. At this point if you want to change the stereochemistry to correspond with the wedge, you must invoke the Model Builder. Otherwise, if you're satisfied with the stereochemistry of the bond, cycle to either of the two wedges that appear without exclamation marks; these refer to the current stereochemistry. (If your system is planar, all wedges will be shown with exclamation marks.)

5. Double-click on the Select Tool icon to invoke the model builder. Notice that any exclamation marks have disappeared because the restraints have now been satisfied by the Model Builder.

If you rotate the molecule, the dashed wedges become solid wedges and vice versa. Their appearance changes according to their orientation with respect to the viewer (the plane of the workspace).

## Wedges Explained

HyperChem displays two types of stereochemistry wedges: solid and dashed. A stereo wedge, which is drawn between two atoms instead of the normal rendering of a bond, specifies a stereochemistry builder constraint describing the arrangement of neighbors about one of the atoms. In every case, the atom at the “point” of the wedge is the atom whose stereochemistry is being specified. Thus the wedge



specifies the stereochemistry about atom A. It does not describe the arrangement of neighbors about atom B. This would be expressed by some other wedge whose point was at atom B. If a wedge (either solid or dashed) is displayed with an exclamation mark “!”, the stereochemistry constraint is not currently satisfied. It will be satisfied after invoking the model builder.

Solid wedges mean “bond up” (out of the screen), while dashed wedges mean “bond down” (behind the screen).

## For Atoms with Four Neighbors

Start by looking at the central atom and its four neighbors as displayed on screen (these five atoms are displayed in the plane of the screen). If one of the bonds is a solid wedge with its point at the central atom, imagine that

- the central atom is in the plane of the screen,
- the wedge neighbor is coming out of the screen, and
- the three other (nonwedge) neighbors are back, behind the screen.

This imagined arrangement specifies the stereochemistry. Remember, if the wedge is annotated with an “!”, the constraint is not currently satisfied (perhaps because the current geometry is planar).

Of course, for a dashed wedge, you must imagine that the wedge bond points back behind the screen, while the others point out of the screen (toward you).

## For Atoms with Three Neighbors

Once again, start with the planar screen coordinates displayed for the central atom and its three neighbors. In this case of only three neighbors, there is some yet-to-be-provided fourth neighbor—probably a hydrogen, or a lone pair in the case of a central nitrogen. To visualize the meaning of a wedge with its point directed at this central atom, imagine that this missing fourth neighbor has exactly the same screen position as the wedge neighbor. That is, this missing atom is directly in front of or directly behind the wedge neighbor. Now, there are four neighbors, including one that is imaginary. Apply the procedure in the previous section for four neighbors.

## Changing Chirality

HyperChem can interactively change the chirality of an atom if two of its neighbors can be freely swapped.

To interactively change chirality:

1. L-click on the Drawing tool icon to get into Drawing mode.
2. Place the Drawing Tool over the atom whose chirality you wish to change, hold down the [Shift] key, and L-click.

This rotates two of the neighbors around the axis midway between their bonds to the atom. This rotation will occur even if the center atom is not chiral. You can check this by using colors or displaying atom indexes (by selecting Numbers for Atoms in the Labels dialog box) and noting that neighbors swap when you [Shift]+L-click on the central atom.

If the neighbors cannot be swapped, with fused rings for instance, you must use wedges and the model builder to change chirality.

To change chirality using wedges:

1. Choose Rendering on the Display menu to open the Rendering dialog box. Choose the Sticks property sheet and check Wedges, then L-click on OK.
2. L-click on the Drawing tool icon to get into Drawing mode.

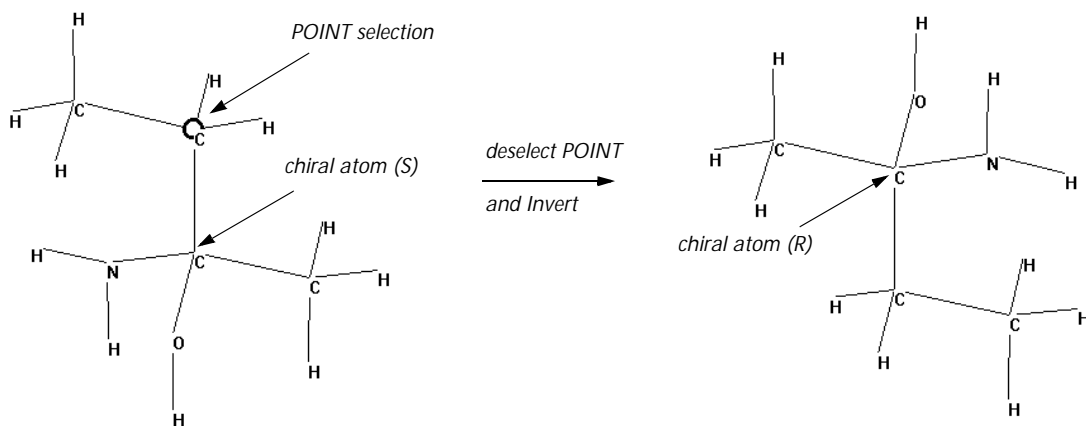
- Place the Drawing Tool over a bond involving the atom whose chirality you wish to change, hold down the [Shift] key, and L-click until a wedge with an exclamation mark appears.
- Double-click on the Selection tool to invoke the Model Builder.

The Model Builder will try to preserve any existing chirality of other centers when you use it.

Both Invert and Reflect on the Edit menu can also be used to change the chirality of atoms, but you cannot always change atoms independently. Inversion takes place through a selection that you name as POINT and reflection takes place through a selection that you call PLANE. You define POINT or PLANE by using Name Selection on the Select menu (see page 237).

To change chirality using Invert or Reflect:

- Select atoms in a molecular system.
- Choose Name Selection on the Select menu and name this selection either POINT or PLANE.
- Deselect all atoms.
- Select the atoms to invert or reflect. Without a selection, HyperChem acts on the whole molecular system.
- Choose Invert or Reflect on the Edit menu.

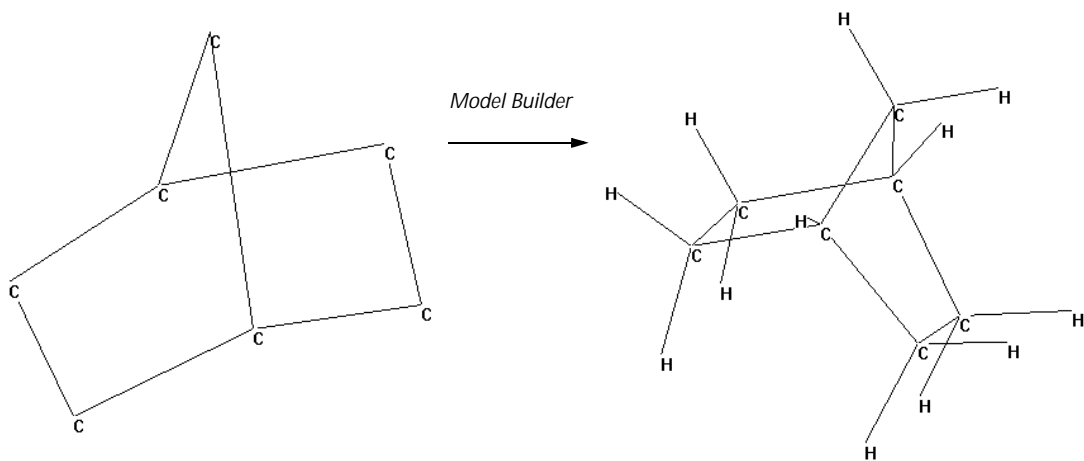


## Converting 2D Drawings into 3D Structures

HyperChem's Model Builder uses predetermined bond lengths, bond angles, and torsion angles to estimate the 3D structure of a molecular system or of selected atoms.

While building a 3D structure, the Model Builder evaluates the chirality of every atom. You can label chiral centers R or S by using Labels on the Display menu.

The Model Builder gives you approximate structures. It cannot anticipate the diversity of all chemical compounds and might produce inaccurate structures for some ring systems, or for compounds containing less common elements. In these cases, use the Model Builder first to construct an approximate structure. Then do a geometry optimization calculation to find a better conformation.



The Model Builder does not add charges (ionic or partial) to molecules. You can add charges to selected atoms by using Set Charge on the Build menu. Or, you can use a semi-empirical or *ab initio* quantum mechanical calculation to determine atomic charges (see "Single Point" on page 360).

### Incremental Model Building

The Model Builder works on selected atoms. You can use this feature to Model Build only part of a molecular system. For example,

you can add a group or a separate molecule to a system that already has an optimized structure (for example, a molecule after a geometry optimization calculation or a residue from a database) and then Model Build only the new atoms.

*Caution:* Select only functional groups or larger units to reduce the possibility of building stressed or abnormal configurations.

## Formal Charge

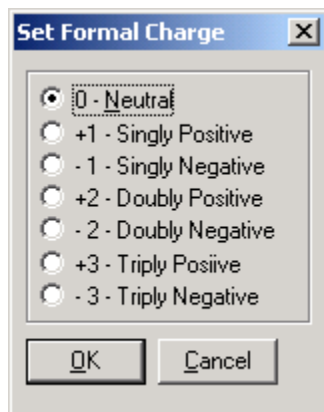
HyperChem contains the concept of formal charge on an atom. The formal charge has nothing to do with the total charge on the system (affecting the total number of electrons for input to quantum mechanical computations) or the atom charge (the net positive or negative partial charge from residue templates or the output of quantum mechanical computations). The formal charge affects only the internal functioning of the model builder plus a minor visual change in the symbol label.

The formal charge affects the valence or number of Hydrogens the model builder expects to add when the menu item, Add H and Model Build, acts. For example, an isolated Nitrogen atom submitted to the model builder normally results in  $\text{NH}_3$  being created. However, if the Nitrogen is given a formal charge of +1, then the model builder produces  $\text{NH}_4^+$  with four rather than three hydrogens being added. The basic rule is that a formal charge of +N increases the valence by N, whereas a formal charge of -N decreases the valence by N. Thus, HyperChem's modelbuilder produces  $\text{CH}_5^+$  and  $\text{CH}_3^-$  but not  $\text{CH}_3^+$  (at least not by default).

The formal charge is used to give an initial guess at the Total-Quantum-Charge variable used as input to quantum mechanical calculations but this variable is also available for explicit setting in the Setup/Options dialog boxes of quantum mechanical calculations.

The formal charge, by default, is zero but can be changed via the following dialog box:



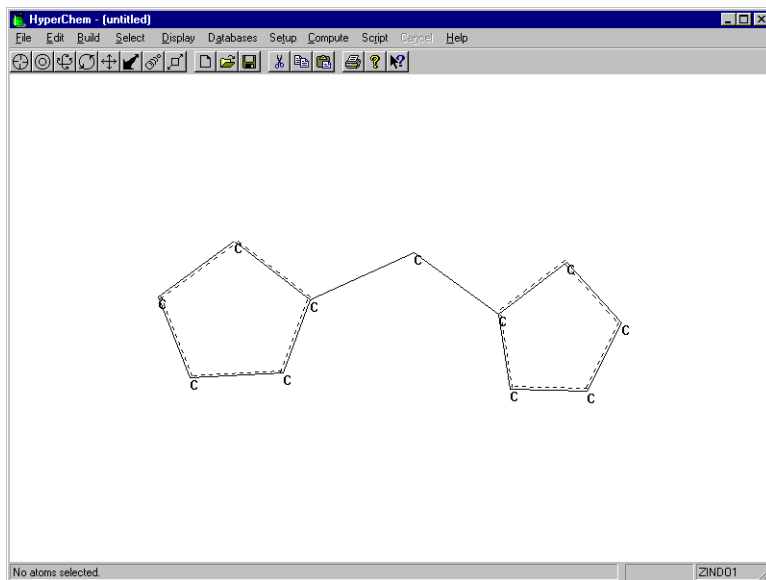


## Building Inorganic Structures

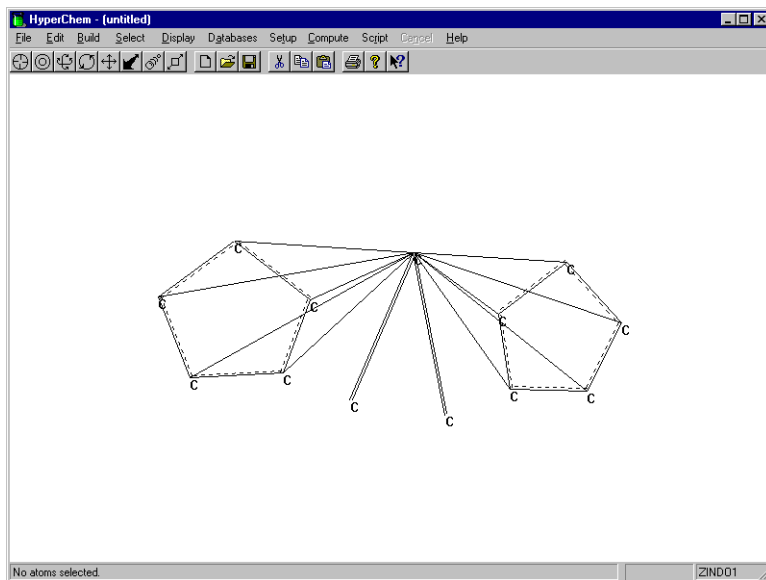
The maximum number of neighbors that the Model Builder can handle is 12. This is particularly useful for building inorganic structures, such as ferrocene.

To see this modeling capability, try the following exercise:

1. On the Build menu, turn on Allow Arbitrary Valence.
2. Draw the following 2D structure:

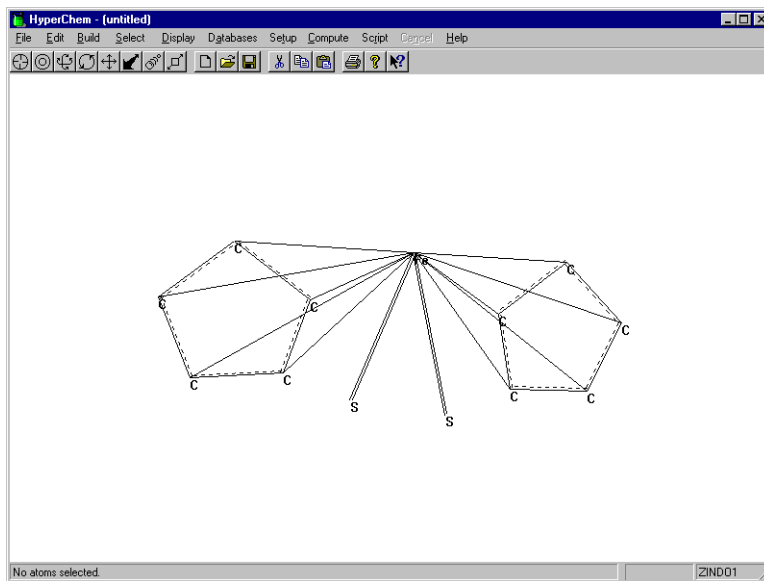


3. Connect all remaining carbon atoms to the central carbon atom and add two more carbon atoms, as shown in the following illustration:

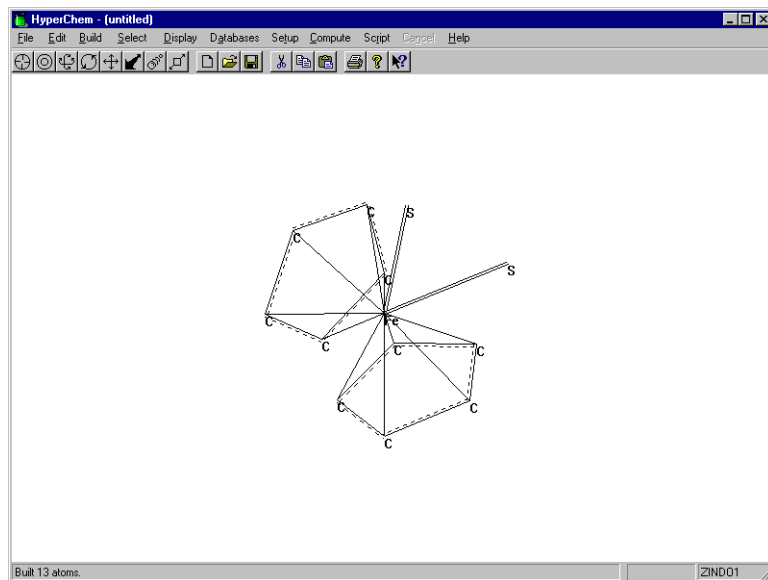


4. Make sure the two newly added C—C bonds are double bonds.

- Change the central carbon atom to iron, and change the two nonring carbons to sulfur atoms. The 2D structure should now look like this:



- Double-click on the Selection tool to invoke the Model Builder. HyperChem builds the following 3D structure:



(If your 3D structure doesn't appear as shown in the previous illustration, use the Rotation tool to reorient the molecule.)

If you wish to include the hydrogen atoms, you must draw them onto the rings explicitly before or after you use the Model Builder.

This demonstrates how easily HyperChem can build a molecule with a highly coordinated metal center.

## Building Molecules from Residues

HyperChem provides databases (templates) containing residues for constructing proteins, polynucleotides, and water environments (solvation). The Databases menu ("Databases Menu" on page 139) contains items for constructing polypeptides and polynucleotides. Periodic Box on the Setup menu (see "Periodic Box" on page 335) supplies equilibrated water molecules to fill a periodic box.

Residues with standard geometries are stored in database files (TPL files for amino and nucleic acids, and BIN files for water). As you construct polymers, you can automatically incorporate standard secondary and tertiary structures, or use your own specifications for conformations.

Proteins and nucleic acids from PDB files are also composed of residues. You can use the methods described in this section to modify those molecules.

The sequence editor described in Chapter 6 can also be used to create protein structures.

## The Databases Menu

Use the items on the Databases menu to construct or modify proteins and nucleic acids.

The first three items, Amino Acids, Make Zwitterion, and Remove Ionic Ends control the building of proteins. The second group of items, Nucleic Acids and Add Counter Ions, provide the tools for constructing polynucleotides, both single and double stranded. The last item, Mutate, changes a residue in an existing protein or polynucleotide.

See “Databases Menu” on page 139 and the corresponding dialog boxes in chapter 3 for more information on using these menu items.

### Proteins

To construct a protein, choose Amino Acids on the Databases menu. You see the Amino Acids dialog box with choices of residues.



For a definition of each residue in this dialog box, see “Amino Acids Dialog Box” on page 142.

The Amino Acids dialog box also gives you choices for the secondary conformation of each residue that you add. You can choose from standard conformations (Alpha or Beta) or devise your own conformation (Other) with specific torsional angles. You can also add D-amino acids instead of normal L-amino acids.

## Constructing a Protein

Proteins are constructed from the N-terminus to the C-terminus. To build a new protein, you must start with one of these conditions:

- An empty workspace.
- A workspace containing proteins with a C-terminal blocking group or in the zwitterionic form. Both conditions block elongation of the existing proteins and force HyperChem to start a new polymer.
- A workspace containing any molecules *except* unblocked proteins (see above) constructed from a database or from a PDB file.

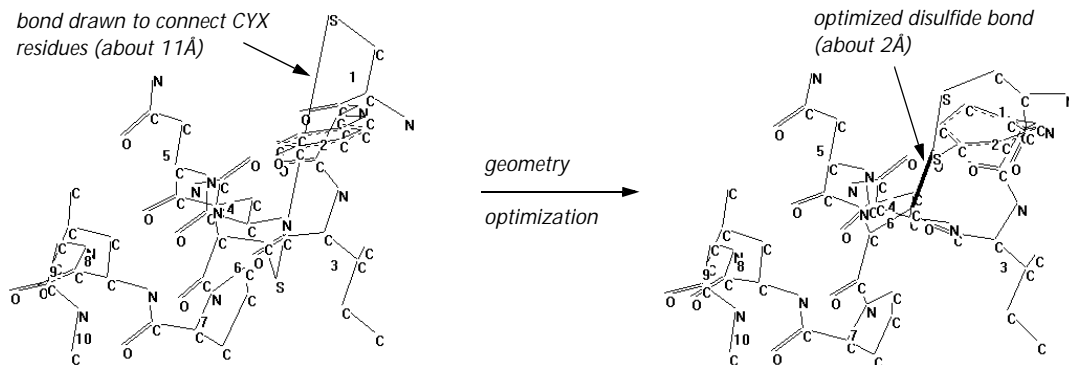
To build a new protein:

1. Choose Amino Acids on the Databases menu. You see the Amino Acids dialog box. You can move this dialog box to one side of the workspace so that you can watch HyperChem construct a new polymer. While this dialog box is on screen, you can use menu items and tools.
2. From the Conformation side of the dialog box, L-click on Alpha, Beta, or Other. L-click on Isomer: D or L. These choices affect each added amino acid until you make new choices. Once you add a residue, you cannot use this dialog box to change its conformation.
3. L-click on Ace if you want an N-terminal cap. If not, go to step 4.
4. L-click on the first amino acid residue.
5. L-click on the rest of the residues. You can get ahead of HyperChem, but it will catch up with you. Remember to change the conformation of each amino acid *before* you choose it.

6. L-click on Nme if you want a C-terminal cap. This cap prevents you from adding more residues.

## Cross-linking with Cyx

HyperChem provides half-cystine residues (Cyx) for disulfide cross-links. As you construct a polypeptide, place the Cyx residues in the correct primary sequence. Then use the Drawing tool to form a disulfide bond between the Cyx residues. To adjust the length of this bond and the conformation of the polypeptide, use a geometry optimization calculation.



## Choosing Conformations

Each amino acid that you add to a polypeptide can have a different secondary conformation. In this way, the polypeptide can have stretches of alpha-helical structure, beta-pleated sheet structure, and other transitional conformations.

For Other, you can use angles from  $-180$  to  $+180$  degrees. Omega, the torsion around the peptide bond, is usually close to  $180$  degrees, or trans.

You can also choose D-amino acids instead of the usual L-isomer.

## Creating Zwitterions

At neutral pH, polypeptides exist in solution as zwitterions, with  $-\text{NH}_3^+$  and  $-\text{COO}^-$  terminal groups. When you construct polypeptides with HyperChem, the ends are incomplete ( $-\text{N}-\text{H}$  and  $-\text{C}-\text{O}$ ). Make Zwitterion on the Databases menu completes the ends ( $-\text{NH}_3^+$  and  $-\text{COO}^-$ ).

If the ends of a polypeptide are modified or capped, you cannot convert it into a zwitterion. If a polypeptide exists as a zwitterion, you cannot add more amino acid residues to an existing polypeptide or start new polypeptides in the same molecular system.

Use Remove Ionic Ends on the Databases menu to reverse Make Zwitterion.

## Mutating Amino Acids

Use Mutate on the Databases menu to change one amino acid to another in a polypeptide. The polypeptide must come from the Databases menu, or a PDB or MOL2 file.

To mutate an amino acid:

1. Choose Residues on the Select menu. This lets the Selection tool act on whole residues if they are present in a molecular system.
2. Select one residue in a protein.
3. Choose Mutate on the Databases menu.
4. From the Mutate dialog box (see “Mutate Dialog Box” on page 191), choose a new residue. Then L-click on OK. HyperChem substitutes the new residue for the selected residue.

To change the conformation of amino acids after they are in a polypeptide, you must select individual torsional angles (see “Selecting” on page 32) or restrain the torsion to a new angle (see “Restraint Forces Dialog Box” on page 340). Alternatively, you can use Set Bond Torsion on the Edit menu (see page 220), and calculate a new optimal geometry (see “Geometry Optimization” on page 362).

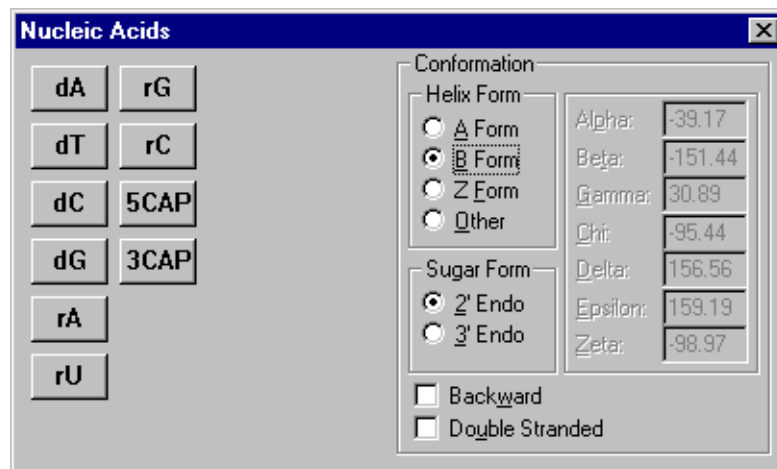
## Adding Unusual Residues

You can add more residues to the Amino Acids dialog box. The information for a residue comes from a template (TPL) file (see “Residue Template Files” on page 727). Whenever you use a residue in this dialog box or when you open a PDB file, HyperChem interprets the structure of residues by using information in a TPL file. See page 739 for how to write a new entry. The new residue can appear as a choice (a button) in the Amino Acids dialog box.



## Nucleic Acids

To construct a polynucleotide, choose Nucleic Acids on the Data-bases menu. You see the Nucleic Acids dialog box with a choice of nucleic acid residues. For a definition of each residue in this dialog box, see “Nucleic Acids Dialog Box” on page 159.



The Nucleic Acids dialog box also has choices for the secondary conformation of each residue that you add. You can choose from standard conformations (A, B, or Z) or devise your own conformation with specific torsional angles. You can also change the pucker of the furanose ring in each residue (Sugar Form), and the number of strands (single or double).

At the start of polymerization, you can choose to begin at the 3' end (Backward), to add nucleotides to the 5' end, or to begin at the 5' end (normal direction of biosynthesis).

### Constructing a Polynucleotide

To build a new polynucleotide, you must start with one of these conditions:

- An empty workspace.
- A workspace containing polynucleotides with a 3' or a 5' blocking group (3CAP or 5CAP). Either of these groups block elongation of the existing nucleic acid (depending on the direction of polymerization) and force HyperChem to start a new polymer.

- A workspace containing any molecules *except* unblocked nucleic acids (see above) constructed from a database or from a PDB file.

**To build a new polynucleotide:**

1. Choose Nucleic Acids on the Databases menu. You see the Nucleic Acids dialog box. You can move this dialog box to a side of the workspace to watch HyperChem construct the new polymer. While this dialog box is on screen, you can use any menu items or tools.
2. Choose the direction of polymerization before you begin a molecule: L-click on Backward (5) to begin from the 3' end. No check mark means that HyperChem starts at the 5' end. If you change the direction of polymerization after you start a molecule, HyperChem begins a new molecule.
3. From the Conformation side of this dialog box, you can make these choices for each residue before you add it:
  - A Form, B Form, Z Form, or Other.
  - HyperChem automatically chooses 3' Endo (Sugar Form) for A Form, or 2' Endo for B and Z Forms. You can choose the alternate sugar pucker.
  - Double Stranded (3) for adding the next residue plus its hydrogen-bonded complement.

These choices affect each added nucleic acid until you make new choices. Once you add a residue, you cannot use this dialog box to change its conformation.

4. L-click on the first nucleic acid residue.
5. L-click on the other residues. You can get ahead of HyperChem, but it will catch up with you. Remember to change the conformation of each residue *before* you choose it.

*Note:* You can mix deoxyribose and ribose nucleotides. You can also mix single- and double-strand structures to form partially base-paired molecules.

6. L-click on 3CAP or 5CAP if you want to block further polymerization at the growing end of this polymer.

## Choosing Conformations

Each nucleic acid residue that you add to a polynucleotide can have a different conformation. In this way, the polynucleotide can have stretches of A form DNA, B form, Z form, or another conformation that you choose.

A and B forms of DNA are normally right-hand helices, while the Z form is a left-hand helix. To change the direction of a helix after constructing it, use Reflect on the Edit menu to connect the helix into its mirror image. You can reflect around any PLANE in the molecular system.

You can also change the conformation of the deoxyribose or the ribose sugar in each residue. The A form of DNA normally contains deoxyribose sugar in the more stable C2'-endo form. The B form of DNA form contains the C3'-endo sugar.

## Adding Counter Ions

HyperChem can supply sodium counter ions to balance the positive charge on each phosphate group in a polynucleotide. When you choose Add Counter Ions on the Databases menu, HyperChem places one Na<sup>+</sup> ion 1.688 Ångstroms equidistant from each oxygen atom in a phosphate group (see page 162).

Use counter ions for chemical calculations of polynucleotides in water (see "Periodic Box" on page 335).

If you choose Add Counter Ions and then add more residues, only the original residues have counter ions. Choose Add Counter Ions again to add Na<sup>+</sup> ions to the new residues.

Use Clear or Cut on the Edit menu to remove Na<sup>+</sup> ions.

If you store the system in a HIN file, the file also contains the counter ions. The Na<sup>+</sup> ions appear at the end of the HIN file as separate molecules, with mol and atom records for each Na<sup>+</sup> ion.

## Mutating Nucleic Acid Residues

Use Mutate on the Databases menu to change one residue to another in a polynucleotide from the Databases menu, or from a PDB or MOL2 file. For a more general replacement of a set of residues with another set, see the description of the Replace operation in Chapter 6.

**To mutate a nucleic acid residue:**

1. Choose Residues on the Select menu. The Selection tool acts on whole residues if they are present in a molecular system.
2. Select one residue in a polynucleotide.
3. Choose Mutate on the Databases menu.
4. From the Mutate dialog box (see “Mutate Dialog Box” on page 191), choose a new residue. Then L-click on OK. HyperChem substitutes the new residue for the selected residue.

*Note:* The Mutate dialog box provides nucleotide residues in addition to those in the Nucleic Acids dialog box. These additional residues have optimized structures but lack atomic charge information.

To change the conformation of nucleic acid residues after they are in a polymer, you select individual torsional angles (see “Selecting” on page 32), restrain the torsion to a new angle (see “Restraint Forces Dialog Box” on page 340), and calculate a new optimal geometry (see “Geometry Optimization” on page 362).

## Creating Unusual Base Pairs

**To build a polynucleotide with unusual base pairing:**

1. Choose Double Stranded in the Nucleic Acids dialog box.
2. Choose one of the residues in the pair. For example, choose rU. This adds rU-rA.
3. Set the Selection tool to the Residues level.
4. Select the incorrect residue. For example, select rA. To help find rA, you can turn on Residue Name labels in the Labels dialog box.
5. Choose Mutate on the Databases menu and then choose a residue to replace the incorrect residue. For example, choose rU to replace rA. This produces an rU-rU pair.

## **Adding Unusual Residues**

You can add more residues to the Nucleic Acids dialog box. The information about residue comes from TPL files. Whenever you use a residue in this dialog box or open a PDB file, HyperChem interprets the structure of residues using information in a TPL file. See page 739 for how to write a new entry. The new residue can appear as a choice (a button) in the Nucleic Acids dialog box.

## **Water Molecules**

HyperChem can place a molecular system in a periodic box containing water molecules. These are standard TIP3P models of water molecules equilibrated at 300 K (25°C) and one atmosphere. Without the periodic box, HyperChem assumes that molecules are in a vacuum. You can use a distance-dependent dielectric constant in calculations to simulate the effect of a solvent (see “Force field Parameters Dialog Box” on page 352).

Since these water molecules have special properties and come from a database, HyperChem treats them like residues. You can select individual water molecules with the Selection tool, which you should set to Residues (see “Residues” on page 225).

Choose Periodic Box on the Setup menu to add water molecules, along with periodic boundary conditions (see “Periodic Box” on page 335), to a molecular system.

If you apply a label to these residues (see “Labels” on page 281), it might appear as HOH or WAT, depending on the source of the molecular system.



# Chapter 10

## Protein Data Bank Files

Structures of proteins and nucleic acids are available in PDB files.<sup>1</sup> These structures are derived from physical studies of molecules (for example, x-ray diffraction or nuclear magnetic resonance (NMR) analyses). HyperChem can interpret and display structures stored in all PDB files that contain atomic coordinates, and can also store new molecular structures in PDB-type files.

### Obtaining PDB Files

The HyperChem distribution disks contain several sample PDB files. You can obtain the complete PDB collection, containing about 17000 structures, from the Research Collaboratory for Structural Bioinformatics (RCSB). Alternatively, you can download a specific PDB file. For information, contact:

[www.rcsb.org/pdb/](http://www.rcsb.org/pdb/)

### Reading PDB Files

HyperChem can read all PDB files that contain atomic coordinates, up to the limit of the computer memory. HyperChem uses the PDB information on atomic coordinates, plus the information in its own TPL files (see page 727), to reconstruct standard amino acid and nucleic acid residues. For HETATM records and atoms that are not included in TPL files, HyperChem uses atomic coordinates

1. The Protein Data Bank is operated by the Research Collaboratory for Structural Bioinformatics (RCSB) and is supported by funds from the National Science Foundation, the Department of Energy, and two units of the National Institutes of Health: the National Institute of General Medical Sciences and the National Library of Medicine.

and CONECT records from the PDB file to reconstruct a molecular system.

HyperChem expects that PDB files will follow the formatting standards described by the maintainers of the PDB archive. Some software creates “PDB” files that do not follow this format properly; for example, the data entries on a line of the file may be in the wrong columns, or shifted left or right within a section. If you use the script command “non-standard-pdb-names = yes”, HyperChem will attempt to make allowances for incorrect formatting. Use of this command may prevent correct reading of files which do conform to the standards.

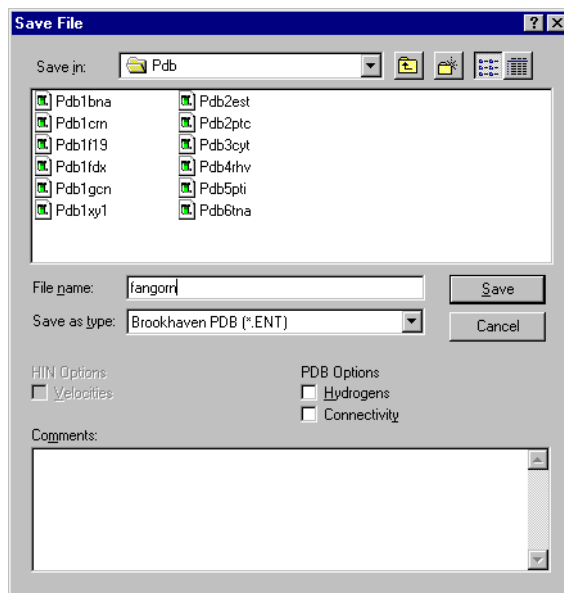
Most PDB files contain all the information that is necessary for HyperChem to regenerate the structure. However, if a PDB file includes non-standard residues which are not in HyperChem’s template files, but does not include information about bonding and/or hydrogen atoms, then HyperChem will not be able to process the structure properly. If the file does not have bonding information, you will probably end up with a number of disconnected atoms in the workspace which you will have to connect to form molecules. (See “Using the Z-Matrix File Filter” on page 81.) If the file does not include hydrogen atoms, you can add them with Build/Add Hydrogens (see page 123).

*Warning:* HyperChem’s internal storage of structures requires that residues within molecules be numbered sequentially starting from 1. Many PDB files include structures whose residue numbers start with values other than 1, and/or have gaps in the numbering. Therefore, the residue numbering in the original file may be different from the numbering in HyperChem when a PDB file is read in. If you save the structure in the PDB format, the residue numbering will correspond to HyperChem’s internal storage numbering, and so may not be the same as the numbering in the original file.

## Creating PDB-type Files

You can save any molecular system in the workspace as a PDB-type file. Choose Save As on the File menu and L-click on PDB in the Save File dialog box.





HyperChem provides the filename extension *.ent*, but you have to enter a new filename in the File Name text box. The PDB file can contain 13 types of records (each record has a title and takes up one line in the file):

HEADER	These are all comment records from a PDB file that you can enter or edit in the Save File dialog box.
COMPND	
SOURCE	
AUTHOR	
REVDAT	
JRNL	<i>Caution:</i> Comments must adhere to the format described in the PDB document, <i>Atomic Coordinates and Bibliographic Entry Format Description</i> , July, 1989.
REMARK	Comments from a HIN file, entered in the Save File dialog box.
SSBOND	Residues involved in disulphide bridges.
ATOM	The identity and coordinates of each heavy atom in the molecular system. These atoms must be part of the standard set of residues that HyperChem recognizes (see page 735 and page 737).
HETATM	The identity and coordinates of all heavy atoms that are not part of standard residues. Standard residues belong to the set that HyperChem recognizes (see page 735 and page 737).

<b>CONNECT</b>	The numbers of atoms directly bonded to each other. These numbers come from the ATOM or HETATM records. The Numbers label (Choose Labels on the Display menu) also uses these numbers. CONNECT records appear for HETATMs and for all atoms if you choose the Connectivity option (see below).
<b>TER</b>	A record marking the end of a chain.
<b>END</b>	The end of the file.

HyperChem has two options for storing PDB-type files. If you store files with both options off (no 3), the file contains only REMARK, ATOM, HETATM, and END records for heavy atoms. These are the options:

<b>Hydrogens</b>	With this option on (3), the file also contains ATOM records for all hydrogen atoms.
<b>Connectivity</b>	With this option on (3), the file also contains CONNECT records.

The FillPDBRecords setting in the Registry or in chem.ini can require HyperChem to fill each record with spaces, to a total of 70 characters (see “Ab Initio Settings” on page 747).

## Sample PDB Files

HyperChem provides these sample PDB files<sup>1</sup>:

Filename	Molecule
PDB1BNA	DNA, B form, single stranded, 290K Daltons
PDB1CRN	Crambin
PDB3CYT	Cytochrome C, albacore, oxidized
PDB2EST	Elastase-TFAP complex, porcine
PDB1F19	IGG, FAB portion, mouse

1. These files are for public use, courtesy of the Protein Data Bank. To obtain other files, see “Obtaining PDB Files” on page 503.

Filename	Molecule
PDB1FDX	Ferredoxin
PDB1GCN	Glucagon
PDB2PTC	Trypsin-trypsin inhibitor complex
PDB5PTI	Pancreatic trypsin inhibitor
PDB4RHV	Rhinovirus 14, human
PDB6TNA	Transfer RNA, yeast phenylalanine
PDB1XY1	Deamino-oxytocin, wet form



# Chapter 11

## Scripts and DDE

HyperChem provides facilities for these tasks:

- Running HyperChem automatically, with scripts, to easily repeat functions and analyses.
- Integrating HyperChem with your own software and with other commercial software.

### Automatic Operation with Scripts

You can use a small program, called a script, to send instructions to HyperChem. These instructions automatically run specific HyperChem activities. For example, you can automate a molecular dynamics calculation so that you can repeat it easily.

A script reproduces most interactive HyperChem functions, plus a few functions that are not yet available with HyperChem.

You write the instructions in a script file using a text editor, like Windows Notepad. You start a script by using the Script menu in HyperChem.

### Integrating HyperChem with DDE

You can easily integrate other programs that also run with Microsoft Windows with HyperChem. HyperChem retains its integrity and functions, while other programs can request information, send instructions, and collect HyperChem results for their own use.

Programs communicate with HyperChem using the Windows Dynamic Data Exchange (DDE).

## Using HyperChem Messages

HyperChem has a unified set of messages for directing its internal operations and for communicating with other programs. When you use HyperChem menus and dialog boxes, HyperChem selects and uses the correct message.

When you use a script file, HyperChem reads the messages in that file and acts on them. If you are communicating with another program, the DDE commands use some HyperChem messages as arguments and generate other HyperChem messages.

HyperChem uses two classes of messages for scripts and DDE commands:

- |      |   |
|------|---|
| IMSG | An input message is a <i>request</i> to HyperChem to change a variable, carry out a function, or respond with information.  |
| OMSG | An output message is a <i>response</i> that HyperChem provides, including information on settings, calculation results, and error messages. An OMSG goes to the screen (by default), to a file, or to the program requesting the information via DDE. |

### Types of Input Messages

HyperChem has three types of input messages (IMSG):

- |          |  |
|----------|--|
| Variable | A variable (also known as a HyperChem State Variable, or HSV) describes a setting in HyperChem. Variables usually have arguments (see page 511). There are two subtypes of variable messages: <ul style="list-style-type: none"><li>Set HSV Changes (writes) a variable.</li><li>Get HSV Requests (reads) the current setting for a variable. HyperChem responds with an OMSG.</li></ul> |
| Command  | This starts a HyperChem function. A Command message includes arguments (see page 516). This is usually equivalent to using a Menu Activator, choosing options in a dialog box, and then L-clicking on OK.  |

**Menu Activator** This is equivalent to choosing an item on a HyperChem menu. It often produces an on-screen dialog box for choosing options.

## Message Syntax

Messages are not case-sensitive; use any combination of upper- and lowercase letters. Where the syntax shows a space, use one or more spaces or a comma. Parentheses and quotes are optional.

### Variables

Set HSV and Get HSV messages have slightly different syntax:

**Set HSV**            *<Variable-name> = <argument>*

*argument* = string, integer, float, Boolean, enum, or a combination of these. If there are spaces or commas in string arguments, you can enclose them in quotes or parentheses. The equals sign is optional.

**Get HSV**            *<Variable-name> ?*  
or  
*query-value <Variable-name>*

These two statements are equivalent. However, a variable name with arguments can use only a query-value statement.

Separate a variable-name and ? with a space.

**Important:** Define all variables before the commands that require these variables. If you do not set a required variable, HyperChem uses the last value for that variable.

#### Examples:

optim-algorithm steepestdescents

This sets the algorithm for a geometry optimization calculation to steepest descent.

show-perspective ?

or

query-value show-perspective

These both request the setting for Show Perspective: on or off? HyperChem sends a response, yes or no, to the screen (by default), to a file, or to the program requesting the information via DDE.

```
query-value coordinates(2, 1)
```

This requests the Cartesian coordinates for atom 2 in molecule 1. Notice that you cannot use a question mark instead of the query-value statement, since this variable requires arguments.

```
coordinates (2, 1) =0.0,0.0,0.0
```

This changes (writes) the x, y, and z coordinates of atom 2 in molecule 1 to 0.0, 0.0, and 0.0.

```
file-format hin  
open-file benzene.hin
```

These two lines from a script file set the file format variable to hin, then use the open-file command to read the file, benzene.hin.

```
multiple-selections yes  
selection-target atoms  
select-atom 7 1  
select-atom 8 1
```

The first two lines from this script set variables for selection. The next two lines are commands that select atoms 7 and 8 in molecule 1.

```
selection-target molecules  
select-atom 3 1
```

The first line sets the selection level to molecules. This takes priority over the atom selection command in the next line. HyperChem selects the entire molecule 1 instead of only atom 3 in molecule 1.

## Behavior of array variables

Array variables are those specified by indices, such as **coordinates(i,j)**, the coordinates of atom **i** in molecule **j**, or **ir-intensity(i)**, the intensity of peak **i**. Arrays are handled rather differently in HyperChem scripts and require care in their use.

## Atom Indices

First, an array index is commonly the index of an atom. In HyperChem, these indices, which would normally be a single index, **i**, become a pair of indices — molecule index and atom index. For



simplicity here, we denote these two indices — the atom index in a particular molecule and the particular molecule index — as a single index, [i], although these must always be written in an actual script as two indices,

$$[i] \equiv k,l$$

## Two-dimensional Arrays

Thus, for example, the coordinates of an atom ought perhaps to be assigned as:

```
coordinates( [i], 1 ) = 10.0
coordinates( [i], 2 ) = -5.0
coordinates( [i], 3 ) = 20.0
```

where **coordinates** is a two-dimensional array with the first index being the atom index and the second index being the direction = 1 ... 3 for x, y, z. In HyperChem, however, two-dimensional arrays are never assigned or queried with two such indices. One always leaves off the second pre-defined index and instead assigns or queries with a list,

```
coordinates( [i] ) = 10.0, -5.0, 20.0
coordinates( [i] ) ?
```

Two-dimensional arrays, such as **coordinates**, can be queried all at once,

```
coordinates ?
```

They may also be assigned without an index as in the following assignment for a two atom molecule, such as H<sub>2</sub>,

```
coordinates = 0.0, 0.0, 0.0, 0.0, 0.0, 0.74
```

## One-dimensional Arrays

One-dimensional arrays have a single index with a range that may be pre-defined to a read-only value, such as **atom-count**, or with a range that is assignable (read/write), such as the one dimensional array, **ir-frequency**, with a range, **ir-band-count**.

Both types of one-dimensional arrays are assignable using an index,

```
atom-charge( [3] ) = 0.5
ir-frequency(2) = 2076
```

or without an index using a list with a required length determined by the range,

**atom-charge** = 0.0, 0.0, 0.5, -0.5

**ir-frequency** = 1723, 2076

One-dimensional arrays with assignable ranges are referred to by the script command **print-variable-list** as vectors.

The vectors in HyperChem are:

range: **ir-band-count**

**ir-frequency**

**ir-intensity**

**ir-intensity-components**

**ir-normal-mode**

range: **uv-band-count**

**uv-dipole-components**

**uv-energy**

**uv-oscillator-strength**

**uv-spin**

**uv-total-dipole**

**uv-transition-dipole**

range: **named-selection-count**

**named-selection-name**

**named-selection-value**

range: **selected-atom-count**

**selected-atom**

range: **orbital-count**

**alpha-scf-eigenvector**

**beta-scf-eigenvector**

**scf-orbital-energy**

range: **graph-horizontal-grid-size**

**graph-data-row**

range: **molecule-count**

**residue-count**

range: 10

**script-menu-caption**  
**script-menu-checked**  
**script-menu-command**  
**script-menu-enabled**  
**script-menu-help-file**  
**script-menu-help-id**  
**script-menu-in-use**  
**script-menu-message**

The following is an example of the usage and syntax for a vector.

*Example:*

First declare the size of the vector:

**ir-band-count = 5**

Next, assignment can be performed using one of the following three methods:

Method 1:

**ir-frequency(1) = 100**  
**ir-frequency(2) = 170**  
**ir-frequency(3) = 200**  
**ir-frequency(4) = 210**  
**ir-frequency(5) = 213**

Method 2:

**ir-frequency = 100, 170, 200, 210, 213**

Method 3:

**ir-frequency = 100 170 200 210 213**

Values for a vector variable can be obtained in the following manner:

**query-value ir-frequency**  
**ir-frequency ?**

Both the above inquiries will yield the following results based on the previous assignments.

**ir-frequency(1) = 100**  
**ir-frequency(2) = 170**  
**ir-frequency(3) = 200**

**ir-frequency(4) = 210**

**ir-frequency(5) = 213**

Individual vector components can be queried in the following manner:

**query-value ir-frequency(2)**

**ir-frequency(2) ?**

Both the above inquiries will yield the 2<sup>nd</sup> component of the **ir-frequency** vector.

You can obtain continual notification of all coordinates in a script by sending the command “**notify-on-update coordinates**”, but you cannot **notify-on-update** for a single atom’s coordinates. The DDE equivalent of **notify-on-update** is a hot, or automatic link, and only the whole array may be addressed in these links. Beware, however: in some cases the output may be voluminous.

## Commands

Command messages can have one or more arguments, each separated by at least one space or a comma:

*<request-name> <argument> <argument> . . .*

*argument* = string, integer, float, Boolean, enum, or a combination of these. Enclose a string in quotes (“ ”) when there are spaces between arguments.

*request-name* = command name

*Examples:*

open-file benzene.hin

This request opens the file benzene.hin.

execute-client graph.exe Graph Form1

This is a request to start and run another DDE application. Graph.exe is the filename for the DDE application. “Graph” is the name of the application. “Form1” is the topic, either a filename or some unit of information, that the application recognizes.

## Arguments

The types of arguments for variables or commands are the following:

Boolean	Yes or no, true or false, 0 or 1.
string	Text (letters, characters, or symbols, in upper- or lowercase, unlimited number of characters). Enclose a string in quotes (“ ”) if it contains spaces, tabs, or newline characters. If a string with quotes includes a backslash (“\”), you must double the backslash (“\\”). Thus, a path might be specified as “C:\\Program Files\\Hypercube”. (You do not need to double the backslash if the string is <i>not</i> enclosed in quotes.) In a string with quotes, you can designate a newline character as “\n”.
filename	A type of string requiring a DOS filename.
enum	A type of string requiring one of a limited set of possibilities.
int	An integer.
float	A floating point (decimal) number. For an angle, the number is in degrees.

Some Get HSV statements (requesting a variable) require only a subset of the possible arguments. For example, setting the atom-name variable requires all arguments (atom number, molecule number, and atom name), but requesting the variable requires only the atom number and molecule number. HyperChem responds with all three variables.

## Menu Activators

Menu activators are text strings with hyphens between each word. Every menu activator starts with “menu-.”

*menu-<menu-name>-<item-name>*

*Example:*

menu-edit-zoom

This requests the Zoom item on the Edit menu. You see the result of choosing this item, the Zoom dialog box, on screen.

## **HyperChem Messages**

The following tables list all variables, commands, and menu activators organized according to menu item, dialog box, and function. Use these messages to write scripts or as arguments in DDE commands.

The numbers in the first column of each table are used in an alphabetic cross-reference to all messages (see “Message Cross-Reference” on page 584).

For specific discussions of scripts and DDE, with examples, see page 602 and page 604.

## File Menu

#	File Variable	Type	R/W <sup>a</sup>	Use
1	current-file-name	filename	R	The filename corresponding to the molecular system displayed in the workspace
2	path	string	R, W	Current directory
3	file-format	enum (hin, pdb, skc, ml2, zmt, mol, chm)	R,W	The file format for a molecular system
4	file-needs-saved	Boolean	R,W	Yes to save the molecular system if it contains changes since it was read in or if it is new
5	velocities-in-hin-file	Boolean	R,W	Yes to store atom velocities in a HIN file
6	view-in-hin-file	Boolean	R, W	Should view be written into a HIN file? (Useful for comparing HIN files)
7	hydrogens-in-pdb-file	Boolean	R,W	Yes to include hydrogen coordinates in a PDB file
8	connectivity-in-pdb-file	Boolean	R,W	Yes to include connectivity information (CONNECT records) in a PDB file
9	non-standard-pdb-names	Boolean	R,W	Enables HyperChem to read some PDB files, produced by other programs, which do not properly conform to the PDB specifications. In particular, it allows the use of left-justified atom names. This may prevent correct reading of standard-conforming files.
10	window-color	enum (mono- chrome or a color <sup>b</sup> )	R,W	The background color for the HyperChem workspace; mono-chrome results in a white background with bonds in color or black

#	File Variable	Type	R/W <sup>a</sup>	Use
11	bond-color	enum (byelement or a <i>color</i> <sup>b</sup> )	R,W	The color used for drawing all atoms (and bonds). Byelement uses a color based on the atomic number set in the Element Color dialog box (on the Display menu)
12	selection-color	enum (thickline or a <i>color</i> <sup>b</sup> )	R,W	Assigns a color or bond appearance to differentiate atoms while they are selected
13	positives-color	(a <i>color</i> <sup>b</sup> )	R, W	Color for positive orbital lobes, contour lines, occupied orbitals, etc.
14	negatives-color	(a <i>color</i> <sup>b</sup> )	R, W	Color for negative orbital lobes, contour lines, unoccupied orbitals, etc.
15	x-y-rotation-cursor	float (>0 . . ≤3600)	R,W	The amount of XY rotation, in degrees, for a drag from one edge of the workspace to the other
16	reorder- selections	Boolean	R, W	Yes to reorder atoms in selections to create proper angles; no to give user direct control over atom order
17	translate-merged- systems	Boolean	R, W	Yes to translate newly merged/pasted structures so that they do not overlap
18	z-rotation-cursor	float (>0 . . ≤3600)	R,W	The amount of Z rotation, in degrees, for a horizontal drag from one edge of the workspace to the other
19	export-dipole	Boolean	R, W	Whether or not to export dipole moment data to .EXT file.
20	export-ir	Boolean	R, W	Whether or not to export IR spectral data to .EXT file.
21	export-orbitals	Boolean	R, W	Whether or not to export orbital data to .EXT file.



#	File Variable	Type	R/W <sup>a</sup>	Use
22	export-uv	Boolean	R, W	Whether or not to export UV spectral data to .EXT file.
23	import-dipole	Boolean	R, W	Whether or not to import dipole moment data from .EXT file.
24	import-ir	Boolean	R, W	Whether or not to import IR spectral data from .EXT file.
25	import-orbitals	Boolean	R, W	Whether or not to import orbital data from .EXT file.
26	import-uv	Boolean	R, W	Whether or not to import UV spectral data from .EXT file.
27	z-translation-cursor	float (>0 . . ≤1000)	R,W	The amount of Z translation, in Ångstroms, for a horizontal drag from one edge of the workspace to the other
28	zoom-cursor	float (>1 . . ≤1000)	R,W	The amount of magnification for a vertical drag across the workspace
29	clip-cursor	float (>0 . . ≤1000)	R,W	The amount of clipping plane movement, in Ångstroms, for a drag vertically across the workspace
30	x-y-rotation-icon-step	float (>0 . . ≤3600)	R,W	The amount of XY rotation, in degrees, from a shift-click at the extreme edge of the Tool icon
31	z-rotation-icon-step	float (>0 . . ≤3600)	R,W	The amount of Z rotation, in degrees, from a shift-click at the extreme edge of the Tool icon
32	x-y-translation-icon-step	float (>0 . . ≤1000)	R,W	The amount of XY translation, in Ångstroms, from a shift-click at the extreme edge of the Tool icon
33	z-translation-icon-step	float (>0 . . ≤1000)	R,W	The amount of Z translation, in Ångstroms, from a shift-click at the extreme edge of the Tool icon

#	File Variable	Type	R/W <sup>a</sup>	Use
34	zoom-icon-step	float (>1 . . ≤1000)	R,W	The amount of magnification from a shift-click at the extreme edge of the Tool icon
35	clip-icon-step	float (>0 . . ≤1000)	R,W	The amount of clipping plane movement, in Ångstroms, from a shift-click at the extreme edge of the Tool icon
36	translate-whole-molecules	Boolean	R,W	Yes to translate only whole molecules. This prevents translating individual atoms of a molecule and distorting the structure. No lets you translate selected atoms in larger molecules
37	use-fast-translation	Boolean	R,W	Yes to increase the speed of XY translation by temporarily substituting a bitmapped image of the molecular system
38	printer-background-white	Boolean	R,W	Yes to force printer background color to white. No to use the current window color
39	image-source-window	enum (toplevel, workspace, hyperchem, fullscreen)	R,W	Sets the extent of the image captured with "edit-menu-copy-image"
40	image-file-bitmap	Boolean	R,W	Yes to save an image in bitmapped format
41	image-file-bitmapRGB	Boolean	R, W	Yes to save bitmap pictures in 24-bit color instead of 8-bit color
42	image-file-metafile	Boolean	R,W	Yes to save an image in metafile format

#	File Variable	Type	R/W <sup>a</sup>	Use
43	image-destination-file	Boolean	R,W	Yes to save an image in chem.bmp and/or chem.wmf, depending on the settings of the previous two variables
44	image-destination-filename	String	R, W	Save image to a given file.
45	image-destination-clipboard	Boolean	R,W	Yes to save an image to the Clipboard
46	image-include-cursor	Boolean	R,W	Yes to include the cursor in an image
47	image-color	Boolean	R,W	Yes to save an image in color

a. R = read only. R,W = read and write.

b. Choose one for color: black, blue, green, cyan, red, violet, yellow, white.

	File Command	Arguments	Use
48	open-file	filename	Opens a file containing a molecular system
49	merge-file	filename	Opens a file containing a molecular system and adds this system to a molecular system already in the workspace
50	write-file	filename	Stores the current molecular system in a file
51	print	none	Print to default printer
52	start-logging	filename, Boolean	Starts recording HyperChem results in log filename. Yes = append data to file of the same name. No = replace file
53	stop-logging	none	Stops recording HyperChem results in a log file
54	log-comment	string	Stores comment (string) in an active log file

	File Command	Arguments	Use
55	import-property-file	filename	Reads quantum mechanics data (dipole moment, orbital data, IR and/or UV/vis spectra) from the named file
56	export-property-file	filename	Writes quantum mechanics data (dipole moment, orbital data, IR and/or UV/vis spectra) to the named file
57	delete-file	filename	Delete the file specified by filename without giving any warning message
58	file-diff-message	filename1, filename2, string1, string2	Compare the contents of filename1 with those of filename2; report string1 if the files are the same and string2 if they are different

#	File Menu Activator	Use	Dialog Box
59	menu-file-new	Clears the workspace	—
60	menu-file-open	Opens a file	Open File
61	menu-file-merge	Merges a file into the workspace	Merge File
62	menu-file-save	Updates the file the system was read from	Save File, if current system is untitled; otherwise, saves file without showing dialog box
63	menu-file-save-as	Saves the system in a different file	Save As...
64	menu-file-start-log	Starts logging computation results in a file	Open Log
65	menu-file-stop-log	Stops logging	—
66	menu-file-log-comments	Stores a comment in the active log file	Log Comments

#	File Menu Activator	Use	Dialog Box
67	menu-file-export	Saves the results of quantum calculations, including IR and UV/visible spectra	Export Properties
68	menu-file-import	Imports values of quantum calculations, including IR and UV/visible spectra	Import Properties
69	menu-file-print	Prints an image of the system	Print
70	menu-file-preferences	Customizes the program	Preferences
71	menu-file-exit	Exits HyperChem	—

## Edit Menu

#	Edit Variable	Type	R/W <sup>a</sup>	Use
72	front-clip	float ( $\geq 0$ . . $\leq 3200$ )	R, W	The distance in Ångstroms of the front clipping plane from the viewer. Must be at least 1 less than back-clip value
73	back-clip	float ( $\geq 0$ . . $\leq 3200$ )	R, W	The distance in Ångstroms of the back clipping plane from the viewer. Must be at least 1 more than front clip value
74	mouse-mode	enum (drawing, selecting, rotating, zrotating, translating, ztranslating, zooming, clipping)	R, W	The function of the cursor (tool) in the workspace

a. R = read only. R,W = read and write.

#	Edit Command	Arguments	Use
75	align-molecule	enum1, enum2, enum3, enum4 (enum1 and 3 = primary, secondary, tertiary; enum2 and 4 = x, y, z, line)	Aligns the inertial axes of selected atoms or a whole molecular system with the Viewer's Coordinate System, or with a LINE in the Molecular Coordinate System. Enum1 and enum2 describe the major alignment. Enum3 and enum4 describe the optional minor alignment
76	align-viewer	enum (x, y, z, line)	Aligns a viewer's z coordinate with an axis or with a LINE in the Molecular Coordinate System
77	delete-selected-atoms	none	Deletes selected atoms. Equivalent to menu-edit-clear

#	Edit Command	Arguments	Use
78	rotate-viewer	enum, float (enum = x, y, z, line)	Rotates the viewer around (axis = x, y, z, or LINE) by angle in degrees
79	rotate-molecules	enum, float (enum = x, y, z, line)	Rotates molecules around (axis = x, y, z, or LINE) by angle in degrees
80	translate-view	float1, float2, float3 ( $\geq 0$ . . $\leq 3200$ )	Translates the viewer (in the Viewer's Coordinate System) by a change in coordinates (dx, dy, dz), in Ångstroms
81	translate-selection	float1, float2, float3 ( $\geq 0$ . . $\leq 3200$ )	Translates the selection in the Molecular Coordinate System to coordinates x, y, and z
82	zoom	float ( $\geq 0.01$ . . $\leq 50$ )	Sets the magnification of the current molecular system. A value of 1.0 fits the whole molecular system into the workspace and corresponds to Scale to Fit (on the Display menu)
83	set-bond-length	float ( $> 0$ . . $\leq 3200$ )	Sets the bond length for the selected bond
84	set-bond-angle	float ( $\geq 0$ . . $\leq 180$ )	Sets the bond angle for the selected angle
85	set-bond-torsion	float ( $\geq -360$ . . $\leq 360$ )	Sets the torsion angle for the selected torsion

#	Edit Menu Activator	Use	Dialog Box
86	menu-edit-clear	Deletes the selection	—
87	menu-edit-cut	Deletes the selection and places a copy of the molecular structure on the Clipboard	—

#	Edit Menu Activator	Use	Dialog Box
88	menu-edit-copy	Copies the selection to the Clipboard	—
89	menu-edit-copy-isis-sketch	Copies the selection to the Clipboard in ISIS Sketch format	—
90	menu-edit-paste	Copies the Clipboard's contents to the workspace	—
91	menu-edit-copy-image	Copies the image of the molecular system to the Clipboard	—
92	menu-edit-setup-image	Sets up the image copying process (type of image)	Setup Image
93	menu-edit-invert	Inverts the selection through POINT	—
94	menu-edit-reflect	Reflects the selection through PLANE	—
95	menu-edit-rotate	Rotates the selection or the viewer	Rotate
96	menu-edit-translate	Translates the selection or the viewer	Translate
97	menu-edit-zoom	Changes the magnification	Zoom
98	menu-edit-z-clip	Changes the Z clipping	Z Clip
99	menu-edit-align-viewer	Aligns the viewer's z axis	Align Viewer
100	menu-edit-align-molecules	Aligns the inertial axis of a molecule(s)	Align Molecules
101	menu-edit-set-bond-length	Changes a bond length	Set Bond Length
102	menu-edit-set-bond-angle	Changes a bond angle	Set Bond Angle
103	menu-edit-set-bond-torsion	Changes a torsion angle	Set Bond Torsion



## Build Menu

#	Build Variable	Type	R/W <sup>a</sup>	Use
104	explicit-hydrogens	Boolean	R, W	Allows the Drawing tool to add hydrogens. If yes, all single bonded atoms are hydrogens. If no, Drawing tool places only the default element (usually a heavy atom)
105	default-element	int (0 . . 103)	R, W	The atomic number of the default element for drawing. 0 = lone pair
106	allow-ions	Boolean	R, W	Allows excess valence on atoms. If yes, you can add up to 12 bonds to any atom. If no, each element has its normal limit of valence electrons
107	builder-enforces-stereo	Boolean	R, W	Specifies whether the Model Builder implicitly enforces existing stereochemistry.

a. R = read only. R,W = read and write.

#	Build Command	Arguments	Use
108	set-atom-type	string	Provides the type (string) for a selected set of atoms in the active force field. Use only a type that the force field recognizes. Use the correct upper- and lowercase characters for atom types (for example CA, not ca or Ca)
109	set-atom-charge	float ( $\geq -100$ . . $\leq 100$ )	Provides the charge (float) for a selected set of atoms
110	set-formal-charge	int (-3 . . +3)	Provides the formal charge (int) for a selected set of atoms
111	set-atom-charges-to-formal	none	Sets all atom charges to the value of formal charge.
112	create-atom	int (0 . . 103)	Draws an atom, with atomic number int, at the origin of the Molecular Coordinate System

#	Build Command	Arguments	Use
113	delete-atom	int1, int2	Deletes atom int1 from molecule int2
114	set-bond	int1, int2, int3, int4, enum (n, h, s, d, t, a)	Draws a bond between atom int1 in molecule int2 and atom int3 in molecule int4. The bond order is n = none (deletes an existing bond), h = hydrogen, s = single, d = double, t = triple, and a = aromatic
115	re-bond-system	none	Destroy all bonds and create new bonds with an algorithm that uses a distance criterion along with atomic number of atoms.
116	constrain-geometry	enum (none, linear, trigonal, tetrahedral, square-planar, bipyramidal, octahedral) or computed	Constrains the bonding to a selected atom during model building. None removes existing constraints
117	constrain-bond-length	float ( $>0 \dots \leq 100$ ) or computed	Constrains a bond length between two selected atoms during model building. Computed removes a constraint and uses the Model Builder's default bond length
118	constrain-bond-angle	float ( $\geq -360 \dots \leq 360$ ) or computed	Constrains a bond angle between three selected atoms during model building. Computed removes a constraint and uses the Model Builder's default bond angle
119	constrain-bond-torsion	float ( $\geq -360 \dots \leq 360$ ) or computed	Constrains a bond torsion angle between four selected atoms during model building. Computed removes a constraint and uses the Model Builder's default torsion angle
120	constrain-drawing	Boolean	Whether to constrain bond lengths and angles to canonicalize drawing of molecule.
121	unconstrain-bond-length	none	Remove any length constraint on the two currently selected atoms
122	unconstrain-bond-angle	none	Remove any angle constraint on the three currently selected atoms

#	Build Command	Arguments	Use
123	unconstrain-bond-torsion	none	Remove any torsion constraint on the four currently selected atoms
124	remove-stereo-constraint	int1, int2	Removes any stereo constraints from atom number int1 into molecule int2
125	remove-all-stereo-constraints	int1, int2	Removes all stereo constraints
126	constrain-bond-up	int1, int2, int3, int4	Constrains a bond to be up. The bond is the one from atom int1 in molecule int2 to atom int3 in molecule int4
127	constrain-bond-down	int1, int2, int3, int4	Constrains a bond to be down. The bond is the one from atom int1 in molecule int2 to atom int3 in molecule int4
128	constrain-fix-stereo	int1, int2	Constrains atom to enforce the current stereochemistry about atom int1 in molecule int2
129	constrain-change-stereo	int1, int2	Constrains atom to change the current stereochemistry about atom int1 in molecule int2
130	cycle-bond-stereo	int1, int2, int3, int4	Advances the stereo constraint along a bond. The bond is the one from atom int1 in molecule int2 to atom int3 in molecule int4
131	cycle-atom-stereo	int1, int2	Advances the stereo constraint about atom int1 in molecule int2
132	change-stereochem	int1, int2	Immediately change the existing stereochemistry about atom int1 in molecule int2

#	Build Menu Activator	Use	Dialog Box
133	menu-build-explicit-hydrogens	Toggles explicit hydrogens for drawing	—

#	Build Menu Activator	Use	Dialog Box
134	menu-build-default-element	Changes the default element for drawing atoms	Default Element
135	menu-build-add-hydrogens	Adds hydrogens to fill the atoms' valences	—
136	menu-build-model-build	Transforms a 2D drawing into a 3D chemical structure	—
137	menu-build-allow-ions	Toggles restrictions on excess valence	—
138	menu-build-united-atoms	Removes hydrogens to form united atoms	—
139	menu-build-all-atoms	Restores hydrogens to united atoms	—
140	menu-build-calculate-types	Assigns atom types for the chosen force field	—
141	menu-build-compile-type-rules	Compiles the chem.rul file into a binary file (typerule.bin)	—
142	menu-build-set-atom-type	Sets the atom type for selected atoms	Set Atom Type
143	menu-build-set-charge	Sets the charge on selected atom	Set Atom Charge
144	menu-build-set-mass	Sets the mass of selected atom	Set Atom Mass
145	menu-build-constrain-geometry	Constrains the geometry for bonds to a selected atom	Constrain Geometry
146	menu-build-constrain-bond-length	Constrains the distance between two selected atoms	Constrain Bond Length
147	menu-build-constrain-bond-angle	Constrains the bond angle between three selected atoms	Constrain Bond Angle
148	menu-build-constrain-bond-torsion	Constrains the torsion angle of four selected atoms	Constrain Bond Torsion

## Select Menu

#	Select Variable	Type	R/W <sup>a</sup>	Use
149	selection-target	enum (molecules, residues, atoms)	R, W	The level of selection or the smallest unit of selection
150	selected-atom-count	int	R	Reports the number of atoms selected
151	selected-atom	int1, int2, int3	R	When you give the number of the selected atom (int1), using this variable reports the sequence number of this atom (int2) in molecule number int3
152	multiple-selections	Boolean	R, W	Yes to accumulate selections, or no to cancel the previous selection with each new selection
153	select-sphere	Boolean	R, W	Yes for group selection with a 3D sphere, or no for group selection with a 2D rectangle
154	named-selection-count	int	R	The count of the named selections
155	named-selection-name	int, string	R	When you give the number of a named selection (int), using this variable gives the name of that selection
156	named-selection-value	int1, float	R	The (read only) "value" of number int1 named selection. The "value" is the x, y, and z coordinates for 1-atom selections, the distance for 2-atom selections, angle for 3, torsion for 4, and "(nil)" for anything else
157	selection-value	float	R	The value of the current selection (bond, angle, etc.)

a. R = read only. R,W = read and write.

#	Select Command	Arguments	Use
158	select-none	none	Deselects all atoms <sup>a</sup>
159	select-name	string	Selects the atoms with the selection name = string <sup>a</sup> , where string has at least one lowercase character (up to 19 total) and no spaces
160	select-atom	int1, int2	Selects atom int1 in molecule int2. Honors current selection level (see selection-target variable) <sup>a,b</sup>
161	un-select-atom	int1, int2	Deselects atom int1 in molecule int2. Honors current selection level (see selection-target variable) <sup>a,b</sup>
162	select-residue	int1, int2	Selects residue int1 in molecule int2. Disregards current selection level, always selecting a residue if present. If there is no residue, reports an error (see selection-target variable) <sup>a</sup>
163	un-select-residue	int1, int2	Deselects residue int1 in molecule int2. Disregards current selection level and always deselects a residue if present. If there is no residue, reports an error (see selection-target variable) <sup>a</sup>
164	name-selection	string	Gives a name to the current selection. String = POINT, LINE, PLANE (in all uppercase) or any set of characters (not all uppercase) up to 19 characters long
165	name-molecule	string	Give name String-1 to molecule of current selection.
166	delete-named-selection	string	Removes a named selection from the list of named selections
167	select-path	int, int, int, int	Selects atoms in path from int-1 in molecule int-2 to int-3 in molecule int-4.

a. Honors multiple-selections (variable) value (yes or no).

b. To select a molecule—for example; molecule 3—use selection-target molecules, and then select-atom. Use a similar procedure with un-select-atom to deselect a molecule.

#	Select Menu Activator	Use	Dialog Box
168	menu-select-atoms	Sets selection level to "Atoms"	—
169	menu-select-residues	Sets selection level to "Residues"	—
170	menu-select-molecules	Sets selection level to "Molecules"	—
171	menu-select-multiple-selections	Toggles accumulation of selections	—
172	menu-select-select-sphere	Toggles between 2D and 3D group selection	—
173	menu-select-select-all	Selects all visible atoms in the molecular system	—
174	menu-select-complement-selection	Complements the current selection	—
175	menu-select-select	Selects atoms by an assigned name or number	Select
176	menu-select-name-selection	Assigns a name to the current selection	Name Selection
177	menu-select-extend-ring	Extends selection to an entire ring	—
178	menu-select-extend-side-chain	Extends a selection to a side chain	—
179	menu-select-extend-to-sp3	Extends a selection to $sp^3$ - $sp^3$ single bonds and singly bonded atoms	—
180	menu-select-select-backbone	Selects all atoms in the backbone of a polymer built from residues	—

## Display Menu

#	Display Variable	Type	R/W <sup>a</sup>	Use
181	show-perspective	Boolean	R, W	Yes to display molecules in perspective, or no for a planar projection
182	show-stereo	Boolean	R, W	Yes to display molecules as a stereo pair, or no for a single model
183	wall-eyed-stereo	Boolean	R, W	Yes sets the stereo pair for wall-eyed viewing; select no for crosseyed viewing
184	show-ribbons	Boolean	R, W	Yes displays a protein backbone with ribbons
185	render-method	enum (sticks, balls, balls-and-cylinders, spheres, dots, sticks-and-dots)	R, W	The rendering method for the display
186	balls-shaded	Boolean	R, W	Whether balls and balls-and-cylinders should be shaded instead of being solidly colored
187	balls-highlighted	Boolean	R, W	Whether balls and balls-and-cylinders should be highlighted when shaded
188	balls-radius-ratio	float (0 . . 1)	R, W	Size of the balls relative to the maximum value
189	spheres-shaded	Boolean	R, W	Whether spheres should be shaded instead of being solidly colored (as disks)
190	spheres-highlighted	Boolean	R, W	Whether spheres should be highlighted when shaded
191	sticks-width	integer (0 . . 25)	R, W	Sticks rendering width in pixels



#	Display Variable	Type	R/W <sup>a</sup>	Use
192	cylinders-width-ratio	float (0 . . 1)	R, W	Size of the cylinders relative to the maximum value
193	cylinders-color-by-element	Boolean	R, W	Color cylinders using element colors (yes) or gray (no).
194	grid-min-value	real	R	The minimum value in the grid of quantum mechanical data
195	grid-max-value	real	R	The maximum value in the grid of quantum mechanical data
196	isosurface-threshold	real (grid-min-value ≤ isosurface-threshold ≤ grid-max-value)	R, W	Sets the value of the isosurface of the quantum mechanical grid data to render
197	isosurface-render-method	enum (wire-mesh, Jorgensen-Salem, lines, flat-surface, shaded-surface, Gouraud-shaded-surface, translucent-surface)	R, W	Rendering method used to draw isosurfaces
198	isosurface-transparency-level	float (0 . . 1)	R, W	The level of transparency of the isosurface (0 = opaque, 1 = transparent)
199	isosurface-hide-molecule	Boolean	R, W	Hide the structure while an isosurface is being displayed
200	dot-surface-angle	float angle (≥-90 . . ≤90)	R, W	The angle of planes in a dot rendering relative to the XZ plane of the Viewer's Coordinate System

#	Display Variable	Type	R/W <sup>a</sup>	Use
201	double-buffered-display	Boolean	R, W	Yes sets up two copies of the display in memory. HyperChem then draws molecules off screen and shows them when complete. No sets HyperChem to display one atom at a time
202	cpk-max-double-buffer-atoms	int (0 . . 32767)	R,W	If double-buffered display is on, HyperChem limits double buffering of a spheres rendering to this number of atoms (int), but turns off double buffering if the number of atoms exceeds this number
203	show-hydrogens	Boolean	R, W	Toggles the display of hydrogen atoms on (yes) or off (no)
204	show-periodic-box	Boolean	R, W	Toggles the display of an existing periodic box on (yes) or off (no). Does not create the box or periodic conditions
205	show-multiple-bonds	Boolean	R, W	Toggles the display of double, triple, and aromatic bonds with multiple lines on (yes) or off (no)
206	show-gradient-vectors	Boolean	R, W	Show atomic gradient vectors.
207	bond-spacing-display-ratio	float ( $\geq 0$ . . $\leq 1$ )	R, W	The distance between lines in multiple bonds. This affects only nonraster devices, like printers, and metafile images
208	show-hydrogen-bonds	Boolean	R, W	Toggles the display of hydrogen bonds on (yes) or off (no)
209	show-isosurface	Boolean	R, W	Toggles the display of a calculated isosurface on (yes) or off (no)

#	Display Variable	Type	R/W <sup>a</sup>	Use
210	show-axes	Boolean	R, W	Toggles the display of inertial axes on (yes) or off (no)
211	show-dipoles	Boolean	R, W	Toggles the display of dipoles on (yes) or off (not)
212	show-aromatic-rings-as-circles	Boolean	R, W	Whether aromatic rings show as circles or dotted bonds.
213	hide-toolbar	Boolean	R, W	Toggles whether the toolbar should be hidden (yes) or visible (no)
214	atom-labels	enum (none, symbol, name, number, type, basisset, mass, charge, chirality)	R, W	Labels apply to visible, selected atoms. If there is no selection, applies specified label to all newly created atoms or atoms read in from a file
215	residue-labels	enum (none, name, sequence, namesequence)	R, W	The label for visible, selected residues. If there is no selection, applies specified label to all newly created residues or to residues read in from a file
216	show-stereochem-wedges	Boolean	R, W	Yes displays stereochemistry constraints on the screen
217	periodic-box-size	float1, float2, float3	R	Returns the size of the periodic box in x, y, z
218	moments-of-inertia	float1, float2, float3	R	Returns the x, y, z moments of inertia of selection or of entire workspace

a. R = read only. R,W = read and write.

#	Display Command	Arguments	Use
219	color-selection	enum (byelement or <i>color</i> <sup>a</sup> )	Sets the persistent color applied to selected atoms when they are deselected. by element uses the colors defined by color-element
220	color-element	int1, enum (default or <i>color</i> <sup>a</sup> )	Assigns a new color (enum) to element int1. Int1 = 1–103. Default = reverts to the color in the HyperChem element color table (not available to user)
221	revert-element-colors	none	Applies default colors to all elements. Same as the default argument for color-element, but applies color for all elements

a. Choose one for color: black, blue, green, cyan, red, violet, yellow, white.

#	Display Menu Activator	Use	Dialog Box
222	menu-display-scale-to-fit	Positions and changes the magnification of the selection to fit it in the center of the workspace	—
223	menu-display-overlay	Superimposes two selected molecules based on a selected bond angle in each	—
224	menu-display-show-all	Shows all atoms	—
225	menu-display-show-selection-only	Shows only those atoms in the current selection	—
226	menu-display-hide-selection	Hides all atoms in the current selection	—
227	menu-display-rendering	Provides six choices of a molecular model	Rendering
228	menu-display-last-rendering	Render using last rendering options	—

#	Display Menu Activator	Use	Dialog Box
229	menu-display-isosurface	Allows user to select a new rendering mode for an isosurface	Plot Molecular Properties, Plot Orbitals
230	menu-display-show-isosurface	Toggles the display of an isosurface	—
231	menu-display-show-hydrogens	Toggles the display of hydrogen atoms	—
232	menu-display-show-periodic-box	Toggles the display of an existing periodic box. Does not create the box or periodic conditions	—
233	menu-display-show-multiple-bonds	Toggles the display of multiple bonds	—
234	menu-display-show-hydrogen-bonds	Toggles the display of hydrogen bonds	—
235	menu-display-recompute-h-bonds	Recomputes hydrogen bonds between the selected atoms	—
236	menu-display-show-inertial-axes	Toggles the displays of inertial axes for selected atoms or the whole molecular system	—
237	menu-display-show-dipole-moment	Toggles display of the dipole moment after it has been calculated for selected atoms or the whole molecular system	—
238	menu-display-labels	Provides a choice of labeling options for selected atoms and residues	Labels
239	menu-display-color	Provides a persistent color to differentiate selected atoms when they are deselected	Selection Color
240	menu-display-element-color	Provides alternative default colors for each element	Element Color

## Databases Menu

#	Databases Variable	Type	R/W <sup>a</sup>	Use
241	amino-phi	float ( $\geq -360$ . . $\leq 360$ )	R, W	Phi ( $\phi$ ) torsion angle of an amino acid residue
242	amino-psi	float ( $\geq -360$ . . $\leq 360$ )	R, W	Psi ( $\psi$ ) torsion angle of an amino acid residue
243	amino-omega	float ( $\geq -360$ . . $\leq 360$ )	R, W	Omega ( $\omega$ ) torsion angle of an amino acid residue—the peptide bond angle
244	amino-isomer	enum (l, d)	R, W	The isomer of an amino acid residue, d or l
245	nucleic-sugar-pucker	enum (2-endo, 3-endo)	R, W	Pucker of ribose sugar in nucleic acid residue
246	nucleic-alpha	float ( $\geq -360$ . . $\leq 360$ )	R, W	Alpha ( $\alpha$ ) torsion angle in a nucleic acid backbone
247	nucleic-beta	float ( $\geq -360$ . . $\leq 360$ )	R, W	Beta ( $\beta$ ) torsion angle in a nucleic acid residue
248	nucleic-gamma	float ( $\geq -360$ . . $\leq 360$ )	R, W	Gamma ( $\gamma$ ) torsion angle in a nucleic acid residue
249	nucleic-delta	float ( $\geq -360$ . . $\leq 360$ )	R, W	Delta ( $\delta$ ) torsion angle in a nucleic acid residue
250	nucleic-epsilon	float ( $\geq -360$ . . $\leq 360$ )	R, W	Epsilon ( $\epsilon$ ) torsion angle in a nucleic acid residue
251	nucleic-zeta	float ( $\geq -360$ . . $\leq 360$ )	R, W	Zeta ( $\zeta$ ) torsion angle in a nucleic acid backbone
252	nucleic-chi	float ( $\geq -360$ . . $\leq 360$ )	R, W	Chi ( $\chi$ ) torsion angle for base (twist angle) in a nucleic acid residue

---

#	Databases Variable	Type	R/W <sup>a</sup>	Use
253	nucleic-backwards	Boolean	R, W	Toggles direction for building a strand; yes for backwards, from 3' to 5'
254	nucleic-double-strand	Boolean	R, W	Toggles number of strands to build into a polynucleotide; yes for double, no for single

---

a. R = read only. R,W = read and write.

#	Databases Command	Arguments	Use
255	add-amino-acid	string	Chooses the next amino acid residue to add. Use the three-character residue name (e.g., ala)
256	amino-alpha-helix	none	Uses alpha-helical torsion angles for the next amino acid residue added
257	amino-beta-sheet	none	Uses beta-sheet torsion angles for the next amino acid residue added
258	add-nucleic-acid	string	Chooses the next nucleic acid residue to add. Use the two-to-four-character residue name (e.g., da)
259	nucleic-a-form	none	Uses A-form torsion angles for the next nucleic acid residue added
260	nucleic-b-form	none	Uses B-form torsion angles for the next nucleic acid residue added
261	nucleic-z-form	none	Uses Z-form torsion angles for the next nucleic acid residue added
262	mutate-residue	string	Changes the selected residue into residue string, where string = residue name (2–3 characters). Select one residue, not a cap. Residues must be of the same type, N or A

#	Databases Menu Activator	Use	Dialog Box
263	menu-databases-amino-acids	Constructs a sequence of amino acids	Amino Acids
264	menu-databases-make-zwitterion	Changes a polypeptide into a zwitterion	—
265	menu-databases-remove-ionic-ends	Changes a zwitterion back to a neutral polypeptide	—



#	Databases Menu Activator	Use	Dialog Box
266	menu-databases-nucleic-acids	Constructs a sequence of nucleotides	Nucleic Acids
267	menu-databases-add-counter-ions	Adds a sodium ion near each phosphate in a polynucleotide	—
268	menu-databases-mutate	Changes one residue into another	Mutate

## Setup Menu

#	Setup Variable	Type	R/W <sup>a</sup>	Use
269	calculation-method	enum (molecular-mechanics, semi-empirical, abinitio, DFT)	R, W	A molecular mechanics, or semi-empirical or <i>ab initio</i> quantum mechanics method for calculations
270	molecular-mechanics-method	enum (mm+, amber, bio+, opls, amber94, charmm22, tndo)	R, W	The type of molecular mechanics force field for calculations
271	semi-empirical-method	enum (extendedhuckel, cndo, indo, mindo3, mndo, am1, pm3, zindo1, zindos, mnod, tndo)	R, W	The type of semi-empirical quantum mechanics method for calculations
272	keep-atom-charges	Boolean	R, W	Keep atom charges when changing calculation methods.
273	mechanics-dielectric	enum1, enum2, enum3, enum4 <sup>b</sup> (constant, distancedependent)	R,W	The method for calculating dielectric permittivity

#	Setup Variable	Type	R/W <sup>a</sup>	Use
274	mechanics-dielectric-scale-factor	float1, float2, float3, float4 <sup>b</sup> (≥1 . . ≤1000)	R,W	The multiplier for dielectric permittivity (see "Force field Parameters Dialog Box" on page 352)
275	mechanics-electrostatic-scale-factor	float1, float2, float3, float4 <sup>b</sup> (≥0 . . ≤1)	R,W	Scale factor for 1–4 electrostatic interactions
276	mechanics-van-der-waals-scale-factor	float1, float2, float3, float4 <sup>b</sup> (≥0 . . ≤1)	R,W	Scale factor for 1–4 van der Waals interactions
277	mechanics-mmp-electrostatics	enum (bondipoles, connectedbond-dipoles, atomiccharges, none)	R, W	Specifies the type of electrostatic interaction to use in MM+ calculations
278	cutoff-type	enum (none, switched, shifted)	R, W	The type of cutoff to apply to all molecular mechanics calculations
279	cutoff-inner-radius	float (≥0 . . < 1e10)	R, W	The distance (in Ångstroms) to begin a switched cutoff
280	cutoff-outer-radius	float (>0 . . < 1e10)	R, W	The distance (in Ångstroms) to end a switched cutoff, or the limit for a shifted cutoff
281	huckel-constant	float (>0 . . ≤10)	R, W	The proportionality factor to use in a Extended Hückel calculation. Usually 1.75
282	huckel-scaling-factor	float (>0 . . ≤100000)	R, W	Scale factor for Coulomb-Coulomb interactions between quantum and classical parts in a mixed model calculation
283	huckel-weighted	Boolean	R, W	A toggle for using a simple Hückel constant (yes), or a constant including a weighting factor for orbital diffuseness (no)

#	Setup Variable	Type	R/W <sup>a</sup>	Use
284	d-orbitals-on-second-row	Boolean	R, W	Yes to include d orbitals from the basis set for the second long row of atoms (Si to Cl), or no to ignore d orbitals (and shorten the calculation). For Extended Hückel only
285	scf-convergence	float (>0 . . ≤100)	R, W	Sets the convergence limit (the difference in electronic energy, in kcal/mol, after two consecutive iterations) for an SCF quantum mechanics calculation
286	max-iterations	int (1 . . 32767)	R, W	Sets the maximum number of SCF iterations allowed in a quantum mechanics calculation
287	accelerate-scf-convergence	Boolean	R, W	Yes uses the DIIS procedure to accelerate a calculation with a quantum-mechanical calculation
288	quantum-total-charge	int (-32767 . . ≤ 32767)	R, W	The total charge of a molecular system for a quantum mechanics calculation. For Extended Hückel or semi-empirical or <i>ab initio</i> SCF calculations, this variable must precede multiplicity (see the next variable)
289	multiplicity	int (1 . . 6)	R, W	The spin multiplicity (1 = singlet, 2 = doublet, 3 = triplet, 4 = quartet, etc.) of an electronic state to compute in a quantum mechanics calculation. For Extended Hückel or semi-empirical or <i>ab initio</i> SCF calculations, this variable must follow quantum-total-charge (see the previous variable)
290	uhf	Boolean	R, W	Yes sets a quantum mechanics calculation to Unrestricted Hartree Fock, or no for Restricted Hartree Fock

#	Setup Variable	Type	R/W <sup>a</sup>	Use
291	excited-state	Boolean	R, W	Yes directs an SCF quantum mechanics calculation for the next lowest electronic state of a given spin multiplicity, No for the lowest state of the given spin multiplicity
292	zindo-1-sigma-sigma	float ( $\geq 0 \dots \leq 10^{10}$ )	R, W	Weight (float) assigned to sigma-sigma overlap integral into ZINDO/1 calculations
293	zindo-1-pi-pi	float ( $\geq 0 \dots \leq 10^{10}$ )	R, W	Weight (float) assigned to pi-pi overlap integral into ZINDO/1 calculations
294	zindo-s-sigma-sigma	float ( $\geq 0 \dots \leq 10^{10}$ )	R, W	Weight (float) assigned to sigma-sigma overlap integral into ZINDO/S calculations
295	zindo-s-pi-pi	float ( $\geq 0 \dots \leq 10^{10}$ )	R, W	Weight (float) assigned to pi-pi overlap integral into ZINDO/S calculations
296	abinitio-mo-initial-guess	enum (core-hamiltonian, projected-huckel, projected-cndo, projected-indo)	R, W	Specify a method to produce an initial guess of the MO coefficients
297	abinitio-cutoff	float (>0)	R, W	Specify a cutoff threshold for two-electron integrals in an <i>ab initio</i> calculation
298	abinitio-buffer-size	int (>50 ... 32000)	R, W	Specify a buffer size to store the two-electron integrals in an <i>ab initio</i> calculation
299	abinitio-integral-format	enum (regular, raffanetti)	R, W	Specify a format to store the two-electron integrals in an <i>ab initio</i> calculation

#	Setup Variable	Type	R/W <sup>a</sup>	Use
300	abinitio-integral-path	string	R, W	Specify a full path, including disk drive and directory (e.g., C:\HYPER), to store the two-electron integrals in an <i>ab initio</i> calculation
301	abinitio-direct-scf	Boolean	R, W	Yes uses Direct SCF calculation to avoid using disk space to store the two-electron integrals in an <i>ab initio</i> calculation
302	abinitio-use-ghost-atoms	Boolean	R, W	Yes uses the "ghost-atoms" selection and treats them as ghost atoms
303	abinitio-scf-convergence	float (>0 ... ≤100)	R,W	Sets the convergence limit (the difference in electronic energy, in kcal/mol, between two successive iterations) for an <i>ab initio</i> calculation
304	abinitio-d-orbitals	Boolean	R, W	Yes uses 6 d orbitals (i.e., d <sub>xx</sub> , d <sub>yy</sub> , d <sub>zz</sub> , d <sub>xy</sub> , d <sub>xz</sub> , d <sub>yz</sub> ).  No uses 5 d orbitals (i.e., d <sub>0</sub> , d <sub>+1</sub> , d <sub>-1</sub> , d <sub>+2</sub> , d <sub>-2</sub> )
305	abinitio-calculate-gradient	Boolean	R, W	Yes computes the first derivatives of the total energy with respect to the coordinates of the atoms. This option applies to single-point calculations only
306	abinitio-mp2-correlation-energy	Boolean	R, W	Yes computes the second order Møller-Plesset perturbation energy. This option applies to single-point calculations only
307	abinitio-mp2-frozen-core	Boolean	R, W	Yes uses the frozen-core approximation in computing the MP2 correlation energy

#	Setup Variable	Type	R/W <sup>a</sup>	Use
308	configuration	int	R, W	Specifies the current selected line in a UV spectrum
309	configuration-interaction	enum (NoCI, SinglyExcited, or Microstate)	R, W	NoCI = no CI calculation requested; SinglyExcited = perform singly excited CI calculation; Microstate = perform a microstate CI calculation
310	ci-occupied-orbitals	int	R, W	The number of molecular orbitals from which electrons are excited to produce excited configuration
311	ci-unoccupied-orbitals	int	R, W	The number of unoccupied molecular orbitals to which electrons are excited to produce excited configurations
312	ci-excitation-energy	float( $\geq 0.. \leq 1e5$ )	R, W	Maximum MO orbital energy (in eV) between the virtual and occupied orbitals which form an active space in a singly excited CI calculation. (This option is only available for singly excited CI, and only applies when the ci-criterion is set to energy)
313	ci-criterion	enum (energy, orbital)	R, W	Use energy criterion or orbital criterion
314	parameter-set-changed	Boolean	R, W	Toggles the state of the backend parameters
315	backend-active	Boolean	R	Yes if calculations are in progress (at backend)
316	backend-communications	enum (remote, local)	R, W	Location of chemical calculations (local: current machine; remote: backend-host-name)
317	backend-host-name	string	R, W	Name of ethernet host to run back-end jobs

#	Setup Variable	Type	R/W <sup>a</sup>	Use
318	backend-user-id	string	R, W	Specifies the user's id
319	backend-user-password	string	R, W	Specifies the password to access the remote computer using the specified user id
320	backend-process-count	int	R, W	Specifies the number of processes running on the remote computer. The current version of HyperChem ignores this option and always uses 1.
321	use-electric-field	Boolean	R, W	Specifies whether to apply an electric field to workspace.
322	use-magnetic-field	Boolean	R, W	Specifies whether to apply a magnetic field to workspace.
323	field-direction	integer (1...3)	R, W	Direction (X,Y or Z) of the static electric field applied to the system.
324	field-strength	float (-1000...1000)	R, W	Strength (a.u.) of the static electric field applied to the system.
325	magnetic-field-direction	integer (1 ... 3)	R, W	Direction (X,Y or Z) of the static magnetic field applied to the system.
326	magnetic-field-strength	float (-1000...1000)	R, W	Strength (a.u) of the static magnetic field applied to system.
327	polar-field-strength	float in range (-1000 ... 1000)	R, W	Electric field strength for polarizability calculations.
328	polarizability-calculated	Boolean	R, W	Calculate ab initio or semi-empirical polarizability polarizability.
329	auxilliary-basis	int (1 .. 3)	R, W	1=A1, 2=A2, 3=P1

#	Setup Variable	Type	R/W <sup>a</sup>	Use
330	correlation-functional	enum(None, Perdew86, VWN, LYP, PZ81, PW91, PBE96, HCTH98)	R, W	Which correlation to use in DFT calculations.
331	exchange-functional	enum(None, Hartree-Fock, Slater, Becke88, PW91, PBE96, HCTH98, B3-LYP, B3-PW91,	R, W	Which exchange functional to use in DFT calculations.
332	read-hcth-coefficients	Boolean	R,W	Whether or not to use the standard coefficients or to read file hcth.coef containing five rows of three coefficients $C_{\chi}$ , $C_{\sigma\sigma}$ , $C_{\alpha\beta}$
333	small-density-cutoff	float in range (0 .. 0.001)	R, W	Value of density used as cutoff value in DFT calculations.
334	state-to-optimize	enum(SCF, MP2, CI)	R, W	Which energy to use for conjugate directions.
335	tndo-version	int	R, W	1, 2, etc. as versions of TNDO
336	use-auxilliary-basis	Boolean	R, W	As opposed to numerical integration.
337	use-mullikan-approximation-to-density	Boolean	R, W	Speed things up in DFT.
338	use-small-density-cutoff	Boolean	R, W	Minimize points in numerical integration.

a. R = read only. R,W = read and write.

b. enum1/float1 for MM+, enum2/float2 for AMBER, enum3/float3 for BIO+, and enum4/float4 for OPLS. This release of HyperChem does not use the MM+ values.



#	Setup Command	Arguments	Use
339	restraint	string, float1 (>0 . . >10 <sup>6</sup> for stretch, >0 . . <180 for bend, ≥-180 . . ≤180 for torsion) or computed, float2 (≥0 . . ≤10 <sup>6</sup> ) or default	Adds a restraint (string = selection-name, float1 = length or angle, float2 = force constant) to molecular mechanics or quantum mechanics calculations. Computed and default are the values that HyperChem supplies
340	restraint-tether	string, enum or float*3 (≥0 . . ≤10 <sup>6</sup> ), float (>0 . . >10 <sup>6</sup> )	Tethers a selection to a desired position in space. String refers to a named selection containing a single atom. Enum is either COMPUTED to use the current value for the position of the selection, or POINT to tether the selection to the coordinates of a selection saved as the named selection POINT. A point in space can be used as a tether point by replacing this enum with the x, y, and z coordinates of a point in space (float * 3). The last argument is the force constant
341	use-restraint	string, Boolean	Controls whether a restraint associated with a named selection is used; requires that you specify the name of a selection and "yes" or "no"
342	use-no-restraints	none	Ignores all restraints
343	use-parameter-set	string	Chooses a parameter set (string) for the current molecular mechanics method. Parameter set must be a CustomName defined in Registry or in chem.ini
344	solvate-system	none	Adds a periodic box and water molecules to a molecular system, using the last default settings
345	solvate-system-in-this-box	float1, float2, float3	Adds a periodic box—with dimensions x, y, and z Ångstroms—and water molecules to a molecular system

#	Setup Command	Arguments	Use
346	set-velocity	enum (POINT, LINE or OTHER), float, float, float (if OTHER specified), float (required)	Sets the velocity for the molecular system or selected atoms. The last argument is the velocity (Ångstrom/ps) and the direction is denoted by POINT (towards the named selection POINT), LINE (parallel to the named selection LINE) or OTHER (with three floats specifying the velocity vector).
347	assign-basisset	enum (STO-3G, STO-3G*, 3-21G*, ...)	Assign a basis set to the selected atom(s) or to all the atoms in <i>ab initio</i> calculations

#	Setup Menu Activator	Use	Dialog Box
348	menu-setup-molecular-mechanics	Chooses a molecular mechanics force field, options, and constraints	Molecular Mechanics Force Field
349	menu-setup-semi-empirical	Chooses a semi-empirical method	Semi-empirical Method
350	menu-setup-ab-initio	Chooses an <i>ab initio</i> method	Ab Initio Method
351	menu-setup-network	Chooses backend calculations running on the local computer or a remote computer	Network
352	menu-setup-periodic-box	Provides a water environment and periodic boundary conditions for a molecular system	Periodic Box Options
353	menu-setup-restraints	Adds computational restraints to named selections	Restraint Forces
354	menu-setup-set-velocity	Sets the velocity vectors for the selected atoms	Set Velocity
355	menu-setup-select-parameter-set	Chooses a new parameter file for molecular mechanics calculations	Force Field Parameters

#	Setup Menu Activator	Use	Dialog Box
356	menu-setup-compile-parameter-file	Converts a text parameter file to a binary file	—
357	menu-setup-reaction-map	Map reactants and products in preparation for a transition state search	Reaction Map

## Compute Menu

#	Compute Variable	Type	R/W <sup>a</sup>	Use
358	atom-spin-density-at-nucleus	array of float	R, W	The electron density of nucleus of atom iat in molecule imol.
359	atom-spin-population	array of float	R, W	The spin density of iat in molecule imol.
360	do-qm-calculation	Boolean	R, W	Yes to compute the wave function for a single point quantum mechanics calculation
361	do-qm-graph	Boolean	R, W	Yes graphs a 2D contour plot of the current quantum mechanics property, and sets do-qm-isosurface to false
362	do-qm-isosurface	Boolean	R, W	Yes plots the isosurface of the quantum mechanical property and sets do-qm-graph to false
363	isosurface-map-function	Boolean	R, W	Yes plots the mapped function isosurface of the quantum mechanical property
364	isosurface-grid-step-size	real (> 0)	R, W	Distance (in Ångstroms) between data points in the grid. Resets the x,y,z nodes
365	isosurface-x-nodes	int (1 < int ≤ 128)	R, W	Number of data grid points to accumulate in the X direction
366	isosurface-y-nodes	int (1 < int ≤ 128)	R, W	Number of data grid points to accumulate in the Y direction
367	isosurface-z-nodes	int (1 < int ≤ 128)	R, W	Number of data grid points to accumulate in the Z direction
368	isosurface-x-min	float (-1e+010 . . . 1e+010)	R, W	Smallest x coordinate of the grid data

#	Compute Variable	Type	R/W <sup>a</sup>	Use
369	isosurface-y-min	float (-1e+010 . . 1e+010)	R, W	Smallest y coordinate of the grid data
370	isosurface-z-min	float (-1e+010 . . 1e+010)	R, W	Smallest z coordinate of the grid data
371	isosurface-map-function-display-legend	Boolean	R, W	Yes to display a legend for the mapped function isosurface
372	isosurface-map-function-range	float1, float2 (If neither of these is 0, float2 = -1 * float1)	R, W	Minimum and maximum values for the display of a mapped function isosurface.
373	graph-data-type	enum (electrostatic, spin-density, charge-density, orbital, orbital-squared)	R, W	Type of data to plot with do-qm-graph or do-qm-isosurface
374	graph-data-row	int, float list	R	The values of the <i>i</i> <sup>th</sup> row of the data graph
375	graph-orbital-selection-type	enum (lumo-plus, homo-minus, orbital-number)	R, W	The procedure for indicating which orbital to plot, either relative to an HOMO/LUMO orbital or an absolute number
376	graph-beta	Boolean	R, W	Yes to plot beta orbital; no to plot alpha orbitals in an Unrestricted Hartree Fock calculation
377	graph-horizontal-grid-size	int (1 . . 16384)	R, W	Number of data grid points for plotting 2D contours in the horizontal direction
378	graph-vertical-grid-size	int (1 . . 16384)	R, W	Number of data grid points for plotting 2D contours in the vertical direction

#	Compute Variable	Type	R/W <sup>a</sup>	Use
379	graph-orbital-offset	int (0 . . 32767)	R, W	The number to use as an offset in describing which orbital to plot. The meaning depends on graph-orbital-selection-type. For orbital-number, it starts at 1 and indicates the orbital number. For homo-minus, it starts at 0 and indicates offset down from HOMO orbital. For lumo-plus, it starts at 0 and indicates offset up from LUMO orbital
380	basisset-count	int	R	Number of coefficients required to describe a molecular orbital
381	graph-contour-levels	int (1 . . 32767)	R, W	The number of contour lines in a plot
382	graph-plane-offset	float ( $\geq -10^{10}$ . . $\leq 10^{10}$ )	R, W	Offset along the viewer's z axis of the plane containing contour lines. Without an offset, contours pass through the center of the selection
383	graph-contour-start	float ( $\geq -10^{10}$ . . $\leq 10^{10}$ )	R, W	Value for the first contour line, if graph-contour-start-other is yes
384	graph-contour-increment	float ( $\geq -10^{10}$ . . $\leq 10^{10}$ )	R, W	Increment for contour lines, if graph-contour-increment-other is yes
385	graph-contour-start-other	Boolean	R,W	No to use default contouring; yes to use value from graph-contour-start
386	graph-contour-increment-other	Boolean	R,W	No to use default contouring; yes to use value for graph-contour-increment

#	Compute Variable	Type	R/W <sup>a</sup>	Use
387	optim-algorithm	enum (steepestdescent, fletcherreeves, polakribiere, newtonraphson, eigenvectorfollow, conjugate directions)	R, W	The algorithm to use for geometry optimization
388	optim-convergence	float (>0 . . ≤100)	R, W	For geometry optimization, the RMS gradient to use as a convergence criterion
389	optim-max-cycles	int (1 . . 32767)	R, W	Maximum number of optimization steps
390	state-to-optimize	enum (SCF, MP2, CI)	R, W	What to optimize when using conjugate Directions Optimizer. SCF is same as other optimizers. MP2 is ground state energy including MP2 correlation energy. CI is energy of Configuration Interaction state.
391	ci-state-to-optimize	int (0 . . 100)	R, W	When using Conjugate Optimizer to optimize CI states, this is the CI state to optimize. Lowest state (ground state) is 0, next state is 1, next is 2, etc.
392	periodic-boundaries	Boolean	R, W	Yes to use periodic boundary conditions; no to turn off these conditions. Set up periodic conditions with the Setup variables solvate-system-in-this-box or solvate-system
393	screen-refresh-period	int (1 . . 32767)	R, W	Frequency for updating the display during a molecular dynamics calculation, in units of data steps.

#	Compute Variable	Type	R/W <sup>a</sup>	Use
394	optim-converged	Boolean	R	Yes means geometry optimization has converged
395	dynamics-heat-time	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Time to change from dynamics-starting-temp to dynamics-simulation-temp
396	dynamics-run-time	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Total time at dynamics-simulation-temp
397	dynamics-cool-time	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Time to change from dynamics-simulation-temp to dynamics-final-temp
398	dynamics-time-step	float ( $> 0. \dots \leq 10^{10}$ )	R, W	Step size, in picoseconds, for the dynamics calculation
399	monte-carlo-heat-steps	float ( $0 < \dots \leq 10^{10}$ )	R, W	Number of steps to change from dynamics-starting-temp to dynamics-simulation-temp
400	monte-carlo-run-steps	float ( $0 < \dots \leq 10^{10}$ )	R, W	Number of steps at dynamics-simulation-temp
401	monte-carlo-cool-steps	float ( $0 < \dots \leq 10^{10}$ )	R, W	Number of steps to change from dynamics-simulation-temp to dynamics-final-temp
402	monte-carlo-max-delta	float ( $0 < \dots \leq 10^{10}$ )	R, W	Maximum allowed size of the displacement step in Ångströms.
403	dynamics-starting-temp	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Starting temperature for a dynamics simulation
404	dynamics-simulation-temp	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Run-time temperature for a dynamics simulation
405	dynamics-final-temp	float ( $\geq 0. \dots \leq 10^{10}$ )	R, W	Final temperature for a dynamics simulation



#	Compute Variable	Type	R/W <sup>a</sup>	Use
406	dynamics-temp-step	float ( $>0. \leq 10^{10}$ )	R, W	Temperature step size (degrees K) for heating and cooling (annealing)
407	dynamics-restart	Boolean	R, W	Yes to restart a dynamics calculation using velocities stored in a file; no to use new equilibrated velocities
408	dynamics-playback	enum (record, playback, none)	R, W	Record = store dynamics simulation in a snapshots file; playback = use dynamics simulation stored in a snapshots file; none = calculate a new trajectory (without snapshots or playback of snapshots)
409	dynamics-playback-start	int (0 . . 32767)	R, W	The snapshot number to begin the playback
410	dynamics-playback-end	int (0 . . 32767)	R, W	The snapshot number to end the playback
411	dynamics-playback-period	int (0 . . 32767)	R, W	The length of the playback period in numbers of snapshots
412	dynamics-constant-temp	Boolean	R, W	Yes to apply temperature regulation during a dynamics simulation
413	dynamics-bath-relaxation-time	float ( $>0. \leq 10^{10}$ )	R, W	The time constant in picoseconds for constant temperature regulation
414	dynamics-friction-coefficient	float (0 . . 1000000)	R, W	Friction coefficient for Langevin dynamics
415	dynamics-collection-period	int (1 . . 32767)	R, W	The period for collecting data from the molecular dynamics calculation in time steps

#	Compute Variable	Type	R/W <sup>a</sup>	Use
416	dynamics-snapshot-filename	filename	R, W	The name of a snapshot file, with the extension <i>.snp</i>
417	dynamics-snapshot-period	int (1 . . 32767)	R, W	The period between snapshots, in units of data steps
418	dynamics-average-period	int (1 . . 32767)	R, W	The period, in units of data steps, for storing data from a molecular dynamics run in a CSV file
419	dynamics-seed	int (-32768 . . 32767)	R, W	Seed for dynamics initialization random number generator
420	mechanics-print-level	int (0 . . 9)	R,W	Sets the level of information that appears in an active log file for a molecular mechanics calculation. 0 = no results, 9 = maximum information
421	quantum-print-level	int (0 . . 9)	R,W	Sets the level of information that appears in an active log file for a semi-empirical or <i>ab initio</i> calculation. 0 = no results, 9 = maximum information
422	warning-type	enum (none, log, message)	R/W	Sets the use of warning messages from backend calculations. None = discard all warning messages; log = place warning messages in the log file if one is active; message = display the warning on screen
423	show-vibrational-vectors	Boolean	R, W	Yes to display vibrational mode displacement vectors
424	animate-vibrations	Boolean	R, W	Controls display of animated vibrations

#	Compute Variable	Type	R/W <sup>a</sup>	Use
425	ir-animate-amplitude	float ( $\geq 1 \dots \leq 10^{10}$ )	R, W	Sets distances in Ångstroms to move the fastest atom during vibrational animations
426	ir-animate-steps	int (3 .. 32767)	R, W	Number of steps to use in each cycle while animating vibrations
427	ir-animate-cycles	int (1 .. 32767)	R, W	Controls how many cycles an animation of a vibrational mode will show before terminating. <b>Caution:</b> If this variable is not set before the animate-vibrations command, the animation will run on without continuing to further script commands until the Cancel item is selected on the menu
428	vibrational-mode	int (1 .. (3 * natoms - 6))	R, W	The index of the current vibrational mode
429	ir-band-count	int. (0 .. 2685)	R, W	The number of IR bands. Setting this variable discards existing IR data
430	ir-frequency	vector of floats	R, W	When you give the index (int) of an IR band, this variable specifies its frequency
431	ir-intensity	vector of floats	R, W	When you give the index (int) of an IR band, this variable specifies its intensity
432	ir-intensity-components	vector of floats	R, W	When you give the index (int) of an IR band, this variable specifies the x, y, and z components of its intensity

#	Compute Variable	Type	R/W <sup>a</sup>	Use
433	ir-normal-mode	vector of floats	R, W	When you give the index (int) of an IR band, this variable specifies for each atom the x, y, and z displacements for the normal mode
434	uv-band-count	int (0 . . 32766)	R, W	The number of UV bands. Changing this value discards any existing UV data
435	uv-energy	vector of floats	R, W	When you give the index (int) of an UV band, this variable specifies the energy of the transition
436	uv-spin	vector of floats	R, W	When you give the index (int) of an UV band, this variable specifies the spin multiplicity of the state
437	uv-total-dipole	vector of floats	R, W	The total dipole moment of excited state (int)
438	uv-dipole-components	vector of floats	R, W	The x, y, and z components of excited state (int)
439	uv-transition-dipole	vector of floats	R, W	The transition dipole between the ground state and excited state (int)
440	uv-oscillator-strength	vector of floats	R, W	The oscillator strength for excited state (int)
441	scf-orbital-energy	vector of floats	R	Gives the energy of orbital (int), in eV. For UHF, the set of beta orbitals are numbered following the alpha

#	Compute Variable	Type	R/W <sup>a</sup>	Use
442	alpha-scf-eigenvector	vector of floats	R	When you give the index of an SCF eigenvector, this variable gives the coefficients of the corresponding molecular orbital. In an RHF calculation this variable is used to specify the indices of the SCF eigenvectors. In a UHF calculation this variable is only used to specify the indices of the alpha SCF eigenvectors
443	beta-scf-eigenvector	vector of floats	R	When you give the index of an SCF eigenvector, this variable gives the coefficients of the corresponding molecular orbital. Available only after UHF computations
444	alpha-orbital-occupancy	vector of floats	R/W	Number of electrons in the i-th MO
445	beta-orbital-occupancy	vector of floats	R/W	Number of electrons in the i-th MO
446	dipole-moment	float	R/W	Gives the dipole moment (Debyes) from a quantum-mechanical or MM+ calculation
447	dipole-moment-components	float, float, float	R	Gives dipole moment (Debyes) x, y, and z components from a quantum-mechanical or MM+ calculation
448	scf-atom-energy	float	R/W	Gives the isolated atom energy (kcal/mol) from a semi-empirical calculation
449	scf-binding-energy	float	R/W	Gives the energy, relative to isolated atoms (kcal/mol), from a quantum-mechanical calculation. This is the value reported on the status line following a quantum-mechanical calculation

#	Compute Variable	Type	R/W <sup>a</sup>	Use
450	scf-core-energy	float	R/W	Gives the core-core interaction energy (kcal/mol) from a quantum-mechanical calculation
451	scf-electronic-energy	float	R/W	Gives the electronic energy (kcal/mol) from a quantum-mechanical calculation
452	heat-of-formation	float	R/W	Gives the heat of formation (kcal/mol) from a Semi-empirical calculation
453	annotation-filled	Boolean	R/W	Fills the annotation layer.
454	annotation-layer-hidden	Boolean	R/W	Hides the annotation layer.
455	annotation-layer-in-front	Boolean	R/W	Places the annotation layer in front.

a. R = read only. R,W = read and write.

#	Compute Command	Arguments	Use
456	do-single-point	none	Performs a single point calculation on a molecular system with the current calculation method
457	do-optimization	none	Performs a structure optimization on a molecular system with the current calculation method
458	do-vibrational-analysis	none	Performs a vibrational analysis on a molecular system with the current calculation method
459	do-molecular-dynamics	none	Performs a molecular dynamics computation on a molecular system with the current calculation method

#	Compute Command	Arguments	Use
460	do-langevin-dynamics	none	Performs a Langevin dynamics computation on a molecular system with the current calculation method
461	do-monte-carlo	none	Performs a Monte Carlo computation on a molecular system with the current calculation method
462	append-dynamics-average	enum (none, EKIN, EPOT, ETOT, TEMP, D EKIN, D EPOT, D ETOT, D TEMP, <i>selection name</i> , <i>D selection name</i> )	Names one selection (containing 2 to 4 atoms) or energetic quantity to average or graph during a molecular dynamics simulation. The prefix D indicates the deviation of the average. One command adds one value to the list of values to average, the other to the list to graph. Use all uppercase as shown here. Use double quotes around arguments containing spaces; for example "D EKIN"
463	append-dynamics-graph		
464	isosurface-mesh-quality	enum (coarse, medium, fine)	Sets variables for isosurface grid to default values

#	Compute Menu Activator	Use	Dialog Box
465	menu-compute-single-point	Calculates the energy and gradient for the current conformation	—
466	menu-compute-geometry-optimization	Finds an optimal conformation for the molecular system	Molecular Mechanics or Semi-empirical or Ab Initio Optimization
467	menu-compute-molecular-dynamics	Runs a molecular dynamics simulation	Molecular Dynamics Options
468	menu-compute-langevin-dynamics	Perform a Langevin Dynamics simulation	Langevin Dynamics Options

#	Compute Menu Activator	Use	Dialog Box
469	menu-compute-monte-carlo	Perform a Monte Carlo simulation	Monte Carlo Options
470	menu-compute-vibrations	Performs a vibrational analysis computation	—
471	menu-compute-transition-state	Perform a transition state computation	Transition State Options
472	menu-compute-plot-molecular-properties	Draws contour plots and isosurfaces of quantum mechanics results	Molecular Properties
473	menu-compute-orbitals	Selects an orbital to graph	Orbitals
474	menu-compute-vibrational-spectrum	Displays vibrational transitions and normal modes	Vibrational Spectrum
475	menu-compute-electronic-spectrum	Displays UV visible spectrum	Electronic Spectrum

## Script Menu

#	Script Variable	Type	R/W <sup>a</sup>	Use
476	notify-with-text	Boolean	R, W	For DDE channels, are notifications to be text (instead of binary)
477	query-response-has-tag	Boolean	R, W	Yes means response to a query is a value x plus a tag (<variable> = x). No means response is just the value x. Tag is a variable name (HSV)



#	Script Variable	Type	R/W <sup>a</sup>	Use
478	ignore-script-errors	Boolean	R, W	Yes to proceed with a script when an error occurs; no to stop HyperChem and show a message box on screen. <i>Caution:</i> Ignoring script errors can produce incorrect results. If you choose this option, use errors-are-omsgs and omsgs-to-file to save error messages for later review
479	script-menu-caption	vector of strings	R, W	Captions for menu buttons
480	script-menu-checked	vector of Booleans	R, W	If checked
481	script-menu-command	vector of strings	R, W	Commands for menu buttons
482	script-menu-enabled	vector of Booleans	R, W	If grayed
483	script-menu-help-file	vector of strings	R, W	Help files for menu items
484	script-menu-help-id	vector of integers	R, W	Context ids for help on buttons
485	script-menu-in-use	vector of Booleans	R, W	Is somebody claiming this menu item?
486	script-menu-message	vector of strings	R, W	Status messages displayed when script menu items clicked with mouse
487	script-refs-in-errors	Boolean	R, W	Whether to include script file line numbers in errors

#	Script Variable	Type	R/W <sup>a</sup>	Use
488	inhibit- redisplay	Boolean	R, W	Determines whether redisplay of the system is inhibited. <b>Caution:</b> If inhibit-redisplay is not reset to No, the HyperChem workspace will not be updated, even after the termination of the script where it is set to Yes
489	global-inhibit- redisplay	Boolean	R	Whether redisplay of the system is inhibited
490	status-message	string	R, W	Displays a message (string) on the status line. When queried, returns the last status message displayed.
491	omsg-file	filename	R, W	Name of file to append OMSGs to

a. R = read only. R,W = read and write.

#	Script Command	Arguments	Use
492	pause-for	int (1 . . . 32767)	Pause the script for a number (int) of seconds
493	notify-on-update	string	Requests a response when there is a change in value for this variable name (string). Use only for scripts to return variable as OMSG; use WM_DDE_ADVISE for DDE communications
494	cancel-notify	string	Cancels notify-on-update for variable (string)
495	omsgs-to-file	filename	Deletes file contents and then sends omsg's to this file while the script is running. Do not use for DDE. See the previous and following commands
496	omsgs-not-to-file	none	Sends omsgs to an OMSG box on screen instead of recording it in a file (see the next command)

#	Script Command	Arguments	Use
497	append-omsgs-to-file	filename	Sends ommsg's to this file, adding them to previous messages. See the previous two commands
498	errors-are-omsgs	none	Treats error messages like ommsg's: sends them to a file or to an OMSG box on screen
499	errors-are-not-omsgs	none	Sends error messages to screen
500	warnings-are-omsgs	none	Treats warning messages like ommsg's: sends them to a file or to an OMSG box on screen
501	warnings-are-not-omsgs	none	Sends warning messages to screen
502	message	string	Treats string like an ommsg: sends it to a file or to an OMSG box on screen
503	error	string	Treats string like an ommsg: sends it to a file or to an Error box on screen
504	warning	string	Treats string like a warning message: sends it to a file or to a Warning box on screen
505	source-refs-in-errors	none	Controls presentation of filename, line number in error messages
506	no-source-refs-in-errors	none	Controls presentation of filename, line number in error messages
507	read-script	filename	Names a script file to read. Use this to start another script from a script
508	read-tcl-script	filename	Names a tcl script file to read.
509	query-value	string	HyperChem treats the value of a variable (string) like an ommsg
510	request	string	Displays string in a dialog box until you L-click on OK. Allows keyboard and mouse input while the script is executing. The script resumes when you L-click OK

#	Script Command	Arguments	Use
511	execute-client	filename, string1, string2	Runs a DDE application. Filename = .exe file for the application, string1 = DDE application (program) name, and string2 = DDE topic (a filename or other string that the client application recognizes). Must have three arguments, even if the last two are null (" ")
512	execute-hyperchem-client	filename	Runs a client application specified by filename that is a Windows executable application. This application should be able to reliably connect to an instance of HyperChem
513	change-user-menuitem	int, string1, filename	Changes the text and script associated with a menu item (Script One, etc.) on the Script menu. int = number of the Script menu item to place text in (1, 2, etc.), string1 = text to place in the menu item (characters limit depends on your display resolution), and filename = name of the script file associated with this menu item
514	exit-script	none	Ends the script. This is usually the last line in a script file, but is not required
515	compile-script-file	filename1, filename2	Compile a script ( <b>filename1</b> ), convert it to binary format, and write it to a file ( <b>filename2</b> )
516	read-binary-script	filename	Execute a compiled binary script file <b>filename</b>
517	execute-string	string	Execute the string variable as a script

#	Script Menu Activator	Use	Dialog Box
518	menu-script-open-script	Opens a script file and runs the script	Open Script
519	menu-script-one . . . menu-script-ten	Selects script program one . . . ten	—

## Cancel

#	Cancel Activator	Use	Dialog Box
520	menu-cancel	Ends a calculation and returns control to user	—

## Data Variables

#	Data Variable	Type	R/W	Use
521	atom-name	int1, int2, string <sup>a</sup>	R, W	If you give int1 = atom number and int2 = molecule number, reports string = atom name
522	atomic-number	int1, int2, int3 <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports atomic number (int3)
523	atomic-symbol	int1, int2, string <sup>a</sup>	R	If you give atom number (int1) and molecule number (int2), reports atomic symbol (string)
524	atom-type	int1, int2, string <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports atom type (string)
525	atom-charge	int1, int2, float <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports atomic charge (float)
526	formal-charge	int1, int2, int3 <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports formal charge (int)
527	atom-mass	int1, int2, float <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports atomic mass (float)
528	atom-basisset	int1, int2, string <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports basis set for this atom

#	Data Variable	Type	R/W	Use
529	atom-extra-basisset	int1, int2, string, float <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports the shell type (string) of an extra basis function and the exponent value (float) of this extra basis function for this atom
530	atom-color	int1, int2, string <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports the atom's current color (string)
531	atom-label-text	int1, int2, string <sup>a</sup>	R	If you give atom number (int1) and molecule number (int2), reports the atom's label text (string)
532	residue-label-text	int1, int2, string <sup>a</sup>	R	If you give residue number (int1) and molecule number (int2), reports the residue's label text (string)
533	is-ring-atom	int1, int2, Boolean <sup>a</sup>	R	If you give atom number (int1) and molecule number (int2), reports true if the atom is in a ring (Boolean)
534	is-extended-hydrogen	int1, int2, Boolean <sup>a</sup>	R	If you give atom number (int1) and molecule number (int2), reports if the atom is an extended hydrogen (Boolean)
535	coordinates	int1, int2, float1, float2, float3 <sup>a</sup>	R, W	If you give atom number (int1) and molecule number (int2), reports the x, y, and z coordinates (float1, 2, and 3)
536	neighbors	int1, int2, int3, int4, enum (s, d, t, h, a) <sup>a</sup>	R	If you give atom number (int1) and molecule number (int2), reports the atoms bonded to this atom through either covalent or hydrogen bonds: int3 = number of neighbor atom, int4 = molecule containing neighbor atom and bond type (enum, s = single, d = double, t = triple, a = aromatic, h = hydrogen). Repeats "int3, int4, enum" for each neighbor atom

#	Data Variable	Type	R/W	Use
537	residue-charge	(i,j) int, int, float	R	Net charge on residue i in molecule j
538	residue-name	(i,j) int, int, string	R	Name of residue i in molecule j
539	residue-coordinates	(i,j) int, int, f, f, f	R	Coordinates of center of mass of residue in i molecule j
540	molecule-count	int	R	The number of molecules (int) in the molecular system
541	atom-count	int1, int2	R	If you give the molecule number (int1), reports the number of atoms (int2) in this molecule
542	residue-count	int1, int2	R	If you give the molecule number (int1), reports the number of residues (int2) in this molecule
543	orbital-count	int	R	Show the number of molecular orbitals used in quantum mechanical calculations
544	coordination	int1, int2, int3	R,W	If you give atom number (int1) and molecule number (int2), reports (int3) the number (from 1 to 12) of bonded neighbors
545	chirality	int1, int2, enum	R, W	If you give atom number (int1) and molecule number (int2), reports (enum) one of A, R, S, or ?, giving the chirality ("A" achiral atom, "R" rectus or right chirality, "S" sinister or left chirality and "?" for 5 or 6 coordinated systems)
546	dynamics-info-total-energy	float	R	Reports the sum of the potential and kinetic energies for a molecular dynamics simulation or its playback

#	Data Variable	Type	R/W	Use
547	dynamics-info-potential-energy	float	R	Reports the potential energy for a molecular dynamics simulation or its playback
548	dynamics-info-kinetic-energy	float	R	Reports the kinetic energy for a molecular dynamics simulation or its playback
549	dynamics-info-temperature	float	R	Reports the temperature value corresponding to the current velocities in a molecular dynamics simulation or its playback.
550	dynamics-info-elapsed-time	float	R	Reports the elapsed running time, in picoseconds, of a molecular dynamics simulation or its playback
551	dynamics-info-last-update	Boolean	R	Yes if the last step of a molecular dynamics simulation or its playback has occurred; no if the simulation is still in progress
552	monte-carlo-info-acceptance-ratio	float (0 . . 1)	R	Computation result from Monte Carlo run
553	velocities	int1, int2, float1, float2, float3	R, W	If you give int1 = atom number and int2 = molecule number, reports float1, 2, and 3, the $v_x$ , $v_y$ , and $v_z$ values (in Ångstroms/picosecond) for this atom
554	rms-gradient	float	R	The RMS gradient, in kcal/Å, derived from a single point or geometry optimization calculation
555	total-energy	float	R/W	The total energy of a molecular system, in kcal/mol (energy on a potential energy surface), either from a molecular mechanics or quantum mechanics calculation



#	Data Variable	Type	R/W	Use
556	stretch-energy	float	R	The bond stretching contribution to the molecular mechanics total energy
557	bend-energy	float	R	The bond angle bending contribution to the molecular mechanics total energy
558	torsion-energy	float	R	Reports the bond torsion contribution to the molecular mechanics total energy
559	nonbond-energy	float	R	Reports the nonbonded interaction contribution to the molecular mechanics total energy
560	hbond-energy	float	R	Reports the hydrogen bonding contribution to the molecular mechanics total energy
561	estatic-energy	float	R	Reports the electrostatic interaction contribution to the molecular mechanics total energy. For MM+, this reports the stretch-bend energy
562	mp2-energy	float	R/W	The MP2 correlation energy, in kcal/mol, in <i>ab initio</i> calculations
563	molecule-name	vector of string	R	The name of the i-th molecule.

a. For a read from HyperChem, use int1 and int2. To write, use all arguments.

## Help Menu

#	Help Variable	Type	R/W	Use
564	version	string	R	The version of HyperChem
565	serial-number	string	R	The serial number of the copy of HyperChem in use

#	Help Command	Arguments	Use
566	help	string	Displays help for the topic string

#	Help Menu Activator	Use	Dialog Box
567	menu-help-index	Displays help index	HyperChem Help
568	menu-help-keyboard	Displays keyboard information	HyperChem Help
569	menu-help-commands	Displays instructions about using HyperChem commands	HyperChem Help
570	menu-help-tools	Displays instructions about using HyperChem tools	HyperChem Help
571	menu-help-scripts-and-dde	Displays instructions on how to use HyperChem script messages	HyperChem Help
572	menu-help-on-context	Displays context sensitive help cursor	HyperChem Help
573	menu-help-glossary	Displays definitions of HyperChem terms	HyperChem Help
574	menu-help-using-help	Displays instructions about using Help	HyperChem Help
575	menu-help-about-hyperchem	Displays information about HyperChem	HyperChem Help

## Miscellaneous Commands

	Command	Arguments	Use
576	write-atom-map	filename	Writes a mapping of backend atom numbers to HyperChem
577	variable-changed	variable name	Causes HyperChem to operate as if the provided variable changed in value, notifying all "watchers" of its current value.
578	declare-integer	string	Declare a new integer variable
579	declare-string	string	Declare a new string variable
580	push	string	Push copy of current value onto stack
581	pop-value	string	Restore pushed value
582	pop-no-value	string	Pop stack, don't restore value
583	toggle	string	Invert value of Boolean variable named in <b>string</b>
584	print-variable-list	filename	Write a summary of the state variables to the <b>filename</b>
585	factory-settings	none	Reset all the options of HyperChem to its out-of-the-box state
586	load-user-menu	filename	Load a customized menu from this filename
587	switch-to-user-menu	none	Use a customized menu
588	load-default-menu	none	Use the default menus

## Miscellaneous Variables

#	Miscellaneous Variable	Type	R/W	Use
589	mechanics-data	special	R	<p>Contains the following information to support a back-end calculation:</p> <ul style="list-style-type: none"> <li>(a) type of calculation [single-point (0), geometry optimization (1), molecular dynamics (2), or vibrations (4)];</li> <li>(b) using restraints [true (1) or false (0)];</li> <li>(c) optimization algorithm [steepest descent (0), Fletcher-Reeves (1), Polak-Ribiere (2), or Block-diagonal Newton-Raphson (3)];</li> <li>(d) (reserved for future use);</li> <li>(e) maximum number of cycles of optimization;</li> <li>(f) desired RMS gradient in optimization;</li> <li>(g) (reserved for future use);</li> <li>(h) whether or not fix the non-selected atoms;</li> <li>(i) whether or not using periodic boundary conditions;</li> <li>(j) whether or not using constant temperature;</li> <li>(k) relaxation time;</li> <li>(l) (reserved for future use);</li> <li>(m) whether or not using restart;</li> <li>(n) random number seed;</li> <li>(o) x, y, z dimensions of the periodic box;</li> <li>(p) <math>\alpha</math>, <math>\beta</math>, <math>\gamma</math> angles of the periodic box (must all be <math>90^\circ</math> in this version);</li> <li>(q) starting temperature for heating time (<math>T_1</math>);</li> <li>(r) starting temperature for running time (<math>T_2</math>);</li> <li>(s) final temperature for cooling time (<math>T_3</math>);</li> <li>(t) time for the <math>T_1 \rightarrow T_2</math> temperature change;</li> <li>(u) time at temperature <math>T_2</math>;</li> </ul>

#	Miscellaneous Variable	Type	R/W	Use
	mechanics-data (continued)			(v) time for the $T_2 \rightarrow T_3$ temperature change; (w) size of the time step for integration; (x) size of temperature steps; (y) Langevin Dynamics friction coefficient (z, aa, bb, cc, dd, ee, ff) (reserved for future use); (gg) data collection period (in time steps); (hh) warning level [none (0), log file (1), or log file and message box (2)]
590	mechanics-info	special	R	Contains the following information to support a molecular-mechanics back-end calculation: (a-d) (reserved for future use); (e) whether or not to use a distance-dependent dielectric; (f) (reserved for future use); (g) electrostatic option for MM+ [bond dipoles (0) or atomic charges (1)]; (h) inverse of dielectric constant; (i-j) (reserved for future use); (k) scale for 1-4 van der Waals; (l) scale for 1-4 electrostatic; (m) type of cutoff; (n) inside cutoff radius; (o) outside cutoff radius; (p-u) reserved for future use; (v) level of detail for printing to log file
591	atom-info	vectors of array	R	Atom information that supports back-end calculations. Each atom corresponds to one vector that contains the following items: index, atomic number, atom type, atom flag, selection flag, ring flag, coordination, bonded atom indices (up to 12), atom charge, x coordinate, y coordinate, z coordinate, and atom mass

#	Miscellaneous Variable	Type	R/W	Use
592	cancel-menu	Boolean	R, W	Whether the 'Cancel' menu is up (other items grayed out) instead of the normal menu
593	info-variable-target	string	R, W	Variable for which info is required
594	one-line-arrays	Boolean	R, W	Whether to emit arrays all on one line.
595	info-access	string	R	Access (R, W, RW) for info-variable-target
596	info-enum-id-of	string	R	Binary id of info-enum-target value for info-variable-target
597	info-enum-list	string	R	If enumerated type, list of enumerated values for info-variable-target
598	info-factory-setting	string	R	Factory setting value for info-variable-target
599	info-id-of	int	R	Binary id of info-variable-target
600	info-type-of	int	R	Type of info-variable-target
601	info-type-of-element	int	R	If info-type-of is array or vector, type of elements
602	custom-title	string	R, W	Appends string to title in program window title bar
603	window-height	int	R, W	Height of program window in pixels
604	window-width	int	R, W	Width of program window in pixels
605	workspace-height	int	R, W	Height of workspace in pixels
606	workspace-width	int	R, W	Width of the workspace of pixels.

#	Miscellaneous Variable	Type	R/W	Use
607	gradient-x	float (1e+010 ... 1e+010)	R, W	Molecular gradient in the x direction
608	gradient-y	float (1e+010 ... 1e+010)	R, W	Molecular gradient in the y direction
609	gradient-z	float (1e+010 ... 1e+010)	R, W	Molecular gradient in the z direction
610	gradients	array of float, float, float	R, W	The x, y, and z gradients of atom iat in molecule imol
611	module-name	string	R	Current name of module.

## Cross Reference of HyperChem Messages

The numbers that appear in the first column of each of the previous tables are used in the following alphabetic cross-reference. Use this cross-reference as an index to help you locate the message you want to use.

### Message Cross-Reference

This alphabetic cross-reference applies to the previous tables containing HyperChem message variables, commands, and menu activators. The numbers listed are the message numbers for the preceding pages; they are **not** page numbers.

You can use the on-line help Search function to find any message by searching for part of the message. For example, you can display a list of all messages that include a word (such as “hin” or “atom”), and then display the definition of the specific message that you want.

**Caution:** Use this cross-reference only for finding messages in the tables. The syntax of a cross-reference might not match the original script message.



## Numerics

3D surface display toggle ..... #230  
 3D surface, display type ..... #229

## A

ab initio results, export ..... #21  
 ab initio results, import ..... #25  
 ab initio setup ..... #350  
 abinitio-buffer-size ..... #298  
 abinitio-calculate-gradient ..... #305  
 abinitio-cutoff ..... #297  
 abinitio-direct-scf ..... #301  
 abinitio-d-orbitals ..... #304  
 abinitio-integral-format ..... #299  
 abinitio-integral-path ..... #300  
 abinitio-mo-initial-guess ..... #296  
 abinitio-mp2-correlation-energy ..... #306  
 abinitio-mp2-frozen-core ..... #307  
 abinitio-scf-convergence ..... #303  
 abinitio-use-ghost-atoms ..... #302  
 about-hyperchem ..... #575  
 accelerate-scf-convergence ..... #287  
 acceptance ratio, Monte Carlo ..... #552  
 add hydrogens ..... #135  
 add-amino-acid ..... #255  
 add-nucleic-acid ..... #258  
 a-form, DNA ..... #259  
 algorithm, geometry optimization ..... #387  
 align molecule ..... #100  
 align viewer ..... #76, #99  
 align-molecule ..... #75  
 all atoms ..... #139  
 allow-ions ..... #106, #137  
 alpha helix ..... #256  
 alpha orbital ..... #376  
 alpha torsion angle, nucleic acid ..... #246  
 alpha-orbital-occupancy ..... #444  
 alpha-scf-eigenvector ..... #442  
 amino acid, add ..... #255, #263  
 amino-alpha-helix ..... #256  
 amino-beta-sheet ..... #257  
 amino-isomer ..... #244  
 amino-omega ..... #243

amino-phi ..... #241  
 amino-psi ..... #242  
 angle, bond ..... #84, #102  
 angle, constrain bond ..... #147  
 angle, constrain torsion ..... #148  
 animate, amplitude ..... #425  
 animate, number of cycles ..... #427  
 animate, number of steps ..... #426  
 animate-vibrations ..... #424  
 append-dynamics-average ..... #462  
 append-dynamics-graph ..... #463  
 append-omsgs-to-file ..... #497  
 array type ..... #601  
 arrays, format ..... #594  
 assign-basisset ..... #347  
 atom charge, set ..... #109  
 atom coordinates ..... #535  
 atom count, selected ..... #150  
 atom map, write ..... #576  
 ATOM records, PDB file ..... #9  
 atom type, set ..... #108, #142  
 atom types, calculate ..... #140  
 atom, create ..... #112  
 atom, cycle stereo ..... #131  
 atom, delete ..... #113  
 atom, deselect ..... #161  
 atom, select ..... #160, #168  
 atom, selected ..... #151  
 atom, stereo ..... #132  
 atom-basisset ..... #528  
 atom-charge ..... #525  
 atom-color ..... #530  
 atom-count ..... #541  
 atom-extra-basisset ..... #529  
 atomic motion, maximum, Monte Carlo ..... #402  
 atomic velocity, set ..... #346  
 atomic-number ..... #522  
 atomic-symbol ..... #523  
 atom-info ..... #591  
 atom-labels ..... #214  
 atom-label-text ..... #531  
 atom-mass ..... #527  
 atom-name ..... #521  
 atoms, all ..... #139  
 atoms, cut ..... #87  
 atoms, delete ..... #77  
 atoms, number selected ..... #150

atoms, paste ..... #90  
atoms, united ..... #138  
atom-spin-density-at-nucleus ..... #358  
atom-spin-population ..... #359  
atom-type ..... #524  
attached atoms ..... #536  
attachments, count ..... #544  
auxilliary-basis ..... #329  
averages period, dynamics ..... #418  
axes, show ..... #210, #236

## B

backbone, select ..... #180  
back-clip ..... #73  
back-end support ..... #589, #590, #591  
backend-active ..... #315  
backend-communications ..... #316  
backend-host-name ..... #317  
backend-process-count ..... #320  
backend-user-id ..... #318  
backend-user-password ..... #319  
background, printer page color ..... #38  
backwards, nucleic acid construction .. #253  
balls-highlighted ..... #187  
balls-radius-ratio ..... #188  
balls-shaded ..... #186  
band count, uv ..... #434  
band, ir count ..... #429  
basis set, assign ..... #347  
basis set, atom ..... #528  
basis set, extra atom ..... #529  
basisset-count ..... #380  
bath relaxation time, dynamics ..... #413  
bend-energy ..... #557  
beta orbital ..... #376  
beta sheet ..... #257  
beta torsion angle, nucleic acid ..... #247  
beta-orbital-occupancy ..... #445  
beta-scf-eigenvector ..... #443  
b-form, DNA ..... #260  
bitmap image ..... #40, #41  
black & white, image ..... #47  
bond angle, constrain .... #118, #122, #147  
bond angle, set ..... #84, #102  
bond color ..... #193

bond functions, ab initio ..... #302  
bond length, constrain .... #117, #121, #146  
bond length, set ..... #83, #101  
bond order ..... #114  
bond torsion ..... #103  
bond torsion, constrain ..... #123  
bond, bond-down constraint ..... #127  
bond, bond-up constraint ..... #126  
bond, create or delete ..... #114  
bond, cycle stereo ..... #130  
bond-color ..... #11  
bonded atoms ..... #536  
bond-spacing-display-ratio ..... #207  
Boolean variable, toggle ..... #583  
boundary conditions, periodic ..... #392  
buffer size, ab initio ..... #298  
build, model ..... #136  
builder-enforces-stereo ..... #107  
buttons, hide ..... #213

## C

calculate types ..... #140  
calculation-method ..... #269  
cancel ..... #520  
cancel-menu ..... #592  
cancel-notify ..... #494  
change-stereochem ..... #132  
change-user-menuitem ..... #513  
charge density ..... #472  
charge, atom ..... #109, #525  
charge, residue ..... #537  
charge, set ..... #109, #143  
chi torsion angle, nucleic acid ..... #252  
chirality ..... #545  
Cl ..... #309  
ci-criterion ..... #313  
ci-excitation-energy ..... #312  
ci-occupied-orbitals ..... #310  
ci-state-to-optimize ..... #391  
ci-unoccupied-orbitals ..... #311  
clear selection ..... #86  
clear workspace ..... #59  
client, execute ..... #511, #512  
clip z ..... #98  
clip, back ..... #73

clip, front	#72	contour graph levels	#381
clipboard	#45	contour graph, data type	#373
clipboard destination, image	#45	contour graph, offset	#382
clip-cursor	#29	contour graph, orbital selection	#375
clip-icon-step	#35	contour graph, start value	#383, #385
color, 24-bit	#41	contour increment	#386
color, atom	#530	contour plot	#472
color, bond	#11	convergence limit, optimization	#388
color, cylinders	#193	convergence report, optimization	#394
color, element	#220, #221, #240	convergence, ab initio SCF	#303
color, image	#47	convergence, accelerate	#287
color, negative orbitals	#14	convergence, SCF	#285
color, positive orbitals	#13	cool steps, monte carlo	#401
color, selection	#12, #239	cool time, dynamics	#397
color, window	#10	coordinate, residue	#539
color-selection	#219	coordinates	#535
commands, help	#569	coordination	#544
comments, log	#54, #66	copy image	#91
compare files	#58	copy selection	#88
compile parameter file	#356	correlation energy	#562
compile type rules	#141	correlation energy, MP2	#306
compile-script-file	#515	correlation-functional	#330
complement selection	#174	count atoms	#541
CONNECT records, PDB file	#8	count molecules	#540
configuration	#308	count orbitals	#543
configuration-interaction	#309	count residues	#542
connectivity-in-pdb-file	#8	count, neighbors	#544
constant temp, dynamics	#412	counter ions, add	#267
constant, huckel	#281	CPK rendering, display buffering	#202
constant, huckel, weighted	#283	cpk-max-double-buffer-atoms	#202
constrain bond angle	#118, #147	create-atom	#112
constrain bond length	#117, #146	current-file-name	#1
constrain bond torsion	#119, #148	cursor, include in image	#46
constrain geometry	#116, #145	cursor, xy rotation	#15
constrain stereochemistry	#126, #127, #128, #129, #130, #131, #132	cursor, z clip	#29
constrain torsion	#148	cursor, z rotation	#18
constrain-bond-up	#126	cursor, z translation	#27
constrain-change-stereo	#129	cursor, zoom	#28
constrain-drawing	#120	custom-title	#602
constrain-fix-stereo	#128	cut selection	#87
constraint, remove	#121, #122, #123	cutoff, ab initio	#297
constraint, remove all stereo constraints	#125	cutoff-inner-radius	#279
constraint, remove stereo	#124	cutoff-outer-radius	#280
constraint-bond-down	#127	cutoff-type	#278
contour color	#13, #14	cycle-atom-stereo	#131
contour graph increment	#384	cycle-bond-stereo	#130
		cylinders-color-by-element	#193

cylinders-width-ratio ..... #192

## D

d orbitals, ab initio ..... #304  
 data collection period, dynamics ..... #415  
 data, graph ..... #374  
 data, mechanics ..... #589, #590  
 DDE, help ..... #571  
 declare-integer ..... #578  
 declare-string ..... #579  
 default element ..... #105, #134  
 default menu, use ..... #588  
 default setting, info ..... #598  
 defaults, restore ..... #585  
 delete-atom ..... #113  
 delete-file ..... #57  
 delete-named-selection ..... #166  
 delete-selected-atoms ..... #77  
 delta torsion angle, nucleic acid ..... #249  
 delta, maximum, Monte Carlo ..... #402  
 deselect all ..... #158  
 diameter, cylinders ..... #192  
 diameters, balls ..... #188  
 dielectric constant, method for ..... #273  
 dielectric constant, scale factor ..... #274  
 DIIS ..... #287  
 dipole moment ..... #446, #447  
 dipole moment, display ..... #237  
 dipole, export ..... #19, #56  
 dipole, import ..... #23, #55  
 dipole, uv components ..... #438  
 dipole, uv total ..... #437  
 dipole, uv transition ..... #439  
 dipole-moment ..... #446  
 dipole-moment-components ..... #447  
 dipoles, show ..... #211  
 direct scf calculation, ab initio ..... #301  
 directory, current ..... #2  
 display buffering ..... #201, #202  
 DNA, A form ..... #259  
 DNA, B form ..... #260  
 DNA, Z form ..... #261  
 do-langevin-dynamics ..... #460  
 do-molecular-dynamics ..... #459  
 do-monte-carlo ..... #461

do-optimization ..... #457  
 do-qm-calculation ..... #360  
 do-qm-graph ..... #361  
 do-qm-isosurface ..... #362  
 d-orbitals-on-second-row (huckel) ..... #284  
 do-single-point ..... #456  
 dot-surface-angle ..... #200  
 double buffered atoms ..... #202  
 double strand, nucleic acid ..... #254  
 double-buffered-display ..... #201  
 do-vibrational-analysis ..... #458  
 draw atom ..... #112  
 dynamics average, append ..... #462  
 dynamics graph, append ..... #463  
 dynamics screen refresh period ..... #393  
 dynamics, Langevin, calculate ..... #460  
 dynamics-average-period ..... #418  
 dynamics-bath-relaxation-time ..... #413  
 dynamics-collection-period ..... #415  
 dynamics-constant-temp ..... #412  
 dynamics-cool-time ..... #397  
 dynamics-final-temp ..... #405  
 dynamics-friction-coefficient ..... #414  
 dynamics-heat-time ..... #395  
 dynamics-info-elapsed-time ..... #550  
 dynamics-info-kinetic-energy ..... #548  
 dynamics-info-last-update ..... #551  
 dynamics-info-potential-energy ..... #547  
 dynamics-info-temperature ..... #549  
 dynamics-info-total-energy ..... #546  
 dynamics-playback ..... #408  
 dynamics-playback-end ..... #410  
 dynamics-playback-period ..... #411  
 dynamics-playback-start ..... #409  
 dynamics-restart ..... #407  
 dynamics-run-time ..... #396  
 dynamics-seed ..... #419  
 dynamics-simulation-temp ..... #404  
 dynamics-snapshot-filename ..... #416  
 dynamics-snapshot-period ..... #417  
 dynamics-starting-temp ..... #403  
 dynamics-temp-step ..... #406  
 dynamics-time-step ..... #398

**E**

eigenvalue, scf . . . . . #441  
 eigenvector, scf alpha . . . . . #442  
 eigenvector, scf beta . . . . . #443  
 elapsed time, molecular dynamics . . . . . #550  
 electronic spectrum, compute . . . . . #475  
 electrostatic potential . . . . . #472  
 electrostatic scale factor . . . . . #275  
 electrostatics, mechanics mmp . . . . . #277  
 element color . . . . . #220, #240  
 element color, revert . . . . . #221  
 element, default . . . . . #105, #134  
 element, symbol . . . . . #523  
 energy, bend . . . . . #557  
 energy, ci excitation . . . . . #312  
 energy, electrostatic . . . . . #561  
 energy, hydrogen bond . . . . . #560  
 energy, MP2 . . . . . #562  
 energy, nonbonded . . . . . #559  
 energy, scf . . . #448, #449, #450, #451, #452  
 energy, stretch . . . . . #556  
 energy, torsion . . . . . #558  
 energy, total . . . . . #555  
 energy, uv . . . . . #435  
 enlarge . . . . . #34, #97  
 ensemble averaging, calculate . . . . . #461  
 enum id, info . . . . . #596  
 enum list, info . . . . . #597  
 epsilon torsion angle, nucleic acid . . . . . #250  
 erase atom . . . . . #113  
 erase atoms . . . . . #77, #86  
 erase file . . . . . #57  
 error . . . . . #503  
 error message format . . . . . #505, #506  
 errors-are-not-omsgs . . . . . #499  
 errors-are-omsgs . . . . . #498  
 estatic-energy . . . . . #561  
 exchange-functional . . . . . #331  
 excited state, electronic . . . . . #309  
 excited-state . . . . . #291  
 execute-client . . . . . #511  
 execute-hyperchem-client . . . . . #512  
 execute-string . . . . . #517  
 exit . . . . . #71  
 exit-script . . . . . #514  
 expand . . . . . #82

explicit hydrogens . . . . . #104, #133  
 export quantum results . . . . . #67  
 export-dipole . . . . . #19  
 export-ir . . . . . #20  
 export-orbitals . . . . . #21  
 export-property-file . . . . . #56  
 export-uv . . . . . #22  
 extend ring . . . . . #177  
 extend side chain . . . . . #178  
 extend to sp3 . . . . . #179  
 extended atom . . . . . #534  
 extra basis set, atom . . . . . #529

**F**

factory setting, info . . . . . #598  
 factory-settings . . . . . #585  
 field-direction . . . . . #323  
 field-strength . . . . . #324  
 file comparison . . . . . #58  
 file destination, image . . . . . #43  
 file merging . . . . . #17  
 file path . . . . . #2  
 file, delete . . . . . #57  
 file, hin . . . . . #5  
 file, merge . . . . . #49, #61  
 file, new . . . . . #59  
 file, open . . . . . #48, #60  
 file, save . . . . . #62  
 file, save-as . . . . . #63  
 file, write . . . . . #50  
 file-diff-message . . . . . #58  
 file-format . . . . . #3  
 filename for current system . . . . . #1  
 filename, snapshots . . . . . #416  
 file-needs-saved . . . . . #4  
 final temp, dynamics . . . . . #405  
 force field parameters changed . . . . . #314  
 format, file (HIN, ENT, etc.) . . . . . #3  
 frequency, ir . . . . . #430  
 friction coefficient, dynamics . . . . . #414  
 front-clip . . . . . #72  
 frozen core, MP2 . . . . . #307

## G

gamma torsion angle, nucleic acid	#248
geometry optimization, algorithm	#387
geometry optimization, compute	#457, #466
geometry, constrain	#116, #145
ghost atoms, ab initio	#302
global-inhibit-redisplay	#489
glossary, help	#573
gradient, calculate ab initio	#305
gradient, rms	#554
gradients	#610
gradient-x	#607
gradient-y	#608
gradient-z	#609
graph contours	#361
graph-beta	#376
graph-contour-increment	#384
graph-contour-increment-other	#386
graph-contour-levels	#381
graph-contour-start	#383
graph-contour-start-other	#385
graph-data-row	#374
graph-data-type	#373
graph-horizontal-grid-size	#377
graph-orbital-offset	#379
graph-orbital-selection-type	#375
graph-plane-offset	#382
graph-vertical-grid-size	#378
greyscale, image	#47
grid data, isosurface	#365, #366, #367, #368, #369, #370
grid step size, isosurface	#364
grid, contour graph	#377, #378
grid, isosurface	#464
grid-max-value	#195
grid-min-value	#194

## H

hbond-energy	#560
heat steps, Monte Carlo	#399
heat time, dynamics	#395
heat-of-formation	#452
height, window	#603

help	#566
help, using	#574
help-commands	#569
help-glossary	#573
help-index	#567
help-keyboard	#568
help-tools	#570
hide selection	#226
hide-toolbar	#213
highlights, balls	#187
highlights, spheres	#190
hin file contents	#5, #6
hin file, velocities	#5
horizontal grid size, contour	#377
host name, backend	#317
huckel method, d-orbitals	#284
huckel-constant	#281
huckel-scaling-factor	#282
huckel-weighted	#283
hydrogen bonds, recompute	#235
hydrogen bonds, show	#208, #234
hydrogen, extended atom	#534
hydrogens, add	#135
hydrogens, explicit	#104, #133
hydrogens, show	#203, #231
hydrogens-in-pdb-file	#7
HyperChem, about	#575

## I

icon, xy rotation	#30
icon, xy translation	#32
icon, z clip	#35
icon, z rotation	#31
icon, z translation	#33
icon, zoom	#34
icons, hide	#213
id, info	#599
ignore-script-errors	#478
image file	#43
image, copy	#91
image, setup	#39, #92
image-color	#47
image-destination-clipboard	#45
image-destination-file	#43
image-destination-filename	#44

image-file-bitmap	#40	ir-band-count	#429
image-file-bitmapRGB	#41	ir-frequency	#430
image-file-metafile	#42	ir-intensity	#431
image-include-cursor	#46	ir-intensity-components	#432
image-source-window	#39	ir-normal-modes	#433
import quantum results	#68	is-extended-hydrogen	#534
import-dipole	#23	isis, copy sketch format	#89
import-ir	#24	isomer, amino acid	#244
import-orbitals	#25	isosurface	#472
import-property-file	#55	isosurface display toggle	#230
import-uv	#26	isosurface, display type	#229
include file	#61	isosurface, plot	#362
include structure	#49	isosurface, show	#209
index, help	#567	isosurface-grid-step-size	#364
inertial axes	#210	isosurface-hide-molecule	#199
inertial axes, show	#236	isosurface-map-function	#363
info, atom	#591	isosurface-map-function-display-legend	#371
info, mechanics	#589, #590	isosurface-map-function-range	#372
info-access	#595	isosurface-mesh-quality	#464
info-enum-id-of	#596	isosurface-render-method	#197
info-enum-list	#597	isosurface-threshold	#196
info-factory-setting	#598	isosurface-transparency-level	#198
info-id-of	#599	isosurface-x-min	#368
info-type-of	#600	isosurface-x-nodes	#365
info-type-of-element	#601	isosurface-y-min	#369
info-variable-target	#593	isosurface-y-nodes	#366
inhibit-redisplay	#488	isosurface-z-min	#370
inner radius, cutoff	#279	isosurface-z-nodes	#367
integer, declare	#578	is-ring-atom	#533
integral buffer, ab initio	#298	iterations, scf	#286
integral format, ab initio	#299		
integral storage path, ab initio	#300, #301	<b>K</b>	
integral threshold, ab initio	#297	keyboard, help	#568
intensity, ir	#431	kinetic energy, molecular dynamics	#548
intensity, ir components	#432		
invert	#93	<b>L</b>	
ionic ends, remove	#265	label, atom	#531
ions, allow	#106, #137	labels	#238
IR spectra, export	#56	labels, atom	#214
IR spectra, import	#55	labels, residue	#215
IR spectrum, export	#20	Langevin dynamics, calculate	#460
IR spectrum, import	#24	Langevin dynamics, compute	#468
IR spectrum, read	#68	last update, molecular dynamics	#551
IR spectrum, save	#67		
ir-animate-amplitude	#425		
ir-animate-cycles	#427		
ir-animate-steps	#426		

legend, isosurface . . . . . #371  
 length, bond . . . . . #83, #101  
 length, constrain bond . . . . . #146  
 level, selection . . . . . #149  
 line width, bonds . . . . . #191  
 load structure . . . . . #48, #49  
 load-default-menu . . . . . #588  
 load-user-menu . . . . . #586  
 log comment . . . . . #54, #66  
 log, mechanics print level . . . . . #420  
 log, quantum print level . . . . . #421  
 log, start . . . . . #52, #64  
 log, stop . . . . . #53, #65

## M

magnetic-field-direction . . . . . #325  
 magnetic-field-strength . . . . . #326  
 magnify . . . . . #82, #97  
 mapped function display range . . . . . #372  
 mapped function legend . . . . . #371  
 mapped function, isosurface . . . . . #363  
 mass, atom . . . . . #527  
 mass, set . . . . . #144  
 max cycles, geometry optimization . . . . . #389  
 max-iterations, scf calculation . . . . . #286  
 mechanics-data . . . . . #589  
 mechanics-dielectric . . . . . #273  
 mechanics-dielectric-scale-factor . . . . . #274  
 mechanics-electrostatic-scale-factor . . . . . #275  
 mechanics-info . . . . . #590  
 mechanics-mmp-electrostatics . . . . . #277  
 mechanics-print-level . . . . . #420  
 mechanics-van-der-waals-scale-factor . . . . . #276  
 menu, load custom . . . . . #586  
 menu, use custom . . . . . #587  
 menu, use default . . . . . #588  
 menu-build-add-hydrogens . . . . . #135  
 menu-build-all-atoms . . . . . #139  
 menu-build-allow-ions . . . . . #137  
 menu-build-calculate-types . . . . . #140  
 menu-build-compile-type-rules . . . . . #141  
 menu-build-constrain-bond-angle . . . . . #147  
 menu-build-constrain-bond-length . . . . . #146  
 menu-build-constrain-bond-torsion . . . . . #148  
 menu-build-constrain-geometry . . . . . #145

menu-build-default-element . . . . . #134  
 menu-build-explicit-hydrogen . . . . . #133  
 menu-build-model-build . . . . . #136  
 menu-build-set-atom-type . . . . . #142  
 menu-build-set-charge . . . . . #143  
 menu-build-set-mass . . . . . #144  
 menu-build-united-atoms . . . . . #138  
 menu-cancel . . . . . #520  
 menu-compute-electronic-spectrum . . . . . #475  
 menu-compute-geometry-optimization . . . . . #466  
 menu-compute-langevin-dynamics . . . . . #468  
 menu-compute-molecular-dynamics . . . . . #467  
 menu-compute-monte-carlo . . . . . #469  
 menu-compute-orbitals . . . . . #473  
 menu-compute-plot-molecular-properties . . . . . #472  
 menu-compute-single-point . . . . . #465  
 menu-compute-transition-state . . . . . #471  
 menu-compute-vibrational-spectrum . . . . . #474  
 menu-compute-vibrations . . . . . #470  
 menu-databases-add-counter-ions . . . . . #267  
 menu-databases-amino-acids . . . . . #263  
 menu-databases-make-zwitterion . . . . . #264  
 menu-databases-mutate . . . . . #268  
 menu-databases-nucleic-acids . . . . . #266  
 menu-databases-remove-ionic-ends . . . . . #265  
 menu-display-color . . . . . #239  
 menu-display-element-color . . . . . #240  
 menu-display-hide-selection . . . . . #226  
 menu-display-isosurface . . . . . #229  
 menu-display-labels . . . . . #238  
 menu-display-last-rendering . . . . . #228  
 menu-display-overlay . . . . . #223  
 menu-display-recompute-h-bonds . . . . . #235  
 menu-display-rendering . . . . . #227  
 menu-display-scale-to-fit . . . . . #222  
 menu-display-show-all . . . . . #224  
 menu-display-show-dipole-moment . . . . . #237  
 menu-display-show-hydrogen-bonds . . . . . #234  
 menu-display-show-hydrogens . . . . . #231  
 menu-display-show-inertial-axes . . . . . #236  
 menu-display-show-isosurface . . . . . #230  
 menu-display-show-multiple-bonds . . . . . #233  
 menu-display-show-periodic-box . . . . . #232  
 menu-display-show-selection-only . . . . . #225  
 menu-edit-align-molecules . . . . . #100  
 menu-edit-align-viewer . . . . . #99  
 menu-edit-clear . . . . . #86



menu-edit-copy	#88	menu-select-name-selection	#176
menu-edit-copy-image	#91	menu-select-residues	#169
menu-edit-copy-isis-sketch	#89	menu-select-select	#175
menu-edit-cut	#87	menu-select-select-all	#173
menu-edit-invert	#93	menu-select-select-backbone	#180
menu-edit-paste	#90	menu-select-select-sphere	#172
menu-edit-reflect	#94	menu-setup-ab-initio	#350
menu-edit-rotate	#95	menu-setup-compile-parameter-file	#356
menu-edit-set-bond-angle	#102	menu-setup-molecular-mechanics	#348
menu-edit-set-bond-length	#101	menu-setup-network	#351
menu-edit-set-bond-torsion	#103	menu-setup-parameter-set	#355
menu-edit-setup-image	#92	menu-setup-periodic-box	#352
menu-edit-translate	#96	menu-setup-reaction-map	#357
menu-edit-z-clip	#98	menu-setup-restraints	#353
menu-edit-zoom	#97	menu-setup-semi-empirical	#349
menu-file-exit	#71	menu-setup-set-velocity	#354
menu-file-export	#67	merge file	#49, #61
menu-file-import	#68	merged systems, translate	#17
menu-file-log-comments	#66	mesh, isosurface	#464
menu-file-merge	#61	message	#502
menu-file-new	#59	metafile image	#42
menu-file-open	#60	method, calculation	#269
menu-file-preferences	#70	method, molecular mechanics	#270
menu-file-print	#69	method, semi-empirical	#271
menu-file-quit	#71	microstate, CI	#309
menu-file-save	#62	mmp, electrostatics	#277
menu-file-save-as	#63	mo initial guess, ab initio	#296
menu-file-start-log	#64	mode, vibrational	#428
menu-file-stop-log	#65	model build	#136
menu-help-about-hyperchem	#575	module-name	#611
menu-help-commands	#569	molecular dynamics, compute	#459, #467
menu-help-glossary	#573	molecular mechanics, setup	#348
menu-help-index	#567	molecular orbital, number of coefficients	#380
menu-help-keyboard	#568	molecular properties, compute	#472
menu-help-on-scripts	#572	molecular-mechanics-method	#270
menu-help-scripts-and-dde	#571	molecule hide	#199
menu-help-tools	#570	molecule, align	#75, #100
menu-help-using-help	#574	molecule-count	#540
menu-script-one . . . ten	#519	molecule-name	#563
menu-script-open-script	#518	molecules, rotate	#79
menu-select-atoms	#168	Monte Carlo, calculate	#461
menu-select-complement-selection	#174	Monte Carlo, compute	#469
menu-select-extend-ring	#177	monte-carlo-cool-steps	#401
menu-select-extend-side-chain	#178	monte-carlo-heat-steps	#399
menu-select-extend-to-sp3	#179	monte-carlo-info-acceptance-ratio	#552
menu-select-molecules	#170	monte-carlo-max-delta	#402
menu-select-multiple-selections	#171	monte-carlo-run-steps	#400

mouse-mode ..... #74  
 move molecules ..... #81, #96  
 move viewer ..... #80, #96  
 MP2 correlation energy ..... #306  
 MP2 frozen core ..... #307  
 mp2-energy ..... #562  
 multiple bonds display ..... #207  
 multiple bonds, show ..... #205, #233  
 multiple selections ..... #171  
 multiple-selections ..... #152  
 multiplicity ..... #289  
 mutate residue ..... #262, #268

## N

name selection ..... #176  
 name, atom ..... #521  
 name, residue ..... #538  
 named selection, delete ..... #166  
 named selection, select ..... #159  
 named-selection-count ..... #154  
 named-selection-name ..... #155  
 named-selection-value ..... #156  
 name-molecule ..... #165  
 name-selection ..... #164  
 negatives-color ..... #14  
 neighbors ..... #536  
 neighbors, count ..... #544  
 network setup ..... #351  
 nonbond-energy ..... #559  
 non-standard-pdb-names ..... #9  
 normal modes, ir ..... #433  
 no-source-refs-in-errors ..... #506  
 notify variable ..... #577  
 notify-on-update ..... #493  
 nucleic acids, add ..... #258, #266  
 nucleic-a-form ..... #259  
 nucleic-alpha ..... #246  
 nucleic-backwards ..... #253  
 nucleic-beta ..... #247  
 nucleic-b-form ..... #260  
 nucleic-chi ..... #252  
 nucleic-delta ..... #249  
 nucleic-double-strand ..... #254  
 nucleic-epsilon ..... #250  
 nucleic-gamma ..... #248

nucleic-sugar-pucker ..... #245  
 nucleic-zeta ..... #251  
 nucleic-z-form ..... #261  
 number of selected atoms ..... #150  
 number, atomic ..... #522  
 number, selected atom ..... #151  
 number, serial ..... #565

## O

occupancy, alpha orbital ..... #444  
 occupancy, beta orbital ..... #445  
 omega torsion angle, amino acid ..... #243  
 omsg string ..... #502  
 omsg to file, append ..... #497  
 omsg-file ..... #491  
 omsgs-not-to-file ..... #496  
 omsgs-to-file ..... #495  
 one-line-arrays ..... #594  
 opacity, isosurface ..... #198  
 open file ..... #48, #60  
 open script ..... #518  
 optim-algorithm ..... #387  
 optim-converged ..... #394  
 optim-convergence ..... #388  
 optimization convergence limit, cycles ..... #389  
 optimization, compute ..... #457, #466  
 optim-max-cycles ..... #389  
 orbital color ..... #13, #14  
 orbital data, export ..... #56  
 orbital data, export, quantumcalculations, export  
     #21  
 orbital data, import ..... #55  
 orbital data, import, quantumcalculations, import  
     #25  
 orbital occupancy ..... #444, #445  
 orbital offset, contour graph ..... #379  
 orbital selection ..... #375  
 orbital, beta or alpha ..... #376  
 orbital, type to graph ..... #375  
 orbital-count ..... #543  
 orbitals, ci occupied ..... #310  
 orbitals, ci unoccupied ..... #311  
 orbitals, compute ..... #473  
 orbitals, read ..... #68  
 orbitals, save ..... #67

orient molecule ..... #75  
 orient molecules ..... #79, #95, #100  
 orient viewer ..... #76, #78, #80, #99  
 oscillator, uv strength ..... #440  
 outer radius, cutoff ..... #280  
 overlay ..... #223

## P

parameter file, compile ..... #356  
 parameter set, select ..... #355  
 parameter set, use ..... #343  
 parameter-set-changed ..... #314  
 password, backend user ..... #319  
 paste ..... #90  
 path ..... #2  
 path, file ..... #2  
 pause-for ..... #492  
 pdb file, connectivity ..... #8  
 pdb file, hydrogens ..... #7  
 pdb file, non-standard names ..... #9  
 periodic boundaries, use conditions ... #344,  
     #345, #352, #392  
 periodic box, add ..... #344  
 periodic box, setup ..... #352  
 periodic box, show ..... #204, #232  
 periodic box, size ..... #345  
 perspective ..... #181  
 phi torsion angle, amino acid ..... #241  
 pi-pi, zindo/1 overlap weighting factor . #293  
 pi-pi, zindo/s overlap weighting factor . #295  
 plane offset, contour graph ..... #382  
 playback end, dynamics ..... #410  
 playback period, dynamics ..... #411  
 playback start, dynamics ..... #409  
 playback, dynamics snaphots ..... #408  
 polar-field-strength ..... #327  
 polarizability-calculated ..... #328  
 pop-no-value ..... #582  
 pop-value ..... #581  
 positives-color ..... #13  
 potential energy, molecular dynamics . #547  
 preferences ..... #70  
 print image ..... #69  
 print level, mechanics ..... #420  
 print level, quantum ..... #421

printer-background-white ..... #38  
 print-variable-list ..... #584  
 process count, backend ..... #320  
 properties, molecular, compute ..... #472  
 property file, export ..... #56  
 property file, import ..... #55  
 psi torsion angle, amino acid ..... #242  
 pucker, nucleotide sugar ..... #245  
 push ..... #580

## Q

qm isosurface, plot ..... #362  
 quantum calculation, do ..... #360  
 quantum data, export ..... #56  
 quantum data, import ..... #55  
 quantum-print-level ..... #421  
 quantum-total-charge ..... #288  
 query-response-has-tag ..... #477  
 query-value ..... #509  
 quit ..... #71

## R

radii, balls ..... #188  
 radius, cylinders ..... #192  
 Raffenetti integral format ..... #299  
 random seed, dynamics ..... #419  
 reaction map setup ..... #357  
 read file ..... #60  
 read quantum results ..... #68  
 read structure ..... #48, #49  
 read/write variables ..... #595  
 read-binary-script ..... #516  
 read-script ..... #507  
 read-tcl-script ..... #508  
 recompute hydrogen bonds ..... #235  
 rectangle, group select ..... #172  
 redisplay, inhibit ..... #488  
 reflect ..... #94  
 refresh period ..... #393  
 relaxation time ..... #413  
 remove-all-stereo-constraints ..... #125  
 remove-stereo-constraint ..... #124  
 rendering method, isosurface ..... #197

rendering type	#227	scf convergence, accelerate	#287
rendering, restore last mode	#228	scf iterations	#286
render-method	#185	scf-atom-energy	#448
reorder-selections	#16	scf-binding-energy	#449
request	#510	scf-convergence	#285
residue labels	#215	scf-core-energy	#450
residue, deselect	#163	scf-electronic-energy	#451
residue, mutate	#262	scf-orbital-energy	#441
residue-charge	#537	screen-refresh-period	#393
residue-coordinate	#539	script errors	#478
residue-count	#542	script errors, ignore	#478
residue-label-text	#532	script file, compile	#515
residue-name	#538	script menu item, change	#513
residues, select	#162, #169	script, execute string variable as	#517
response tag, query	#477	script, exit	#514
response to query has tag	#477	script, one . . . ten	#519
restart, dynamics	#407	script, open	#518
restraint, use	#341	script, pause in	#492
restraints	#339, #340	script, read	#507, #508
restraints, ignore	#342	script, read binary	#516
restraints, setup	#353	script, run from menu item	#519
restraint-tether	#340	script-menu-caption	#479
revert-element-colors	#221	script-menu-checked	#480
rhf/uhf	#290	script-menu-command	#481
ribbons, rendering	#184	script-menu-enabled	#482
ring, atom	#533	script-menu-help-file	#483
ring, extend selection	#177	script-menu-help-id	#484
rms gradient	#554	script-menu-in-use	#485
rotate molecules	#79, #95	script-menu-message	#486
rotate-viewer	#78	script-refs-in-errors	#487
rotation	#15	scripts, help	#571
run steps, monte carlo	#400	scripts, help on context sensitive	#572
run time, dynamics	#396	select all	#173
		select atoms	#168
		select backbone	#180
		select by name	#159
		select by name or number	#175
		select by named selection	#159
		select molecules	#170
		select residues	#169
		select sphere	#172
		select-atom	#160
		selected atoms, delete	#77
		selected-atom, sequence number	#151
		selected-atom-count	#150
		selection color	#219
		selection level	#149, #168, #169, #170
<b>S</b>			
save as, file	#63		
save file	#4, #62		
save quantum results	#67		
save/save as, file	#50		
scale factor, dielectric constant	#274		
scale factor, electrostatic	#275		
scale factor, van der Waals	#276		
scale to fit	#222		
scaling, huckel factor	#282		
scf convergence, ab initio	#303		

selection, complement	#174	show all	#224
selection, delete named	#166	show axes	#210, #236
selection, hide	#226	show hydrogen bonds	#208, #234
selection, name	#164	show hydrogens	#203, #231
selection, named count	#154	show inertial axes	#236
selection, named selection	#155	show multiple bonds	#205, #233
selection, named value	#156	show periodic box	#204, #232
selection, show only	#225	show perspective	#181
selection, translate	#81	show selection	#225
selection-color	#12	show-aromatic-rings-as-circles	#212
selections, multiple	#152, #171	show-dipoles	#211
selections, reorder	#16	show-gradient-vectors	#206
selection-target	#149	show-isosurface	#209
selection-value	#157	show-ribbons	#184
select-name	#159	show-stereo	#182
select-none	#158	show-stereochem-wedges	#216
select-path	#167	show-vibrational-vectors	#423
select-residue	#162	shrink	#34, #82, #97
select-sphere	#153	side chain, extend selection	#178
semi-empirical calculation	#360	sigma-sigma, zindo/1 overlap weighting factor	#292
semi-empirical data, export	#56	sigma-sigma, zindo/s overlap weighting factor	#294
semi-empirical data, import	#55	simulation temp, dynamics	#404
semi-empirical results, export	#21	single point, compute	#456
semi-empirical results, import	#25	single-point, compute	#465
semi-empirical setup	#349	size, balls	#188
semi-empirical-method	#271	size, window	#603, #604
sequence number, selected atom	#151	sketch, copy format to clipboard	#89
serial-number	#565	small-density-cutoff	#333
set atom charge	#109, #143	snapshots, store	#408
set atom type	#108, #142	snapshot filename, dynamics	#416
set bond angle	#84, #102	snapshot period, dynamics	#417
set bond length	#101	snapshot playback period	#411
set bond torsion	#103	snapshots, playback	#408
set mess	#144	solvate-system	#344
set velocity, setup	#354	solvate-system-in-this-box	#345
set-atom-charge	#109	source-refs-in-errors	#505
set-atom-charges-to-formal	#111	sp3, extend selection	#179
set-atom-type	#108	spectra, IR, export	#56
set-bond	#114	spectra, IR, import	#55
set-bond-length	#83	spectra, UV, export	#56
set-bond-torsion	#85	spectra, UV, import	#55
settings, restore default	#585	spectrum, compute electronic	#475
setup image	#39, #92	spectrum, compute vibrational	#474
set-velocity	#346	spectrum, IR, export	#20
shading, balls	#186	spectrum, IR, import	#24
shading, spheres	#189		
shifted cutoff	#278		

spectrum, UV, export	#22
spectrum, UV, import	#26
sphere, group select	#172
sphere, select	#153
spheres-highlighted	#190
spheres-shaded	#189
spin density	#472
spin, uv	#436
stack, pop	#581, #582
stack, push	#580
starting temp, dynamics	#403
start-logging	#52, #64
state-to-optimize	#334
status-message	#490
stereo, change constraint	#129
stereo, change stereo chemistry	#132
stereo, cycle atom	#131
stereo, cycle bond	#130
stereo, fix stereo constraint	#128
stereo, remove all constraints	#125
stereo, remove constraints	#124
stereo, show	#182
stereo, show wedges	#216
stereo, wall-eyed or crosseyed	#183
stereochemistry, atom	#545
stereochemistry, constrain	#126, #127, #128, #129, #130, #131, #132
stereochemistry, enforced by Model Builder	#107
sticks-width	#191
stop-logging	#53, #65
stretch-energy	#556
string, declare	#579
structural optimization, algorithm	#387
structure hide	#199
structure merging	#17
sugar-pucker	#245
switched cutoff	#278
switch-to-user-menu	#587
system altered, save file	#4

## T

tcl script, read	#508
temperature step, dynamics	#406
temperature, molecular dynamics	#549

threshold, ab initio	#297
threshold, isosurface	#196
time step, dynamics	#398
title, customized	#602
tndo-version	#335
toggle	#583
tool selection	#74
tool, xy rotation	#15, #30, #32
tool, z clip	#29, #35
tool, z rotation	#18, #31
tool, z translation	#27, #33
tool, zoom	#28, #34
toolbar, hide	#213
tools, help	#570
torsion angle omega	#243
torsion angle phi	#241
torsion angle psi	#242
torsion angle, constrain	#119, #148
torsion angle, set	#85, #103
torsion, bond	#85
torsion-energy	#558
total charge, quantum	#288
total energy, dynamics	#546
total-energy, any calculation	#555
transition state setup	#357
transition state, compute	#471
translate	#96
translate-merged-systems	#17
translate-selection	#81
translate-view	#80
translate-whole-molecules	#36
translation	#37
transparency, isosurface	#198
type rules, compile	#141
type, atom	#108, #524
type, set atom	#142
types, calculate atom	#140

## U

uhf	#290
uhf orbitals	#376
unconstrain-bond-angle	#122
unconstrain-bond-length	#121
unconstrain-bond-torsion	#123
united atom	#534

united atoms ..... #138  
 unselect all ..... #158  
 un-select-atom ..... #161  
 un-select-residue ..... #163  
 update dynamics, last step ..... #551  
 update variable ..... #493  
 update variable, cancel ..... #494  
 use-auxilliary-basis ..... #336  
 use-electric-field ..... #321  
 use-fast-translation ..... #37  
 use-magnetic-field ..... #322  
 use-mullikan-approximation-to-density ..... #337  
 use-no-restraints ..... #342  
 use-parameter-set ..... #343  
 user id, backend ..... #318  
 user menu, load ..... #586  
 user menu, use ..... #587  
 user menuitem (for scripts) ..... #513  
 user password, backend ..... #319  
 use-restraint ..... #341  
 use-small-density-cutoff ..... #338  
 using-help ..... #574  
 UV spectra, export ..... #56  
 UV spectra, import ..... #55  
 UV spectrum ..... #308  
 UV spectrum, export ..... #22  
 UV spectrum, import ..... #26  
 UV/visible spectrum, read ..... #68  
 UV/visible spectrum, save ..... #67  
 uv-band-count ..... #434  
 uv-dipole-components ..... #438  
 uv-energy ..... #435  
 uv-oscillator-strength ..... #440  
 uv-spin ..... #436  
 uv-total-dipole ..... #437  
 uv-transition-dipole ..... #439

## V

value, named selection ..... #156  
 value, query ..... #509  
 value, selection ..... #157  
 van der Waals scale factor ..... #276  
 variable list, print ..... #584  
 variable, declare ..... #578, #579  
 variable, info ..... #593

variable, toggle ..... #583  
 variable-changed ..... #577  
 vector, type ..... #601  
 vectors, show vibrations ..... #423  
 velocities, atom ..... #553  
 velocities-in-hin-file ..... #5  
 velocity, set atomic ..... #346, #354  
 version ..... #564  
 vertical grid size, contour ..... #378  
 vibrational analysis, compute ..... #458  
 vibrational spectrum, compute ..... #474  
 vibrational, show vectors ..... #423  
 vibrational-mode ..... #428  
 vibrations, animate ..... #424  
 vibrations, compute ..... #470  
 view, enlarge or shrink ..... #34  
 viewer, align ..... #76, #99  
 viewer, rotate ..... #78  
 viewer, translate ..... #80  
 view-in-hin-file ..... #6

## W

wall-eyed-stereo ..... #183  
 warning ..... #504  
 warning messages, backend ..... #422  
 warnings-are-not-omsgs ..... #501  
 warnings-are-omsgs ..... #500  
 warning-type ..... #422  
 wedges, stereochemistry ..... #216  
 whole molecules, translate ..... #36  
 width, sticks ..... #191  
 width, window ..... #604  
 window-color ..... #10  
 window-height ..... #603  
 window-width ..... #604  
 workspace clear ..... #59  
 workspace-height ..... #605  
 workspace-width ..... #606  
 write-atom-map ..... #576  
 write-file ..... #50

## X

x-y-rotation-cursor ..... #15

x-y-rotation-icon-step . . . . . #30  
 x-y-translation-icon-step . . . . . #32

**Z**

z clipping . . . . . #29  
 Z form, DNA . . . . . #261  
 z-clip . . . . . #98  
 zeta torsion angle, nucleic acid . . . . . #251  
 zindo-1-pi-pi . . . . . #293  
 zindo-1-sigma-sigma . . . . . #292  
 zindo-s-pi-pi . . . . . #295  
 zindo-s-sigma-sigma . . . . . #294  
 zoom . . . . . #82  
 zoom, set factor . . . . . #97  
 zoom-cursor . . . . . #28  
 zoom-icon-step . . . . . #34  
 z-rotation-cursor . . . . . #18  
 z-rotation-icon-step . . . . . #31  
 z-translation-cursor . . . . . #27  
 z-translation-icon-step . . . . . #33  
 zwitterion, make . . . . . #264



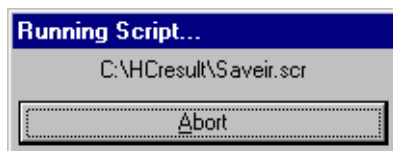


## Scripts

Scripts are text files containing a set of statements that generate input messages (IMSGs) to run HyperChem. Scripts replace HyperChem mouse and keyboard operations, so that you can run HyperChem unattended. You can also interrupt a script to interact manually with HyperChem. Use scripts to repeat HyperChem operations that you do frequently, to do batch analyses, and to set up HyperChem conditions.

Write scripts using a text editor, like Windows Notepad. These files should be unformatted text files, with the extension *.scr*. When you open a script file, using the Script menu in HyperChem, HyperChem automatically executes the instructions in the file.

A script runs to completion, unless it finds an error (you should include the instructions in the script to report errors to screen). It responds to a menu activator by displaying a dialog box, or you use the Abort option in the Running Script . . . message box.



*Example:*

This script performs a geometry optimization of anthracene. The numbers ①, ②, and ③ and so on do not belong to the script, but are present for this discussion.

```
; Test amber optimization of anthracene. ①
calculation-method molecularmechanics ②
molecular-mechanics-method amber ③
optim-max-cycles 200 ④
optim-convergence 0.1 ④
optim-algorithm polakribiere ④
start-logging test.log no ⑤
open-file anth.hin ⑥
do-optimization ⑦
```

write-file test.hin ⑧  
append-omsgs-to-file test.log ⑨  
query-value stretch-energy ⑩  
omsgs-not-to-file ⑪  
stop-logging ⑫  
exit-script ⑬

① A comment. All comment lines must start with a semicolon. Comments appear in the Comments section of the Open Script dialog box.

② Sets the calculation method (variable) to molecular mechanics.

③ Chooses the AMBER force field (variable) for the calculation.

④ Sets parameters (variables) for the calculation. If you do not set these variables, HyperChem uses the last settings.

⑤ Starts a log file, named test.log, to record results of the calculation. No instructs HyperChem to replace the contents of any existing log file having the name test.log. Quotes can surround individual arguments, for clarity. You could also use the menu activator “menu-file-start-log”, but it requires that you respond to the Open Log dialog box.

⑥ Opens the file anth.hin and displays the anthracene molecule in the workspace.

⑦ Performs the geometry optimization calculation.

⑧ Stores the optimized molecule in a file, test.hin.

⑨ Adds error or warning messages, plus the results of the query-value command to the log file. Without this command, these messages go to a dialog box and can interrupt the script.

⑩ Requests the final value of bond stretching energy (a component of total energy) for the optimized molecule. Query values are OMSGs, so they go into the log file. An equivalent statement is stretch-energy ?.

⑪ Stops sending OMSGs to the log file.

⑫ Closes the log file. You could also use the menu activator menu-file-stop-log.

⑬ Ends the script. This is optional.

## Dynamic Data Exchange (DDE)

DDE is a communications system available to all programs (applications) running under Microsoft Windows. Applications can use DDE messages for these functions:

- To start another application
- To establish a communications channel with another application
- To send information
- To request and receive information
- To receive information automatically, when it is updated

In any DDE interaction between two Windows applications, one application initiates the conversation, taking on the role of the “client”; the responding application is the “server.” HyperChem can act only as a server: it can supply information when the client application asks for it or, by pre-arrangement, when it updates the information.

DDE conversations use messages that can include HyperChem messages as arguments. HyperChem converts DDE messages to MSGs and responds with an action or with data.

## Other Applications to HyperChem

HyperChem can respond to six types of DDE messages from client applications:

**DDE\_INITIATE** Establishes a communications channel with HyperChem. The arguments to this command must be the DDE Application name (“HyperChem”) and Topic (“System”).

**DDE\_EXECUTE** Tells HyperChem to carry out a function, for example, turning on Multiple Selections. The arguments are the same as HyperChem script variables (Set HSV, see “Variables” on page 511), commands (see “Commands” on page 516), and menu activators (see “Menu Activators” on page 517).

**DDE\_REQUEST** Requests information from HyperChem. The arguments are the same as the request variables (Get HSV) for HyperChem scripts (see “Variables” on page 511). HyperChem responds using a DDE\_DATA message.

**DDE\_TERMINATE** Ends the conversation with HyperChem.

**DDE\_ADVISE** Requests HyperChem to supply, through a “hot” data link, an update of data whenever it changes. The argument is a variable (HSV) name.

**DDE\_UNADVISE** Removes the hot link for the HSV name in the argument.

Each application can have its own version of these DDE messages. For example, Microsoft Excel can use =Execute (Channel, “[message (argument)]”) and Microsoft Visual Basic™ can use Textn.LinkExecute “[message (argument)]” for a DDE\_EXECUTE message. HyperChem can recognize any version of these DDE messages.

To write DDE conversations with HyperChem, consult the reference manuals for applications.

## HyperChem to Other Applications

HyperChem responds to DDE messages from applications by carrying out functions or by returning information. Information returns to an application in DDE\_DATA messages.

HyperChem has one special way of using a script to initiate communication with a DDE application. The script command `execute-client` (see message cross-reference number 297) can start another application and initiate a communications channel for a particular “Topic.” The other application then controls the DDE conversation.

## HyperChem and Excel

Microsoft Excel is a spreadsheet program available for Windows. It provides a macro language for writing DDE messages, which provides for looping, conditional statements, batch calculations, and other programming conveniences. Macro files have the extension *.xlm*.

Excel spreadsheets (with the extension *.xls*) can hold the input information for HyperChem calculations and receive output data. Excel provides tools for analyzing and plotting this data.

Excel uses these messages for communicating with HyperChem:

INITIATE	Opens a communications channel between Excel and HyperChem. The only acceptable argument is "HyperChem", "System".
REQUEST	Asks for information from HyperChem. The arguments are Channel (the channel opened by INITIATE) and "request string" (the same as a Get HSV message for a HyperChem script).
EXECUTE	Starts a process in HyperChem. The arguments are requests that include variables (Set HSV), commands, and menu activators.
TERMINATE	Closes a DDE channel at the end of a communication. The only argument is Channel (the channel opened by INITIATE).

In addition to these commands, Excel also provides various commands for managing data and programming the interaction with HyperChem.

For more information about writing and using Excel command macros, see the *Microsoft Excel User's Guide* and the *Microsoft Excel Functions Reference*. Several important aspects of the example macro below cannot be seen in a simple printout. For example, simply typing the text that you see below into a macro sheet will not give a working macro; you must also Define Names for some of the cells (as indicated in Column A).

*Example:*

This example does single point calculations on a related set of molecules and stores the results in an Excel spreadsheet file (plot.xls). This macro is supplied with HyperChem; the file is plot.xlm .

	B	C
1	<b>Compute.Results</b>	<i>Compute energies for a table</i>
2		
3	=OpenFile()	<i>Call subroutine to open DDE channel</i>
4	=IF(ISERR(OR(Channel)))	<i>If problem opening a DDE channel</i>
5	= RETURN()	<i>exit this macro</i>
6	=END.IF()	<i>End of IF condition</i>
7		
8	=EXECUTE(Channel,"[query-response-has-tag(no)]")	<i>Do not return data with name of variable</i>
9	=EXECUTE(Channel,"[file-format(hin)]")	<i>Opens all files in "hin" format</i>
10		
11	=WHILE(NOT(ISBLANK(SELECTION())))	<i>While there is still a molecule to work on:</i>
12	=EXECUTE(Channel,"[open-file("&SELECTION()&"hin)][do-single-point]")	<i>Open the file and do single point calc</i>
13	=FORMULA.ARRAY(REQUEST(Channel,"total-energy"),"rc[1]")	<i>Request for data and put it in column 1</i>
14	=FORMULA.ARRAY(REQUEST(Channel,"stretch-energy"),"rc[2]")	
15	=FORMULA.ARRAY(REQUEST(Channel,"bend-energy"),"rc[3]")	
16	=FORMULA.ARRAY(REQUEST(Channel,"torsion-energy"),"rc[4]")	
17	=FORMULA.ARRAY(REQUEST(Channel,"nonbond-energy"),"rc[5]")	
18	=SELECT("r[1]c")	<i>Select molecule in next spreadsheet row</i>
19	=NEXT()	<i>End of WHILE: go back to start of loop</i>
20		
21	=TERMINATE(Channel)	<i>Close DDE channel</i>
22	=RETURN()	
23		
24	<b>OpenFile</b>	<i>Subroutine to open DDE channel</i>
25	=INITIATE("HyperChem","System")	<i>Open a DDE channel</i>
26	=IF(ISERR(OR(NewChan)))	<i>If HyperChem is not open</i>
27	= IF(ISERR(OR(EXEC("c:\chem\ship\chem",1))))	<i>start HyperChem</i>
28	= RETURN(NewChan)	<i>return if there is a problem</i>
29	= END.IF()	<i>End of IF condition</i>
30	= RETURN(INITIA("HyperChem","System"))	<i>Now open a DDE channel to new HC</i>
31	=END.IF()	<i>End of IF condition</i>
32	=RETURN(NewChan)	<i>Return from subroutine</i>

### The Excel macro contains these instructions:

Line	Comment
1	Press [Ctrl] + [R] or choose Run on Excel's Macro menu to start this macro
3	Uses the subroutine, OpenFile, starting on line 25
4 - 5	If there is a problem opening a DDE channel (an error returns from the OpenFile subroutine), the macro ends. If the subroutine reports the number of the open channel, Excel goes to the next step (line 6)
6	Ends the conditional (If) statement
8	Instructs HyperChem to send back data without the name of each variable (tag)

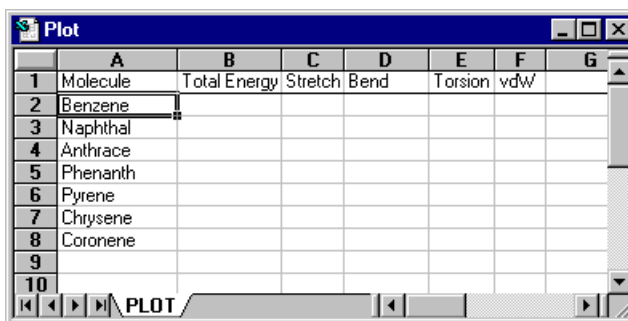
Line	Comment
9	Instructs HyperChem that the files are in HIN format
11	Starts this conditional loop (While statement): while there is still a molecule to calculate, do steps (lines) 12-19
12	Opens the HIN file indicated by the highlighted cell in the spreadsheet plot.xls; runs a single point calculation with the present HyperChem settings
13-17	Requests the indicated information from HyperChem and puts it in columns 1-5 (cells B-F) of plot.xls
18	Selects the molecule in the next row of the spreadsheet
19	Returns to step (line) 11
21	Closes the DDE channel
22	Ends the macro
24	Names the subroutine, OpenFile
25	Opens a DDE channel to HyperChem
26	If the macro cannot open a channel to HyperChem, it goes to the next line (27) to first open a HyperChem window. If a channel opens, it goes to line 32
27-28	Opens a HyperChem program and a HyperChem window. Line 27 specifies the full directory path for HyperChem; you will need to change this path if you installed HyperChem in a different directory than the one that is specified. If this attempt to start HyperChem does not work, the macro returns an error message
29	Ends the second If statement
30	Opens a DDE channel to the new HyperChem window
31	Ends the first If statement



Line	Comment
32	Returns the number of the open DDE channel to the main program

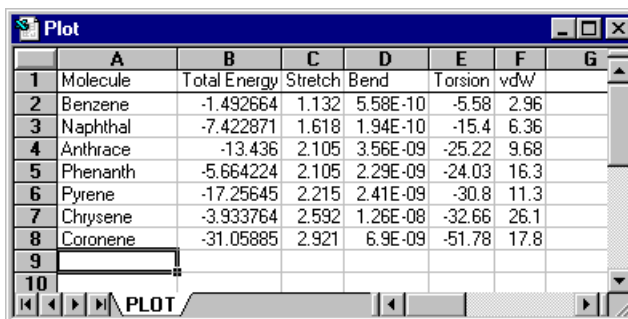
To use this Excel macro:

1. Start Excel and open plot.xlm.
2. Open the spreadsheet file plot.xls.



	A	B	C	D	E	F	G
1	Molecule	Total Energy	Stretch	Bend	Torsion	vdW	
2	Benzene						
3	Naphthal						
4	Anthrace						
5	Phenanth						
6	Pyrene						
7	Chrysene						
8	Coronene						
9							
10							

3. Choose cell 2A (benzene) in plot.xls.
4. Press [Ctrl] + [R] or choose Run on Excel's Macro menu to start the macro.



	A	B	C	D	E	F	G
1	Molecule	Total Energy	Stretch	Bend	Torsion	vdW	
2	Benzene	-1.492664	1.132	5.58E-10	-5.58	2.96	
3	Naphthal	-7.422871	1.618	1.94E-10	-15.4	6.36	
4	Anthrace	-13.436	2.105	3.56E-09	-25.22	9.68	
5	Phenanth	-5.664224	2.105	2.29E-09	-24.03	16.3	
6	Pyrene	-17.25645	2.215	2.41E-09	-30.8	11.3	
7	Chrysene	-3.933764	2.592	1.26E-08	-32.66	26.1	
8	Coronene	-31.05885	2.921	6.9E-09	-51.78	17.8	
9							
10							

As the macro runs, you can see the molecules in the HyperChem window and the values from the single point calculations in plot.xls.

**Note:** For this calculation, HyperChem uses the most recent settings for calculation method, and so on. The results in this example use a molecular mechanics calculation, the MM+ force field (set by opening the files), no cutoffs, and bond dipoles.



# Chapter 12

## Using HyperChem Results

You can use results from HyperChem sessions with other programs. For example, you could do the following:

- Prepare illustrations for publications, presentations, and reports. You can transfer images of molecular models, through the Windows Clipboard or through files, to other programs.
- Analyze the results of HyperChem calculations in other programs that run with Windows (Windows “applications”).
- Save the results of quantum mechanics calculations in an Export file with File/Export, and then process those data with other software.
- Save molecular structures in PDB files for viewing or analysis by using other software that can read this type of file.

### Setting Display Conditions

Before you copy a HyperChem graphic to the Clipboard or a file, you might want to change the appearance of the HyperChem workspace. For example, you can choose Preferences on the File menu (see page 95) to change the HyperChem window color to improve the quality of illustrations. You can also change how selected atoms look so that they have a distinctive appearance.

For reproduction in black and white, use White for Window, Black for Bond, and Thick line for Selection. With these choices, HyperChem substitutes dotted lines for solid lines where it uses color for special information (for example, negative versus positive contour lines in density plots of electronic values).

## Transferring Graphics

Use [F9] or Copy Image on the Edit menu to copy a HyperChem display to the Clipboard or a file. The Clipboard is a temporary storage place that can hold one graphic at a time. You can then use this graphic in a publication, presentation, or report.

### Setting Up a Graphics Transfer

Choose Setup Image on the Edit menu to set up the conditions for transferring graphics from HyperChem to the Windows Clipboard or to a file. In the Setup Image dialog box, you can choose part of the screen to copy, the type of image (bitmap or metafile), and the destination (Clipboard or file).

The settings in this dialog box regulate the results of pressing [F9], Copy Image.

### Choosing the Extent of the Image

To adjust the extent of the image, choose one of these from the Setup Image dialog box:

- |                    |  |
|--------------------|--|
| <b>Top level</b>   | This copies the active HyperChem window, any active pull-down menu, or any visible dialog box. You must use [F9] instead of Copy Image to save images of menus and dialog boxes. |
| <b>Workspace</b>   | This copies only the workspace of the active HyperChem window.   |
| <b>HyperChem</b>   | This copies the entire, active HyperChem window, including its borders.  |
| <b>Full screen</b> | This copies everything on screen, including any other windows.   |

This setting also affects the content of the image:

- |               |   |
|---------------|---|
| <b>Cursor</b> | Check this box (5) to include the cursor in the image. The cursor can be the mouse pointer, a HyperChem tool, or a text cursor (in a dialog box). |
|---------------|---|

## Choosing File Formats

HyperChem can store images in two types of files, either bit-mapped (BMP) or Windows metafile (WMF), or both. These are standard Windows file types that many other programs can use. Check the manual for each program to see if it can accept BMP or WMF graphics files. The commands in other programs for incorporating these files are often called Import or Place.

Choose from the following file formats:

- |                 |   |
|-----------------|---|
| <b>Bitmap</b>   | This saves the graphic as a reproduction of the pixels on screen, in the file name by Filename.bmp as described below. The resolution of this image matches the resolution of your monitor, which you determine when setting up Windows. Use Bitmap to save Spheres renderings or contour plots from a single point, semi-empirical calculation (see page 399).   |
| <b>Metafile</b> | This saves the graphic as a line drawing, in the file Filename.wmf. Metafile is best for Sticks, Ball and Cylinder, Dots, and Sticks & Dots renderings. Metafiles require about 10 times less storage space than bitmap files containing the same renderings. The resolution of a metafile graphic does not depend on the monitor, but on the resolution of the output device. For example, if you print a metafile on a standard laser printer, the resolution might be 1200dpi. |

If you choose both Bitmap and Metafile, HyperChem saves the graphic in both types of file.

## Choosing the Destination

You can choose to store a graphic on the Windows Clipboard, in a file, or both:

- |             |  |
|-------------|--|
| <b>File</b> | HyperChem stores the graphic in the files Filename.bmp and Filename.wmf, or both. Each file can hold only one image at a time. |
|-------------|--|

- Clipboard** HyperChem stores the graphic either as a bitmap, or as a metafile on the Windows Clipboard. The Display menu in the Windows Clipboard program refers to Metafile as Picture. The Clipboard can hold only one graphic of each type at a time.
- Filename** This is where you give the path and root name of the file where either the bitmap or metafile will be stored when you request the image be stored in a file. The extension .bmp or .wmf will be added to Filename and should not be given explicitly. A default Filename will be suggested but can be changed as desired.

## Saving Color Images

Check the Color box (5) to store a graphic in full color. Turn Color off to produce a black-and-white picture and to substitute dotted lines for certain colored lines (see “Using Black-and-white Output” on page 105). For a BMP file format, color results in a file at least four times as large. For a WMF file, Color produces only a small difference in file size.

## Using Copy Image (or [F9])

After you set up the graphic that you want to save (see the previous section), press the function key [F9] or choose Copy Image on the Edit menu. Use [F9] if you want a graphic of a visible pull-down menu or dialog box. Since pulling down a menu or opening a dialog box usually requires the cursor, you cannot use the cursor to choose Copy Image.

## Using the Clipboard

HyperChem shares the Clipboard with other programs that work with Windows. From HyperChem you can copy a graphic to the Clipboard. Using the Paste command in another program, like Microsoft Word®, you can transfer graphics from the Clipboard into documents.

**Caution:** You cannot transfer HyperChem graphics to the Clipboard by using Cut or Copy on the HyperChem Edit menu. You

must use [F9] or Copy Image. Using Cut or Copy places a text description of the molecular system on the Clipboard.

Some programs that share the Clipboard with HyperChem (these are Windows Applications and Non-Windows Applications in the Windows Program Manager) might not be able to use either bitmap or metafile graphics directly from the Clipboard. These programs might require that you store graphics first in a BMP or WMF file (see the next section). For specific requirements, see the user manuals for each program.

Remember that HyperChem can store only one metafile and bitmap graphic at a time on the Clipboard. The next time you press [F9] to store graphics, you replace the previous graphic.

To collect several graphics, paste each one from the Clipboard into a file, using another program that accepts metafile or bitmap images. For example, a Windows Write file can store both bitmap and metafile graphics.

To see the contents of the Clipboard, double-click on the Clipboard icon in the Windows Program manager (Main Group). If you copied both metafile and bitmap images to the Clipboard, you see the metafile version. To see the bitmap version, choose Bitmap on the Clipboard Display menu.

## Using BMP or WMF Files

Most Windows Applications and many other programs can use graphics from metafiles (WMF files) by importing or placing them. An increasing number of programs can also use bitmap graphics from BMP files. For specific requirements, see the user manuals for each program.

Remember that HyperChem can store only one WMF and BMP graphic at a time, in the files name by Filename (probably chem.wmf and chem.bmp by default). The next time you press [F9] to store graphics in these files, you replace the previous graphic unless you change Filename.

## Printing a Molecular System

To print the molecular system in the workspace, choose Print on the File menu.

Microsoft Windows controls the type of printer and its connections. To check these settings, click on the Start button and then Settings/Printers.

Be sure that your printer is on and ready. HyperChem does not inform you of printer problems, but relies on Windows to manage printing.

*Note:* You can speed up printing by choosing Sticks for the Rendering option on the Display menu.

## Storing Results in a Log File

You can store the detailed results of chemical calculations in log files. The file stores all results that appear on the status line during and after the calculation, plus many of the initial conditions and final results. The amount of information placed in the log file depends on the values of Mechanics Print Level and Quantum Print Level. Experiment with values between 0 and 9 for these to obtain the printing you want.

*Note:* The maximum level of printing will not necessarily be obtained with the value 9 for these quantities.

Choose Start Log on the File menu to name a log file and begin recording results. The default filename is chem.log. If you use a name for an existing log file, HyperChem lets you add on to the same file (Append) or replace the contents of the file.

You can start only one log file at a time, but you can store any number of files with different names. When you start a log file, you can have available two new items on the File menu, Stop Log and Log Comments.

## Stopping a Log

Stop Log turns off the recording of a log file and adds the date and time you stopped recording.



To see or print the contents of a log, you can use a text editor such as Windows Notepad or Wordpad. Enter the filename (for example, \hyper7\chem.log), L-click on OK, and then choose No conversion in the next dialog box that appears.

## Adding Comments to a Log

Use Log Comments to add notes to a log file. You can use this item anytime a log file is open and recording results.

You see the Log Comments dialog box (see page 91) to enter your comments.

To add or change text:

- L-click with the text cursor (I-shaped) anywhere in the text to move the insertion point.
- L-click-drag across text to select text that you want to delete or replace.
- To delete text, select it and press [Del].
- To replace text, select it and start entering new text.

You can use these keys for editing:

[Return]	Moves the text cursor to the next line.
[Backspace]	Erases the character (or selected text) to the left of the insertion point (text cursor).
[Delete]	Erases the character (or selected text) to the right of the insertion point (text cursor).
[3] [4]	Moves the insertion point (text cursor).
[1] [2]	

## Storing Results of a Molecular Dynamics Simulation

You can store the progress of a molecular dynamics simulation in a snapshots file. This file contains a complete description of the molecular system for each snapshot. You can request snapshots at regular time intervals through the simulation. The snapshot file has the same name as the current HIN file. HyperChem also creates another file with the same name, but with the extension *.snp*. If

the HIN file is still untitled, HyperChem uses chem.snp. The snapshot file is pointed to by a record in the accompanying HIN file.

When you run a molecular dynamics simulation, you can average up to eight values (see “Molecular Dynamics Averages Dialog Box” on page 377). The instantaneous values and averages are stored in a file that has the extension .csv (see “Plot and Averages Files” on page 629). You can use a text editor to view these files or you can transfer these values to another program for analysis. The file has the same name as the current HIN file, or chem.csv if the HIN file is still untitled.

## Analyzing HyperChem Results

You can transfer many of the results of HyperChem calculations to other Windows applications; for example, to Microsoft Excel for analysis and plotting results. You can do this transfer by using the Windows Clipboard, usually by selecting text by dragging the mouse cursor over it and pressing [Ctrl]+[C] to copy to the Clipboard, and then moving to the new application and pressing [Ctrl]+[V] or Paste on an Edit menu to add the information to it or by Windows Dynamic Data Exchange (see “Dynamic Data Exchange (DDE)” on page 604).

# Appendix A

## HyperChem Files

HyperChem includes these files:

File Type	Filenames	Refer to:
Force field calculations	chem.rul typerule.bin *.txt *.dbf *.par	Appendix B
Semi-empirical calculation	*.abp	Appendix C
<i>Ab initio</i> or <i>DFT</i> calculations	*.bas	Appendix C
HyperChem Input (data storage)	*.hin	Appendix D
Residue templates	chem.tpl chemct.tpl chemnt.tpl	Appendix E
HyperChem default settings	Registry	Appendix F
Water for periodic boxes	wat216c.bin wat5832c.bin	This chapter

File Type	Filenames	Refer to:
HyperChem program files	chem.exe mmplus.exe newton.exe eht.exe ndo.exe tndo.exe gauss.exe dft.exe tclsh80.exe wish80.exe	This chapter
Essential DLL files	chemstr.dll hcqmart.dll	This chapter
Module executable files	HCConformation.exe HCCrystal.exe HCDData7.exe HCNMR7.exe HCPolymer.exe HCQsar.exe HCRay.exe HCRmsfit.exe HCScript.exe HCSequence.exe HCSugar.exe NMRCalc.exe	This chapter
Help text	chem.hlp	This chapter
Script files	chem.scr *.scr *.tcl	This chapter
Snapshot Files	chem.snp *.snp	This chapter
Plot files	chem.csv *.csv	This chapter
Graphics files	*.wmf *.bmp	This chapter

File Type	Filenames	Refer to:
Log files	chem.log *.log	This chapter

## Water Files

HyperChem uses TIP3P models of water molecules (Jorgenson *et al.*, *J. Chem. Phys.* 79(2):926-935, 1983) equilibrated at 300 K (25°C) and one atmosphere to solvate a molecular system in a periodic box. The binary file wat216c.bin provides up to 216 water molecules, enough to fill a standard-sized periodic box 18.622 Ångstroms on a side. The file wat5832c.bin provides up to 5832 water molecules, enough to fill a box three times the standard size (55.866 Ångstroms on a side). Wat5832c.bin contains the same contents as wat216c.bin, repeated 27 times.

## Program Files

These are the HyperChem program files:

chem.exe	Code for the HyperChem user interface, including the Model Builder, all display functions, and preparations for chemical calculations. This is the HyperChem “frontend,” designed to run under Windows on a PC. The other program files constitute the “backend,” which can run on a PC or another processor.
mmplus.exe	Code for calculations using the MM+ force field.
newton.exe	Code for calculations using AMBER, BIO+, and OPLS force fields.
eht.exe	Code for calculations using the Extended Hückel method.
ndo.exe	Code for calculations using CNDO, INDO, MINDO3, MNDO, AM1, PM3, ZINDO/1, and ZINDO/S methods.
tnndo.exe	Code for calculations using TNDO method.

<code>gauss.exe</code>	Code for calculations using <i>ab initio</i> methods.
<code>dft.exe</code>	Code for calculations using density functional theory.

You cannot modify HyperChem program files. You can complement HyperChem functions by developing programs that interact with HyperChem through DDE links (see “Dynamic Data Exchange (DDE)” on page 604).

## Essential DLL Files

HyperChem contains a number of DLL files that accompany the executable files. The DLL's, apart from those involved with licensing, that are most essential are:

<code>chemstr.dll</code>	Contains strings
<code>hcqmart.dll</code>	Contains code for quantum mechanical expectation values

## Module Executable Files

The files of this type listed above contain executable code associated with modules that interact with chem.exe.

## Help Text File

All Help text available on screen is in one file, `chem.hlp`. You can read this text using the Help menu, by activating the Help window with [F1], or by activating the Help cursor (context-sensitive help) with [Shift]+[F1].

## Script Files

Script files contain instructions for running HyperChem. You can write Script files (see “Scripts” on page 602) using a text editor, like Windows Notepad. These files can have any filename or extension, but `.scr` is recommended for scripts in the HyperChem Command Language and `.tcl` for scripts in the Tool Command Language. You can use the Script Editor available on the Script menu to assist in

writing scripts (HyperChem Command Language only). You can run scripts using the Open Script command on the Script menu (see page 105). The HyperChem Chemist's Develop Kit Manual contains a full description of the scripting capability within HyperChem.

## Snapshot Files

During a molecular dynamics simulation, HyperChem can store the trajectory of a molecular system at regular, timed intervals, as snapshots. These results go into both the HIN file and an SNP file, with the extension *.snp*. The SNP file normally has the same name as the HIN file. If there is not yet a HIN file, the filename is *chem.snp*.

You can playback the snapshots using HyperChem, to continue analysis of the simulation. You can also use other applications to modify or analyze the data in the SNP file.

A binary SNP file is composed of three sections:

1. Fixed sized header that contains calculation parameters
2. Variable sized header that contains atom information
3. Coordinate and velocity data.

*Note:* Boolean values are 1 for true, 0 for false. All values are four bytes long.

### Fixed Sized Header

Type	Address	Description
int	0-3	Number of snapshots taken
int	4-7	File offset (in bytes) to first record of coordinate and velocity data
int	8-11	Number of atoms in snapshot
int	12-15	Snapshot period (in data collection units)

Type	Address	Description
	16-19	Reserved
Boolean	20-23	True if restraints were used
int	24-27	Optimizer type: 0: Steepest Descent 1: Fletcher-Reeves 2: Polak-Ribiere 3: Newton-Raphson
	28-47	Reserved
Boolean	48-51	True if periodic boundary conditions were used
Boolean	52-55	Constant temperature?
real	56-59	Temperature relaxation time for coupling to bath (picoseconds)
	60-63	Reserved
Boolean	64-67	Previously saved velocities used (for restart)?
int	68-71	Random number seed
real	72-75	X dimension of periodic box (Ångstroms)
real	76-79	Y dimension of periodic box (Ångstroms)
real	80-83	Z dimension of periodic box (Ångstroms)
	84-95	Reserved
real	96-99	Starting temperature (K)
real	100-103	Peak temperature (K)
real	104-107	Low temperature (K)



Type	Address	Description
real	108-111	Heat time (picoseconds)
real	112-115	Run time (picoseconds)
real	116-119	Cool time (picoseconds)
real	120-123	Time step (picoseconds)
real	124-127	Temperature step (K)
	128-159	Reserved
int	160-163	Data collection period (in time steps)
int	164-167	Warning level: 0: None 1: Log file 2: Log file and message box
int	168-171	Number of atoms in snapshot

## Variable Sized Header

The following header information is repeated for each atom. This data is sequential. Offsets are given relative to the start of each atom's header.

Type	Address	Description
int	0-3	Index of atom (starting from 0)
int	4-7	Atomic number (0-103, 0 for lone pair)
int	8-11	Atom type index
	12-15	Reserved

Type	Address	Description
int	16-19	True if atom is fixed
	20-23	Reserved
int	24-27	Coordination (1 to 12 neighbors)
int	28-31	Index of first neighbor (-1 for none)
int	32-35	Index of second neighbor (-1 for none)
int	36-39	Index of third neighbor (-1 for none)
int	40-43	Index of fourth neighbor (-1 for none)
int	44-47	Index of fifth neighbor (-1 for none)
int	48-51	Index of sixth neighbor (-1 for none)
int	52-55	Index of seventh neighbor (-1 for none)
int	56-59	Index of eighth neighbor (-1 for none)
int	60-63	Index of ninth neighbor (-1 for none)
int	64-67	Index of tenth neighbor (-1 for none)
int	68-71	Index of eleventh neighbor (-1 for none)
int	72-75	Index of twelfth neighbor (-1 for none)
real	76-79	Charge
real	80-83	X coordinate
real	84-87	Y coordinate
real	88-91	Z coordinate

Type	Address	Description
real	92-95	Mass

## Coordinate and Velocity Data

The data from the simulation follows immediately after the variable sized header, as a series of records with one record per recorded timestep. Each record contains the coordinates of each atom (sequentially), the components of the atom's velocity, and a fixed sized trailer containing dynamics information.

The information repeated for each atom follow. Offsets are relative to each atom's data.

Type	Address	Description
real	0-3	X coordinate
real	4-7	Y coordinate
real	8-11	Z coordinate
real	12-15	X component of velocity (Ångstroms/picosecond)
real	6-19	Y component of velocity (Ångstroms/picosecond)
real	20-23	Z component of velocity (Ångstroms/picosecond)

The trailer for each record follows. Offsets are relative to the start of the trailer.

Type	Address	Description
real	0-3	Total energy

Type	Address	Description
real	4-7	Stretching energy
real	8-11	Bending energy
real	12-15	Torsional energy
real	16-19	Van der Waals energy
	20-23	Reserved
real	24-27	Electrostatic energy
	28-35	Reserved
real	36-39	1-4 van der Waals energy
real	40-43	1-4 electrostatic energy
	44-63	Reserved
real	64-67	Semi-empirical binding energy
real	68-71	Semi-empirical atomic energy
real	72-75	Semi-empirical electronic energy
real	76-79	Semi-empirical core energy
real	80-83	Semi-empirical heat of formation
	84-103	Reserved
real	104-107	Total dipole moment
real	108-111	X component of dipole moment
real	112-115	Y component of dipole moment

Type	Address	Description
real	116-119	Z component of dipole moment
real	120-123	RMS gradient
real	124-127	Temperature (K)
real	128-131	Potential energy
real	132-135	Kinetic energy
	136-143	Reserved
real	144-147	Elapsed time (picoseconds)
	148-171	Reserved

## Plot and Averages Files

If you choose to average (or average and plot) values during a molecular dynamics calculation (see “Molecular Dynamics Averages Dialog Box” on page 377), HyperChem saves the instantaneous values and RMS deviations ( $Dx$ )<sup>1</sup> in a file with the same name as the HIN file, but with the extension *.csv*. If the molecular system is not yet associated with a HIN file, HyperChem stores the values in *chem.csv*. The file is saved to the current working directory, which you must be able to write to (*e.g.*, not on a CD-ROM).

CSV files provide the information for recreating a plot of the stored values. To do this, you must run the molecular dynamics calculation with Snapshots on (see “Molecular Dynamics Options Dialog Box” on page 369). When you “Playback” the snapshots (see page 369), you can plot the stored values or choose, from the

1.  $Dx = \sqrt{\langle x_i^2 \rangle - \langle x_i \rangle^2}$ , where  $x_i$  =instantaneous value of EKIN, ETOT, EPOT, or TEMP,

$$\langle x_i^2 \rangle = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right), \text{ and } \langle x_i \rangle^2 = \left( \frac{1}{n} \sum_{i=1}^n x_i \right)^2.$$

Molecular Dynamics Averages dialog box (see page 377), a subset of the original values or their deviations.

*Example:*

If you average total energy (ETOT) and the RMS deviation for total energy (D ETOT), the CSV file contains this type of data:

```
time, ETOT, D ETOT
0,10.3854,5.19271
0.001,9.73882,6.70808
0.002,9.23686,7.34028
0.003,8.96351,7.34028
. . .
0.999,10.0421,10.5366
1,10.3975,10.5365
```

The first line is the header, then there is a line for each time point. ETOT (in the second column) contains the instantaneous values of ETOT. D ETOT (in the third column) contains the instantaneous RMS deviations.

You can read these files into another program, like a spreadsheet or plotting program, that can use data in a comma-separated-value (csv) format.

## Graphics Files

The graphics files Filename.wmf or Filename.bmp (default names chem.wmf and chem.bmp) contain images of the screen obtained with Copy Image on the Edit menu (see page 206).

## Metafile Graphics

Filename.wmf stores an image as instructions for a line drawing. Metafile is best for Sticks, Balls and Cylinders, Dots, and Sticks & Dots renderings. Metafiles require about 10 times less storage space than bitmap files containing the same renderings. The resolution of a metafile image does not depend on the monitor, but on the resolution of the output device. For example, if you print a metafile on a laser printer, the resolution might be 300 dpi.

## Bitmapped Graphics

Filename.bmp stores an image as a reproduction of the pixels on screen. The image resolution is the same as the resolution of your monitor, which you choose when setting up Windows. Use Bitmap to save Spheres renderings or contour plots from a single point, semi-empirical calculation (see “Single Point” on page 360).

## Log Files

HyperChem accumulates the results of calculations in a log file if you choose Start Log on the File menu. In the Open Log dialog box (see page 90), you can give a log file any filename or extension, but the extension *.log* is recommended.

You can save information about a calculation by opening a log file using Start Log on the File Menu. The content and amount of information saved in the log file is determined by settings that you use for the Mechanics Print Level and Quantum Print Level entries in the Start Log dialog box or in the Registry.

This appendix describes the information that is saved when you use different settings for these entries. The settings range from 0–9. Generally, we recommend that you use a setting of 1–3. If you use a setting of 4–9 for either entry, a large amount of information is written into the log file. The log file can take up many megabytes of disk space if you use a setting of 4–9 during a geometry optimization.

## Extended Hückel Calculations

QuantumPrintLevel	Information saved in log file
0	start log time stamp type of calculation total number of electrons total number of orbitals doubly and singly occupied orbitals total energy (kcal/mol and a.u.) eigenvalues atomic orbital electron populations net charges and coordinates
1	All information when QuantumPrintLevel=0 orbital eigenvalues (energies), occupancies and eigenvectors (wavefunctions)
2	All information when QuantumPrintLevel=0 total density matrix
3	All information when QuantumPrintLevel is 0, 1, or 2
4, 5, 6, 7, 8, or 9	All information when QuantumPrintLevel is 0, 1, or 2 overlap matrix core Hamiltonian matrix product of density matrix and overlap matrix



## NDO Calculations

The following table shows the information saved in the log file when you run calculations using NDO methods (all methods in the Semi-empirical Method dialog box, except Extended Hückel).

QuantumPrintLevel	Information saved in log file
0	start log time stamp type of calculation total number of electrons total number of orbitals double and single occupied orbitals in RHF, or occupied alpha and beta orbitals in UHF occupied and unoccupied orbitals when using orbital criterion <sup>a</sup> maximum excitation energy when using energy criterion <sup>a</sup> iteration list total energy (kcal/mol and eV) binding energy isolated atomic energy electronic energy core-core interaction number of configurations heat of formation gradient of reference configuration eigenvalues of reference configuration atomic orbital electron populations net charges and coordinates dipoles lowest 10 state energies, multiplicities, and spins <sup>b</sup> CI energy <sup>b</sup> UV spectra (frequencies, intensities, state dipole moments, etc.) <sup>a</sup> IR spectra (frequencies, intensities, etc.) <sup>c</sup>
1	All information when QuantumPrintLevel=0 orbital eigenvalues (energies), occupancies and eigenvectors (wavefunctions)
2	All information when QuantumPrintLevel=0 density matrix in RHF or density matrices in UHF

QuantumPrintLevel	Information saved in log file
3	All information when QuantumPrintLevel is 0, 1, and 2
4	All information when QuantumPrintLevel=0 overlap Hamiltonian matrix Fock matrices, eigenvalues, eigenvectors, density matrices for each interaction
5	All information when QuantumPrintLevel=0 Slater determinants (microstates) <sup>a</sup>
6	All information when QuantumPrintLevel=0 eigenvalues and eigenvectors of CI secular equation
7	All information when QuantumPrintLevel is 0, 5, and 6
8	All information when QuantumPrintLevel is 0, 3, 5, and 6
9	All information when QuantumPrintLevel is 0 and 3 frequencies and eigenvectors of mass-weighted force matrix (normal modes) <sup>c</sup>

- a. Applies only when you do a singly excited CI calculation.  
b. Applies only when you do a microstate CI calculation.  
c. Applies only when you do a vibrational analysis calculation.

## ***Ab initio* calculations**

This section describes the information that is saved when you use the *ab initio* quantum mechanics method.

Quantum PrintLevel	Information saved in log file
0	start log time stamp; starting options: convergence limit iteration limit accelerate convergence flag method for initial guess of MO basis set type RHF or UHF spin-multiplicity extra charge on the system total number of electrons; total number of orbitals (basis functions); total number of primitive Gaussians; double and single occupied orbitals in RHF, or occupied alpha and beta orbitals in UHF; occupied and unoccupied orbitals when using orbital criterion; <sup>a</sup> iteration list; total energy (kcal/mol and a.u.); total kinetic energy (kcal/mol and a.u.); the Virial factor; electron-electron and electron nuclear interaction energy; nuclear repulsion energy; number of configurations; gradient or gradient of reference configuration; <sup>a</sup> eigenvalues or eigenvalues of reference configuration; <sup>a</sup> atomic orbital electron populations net charges and coordinates; dipole moments; quadrupole moments; octapole moments; hexadecapole moments; components of first derivatives; UV spectra (frequencies, intensities, state dipole moments, etc.); <sup>a</sup> IR spectra (frequencies, intensities, etc.) <sup>b</sup>

Quantum PrintLevel	Information saved in log file
1	All information when QuantumPrintLevel=0; input basis function information; orbital eigenvalues (energies) and eigenvectors (wavefunctions); Mulliken population analysis.
2	All information when QuantumPrintLevel=0; input basis function information; density matrix in RHF or density matrices in UHF; Mulliken population analysis.
3	All information when QuantumPrintLevel is 0, 1, and 2; initial guess of MO coefficients.
4	All information when QuantumPrintLevel=0; core Hamiltonian matrix; initial guess of MO coefficients; Fock matrices, eigenvalues, eigenvectors, density matrices for each interaction; Mulliken population analysis.
5	All information when QuantumPrintLevel=0; input basis function information; all the two-electron repulsion integrals; Mulliken population analysis.
6	All information when QuantumPrintLevel=0; input basis function information; initial guess of MO coefficients; Slater determinants; <sup>a</sup> Mulliken population analysis.
7	All information when QuantumPrintLevel=0; input basis function information; initial guess of MO coefficients; eigenvalues and eigenvectors of singly excited CI matrix; <sup>a</sup> Mulliken population analysis.

---

Quantum PrintLevel	Information saved in log file
8	All information when QuantumPrintLevel=0; input basis function information; initial guess of MO coefficients; Slater determinants; <sup>a</sup> eigenvalues and eigenvectors of singly excited CI matrix; <sup>a</sup> eigenvalues, eigenvectors and density matrix of the reference configuration in the SCF; Mulliken population analysis.
9	All information when QuantumPrintLevel=0; input basis function information; initial guess of MO coefficients; eigenvalues, eigenvectors, and density matrix; Mulliken population analysis; frequencies and eigenvectors of mass-weighted force matrix (normal modes) <sup>b</sup>

---

**a. Applies only when you do a singly-excited CI calculation**  
**b. Applies only when you do vibrational analysis calculation**

## Molecular Mechanics Calculations

The following table shows the information saved in the log file when you run a Single Point calculation using the MM+ force field.

MechanicsPrintLevel	Information saved in log file
0	start log time stamp type of calculation energy components
1	All information when MechanicsPrintLevel is 0 center of mass force-field coordinates principal moments of inertia bond length and stretching energy bond angles and bending energy bond torsions and torsional energy cubic stretch constant stretching parameters equilibrium bond length force constant dipole interaction energy
2	All information when MechanicsPrintLevel is 1 dielectric constant dipole moment
3	All information when MechanicsPrintLevel is 2 van der Waals energy

The following table shows the information saved in the log file when you run other molecular mechanics calculations.

MechanicsPrintLevel	Information saved in log file
0-9	start log time stamp type of calculation energy components

## Annotated Example Log File Output

A user can get extra information about a calculation by saving and reading a log file. Below we describe the information that is put in the log file for representative calculations with MM+, AMBER, Extended Hückel and CNDO. The example is an ethane molecule calculation with single point, optimization, and five times steps of dynamics where appropriate. For CNDO a configuration interaction calculation is also shown.

Annotations to the log files are shown in brackets, as in <annotation>.

### MM<sup>+</sup>

<A time stamp for the beginning of logging is added automatically to the log file.>

```
HyperChem log start -- Sat Jan 25 22:45:54 1992
```

<The following comments below were added via the File/Log Comments command.>

```
Begin comments added by user prior to calculations
```

1. Single point
2. Geometry Optimization
3. 0.005 picoseconds of MD

```
End comments
```

<Log file entries below were triggered by performing a single point calculation using MM+ as the force field.>

```
Single Point, MolecularMechanics, molecule = ETHANE.HIN.
```

<The following section of output is consistent with the usual MM+ output. HyperChem does not make use of these coordinates and they are given only for comparison with MM2 output from Allinger's program. The molecule is aligned and translated to the center of mass to give the following coordinates and to calculate the principal moments of inertia. HyperChem retains the coordinates as they are on the screen for continuity purposes and the following coordinates are not those used by HyperChem. The internal coordinates are, of course, identical to those used by HyperChem. The units here for the principal moments of inertia are also different than those used by HyperChem, associated with request-

ing the display of the principal inertial axes. There, moments of inertia are given in amu-Ångstroms<sup>2</sup>.>

```
Center of Mass
x = -0.53797 y = 0.29747 z = 0.00000
MM+ coordinates of an aligned molecule at center of mass.
C4( 1) -0.77000 0.00000 0.00000 ( 1) <last column is
the MM+ type>
C4( 2) 0.77000 0.00000 0.00000 ( 1)
H( 3) -1.13333 -0.01367 1.02757 ( 5)
H( 4) -1.13333 0.89674 -0.50194 ( 5)
H( 5) -1.13334 -0.88306 -0.52563 ( 5)
H( 6) 1.13333 0.01367 -1.02757 ( 5)
H( 7) 1.13333 0.88307 0.52562 ( 5)
H( 8) 1.13334 -0.89674 0.50195 ( 5)
Principal moments of inertia
(units = 10**(-39)*gm*cm**2)
ix = 1.0605 iy = 4.1827 iz = 4.1827.
```

<Below, the results of bond stretching calculations are reported.>

```
Bond lengths and stretching energy (7 bonds)
Energy = 71.94(Kr)(dr)(dr)(1+(cs)(dr))
dr = R-R0
cs = -2.000 <cubic stretch constant>
```

<The values below are all the stretches. R0 and Kr are the stretching parameters — equilibrium bond length and force constant.>

Bond	R	R0	Kr	Energy
C4( 1) - C4( 2)	1.5400	1.5230	4.4000	0.0884
C4( 1) - H( 3)	1.0900	1.1130	4.6000	0.1831
C4( 1) - H( 4)	1.0900	1.1130	4.6000	0.1831
C4( 1) - H( 5)	1.0900	1.1130	4.6000	0.1831
C4( 2) - H( 6)	1.0900	1.1130	4.6000	0.1831
C4( 2) - H( 7)	1.0900	1.1130	4.6000	0.1831
C4( 2) - H( 8)	1.0900	1.1130	4.6000	0.1831

```
Dipole Interaction energy (Dielectric constant = 1.500000)
Dipole(1) Mu(1) Dipole(2) Mu(2) R12 E(kcal)
```

<These are no dipole terms for ethane so no output appears here. Normally, the individual dipole interaction terms appear.>

```
Dipole Moment: x=0.000000 y=0.000000 z=0.000000 total=0.000000
```

<The Van der Waals calculations start here.>

```
Van der Waals energy
9 vdw interactions (1,3 excluded)
```



```

Energy = Kv*(2.90(10**5)exp(-12.50/P) - 2.25(P**6))
Rv = Rvdw(i) + Rvdw(j)
Kv = sqrt(Eps(i)*Eps(j))
P = (Rv/R) or (Rv/R#)
(if P.gt.3.311, Energy = Kv(336.176)(P**2))
# in the vdw calculations the hydrogen atoms are relocated
so that the attached hydrogen distance is reduced by 0.915

```

<Asterisk denotes a vicinal interaction, R# is the distance between the relocated centers. The parameters are Rv and Kv.>

Atom Pair	R	R#	Rv	Kv	Energy	(1,4)
H( 3), H( 6)	3.0598	2.8980	3.000	0.0470	-0.0524	*
H( 3), H( 7)	2.4887	2.3970	3.000	0.0470	0.2201	*
H( 3), H( 8)	2.4887	2.3970	3.000	0.0470	0.2201	*
H( 4), H( 6)	2.4887	2.3970	3.000	0.0470	0.2201	*
H( 4), H( 7)	2.4887	2.3970	3.000	0.0470	0.2201	*
H( 4), H( 8)	3.0598	2.8980	3.000	0.0470	-0.0524	*
H( 5), H( 6)	2.4887	2.3970	3.000	0.0470	0.2201	*
H( 5), H( 7)	3.0598	2.8980	3.000	0.0470	-0.0524	*
H( 5), H( 8)	2.4888	2.3970	3.000	0.0470	0.2201	*

<Bending calculations start here, including stretch-bend interactions.>

```

Bond angles, bending and stretch-bend energies ( 12 angles)
Eb = 0.043828(Kb/2)(dt)(dt)(1+sf*dt**4)
dt = Theta-Theta0
sf = 0.00700e-5 <the sextic scale factor>
Esb = 2.51118 (Ksb)(dt)(dr1+dr2)
dr(i) = R(i) - R0(i) <the two bond deformations involved in the angle>
(x,y) = f or s

```

<Four built-in stretch-bend interactions.>

```

Ksb(1) = 0.120 x-f-y f = 1st row atom
Ksb(2) = 0.250 x-s-y s = 2nd row atom
Ksb(3) = 0.090 x-f-h (dr2 = 0)
Ksb(4) = -0.400 x-s-h (dr2 = 0)

```

<The bending interactions are listed below.>

Atoms	Theta	Theta0	Kb	Eb	Ksb	Esb
C4( 2)-C4( 1)- H(3)	109.471	110.0000	0.090	-0.0020	0.360	0.0022
C4( 2)-C4( 1)- H(4)	109.471	110.0000	0.090	-0.0020	0.360	0.0022
C4( 2)-C4( 1)- H(5)	109.471	110.0000	0.090	-0.0020	0.360	0.0022
H( 3)-C4( 1)- H(4)	109.471	109.0000	0.320	0.0016		
H( 3)-C4( 1)- H(5)	109.471	109.0000	0.320	0.0016		

```

H( 4)-C4( 1)- H(5) 109.471 109.0000.320 0.0016
C4( 1)-C4( 2)- H(6) 109.471 110.0000.090 -0.0020 0.360 0.0022
C4( 1)-C4( 2)- H(7) 109.471 110.0000.090 -0.0020 0.360 0.0022
C4( 1)-C4( 2)- H(8) 109.471 110.0000.090 -0.0020 0.360 0.0022
H( 6)-C4( 2)- H(7) 109.471 109.0000.320 0.0016
H( 6)-C4( 2)- H(8) 109.471 109.0000.320 0.0016
H( 7)-C4( 2)- H(8) 109.471 109.0000.320 0.0016

```

<Torsion calculations start here.>

Dihedral angles, torsional energy (Et) ( 9 angles)  
 $Et = (V1/2)(1+\cos(w))+(V2/2)(1-\cos(2w))+(V3/2)(1+\cos(3w))$   
 Sign of angle a-b-c-d is negative, when looking through b toward c,  
 if d is counterclockwise from a.

<The torsional interactions are listed below.>

Atoms	omega	V1	V2	V3	et
H( 3)C4( 1)C4( 2) H( 6)	180.000	0.000	0.000	0.237	0.000
H( 3)C4( 1)C4( 2) H( 7)	60.001	0.000	0.000	0.237	0.000
H( 3)C4( 1)C4( 2) H( 8)	60.000	0.000	0.000	0.237	0.000
H( 4)C4( 1)C4( 2) H( 6)	60.001	0.000	0.000	0.237	0.000
H( 4)C4( 1)C4( 2) H( 7)	59.999	0.000	0.000	0.237	0.000
H( 4)C4( 1)C4( 2) H( 8)	179.999	0.000	0.000	0.237	0.000
H( 5)C4( 1)C4( 2) H( 6)	60.000	0.000	0.000	0.237	0.000
H( 5)C4( 1)C4( 2) H( 7)	179.999	0.000	0.000	0.237	0.000
H( 5)C4( 1)C4( 2) H( 8)	60.001	0.000	0.000	0.237	0.000

Total Energy=2.360644 Gradient=11.839364.

<Below is a summary of the interaction terms in a single point calculation.>

Bond=1.18704 Angle=0.0225845 Dihedral=5.607e-010 Vdw=1.16321  
 Stretch-bend=-0.0121913 Electrostatic=0.

<Log file entries below were triggered by a geometry optimization calculation.>

Geometry optimization, MolecularMechanics, molecule = ETHANE.HIN.  
 mm+  
 NewtonRaphson optimizer

<First line comes from back end, second from front end. GradientSquared is the sum of the squares of all gradient components, while Gradient reported below is the RMS gradient (first number divided by 24 for ethane and the square root taken).>

Iter= 1 Energy=1.13315 AvgMovement= 0.01131  
 GradientSquared=1858.67369.

Energy=1.133152 Gradient=8.800269 Converged=NO (1 cycles 1 points).  
 Iter= 2 Energy=0.81988 AvgMovement= 0.00485 GradientSquared=  
 571.54119.  
 Energy=0.819878 Gradient=4.879981 Converged=NO (2 cycles 2 points).  
 Iter= 3 Energy=0.81615 AvgMovement= 0.00074 GradientSquared=  
 6.27436.  
 Energy=0.816150 Gradient=0.511304 Converged=NO (3 cycles 3 points).  
 Energy converged.  
 Energy=0.816150 Gradient=0.061118 Converged=YES (4 cycles 4 points).  
 Bond=0.027583 Angle=0.0991741 Dihedral=0.000255269 Vdw=0.677749  
 Stretch-bend=0.0113892 Electrostatic=0.

**<Log file entries below were triggered by molecular dynamics calculation.>**

Molecular dynamics, MolecularMechanics, molecule = ETHANE.HIN.  
 mm+  
 MechanicsData: RunTime=0.005000 ps SimulationTemperature=300.000000 K  
 TimeStep=0.001000 ps StepsToRefresh=1.  
 Time=0 ps Potential=0.815953 Kinetic=7.15412 TotalEnergy=7.97008  
 kcal/mol T=300.01 K.  
 Time=0.001 ps Potential=1.93134 Kinetic=6.17928 TotalEnergy=8.11062  
 kcal/mol T=259.13 K.  
 Time=0.002 ps Potential=4.17453 Kinetic=4.30619 TotalEnergy=8.48072  
 kcal/mol T=180.581 K.  
 Time=0.003 ps Potential=5.00905 Kinetic=3.58626 TotalEnergy=8.59531  
 kcal/mol T=150.391 K.  
 Time=0.004 ps Potential=3.82956 Kinetic=4.4882 TotalEnergy=8.31776  
 kcal/mol T=188.214 K.  
 Time=0.005 ps Potential=2.8143 Kinetic=5.29967 TotalEnergy=8.11396  
 kcal/mol T=222.243 K.

**<Time stamp triggered by Stop Log menu item.>**

HyperChem  
 log stop -- Sat Jan 25 22:48:44 1992

## AMBER

**<The log file of a HyperNewton calculation contains basically the same information as that of a HyperMM+ calculation except that the detailed description of individual interactions given for MM+, for consistency with Allinger's MM2, are not given here.>**

HyperChem log start -- Tue Dec 17 23:54:09 1991  
 Single Point, MolecularMechanics, molecule = 'ETHANE.HIN'  
 amber

```

Total Energy=0.217552 Gradient=2.514611
bond=0.060759 angle=0.000105982 dihedral=6.86196e-010 vdw=0.156687
H-bond=0 electrostatic=0
Geometry optimization, MolecularMechanics, molecule = 'ETHANE.HIN'
amber
PolakRibiere optimizer
Energy=0.176069 Gradient=0.465800 Converged=NO (1 cycles 4 points)
Energy=0.171165 Gradient=0.120952 Converged=NO (2 cycles 7 points)
Energy=0.169494 Gradient=0.013383 Converged=NO (3 cycles 12 points)
Energy=0.169494 Gradient=0.013383 Converged=YES (3 cycles 12 points)
bond=0.00150512 angle=0.00357023 dihedral=4.23819e-010 vdw=0.164419
H-bond=0 electrostatic=0
Molecular dynamics, MolecularMechanics, molecule = 'ETHANE.HIN'
amber
MechanicsData: RunTime=0.005000 ps SimulationTemperature=300.000000 K
TimeStep=0.001000 ps StepsToRefresh=1
Time=0 ps Potential=0.169494 Kinetic=7.15408 TotalEnergy=7.32357
kcal/mol T=300.008 K
Time=0.001 ps Potential=1.23804 Kinetic=6.2294 TotalEnergy=7.46744
kcal/mol T=261.231 K
Time=0.002 ps Potential=3.37796 Kinetic=4.36028 TotalEnergy=7.73824
kcal/mol T=182.849 K
Time=0.003 ps Potential=4.56812 Kinetic=3.28064 TotalEnergy=7.84877
kcal/mol T=137.574 K
Time=0.004 ps Potential=3.9793 Kinetic=3.71141 TotalEnergy=7.69071
kcal/mol T=155.639 K
Time=0.005 ps Potential=2.77903 Kinetic=4.66901 TotalEnergy=7.44804
kcal/mol T=195.796 K
HyperChem
log stop -- Tue Dec 17 23:55:48 1991

```

## Extended Hückel

<The results from Extended Hückel Theory calculations on model-built ethane appear below. Only a single point calculation is performed as geometry optimization and molecular dynamics are undefined for EHT.>

```

HyperChem log start -- Sun Jan 26 19:39:04 1992
Single Point, SemiEmpirical, molecule = ETHANE.HIN.
ExtendedHuckel
Extended-Huckel Calculation:
Singlet State Calculation
Number of Electrons = 14
Starting Extended-Huckel calculation with 14 orbitals
Charge on the System = 0

```

Total Orbitals = 14  
 Number of Double Occupied Levels = 7  
 Unweighted Huckel Constant = 1.7500000  
 d Orbitals on Si...Cl If Any Not Included  
 Energy=-5724.018299

<The logging of results corresponds to the smallest PrintLevel=0.>

----- RESULTS -----

<Total energies in EHT are sums of occupied orbital energies.>

Sum of One-Electron Energies = -5724.0182994 (kcal/mol)  
 Sum of One-Electron Energies = -9.122089647 (a.u.)

EIGENVALUES(eV) <These are the orbital energies, lowest energy orbitals occupied first.>

-26.485626	-21.621887	-16.121031	-16.121025	-14.782691
-14.782681	-14.191090	2.383136	3.373500	3.373533
7.076813	7.076891	24.606670	40.313354	

ATOMIC ORBITAL ELECTRON POPULATIONS <Number of electrons occupying each atomic orbital>

1.224238	0.950610	0.951825	0.950609	1.224237
0.950609	0.951825	0.950609	0.974238	0.974227
0.974254	0.974242	0.974225	0.974252	

NET CHARGES AND COORDINATES <Charge is total charge on an atom>

Atom	Z	Charge	Coordinates(a.u.)		
			x	y	z
1	6	-0.077281	-1.01663	-0.89296	0.00000
2	6	-0.077281	-1.01663	2.01722	0.00000
3	1	0.025762	0.92538	-1.57955	0.00000
4	1	0.025773	-1.98761	-1.57955	1.68183
5	1	0.025746	-1.98763	-1.57956	-1.68182
6	1	0.025758	-2.95863	2.70382	0.00000
7	1	0.025775	-0.04564	2.70381	1.68183
8	1	0.025748	-0.04562	2.70383	-1.68182

HyperChem

log stop -- Sun Jan 26 19:39:54 1992

## CNDO

<The following is the result for a successive single point calculation, a geometry optimization and .005 picoseconds of molecular dynamics, for CNDO ethane starting at model builder geometry.>

These HyperChem log start -- Sun Jan 26 19:58:41 1992  
 Single Point, SemiEmpirical, molecule = ETHANE.HIN.  
 CNDO  
 Convergence limit = 0.0100000 Iteration limit = 50  
 RHF Calculation:  
 Singlet state calculation  
 Number of electrons = 14  
 Number of Double Occupied Levels = 7  
 Charge on the System = 0  
 Total Orbitals = 14  
 Starting CNDO calculation with 14 orbitals

**<SCF iterations show differences in energy (kcal/mol) at this iteration from previous iteration. First iteration describes the second calculation so that a Difference is defined.>**

Iteration = 1 Difference = 4413.32086  
 Iteration = 2 Difference = 2.12419  
 Iteration = 3 Difference = 0.24927  
 Iteration = 4 Difference = 0.10723  
 Iteration = 5 Difference = 0.00046  
 Energy=-1659.715468 Gradient=64.939762 <Gradient is RMS value>

**<The isolated atomic energies below come from CINDO.ABP parameter file. Binding energies defined as differences between total energy and value for isolated atoms.>**

ENERGIES AND GRADIENT	Total Energy	=	-11801.4373094
(kcal/mol)			
Total Energy		=	-18.807376823 (a.u.)
Binding Energy		=	-1659.7154677 (kcal/mol)
Isolated Atomic Energy		=	-10141.7218417 (kcal/mol)
Electronic Energy		=	-28492.2957749 (kcal/mol)
Core-Core Interaction		=	16690.8584655 (kcal/mol)
Gradient		=	64.9397615 (kcal/mol/Ang)

## EIGENVALUES(eV)

-40.386066	-29.239080	-23.532267	-23.532249	-18.191881
-16.373669	-16.373646	7.900207	7.900218	8.076718
8.596774	10.167131	10.167147	10.711308	

## ATOMIC ORBITAL ELECTRON POPULATIONS

1.014663	0.977262	1.038487	0.977262	1.014663
0.977262	1.038487	0.977262	0.997442	0.997442
0.997442	0.997442	0.997442	0.997442	

## NET CHARGES AND COORDINATES

Atom	Z	Charge	Coordinates(a.u.)
------	---	--------	-------------------

			x	y	z
1	6	-0.007673	-1.01663	-0.89296	0.00000
2	6	-0.007673	-1.01663	2.01722	0.00000
3	1	0.002558	0.92538	-1.57955	0.00000
4	1	0.002558	-1.98761	-1.57955	1.68183
5	1	0.002558	-1.98763	-1.57956	-1.68182
6	1	0.002558	-2.95863	2.70382	0.00000
7	1	0.002558	-0.04564	2.70381	1.68183
8	1	0.002558	-0.04562	2.70383	-1.68182

<Dipole moment has a point charge contribution as well as a contribution for sp polarization (hybridization).>

Dipole (Debyes)	x	y	z	Total
Point-Chg.	0.000	0.000	0.000	0.000
Hybrid	0.000	0.000	0.000	0.000
Sum	0.000	0.000	0.000	0.000

Geometry optimization, SemiEmpirical, molecule = ETHANE.HIN.  
CNDO

PolakRibiere optimizer

Convergence limit = 0.0100000 Iteration limit = 50  
Optimization algorithm = Polak-Ribiere  
Criterion of RMS gradient = 0.1000 kcal/(A mol) Maximum cycles = 120  
RHF Calculation:

Singlet state calculation  
Number of electrons = 14  
Number of Double Occupied Levels = 7  
Charge on the System = 0  
Total Orbitals = 14  
Starting CNDO calculation with 14 orbitals

<The square brackets describe the inner iteration of the outer iteration.>

E=0.0000 Grad=0.000 Conv=NO(0 cycles 0 points) [Iter=1 Diff=4413.32086]  
E=0.0000 Grad=0.000 Conv=NO(0 cycles 0 points) [Iter=2 Diff=2.12419]  
E=0.0000 Grad=0.000 Conv=NO(0 cycles 0 points) [Iter=3 Diff=0.24927]  
E=0.0000 Grad=0.000 Conv=NO(0 cycles 0 points) [Iter=4 Diff=0.10723]  
E=0.0000 Grad=0.000 Conv=NO(0 cycles 0 points) [Iter=5 Diff=0.00046]  
E=-1659.7155 Grad=64.940 Conv=NO(0 cycles 1 points) [Iter=1 Diff=6.28550]  
E=-1659.7155 Grad=64.940 Conv=NO(0 cycles 1 points) [Iter=2 Diff=0.48315]  
E=-1659.7155 Grad=64.940 Conv=NO(0 cycles 1 points) [Iter=3 Diff=0.05823]  
E=-1659.7155 Grad=64.940 Conv=NO(0 cycles 1 points) [Iter=4 Diff=0.01519]  
E=-1659.7155 Grad=64.940 Conv=NO(0 cycles 1 points) [Iter=5 Diff=0.00003]  
E=-1669.7328 Grad=24.582 Conv=NO(0 cycles 2 points) [Iter=1 Diff=0.38656]

```

E=-1669.7328 Grad=24.582 Conv=NO(0 cycles 2 points) [Iter=2 Diff=0.02926]
E=-1669.7328 Grad=24.582 Conv=NO(0 cycles 2 points) [Iter=3 Diff=0.00232]
E=-1670.9449 Grad=3.839 Conv=NO(1 cycles 3 points) [Iter=1 Diff=0.00153]
E=-1671.0237 Grad=2.292 Conv=NO(1 cycles 4 points) [Iter=1 Diff=0.00154]
E=-1671.0615 Grad=1.155 Conv=NO(1 cycles 5 points) [Iter=1 Diff=0.00597]
E=-1671.0162 Grad=3.123 Conv=NO(1 cycles 6 points) [Iter=1 Diff=0.00339]
E=-1671.0653 Grad=1.143 Conv=NO(2 cycles 7 points) [Iter=1 Diff=0.01133]
E=-1671.0653 Grad=1.143 Conv=NO(2 cycles 7 points) [Iter=2 Diff=0.00096]
E=-1671.0725 Grad=2.754 Conv=NO(2 cycles 8 points) [Iter=1 Diff=0.00140]
E=-1671.0762 Grad=1.680 Conv=NO(3 cycles 9 points) [Iter=1 Diff=0.00172]
E=-1671.1079 Grad=0.968 Conv=NO(3 cycles 10 points) [Iter=1 Diff=0.00199]
E=-1671.1213 Grad=0.275 Conv=NO(3 cycles 11 points) [Iter=1 Diff=0.00756]
E=-1671.0930 Grad=1.184 Conv=NO(3 cycles 12 points) [Iter=1 Diff=0.00488]
E=-1671.1215 Grad=0.153 Conv=NO(4 cycles 13 points) [Iter=1 Diff=0.00084]
E=-1671.1133 Grad=1.515 Conv=NO(4 cycles 14 points) [Iter=1 Diff=0.00057]
E=-1671.1223 Grad=0.102 Conv=NO(5 cycles 15 points) [Iter=1 Diff=0.00014]
E=-1671.1222 Grad=0.281 Conv=NO(5 cycles 16 points) [Iter=1 Diff=0.00006]
E=-1671.1224 Grad=0.024 Conv=YES(6 cycles 17 points)[Iter=1 Diff=0.00000]

```

<Values for optimized structure are given below.>

#### ENERGIES AND GRADIENT

```

Total Energy           = -11812.8442711 (kcal/mol)
Total Energy           = -18.825555543 (a.u.)
Binding Energy         = -1671.1224294 (kcal/mol)
Isolated Atomic Energy = -10141.7218417 (kcal/mol)
Electronic Energy      = -28505.2114559 (kcal/mol)
Core-Core Interaction   = 16692.3671848 (kcal/mol)
Gradient               = 0.0248510 (kcal/mol/Ang)

```

#### EIGENVALUES(eV)

```

-40.735325  -29.079802  -23.688864  -23.688860  -18.800158
-15.713195  -15.713187   7.124108   7.124109   7.752248
 8.381527   10.037412   10.037416  11.649177

```

#### ATOMIC ORBITAL ELECTRON POPULATIONS

```

1.010628  0.971429  1.035717  0.971429  1.010628
0.971429  1.035717  0.971429  1.003599  1.003599
1.003599  1.003599  1.003599  1.003599

```

#### NET CHARGES AND COORDINATES

Atom	Z	Charge	Coordinates(a.u.)		
			x	y	z
1	6	0.010797	-1.01662	-0.81492	0.00000
2	6	0.010797	-1.01663	1.93919	0.00000
3	1	-0.003599	0.94470	-1.61103	-0.00001
4	1	-0.003599	-1.99728	-1.61103	1.69856
5	1	-0.003599	-1.99729	-1.61104	-1.69855



6	1	-0.003599	-2.97795	2.73530	-0.00001
7	1	-0.003599	-0.03597	2.73530	1.69856
8	1	-0.003599	-0.03596	2.73530	-1.69855

Dipole (Debyes)	x	y	z	Total
Point-Chg.	0.000	0.000	0.000	0.000
Hybrid	0.000	0.000	0.000	0.000
Sum	0.000	0.000	0.000	0.000

**<Beginning of molecular dynamics calculation.>**

Molecular dynamics, SemiEmpirical, molecule = ETHANE.HIN.  
CNDO

Convergence limit = 0.0100000 Iteration limit = 50  
Heat time = 0.0000 ps Run time = 0.0050 ps  
Cool time = 0.0000 ps Time step = 0.0010 ps  
Starting temperature = 0.0000 K Simulation temperature = 300.0000 K  
Final temperature = 0.0000 K Temperature step = 0.0000 K  
RHF Calculation:

Singlet state calculation  
Number of electrons = 14  
Number of Double Occupied Levels = 7  
Charge on the System = 0  
Total Orbitals = 14  
Starting CNDO calculation with 14 orbitals

**<Square brackets show inner iteration of trajectory.>**

Time=0.0000 ps Total Energy=0.0000 kcal/mol T=0.00 K [Iter=1  
Diff=4387.33259]  
Time=0.0000 ps Total Energy=0.0000 kcal/mol T=0.00 K [Iter=2  
Diff=2.77043]  
Time=0.0000 ps Total Energy=0.0000 kcal/mol T=0.00 K [Iter=3  
Diff=0.23693]  
Time=0.0000 ps Total Energy=0.0000 kcal/mol T=0.00 K [Iter=4  
Diff=0.02929]  
Time=0.0000 ps Total Energy=0.0000 kcal/mol T=0.00 K [Iter=5  
Diff=0.00029]  
Time=0.0000 ps Total Energy=-1663.9692 kcal/mol T=299.97 K [Iter=1  
Diff=0.02540]  
Time=0.0000 ps Total Energy=-1663.9692 kcal/mol T=299.97 K [Iter=2  
Diff=0.00253]  
Time=0.0010 ps Total Energy=-1664.2238 kcal/mol T=284.66 K [Iter=1  
Diff=0.02273]  
Time=0.0010 ps Total Energy=-1664.2238 kcal/mol T=284.66 K [Iter=2  
Diff=0.00229]

Time=0.0020 ps Total Energy=-1664.8376 kcal/mol T=247.75 K [Iter=1  
Diff=0.01758]  
Time=0.0020 ps Total Energy=-1664.8376 kcal/mol T=247.75 K [Iter=2  
Diff=0.00178]  
Time=0.0030 ps Total Energy=-1665.4801 kcal/mol T=209.23 K [Iter=1  
Diff=0.01282]  
Time=0.0030 ps Total Energy=-1665.4801 kcal/mol T=209.23 K [Iter=2  
Diff=0.00130]  
Time=0.0040 ps Total Energy=-1665.8840 kcal/mol T=185.31 K [Iter=1  
Diff=0.00989]

<Values for termination of trajectory are shown below.>

## ENERGIES AND GRADIENT

Potential Energy = -11811.9756909 (kcal/mol)  
Potential Energy = -18.824171329 (a.u.)  
Binding Energy = -1670.2538492 (kcal/mol)  
Isolated Atomic Energy = -10141.7218417 (kcal/mol)  
Electronic Energy = -28579.1119570 (kcal/mol)  
Core-Core Interaction = 16767.1362662 (kcal/mol)  
Gradient = 15.9022448 (kcal/mol/Ang)

## EIGENVALUES(eV)

-40.993458	-29.136822	-23.837172	-23.772387	-18.899385
-15.648407	-15.521277	7.007665	7.054772	7.770212
8.417984	10.039606	10.131248	11.863716	

## ATOMIC ORBITAL ELECTRON POPULATIONS

1.007651	0.969464	1.035971	0.972535	1.006693
0.972557	1.037679	0.969771	1.001260	1.007314
1.006463	1.004401	1.002980	1.005261	

## NET CHARGES AND COORDINATES

Atom	Z	Charge	Coordinates(a.u.)		
			x	y	z
1	6	0.014380	-0.98866	-0.79542	-0.01013
2	6	0.013300	-1.03241	1.92194	0.01288
3	1	-0.001260	0.93860	-1.62845	-0.00970
4	1	-0.007314	-2.01437	-1.60907	1.66915
5	1	-0.006463	-2.01617	-1.59763	-1.69357
6	1	-0.004401	-2.99806	2.73141	-0.00538
7	1	-0.002980	-0.07517	2.72392	1.70330
8	1	-0.005261	-0.07984	2.72577	-1.69653

Dipole (Debyes)	x	y	z	Total
Point-Chg.	0.032	0.010	0.007	0.034
Hybrid	-0.021	-0.006	-0.004	0.022
Sum	0.011	0.003	0.002	0.012

Time = 0.0050 ps  
 Kinetic Energy = 4.2293 kcal/mol  
 Total Energy = -1666.0245 kcal/mol  
 Temperature = 177.36 K  
 Time=0.0050 ps TotalEnergy=-1666.0245 kcal/mol T=177.36 K [final]  
 HyperChem  
 log stop -- Sun Jan 26 20:01:15 1992

## CNDO Configuration Interaction

<Below are the results of a microstate configuration interaction (CI) calculation.>

HyperChem log start -- Sun Jan 26 20:31:04 1992  
 Single Point, SemiEmpirical, molecule = ETHANE.HIN.  
 CNDO  
 Convergence limit = 0.0100000 Iteration limit = 50  
 RHF Calculation:  
 Singlet state calculation  
 Configuration interaction will be used  
 Number of electrons = 14  
 Number of Double Occupied Levels = 7  
 Charge on the System = 0  
 Total Orbitals = 14  
 Starting CNDO calculation with 14 orbitals

<First of all perform an SCF calculation to obtain a reference configuration.>

Iteration = 1 Difference = 4413.32086  
 Iteration = 2 Difference = 2.12419  
 Iteration = 3 Difference = 0.24927  
 Iteration = 4 Difference = 0.10723  
 Iteration = 5 Difference = 0.00046

<Now perform a configuration interaction calculation with the lowest 100 microstates (single determinants). Report results for only the 10 lowest resulting configurations.>

### CONFIGURATION INTERACTION

The Lowest Ten State Energies, Expectation Values of S(S+1) and S

State	Absolute E(kcal/mol)	Relative E(kcal/mol)	S(S+1)	S
1	-11803.7234927	0.0000000	0.00	0.00
2	-11480.3053185	323.4181742	2.00	1.00
3	-11466.9040318	336.8194609	2.00	1.00
4	-11466.9025478	336.8209449	2.00	1.00

5	-11449.3794912	354.3440015	2.00	1.00
6	-11449.3682000	354.3552927	0.00	0.00
7	-11440.2197333	363.5037594	0.00	0.00
8	-11440.2196214	363.5038713	0.00	0.00
9	-11437.3047164	366.4187763	2.00	1.00
10	-11437.3046177	366.4188750	2.00	1.00

Energy=-1662.001651 Gradient=65.321671

## ENERGIES AND GRADIENT

Total Energy	=	-11803.7234927	(kcal/mol)
Total Energy	=	-18.811020202	(a.u.)
Binding Energy	=	-1662.0016510	(kcal/mol)
Isolated Atomic Energy	=	-10141.7218417	(kcal/mol)
Electronic Energy	=	-28494.5819582	(kcal/mol)
Core-Core Interaction	=	16690.8584655	(kcal/mol)
CI Energy	=	-2.2861833	(kcal/mol)
Number of Configurations Used	=	100	
Gradient of Reference Configuration	=	65.3216714	(kcal/mol/Ang)

## EIGENVALUES OF THE REFERENCE CONFIGURATION(eV)

-40.386066	-29.239080	-23.532267	-23.532249	-18.191881
-16.373669	-16.373646	7.900207	7.900218	8.076718
8.596774	10.167131	10.167147	10.711308	

## ATOMIC ORBITAL ELECTRON POPULATIONS

1.014662	0.977380	1.038151	0.977380	1.014662
0.977380	1.038151	0.977380	0.997476	0.997476
0.997476	0.997476	0.997476	0.997476	

## NET CHARGES AND COORDINATES

Atom	Z	Charge	Coordinates(a.u.)		
			x	y	z
1	6	-0.007572	-1.01663	-0.89296	0.00000
2	6	-0.007572	-1.01663	2.01722	0.00000
3	1	0.002524	0.92538	-1.57955	0.00000
4	1	0.002524	-1.98761	-1.57955	1.68183
5	1	0.002524	-1.98763	-1.57956	-1.68182
6	1	0.002524	-2.95863	2.70382	0.00000
7	1	0.002524	-0.04564	2.70381	1.68183
8	1	0.002524	-0.04562	2.70383	-1.68182

Dipole (Debyes)	x	y	z	Total
Point-Chg.	0.000	0.000	0.000	0.000
Hybrid	0.000	0.000	0.000	0.000
Sum	0.000	0.000	0.000	0.000

HyperChem

log stop -- Sun Jan 26 20:32:17 1992

# Appendix B

## Force Field Files

HyperChem provides several files that you need for chemical calculations with molecular mechanics force fields (MM+, AMBER, BIO+ (CHARMM), and OPLS):

chem.rul	A text file containing a set of rules for assigning atom types for each force field. HyperChem requires the binary version of this file, typerule.bin.
typerule.bin	The binary version of chem.rul. HyperChem uses this file to assign atom types to atoms.
*.txt	Force field parameter files in text format. You can change these files with most text editors. There is at least one set of TXT files for each force field.
*.dbf	Force field parameter files in dBase (III or IV) format. You can change these files with most database programs. There is at least one set of DBF files for each force field. These types of files are redundant as same parameters are in TXT files. These files are being phased out and new parameters or parameter sets are only being created for TXT files.
*.par	The binary version of parameter files. HyperChem uses these files to direct molecular mechanics calculations. HyperChem combines a complete set of parameter files for any force field into one PAR file. For the molecular mechanics force field MM+, the name of the binary parameter file is mmplus.par.

*Note:* You can use any filenames and extensions for parameter files. The settings in the Registry (see Appendix F) direct HyperChem to the correct parameter files.

## Atom Types

For molecular mechanics calculations, each atom in a molecular system should have an atom type. Atom types represent sets of atoms of the same element that have similar chemical environments. Molecular mechanics calculations treat all atoms of the same type in similar ways. Each force field has a different set of atom types.

Atom types are generally required for molecular mechanics calculations, but not for semi-empirical calculations. The text file `chem.rul` contains rules that assign atom types to atoms, depending on the environment of the atoms. Each atom type has a name that might be different for each force field. Force fields require atom types to assign calculation parameters. However, all force fields in HyperChem can now deal with atoms that lack types (with the unknown type `**`) by using an alternative default method for assigning parameters. This can be a useful feature for working around limitations of force fields that determine bonding characteristics on the basis of atom types. For example, MM+ makes the single bond in butadiene too short, since by default a bond between two C3-type atoms should have the length of a double bond. Similarly, MM+ tries to make biphenyl planar, though experimentally the rings are not coplanar. By setting the two carbon atoms that are singly bonded to type `**`, these problems are avoided.

When you draw a new molecule, HyperChem gives each atom the type name `**`, indicating no atom type is yet assigned. When you choose Model Build or Calculate Types on the Build menu, or change the force field (using Molecular Mechanics on the Setup menu), HyperChem assigns appropriate atom types, using the force field that you chose in the Molecular Mechanics Force Field dialog box. After assigning types, a remaining `**` means that there is no appropriate atom type available.

If you open a HIN file, HyperChem uses the force field that was in effect when you saved the file and assigns atom types stored in the file.

HyperChem requires a binary version of `chem.rul`, named `typerule.bin`, both of which come with HyperChem. If you alter the text file `chem.rul`, choose Compile Type Rules on the Build menu to convert the new `chem.rul` to a new `typerule.bin` file.

## Atom Typing Rules

You can add or modify the atom typing rules found in chem.rul, which contains rules for all force fields. The file has a hierarchical decision structure, including these levels:

- Force field
- Element
- Clause and tests
- Assignment statement

For each atom in a molecular structure, HyperChem scans this file and makes these decisions:

- What is the force field?
- What is the element?
- For this element, does the atom satisfy the tests in the first clause? If not, in the second clause, and so on?
- When the atom satisfies the tests in a clause, HyperChem assigns the atom type.
- HyperChem goes to the next atom.

*Caution:* Changes to chem.rul can cause fatal errors in HyperChem. Make a backup copy of chem.rul before you change it.

*Example:*

S: connected to (H)?

=SH.

=S.

In this example for a sulphur atom (S), from the OPLS force field, the first clause contains one test: is the sulfur atom connected to a hydrogen? If true, the atom type is SH. If false, HyperChem goes to the next clause, which has no tests, only a statement that assigns S as the atom type.

## Syntax

This is the structure of chem.rul:

forcefield(<force-field-name>)

<element-symbol>:

; <comment>

<clause>

    <test>?

    <test>?

    ...

    =<assignment-statement>.

<clause>

...

*Note:* Entries in chem.rul are not case-sensitive, except for element symbols. HyperChem recognizes, for example, the difference between SI and Si. Spaces between words are optional.

### forcefield

<force-field-name> is amber, opl, bio+, or mm+.

### <element-symbol>:

This is the chemical symbol for an element, for example, S, Cl, and Na.

A colon must follow the symbol. Each element can have only one symbol. Element symbols include LP for lone pair and HX for extended hydrogen.

### ; <comment>

Comments can occur anywhere in the file. A comment begins with a semicolon and ends at the end of a line.

### <clause>

A clause is a set of tests applied to each atom plus an atom type assignment. Each clause ends with an assignment statement.



**Note:** HyperChem checks clauses from top to bottom, so their order is important. Place more specific clauses before general ones.

If one test in a clause fails, HyperChem goes to the next clause. If all the tests in a clause are evaluated as true, HyperChem assigns an atom type. Some elements have a final clause without tests. This assigns an atom type if all previous clauses failed. If the force field does not have a clause without tests and if all clauses failed, HyperChem assigns the atom type \*\*, meaning unknown.

### <test>?

A question mark must follow each test. A clause can contain these tests:

Test	To pass test
Is it sp3?	True if the atom has no double, triple, or aromatic bonds
Is it sp2?	True if the atom has one double or two or more aromatic bonds
Is it sp?	True if the atom has one triple or two double bonds
In 5 ring?	True if the atom is part of a five-membered ring
In 6 ring?	True if the atom is part of a six-membered ring
Has aromatic bond?	True if the atom has at least one aromatic bond
Connected to <expression>?	True if the atom is bonded to atoms described in <expression>. See the next section

### Connected to Expressions

Connected to expressions include the type of connecting bond, the connected atoms, and bonds between the connected atoms. These symbols represent bonds:

-	single bond
=	double bond

#	triple bond
~	aromatic bond
no symbol	any bond type

Connected to expressions use element symbols to represent connected atoms. You can use any standard symbols from the periodic table, plus these:

Lp	Lone pair
Hx	Extended hydrogen
R	Any element, including Lp, except H
X	Any halogen (F, Cl, Br, I)
*	Any element except Lp

**Note:** Keep connected to expressions as short as possible. Atom types are usually unaffected by atoms more than two or three bonds away. Also, longer expressions increase the time required to assign types.

**Example:**

C

The atom is connected to a carbon through any type of bond.

-C=N-C

The atom is connected to the CNC group through a single bond. The group contains a double and single bond.

-CH3

The atom is connected to a methyl group through a single bond.

-C(-CH3)-N

The atom is connected to a carbon through a single bond. The carbon has a methyl side chain (in parentheses) and a singly bonded nitrogen.

(-CH3)-N

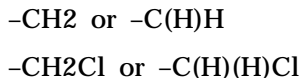
The atom is connected to a methyl group by a single bond and a nitrogen by a single bond.

**Note:** HyperChem compares atom by atom the environment around an atom to the terms on a connected to expression. It does

not retrace its path. This prevents a simple carbon-carbon bond from satisfying expressions like  $-C-C-C-C-C$ .

## Side Groups

Specify side chains either by placing them in parentheses or by using numbers. For example:



## Rings

The atom being typed can be connected to or part of a ring. Pairs of tags (@1 to @9) placed after atoms in the ring describe closure points, for example:



This describes a connection to a six-membered ring, with the tag @1 after  $C_1$  and  $C_7$ .  $C_1$  and  $C_7$  represent the same atom in the ring.



This describes a cyclopropane ring, with the atom being typed as the first atom ( $C_1$ ) in the ring (@1 at the beginning of the expression modifies the atom being typed).  $C_1$  and  $C_4$  are the same atom.

### <assignment-statement>.

A clause must end with an assignment statement that contains an atom type belonging to the force field. These types appear in the files \*typ.txt (see the next section). The assignment statement must end with a period.

## Force Field Parameters

Each force field can have several parameter sets. Each parameter set has a set of parameter files that define the interactions between atoms and the force constants for these interactions. HyperChem supplies each file in two formats, text (ASCII) and dBase III or IV. You can use either a text editor (like Windows Notepad) or a database program (like Excel Q+E) to view and edit these files. The type of file that HyperChem uses can be different for each force field and parameter set, either TXT or DBF. The Registry (see, for exam-

ple, “MM+ Settings” on page 769) defines the file type for each force field and parameter set in this record:

FileFormat=Text (or DBF for DBASE)

*Note:* DBF files for parameters are being phased out as redundant and you should now use only Text files in the above setting.

## Parameter Files

HyperChem supplies the following parameter files for the force fields:

*typ.txt	List of atom types and atomic masses.
*typ.dbf	
*str.txt	Parameters for bond stretching.
str.dbf	
ben.txt	Parameters for bond angle bending.
ben.dbf	
mmpben3.txt	For MM+ only, bond angle parameters for three-membered rings.
mmpben3.dbf	
mmpben4.txt	For MM+ only, bond angle parameters for four-membered rings.
mmpben4.dbf	
tor.txt	Parameters for torsion angle rotation.
tor.dbf	
mmptor3.txt	For MM+ only, bond torsion parameters for three-membered rings.
mmptor3.dbf	
mmptor4.txt	For MM+ only, bond torsion parameters for four-membered rings.
mmptor4.dbf	
*imp.txt	For AMBER, BIO+, or OPLS, parameters for improper torsion angles (see the Glossary) involving planarity constraints.
*imp.dbf	
mmpoop.txt	For MM+ only, out-of-plane bending parameters for an sp <sup>2</sup> hybridized atom. Other force fields use improper torsion angles (see the Glossary) for these cases.
mmpoop.dbf	
nbd.txt	Parameters for nonbonded interactions between pairs of atoms (atom types).
nbd.dbf	

*npr.txt	Parameters for nonbonded, 1–4 interactions between specific pairs of atoms. At this time, only the bio85 and charmm* parameter sets (see the next section) include this file.
*npr.dbf	
am*hbd.txt	For AMBER only, parameters for hydrogen bonding.
am*hbd.dbf	
.par	A binary file combining the parameters files for one parameter set. A force field can have one PAR file for each parameter set (see the next section).

HyperChem requires at least one binary (PAR) file for each force field. All PAR files are supplied on the HyperChem release disks. If you alter the parameter files (TXT or DBF), choose the parameter set (using Select Parameter Set on the Setup menu), and then choose Compile Parameter Set on the Setup menu to convert this parameter set to a PAR file.

To add a new parameter set, enter the information using a text editor or a database program. Then add to the Registry (see Appendix F) a description of the parameter set.

## Parameter Sets

Each force field can have more than one parameter set and a corresponding PAR file. This version of HyperChem has the following parameter sets available:

Parameter Set	Parameter Files Included	Reference
mm+	mmp*.*	Allinger, N.L., <i>J. Am. Chem. Soc.</i> 99:8127-8134, 1977  Quantum Chemistry Program Exchange (QCPE) Program #395 (MM2), Indiana University Department of Chemistry, Bloomington, IN
amber2	amber*.*	Weiner, S.J., <i>et al.</i> , <i>J. Comp. Chem.</i> 7:230-252, 1986

Parameter Set	Parameter Files Included	Reference
amber3	am89a*.*	Kollman, P.A., Department of Pharmaceutical Chemistry, University of California, San Francisco, by request. For AMBER Release 3.0A
ambers	am90s*.*	Hohmans, S.W., <i>Biochemistry</i> 29:9110 (1990). A force field optimized for polysaccharide simulation
amber94	am94a*.*	Cornell, W.D., Cieplak, P., Bayly C.I., Gould I.R., Merz, K.M. Jr., Ferguson, D.M., Spellmeyer, D.C., Fox, T., Caldwell, J.W. and Kollman, P.A., <i>Journal of the American Chemical Society</i> <b>117</b> , 5179-5197 (1995).
amber96	am96ator.txt (only torsions changed from amber94)	<i>Computer Simulation of Biomolecular Systems</i> , A. Wilkinson, P. Weiner, W. Van Gunsteren, eds. Volume 3, p. 83-96, P. Kollman, R. Dixon, W. Cornell, T. Fox, C. Chipot and A. Pohorille
bio83	bio83*.*	Brooks, B.R., <i>et al. J. Comp. Chem.</i> 4:187-217, 1983. Since this version of CHARMM does not consider all dihedral angles, torsion force constants for the missing angles are set to zero in BIO+
bio85	bio85*.*	Reiher, W.E., III, <i>Theoretical Studies of Hydrogen Bonding</i> , Ph.D. thesis, Harvard University, 1985. Since this version of CHARMM does not consider all dihedral angles, torsion force constants are scaled to include all dihedral angles

Parameter Set	Parameter Files Included	Reference
charmm19	charmm19*.*	Reiher, III., W.E. <i>Theoretical Studies of Hydrogen Bonding</i> , Ph.D. Thesis, Department of Chemistry, Harvard University, Cambridge, MA, USA, 1985
charmm22	charmm22*.*	MacKerell, Jr., A. D. et. al., All-atom empirical potential for molecular modeling and dynamics Studies of proteins. <i>Journal of Physical Chemistry B</i> , 1998, <b>102</b> , 3586-3616
charmm27	charmm27*.*	Foloppe, N. and MacKerell, Jr., A.D. "All-Atom Empirical Force Field for Nucleic Acids: 1) Parameter Optimization Based on Small Molecule and Condensed Phase Macromolecular Target Data. <i>Journal of Computational Chemistry</i> , 2000, <b>21</b> :86-104
opls	opls*.*	Jorgensen, W.L. and Tirado-Rives, J. <i>J. Am. Chem. Soc.</i> 110:1657-1666, 1988  Pranata, J. et al., <i>J. Am. Chem. Soc.</i> 113:2810-2819, 1991  Jorgensen, W.L. and Severance, D.L. <i>J. Am. Chem. Soc.</i> 112:4768-4774, 1980

The Registry lists the alternative parameter sets in the record CustomNames; for example: CustomNames=amber2, amber3. This file also gives the default conditions for each parameter set (see, for example, "AMBER Settings" on page 749).

To choose a parameter set:

1. Choose Molecular Mechanics on the Setup menu, then choose a force field.
2. Choose Select Parameter Set on the Setup menu, then choose a parameter set for the chosen force field.

If you change the contents of a parameter file or add a new set of parameters, choose the parameter set (using Select Parameter Set on the Setup menu), and then choose Compile Parameter Set on the Setup menu to convert this parameter set to a PAR file.

## Newer Force Fields for Amber and BIO+(CHARMM)

HyperChem includes two force fields for both the Amber and Bio+ (Charmm) molecular mechanics method.

The older Amber force field is represented by the parameter sets Amber2, Amber3, and Ambers. The newer force field (without a Hydrogen-bonding term, for example) is represented by the parameter sets Amber94, Amber96, and beyond. Choosing the newer parameter sets sets a flag that automatically chooses the new force field (sometimes referred to by the Amber group as the Cornell et. al force field). This new force field has the name Amber94 (in HIN files, for example) although it does not appear as an explicit choice.

A similar scheme is in place with Bio+ (Charmm). Choosing a parameter set charmm22, charmm27, and beyond automatically gives you the newer force field (without hydrogen bonding and with a Urey-Bradley term, etc. as described in the definitive paper on Charmm (*J. Phys. Chem. B.*, **102**, 3586, 1998). This new force field has the name Charmm2 (in HIN files, for example) although it does not appear as an explicit choice. Choosing a bio83 or bio85 parameter set or charmm19 parameter set (which differs trivially from the bio85 parameter set) causes the older force field to be used.

There are thus, in effect 6 force fields in HyperChem - MM+, Amber, Amber94, Bio+, Charmm2, OPLS, and the default force field used when all types are \*\*.

**Caution:** The last two paragraphs imply a restriction on custom user-implemented parameter sets for the older force fields. HyperChem distinguishes the newer force field by a parameter set index of 4 or greater (starting at 1). Thus one can add parameter sets at will for the newer force fields but there must remain only 3 parameter sets for the older force fields or an indexing problem will result.



## Type Parameter File (\*typ)

This file contains all atom types, and corresponding atomic masses, for a force field. The Registry, or the chem.ini file, specifies the names of these parameter files in AtomTypeMass records. The parameter file has three columns, separated by single tabs (in text files):

Column	Content	Field Type <sup>a</sup>
ATOM	Atom type	C
MASS	Mass in grams/mol	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* MG 24.3050 magnesium

## Bond Length Parameter File (str)

This file contains the bond stretch parameters for AMBER, BIO+ (CHARMM), or OPLS. The Registry specifies the names of these parameter files in QuadraticStretch records. The parameter file has five columns:

Column	Content	Field Type <sup>a</sup>
B1	Atom 1 (atom type) involved in bond	C
B2	Atom 2 (atom type) involved in bond	C
KR	Stretching force constant in kcal/mol Å <sup>2</sup>	N
REQ	Equilibrium distance in Ångstroms	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* A B 469.0000 1.4000 TYR

## MM+ Bond Length Parameter file

The chem.ini file specifies the name of this parameter file in the MM+Stretch record. The parameter file has seven columns:

Column	Content	Field Type <sup>a</sup>
T1	Atom 1 (atom type) involved in bond	C
T2	Atom 2 (atom type) involved in bond	C
KS	Stretching force constant in millidynes/Å	N
L0	Minimum energy bond length in Å	N
L1	Alternative minimum energy bond length in Å, used only if atoms 1 and 2 have less than two hydrogens attached to each	N
DIPOLE	Bond dipole moment in Debyes	N
REMARK	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* c4 c3 4.300 1.522 0.000 0.300 "my test parameter"

## Bond Angle Parameter File (ben)

This file contains all bending (bond angle) parameters for the AMBER, BIO+, and OPLS force fields. The Registry specifies the names of these parameter files in the QuadraticBend records. The parameter file has six columns:

Column	Content	Field Type <sup>a</sup>
A1	Atom 1 (atom type) involved in angle	C
A2	Atom 2 (atom type), the central atom, involved in angle	C
A3	Atom 3 (atom type) involved in angle	C
KTHETA	Bending force constant in kcal/mol rad <sup>2</sup>	N
THETAEQ	Equilibrium angle in degrees	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* C2 C2 NT 80.0000 111.2000 pro jpc 76, 1439

## MM+ Bond Angle Parameter Files

For MM+, there are three bend parameter files, the files mmpben.\* for most bond angles and mmpben3.\* and mmpben4.\* for three- and four-membered rings. The Registry specifies the names of these parameter files in the records MMPBend, MMPBend3, and MMPBend4, respectively. The parameter files have eight columns:

Column	Content	Field Type <sup>a</sup>
T1	Atom type for end atom of bond angle	C
T2	Atom type for central atom of bond angle	C
T3	Atom type for other end atom of bond angle	C
KS	Bending force constant in millidyne Å/rad <sup>2</sup>	N

Column	Content	Field Type <sup>a</sup>
TYPE1	Standard bond angle. MM+ uses this angle unless TYPE2 and TYPE3 are nonzero and their conditions are satisfied	N
TYPE2	Bond angle if one end atom is hydrogen	N
TYPE3	Bond angle if both end atoms are hydrogens	N
REMARKS	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* c4 c3 c4 0.440 118.20 0.00 0.00 “my test parameter”

## Urey-Bradley Bend Parameter File

The newer Charmm22 force field includes a possible Urey-Bradley term (an angle term that depends on the distance between the two terminal atoms) that assists in the description of vibrational spectra. The parameter files for this term have the names charmm\*ub.txt. The Registry specifies the name for these files.

Column	Content	Field Type <sup>a</sup>
A1	Atom 1 (atom type) involved in angle	C
A2	Atom 2 (atom type), the central atom, involved in angle	C
A3	Atom 3 (atom type) involved in angle	C
KUB	Distance force constant in kcal/mol Å <sup>2</sup> involving A1 and A2	N
S0	Equilibrium distance in Ångstroms	N
REMARKS	Comments and literature references	R

a. C = four-character string, N = number, R = character string

*Example:* CA CA CA 35.0 2.4162 “MacKerell web site”

## Torsion Parameter File (tor)

This file contains torsional parameters for the AMBER, BIO+ (CHARMM), and OPLS force fields. The Registry specifies the names of these parameter files in FourierTorsion records. The file has nine columns:

Field Name	Content	Field Type <sup>a</sup>
T1	Atom 1 (atom type) involved in torsion	C
T2	Atom 2 (atom type), a central atom, involved in torsion	C
T3	Atom 3 (atom type), a central atom, involved in torsion	C
T4	Atom 4 (atom type) involved in torsion	C
MULT	Multiplicity	N
HALF_VN	(Energy of torsional barrier N)/2, in kcal/mol	N
GAMMA	Phase offset in degrees	N
N	Periodicity of the Fourier term, from 1 to 6	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

**Note:** If there are multiple parameters for the same T1, T2, T3, and T4 that differ in N (periodicity), the Fourier series uses all of them.

**Example:** N\* CM CM CA 1 9.5100 180.000 2

## MM+ Torsion Angle Parameter Files

For MM+, there are two torsion parameter files, mmptor.\* for most torsions and mmptor4.\* for torsions involving four-membered rings. The Registry specifies the names of these parameter files in

the MMPTorsion and MMPTorsion4 records. The parameter files have eight columns:

Column	Content	Field Type <sup>a</sup>
T1	Atom 1 (atom type) involved in torsion	C
T2	Atom 2 (atom type), a central atom, involved in torsion	C
T3	Atom 3 (atom type), a central atom, involved in torsion	C
T4	Atom 4 (atom type) involved in torsion	C
V <sub>1</sub>	Energy of torsional barrier V <sub>1</sub> , in kcal/mol	N
V <sub>2</sub>	Energy of torsional barrier V <sub>2</sub> , in kcal/mol	N
V <sub>3</sub>	Energy of torsional barrier V <sub>3</sub> , in kcal/mol	N
REMARK	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* o1 co o2 c4 -1.650 8.880 0.000 "my test parameter"

## Improper Torsion Parameter File (\*imp)

This file contains all improper torsional parameters for AMBER or OPLS force fields. The Registry specifies the names of these parameter files in the FourierImpTorsion records. The central atom in an improper torsion is either atom 2 or atom 3. The parameter file has eight columns:

Column	Content	Field Type <sup>a</sup>
T1	Atom 1 (atom type) involved in torsion	C

Column	Content	Field Type <sup>a</sup>
T2	Atom 2 (atom type) involved in torsion	C
T3	Atom 3 (atom type) involved in torsion	C
T4	Atom 4 (atom type) involved in torsion	C
HALF_VN	(Energy of torsional barrier N)/2, in kcal/mol	N
GAMMA	Phase offset in degrees	N
N	Periodicity of the Fourier term, from 1 to 6	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* C3 CH NT C 114.00 180.00 3

## BIO+ Improper Torsion Parameter File

The BIO+ force field uses a quadratic form for improper torsions. The Registry gives the names of this parameter file in the QuadraticImpTorsion record. The parameter file has seven columns:

Column	Content	Field Type <sup>a</sup>
T1	Atom 1 (atom type) involved in torsion	C
T2	Atom 2 (atom type) involved in torsion	C
T3	Atom 3 (atom type) involved in torsion	C
T4	Atom 4 (atom type) involved in torsion	C
KOMEGA	Improper torsion force constant in kcal/mol rad <sup>2</sup>	N
OMEGA0	Equilibrium improper torsion angle in degrees	N

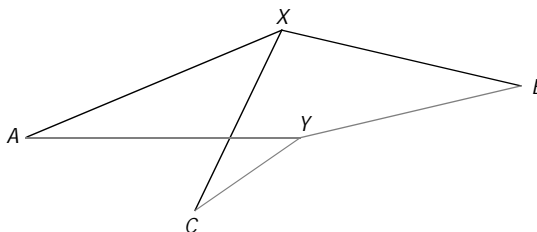
Column	Content	Field Type <sup>a</sup>
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R l= character string.

*Example:* N CH1E CH2E C 45.00 0.00 “J. Comput. Chem. 4, 187 (1983)”

## Out-of-plane Bending Parameter File (mmpoop)

This file contains parameters for the MM+ force field only, describing the planar arrangement of atoms bonded to an  $sp^2$  atom.



If X is an  $sp^2$  hybridized atom, the attached atoms A, B, and C tend to be in the same plane as X. Y is the projection of X onto the plane ABC. The out-of-plane parameters involve the angles XAY, XBY, and XCY. The MM+ calculation also uses standard bond angle parameters (see page 666) for angles AYB, AYC, and BYC, where the atom type for Y is the same as X.

The Registry gives the name of this parameter file in the MMPOOPBend record. The parameter file has four columns:

Column	Content	Field Type <sup>a</sup>
C	Atom type for central, $sp^2$ hybridized atom (atom X in diagram)	C
A	Atom type for one of the atoms bonded to atom C (atom A, B, or C in diagram)	C



Column	Content	Field Type <sup>a</sup>
COPB	Out-of-plane bending force constant, in millidyne Å/rad <sup>2</sup> (for angle XAY, XBY, or XCY in the diagram)	N
REMARK	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

*Example:* c3 c4 0.050 “my test parameter”

## Nonbonded Parameter Files (nbd)

HyperChem calculates nonbonded interactions (which include van der Waals attractions) between pairs of atoms (actually, atom types). The Registry specifies the names of these parameter files in the 6-12AtomVDW records.

There are three methods for calculating pair-wise interactions of atom types: R Star Epsilon, Sigma Epsilon, or Slater-Kirkwood (see *HyperChem Computational Chemistry* for a discussion of these methods). The contents of a parameter file depend on the combination method that you specify in the 6-12AtomVDWFormat records in the Registry for AMBER, BIO+, and OPLS parameter sets.

The default combination method for MM+ and AMBER is R Star Epsilon. You cannot change the combination method for MM+. The default methods for BIO+ and OPLS are R Star Epsilon and Sigma Epsilon, respectively.

This is the format of a \*NBD file using an R Star Epsilon calculation (four columns):

Column	Content	Field Type <sup>a</sup>
ATOM	Atom type	C
R_STAR	Half the minimum energy nuclear separation, in Ångstroms (van der Waals radius, $r_i^*$ ). Uses an arithmetic mean for pair values	N

Column	Content	Field Type <sup>a</sup>
EPS	Well depth in kcal/mol (minimum energy)	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

**Example:** CA 1.8500 0.1200 Equivalent to C

This is the format of a \*NBD file using a Sigma Epsilon calculation (four columns):

Column	Content	Field Type <sup>a</sup>
ATOM	Atom type	C
SIGMA	Nuclear separation where energy is zero. Uses a geometric mean for pair values	N
EPS	Well depth in kcal/mol (minimum energy)	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

This is the format of a \*NBD file using a Slater-Kirkwood calculation (five columns):

Column	Content	Field Type <sup>a</sup>
TYPE	Atom type	C
ALPHA	Atomic polarizability	N
NEFF	Effective number of valence electrons	N
RVDW	van der Waals radius in Ångstroms	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

## Pair-based Nonbonded Parameter Files

Each parameter set (except for MM+) can have an optional file containing parameters for nonbonded interactions between specific pairs of atoms. If these parameters are present, HyperChem uses them instead of calculating pair-wise interactions for specific atom types (see the previous section on \*NBD files). In the present release of HyperChem, only BIO+ (CHARMM) and OPLS have pair-based nonbonded parameter files, charmm\*npr and oplsnpr. The Registry specifies the names of these parameter files in 6-12PairVDW records.

There are three methods for calculating parameters using the data in these files: R Star Epsilon, Sigma Epsilon, or  $A_{ij}B_{ij}$  (see *HyperChem Computational Chemistry* for a discussion of these methods). The contents of a parameter file depend on the calculation method that you specify in the 6-12AtomVDWFormat records in the Registry for AMBER, BIO+ (CHARMM), and OPLS parameter sets.

This is the format of a \*NPR file using an R Star Epsilon calculation (five columns):

Field Name	Content	Field Type <sup>a</sup>
A1	Atom type of atom 1	C
A2	Atom type of atom 2	C
RMIN	The minimum energy separation in Å	N
EPSILON	Well depth in kcal/mol	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

This is the format of a \*NPR file using a Sigma Epsilon calculation (five columns):

Field Name	Content	Field Type <sup>a</sup>
A1	Atom type of atom 1	C
A2	Atom type of atom 2	C
SIGMA	The separation where energy is zero	N
EPSILON	Well depth in kcal/mol	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

This is the format of a \*NPR file using an  $A_{ij}B_{ij}$  calculation (five columns):

Field Name	Content	Field Type <sup>a</sup>
A1	Atom type of atom 1	C

Field Name	Content	Field Type <sup>a</sup>
A2	Atom type of atom 2	C
A <sub>ij</sub>	The coefficient for the r <sup>-12</sup> term (kcal/mol Å <sup>12</sup> )	N
B <sub>ij</sub>	The coefficient for the r <sup>-6</sup> term (kcal/mol Å <sup>6</sup> )	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

BIO+ can use an additional pair-based nonbonded parameter file for 1-4, nonbonded interactions. The Registry specifies the name of this parameter file in the 6-12Atom14VDW record. This file is not supplied in this release of HyperChem.

## Hydrogen Bond Parameter File (am\*hbd)

This file contains the hydrogen bonding parameters for the AMBER force field. The Registry specifies the name of this parameter file in the 10-12PairHBond record. The file has five columns:

Field Name	Content	Field Type <sup>a</sup>
ACC	Atom type of acceptor	C
DON	Atom type of donor	C
C	Parameter <sup>b</sup>	N
D	Parameter <sup>b</sup>	N
REFERENCE	Comments and literature references	R

a. C = four-character string, N = number, R = character string.

b.  $E_{HBOND} = \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^9}$ , where i and j represent the interacting atoms.

**Example:** H NB 7557.00 2385.000 "Raphael"



# Appendix C

## Quantum Mechanics Files

### Semi-Empirical Files

HyperChem provides several files that are necessary for semi-empirical quantum mechanics calculations (Extended Hückel, CNDO, INDO, MINDO3, MNDO, MNDO/d, AM1, PM3, ZINDO/1, ZINDO/S, and TNDO). These files contain parameters that simplify solutions of the wave equations:

Method	Filename	Contents	Reference
Extended Hückel	exhuckel.abp	Alpha and beta parameters, orbital exponents	Howell, J. <i>et al.</i> , QCPE #344 <sup>a</sup> ; see also following table
CNDO and INDO	cindo.abp	Alpha and beta parameters, orbital exponents	Pople, J. A. and Beveridge, D.L., <i>Approximate Molecular Orbital Theory</i> , McGraw-Hill, 1970.
MINDO3	mind03_1.abp	Core-electron attraction parameters, orbital exponents	Dewar, M.J.S. <i>et al.</i> , <i>J. Am. Chem. Soc.</i> 97:1285, 1975.
	mind03_2.abp	Electron repulsion parameters and ionization potentials	
	mind03_3.abp	Core-core repulsion and resonance parameter	Dewar, M.J.S. <i>et al.</i> , QCPE #506 <sup>a</sup> .
MNDO	mndo_1.abp	Alpha and beta parameters, orbital exponents	Dewar, M.J.S. and Theil, W., <i>J. Am. Chem. Soc.</i> 99:4899, 1977.
	mndo_2.abp	Electron repulsion parameters	Dewar, M.J.S. <i>et al.</i> , QCPE #506 <sup>a</sup> .

Method	Filename	Contents	Reference
AM1	am1_1.abp	Core-electron attraction and resonance parameters, orbital exponents	Dewar, M.J.S., et al., <i>J. Am. Chem. Soc.</i> 107:3902, 1985; Dewar, M.J.S., et al., QCPE #506 <sup>a</sup> .
	am1_2.abp	Electron repulsion parameters	
	am1_3.abp	Core-core repulsion parameters	
PM3	PM3_1.abp	Core-electron attraction parameters, orbital exponents	Stewart, J.J.P., <i>J. Comp. Chem.</i> 12:320, 1991; Stewart, J.J.P., et al., QCPE #506 <sup>a</sup> .
	PM3_2.abp	Electron repulsion parameters and ionization potentials	
	PM3_3.abp	Core-core repulsion and resonance parameters	
ZINDO/1	zindo1_1.abp	Orbital exponents, ionization potentials, etc.	Bacon, A.D.; Zerner, M.C., <i>Theor. Chim. Acta</i> 53: 21-54, 1979 and Anderson, W.P.; Edwards, W.D.; Zerner, M.C., <i>Inorgan. Chem.</i> 25: 2728, 1986
	zindo1_2.abp	Slater-Condon parameters, total electronic energies, and heat of formation of free atoms	
	zindo1_3.abp	ionization potentials and valence bond mixing coefficients	
ZINDO/S	zindos_1.abp	Orbital exponents, ionization potentials, etc.	Ridley, J.; Zerner, M.C., <i>Theor. Chim. Acta</i> 32:111-134, 1973; 42: 223-236, 1976; Zerner, et al., <i>J. Am. Chem. Soc.</i> 102:589, 1980
	zindos_2.abp	Slater-Condon parameters, total electronic energies, and heat of formation of free atoms	
	zindos_3.abp	ionization potentials and valence bond mixing coefficients	
	zindos_4.abp	Empirical coulomb integrals	
TNDO	default.abp or a user defined parameter file	Identical in form to CNDO/INDO files except they are indexed by atom type rather than atomic number	HyperChem Computational Chemistry Manual

a. Quantum Chemistry Program Exchange, Indiana University, Bloomington, IN

You may want to modify the parameter files when new and better parameters become available or you might add some new parame-



ters for testing purposes. The current parameter files include most of the parameters published in the literature. The contents of the files are described on the following pages.

## Extended Hückel Parameter File

The parameter file, `exhuckel.abp`, includes the parameters of 105 real elements and 105 phoney fluorines. Each line in the file corresponds to the parameters for a particular element indexed by its atom number. The phoney fluorines are used in mixed mode calculations. To distinguish phoney fluorines from the real elements, the phoney fluorines are labeled with negative atomic numbers.

The parameters for each element have also been extended and include nine data items because of the use of double- $\zeta$  for the Slater d-orbitals for the transition metals.

### `exhuckel.abp` File Format

The description of each column in the `exhuckel.abp` parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number, negative values correspond to capping atoms
2	$\zeta_s$	s orbital Slater exponent (a.u.)
3	$\zeta_p$	p orbital Slater exponent (a.u.)
4	$\zeta_d$	d orbital Slater exponent (a.u.)
5	$I_s$	s orbital ionization potential (eV)
6	$I_p$	p orbital ionization potential (eV)
7	$I_d$	d orbital ionization potential (eV)
8	$C_1$	coefficient of Slater exponent $\zeta_d$

9	$\zeta_{d2}$	d orbital Slater exponent (a.u.)
10	$C_2$	coefficient of Slater exponent $\zeta_{d2}$

### References for Parameters in **exhuckel.abp**

1. R.Hoffmann, *J. Chem. Phys.* **39** (1963), 1397.
2. A.B.Anderson, R.Hoffmann, *J. Chem. Phys.* **60** (1974), 4271.
3. A.R.Rossi, R.Hoffmann, *Inorg. Chem.* **14** (1975), 365.
4. P.J.Hay, J.C.Thibeault, R.Hoffmann, *J. Am. Chem. Soc.* **97** (1975), 4884.
5. M.Elian, R.Hoffmann, *Inorg. Chem.* **14** (1975), 1058.
6. R.H.Summerzille, R.Hoffmann, *J. Am. Chem. Soc.* **98** (1976), 7240.
7. J.W.Lauher, R.Hoffmann, *J. Am. Chem. Soc.* **98** (1976), 1729.
8. M.M.L.Chen, R.Hoffmann, *J. Am. Chem. Soc.* **98** (1976), 1647.
9. J.W.Lauher, M.Elian, R.H.Summerville, R.Hoffmann, *J. Am. Chem. Soc.* **98** (1976), 3219.
10. T.A.Albright, P.Hofmann, R.Hoffmann, *J. Am. Chem. Soc.* **99** (1977), 7546.
11. D.L.Dubois, R.Hoffmann, *Nouv. J. Chim.* **1** (1977), 479.
12. S.Komiya, T.A.Albright, R.Hoffmann, J.K.Kochi, *J. Am. Chem. Soc.* **99** (1977), 8440.
13. D.L.Thorn, R.Hoffmann, *Inorg. Chem.* **17** (1978), 126.
14. N.Trong Anh, M.Elian, Hoffmann, *J. Am. Chem. Soc.* **100** (1978), 110.
15. L.L.Lohr, P.Pykko, *Chem. Phys. Lett.* **62** (1979), 333.
16. A.Dedieu, T.A.Albright, R.Hoffmann, *J. Am. Chem. Soc.* **101** (1979), 3141.
17. R.Hoffmann, S.Shaik, J.C.Scott, M.-H.Whangbo, M.J.Foshee, *J. Solid State Chem.* **34** (1980), 263.
18. K.Tatsummi, R.Hoffmann, *Inorg. Chem.* **19** (1980), 2656.

19. K.Tatsumi, R.Hoffmann, A.Yamamoto, J.K.Stille, *Bull. Chem. Soc. Jpn.* **54** (1981), 1857.
20. J.Hinze. H.H.Jaffe, *J. Chem. Phys.* **67** (1963), 1501.
21. K.Tatsumi, R.Hoffmann, *J. Am. Chem. Soc.* **103** (1981), 3340.
22. P.Kubacek, R.Hoffmann, Z.Havlas, *Organometallics* **1** (1982), 180.
23. D.M.Hoffman, R.Hoffmann, C.R.Fiesel, *J. Am. Chem. Soc.* **104** (1982) 3856.
24. T.Hughbanks, R.Hoffmann, M.-H.Whangbo, K.R.Stewart, O. Eisenstein, E.Canadell, *J. Am. Chem. Soc.* **104** (1982) 3876.
25. E.Canadell, O.Eisenstein, *Inorg. Chem.* **22** (1983) 2398.
26. J.Silvestre. T.A.Albright, *Isr. J. Chem.* **23** (1983), 139.
27. E.Canadell, O.Eisenstein, *Inorg. Chem.* **22** (1983), 3856.
28. G.Trinquier, R.Hoffmann, *J. Phys. Chem.* **88** (1984), 6696.
29. M.Kertesz, R.Hoffmann, *J. Am. Chem. Soc.* **106** (1984), 3453.
30. R.Hoffmann, C.Minot, H.B.Gray, *J. Am. Chem. Soc.* **106** (1984), 2001.
31. M.-H.Whangbo, P.Gressier, *Inorg. Chem.* **23** (1984), 1228.
32. J.-Y Saillard, R.Hoffmann, *J. Am. Chem. Soc.* **106** (1984), 2006.
33. E.Canadell, O.Eisenstein, J.Rubio, *Organometallics* **3** (1984), 759.
34. E.Canadell, O.Eisenstein, T.Hughbanks, *Inorg. Chem.* **23** (1984), 2435.
35. M.-H.Whangbo, W.M.Walsh, R.C.Haddon, F.Wudl, *Solid State Comm.* **43** (1982), 637.
36. O.P.Charkin, *Russ. J. Inorg. Chem. (Engl. Transl.)* **19** (1974), 1590.
37. S.P.McGlynn, L.G.Vanquickenborne, M.Kinoshita, D.G.Carroll, *Introduction to Applied Quantum Chemistry*, Holt, Rinehart and Winston, Inc. New York, 1977, pp. 423-431.
38. W.A.Harrison, *Electronic Structure and the Properties of Solids*, W.Freeman and Co., San Francisco, 1980, p. 551.

## CNDO and INDO Parameter File

This parameter file, `cindo.abp`, includes the parameters of 54 real elements and 54 phoney fluorines. The phoney fluorines are used in mixed mode calculations. The number of parameters for each element is 12.

Each line in `cindo.abp` corresponds to the parameters for a particular element indexed by its atom number. The parameters with negative atomic numbers correspond to the parameters for the capping atoms.

### `cindo.abp` File Format

The description of each column of the parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number, negative atomic numbers for capping atoms <sup>a</sup>
2	$\zeta_s$	s orbital Slater exponent (a.u.)
3	$\zeta_p$	p orbital Slater exponent (a.u.)
4	$\zeta_d$	d orbital Slater exponent (a.u.)
5	$-\frac{1}{2}(I_s + A_s)$	Negative of averaged ionization potential and electron affinity of s orbital (eV)
6	$-\frac{1}{2}(I_p + A_p)$	Negative of averaged ionization potential and electron affinity of p orbital (eV)
7	$-\frac{1}{2}(I_d + A_d)$	Negative of averaged ionization potential and electron affinity of d orbital (eV)
8	b	Bonding parameters (eV)
9	F <sup>2</sup>	Slater-Condon parameters F <sup>2</sup> (p,p) (a.u.)
10	G <sup>1</sup>	Slater-Condon parameters G <sup>1</sup> (s,p) (a.u.)

Column	Symbol	Content
11	$E^{\text{atom}}$	Electron energy of isolated atom for CNDO <sup>b</sup> (eV)
12	$E^{\text{atom}}$	Electron energy of isolated atom for INDO <sup>b</sup> (eV)
13	$H^{\text{atom}}$	Heat of formation of free atom (kcal/mol)

- a. Parameterized fluorine atoms that represent atoms bonded to the atoms selected for the calculation (see *HyperChem Computational Chemistry, Theory and Methods*).
- b. Calculated from the parameters listed in this file for the corresponding ground states.

The electronic energy of isolated atoms depends on the parameters in both CNDO and INDO. The electronic energies of isolated atoms in the `cindo.abp` file were calculated in terms of the parameters listed in this parameter file for the corresponding ground states, that is, the spin multiplicities are 1 for Be, 2 for H, Li, B, F, Na, Al, and Cl, 3 for C, O, Si, and S, and 4 for N in the RHF method in both CNDO and INDO.

## MINDO/3 Parameter Files

The parameter files for this method include 54 real elements and 54 phoney fluorines. The phoney fluorines are used in mixed mode calculations.

The third parameter file `mind3_3.abp`, only includes the bond parameters in the MINDO3 approximation level published in the literature. The first two entries in `mind3_3.abp` define a pair of elements corresponding to the bond parameters. Negative numbers define phoney fluorines.

If the bond parameters appear more than once in this parameter file for the same pair of atoms, the parameters that appear last are used in MINDO3 calculations. The maximum number of pairs of atoms can be  $108 * (108 - 1) / 2 + 107$ . The absolute value of the atomic number must be larger than zero and less than or equal to 54.

**mind03\_1.abp File Format**

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number, negative values denote capping atoms
2	$U_{ss}$	One-center core-electron attraction, plus kinetic energy
3	$U_{pp}$	One-center core-electron attraction, plus kinetic energy
4	$U_{dd}$	One-center core-electron attraction, plus kinetic energy
5		reserved
6		reserved
7		reserved
8	$\zeta_s$	s orbital Slater exponent
9	$\zeta_p$	p orbital Slater exponent
10	$\zeta_d$	d orbital Slater exponent
11		reserved
12	$E^{\text{atom}}$	Ground state electronic energy of isolated atom
13	$H^{\text{atom}}$	Heat of formation of free atom

## mind03\_2.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number, negative values denote capping atoms
2	$G_{ss}$	One-center electron repulsion integral, $G_{ss} = (ss   ss)$
3	$G_{sp}$	One-center electron repulsion integral, $G_{sp} = (ss   pp)$
4	$G_{pp}$	One-center electron repulsion integral, $G_{pp} = (pp   pp)$
5	$G_{p2}$	One-center electron repulsion integral, $G_{p2} = (pp   p'p')$
6	$H_{sp}$	One-center electron repulsion integral, $H_{sp} = (sp   sp)$
7		reserved
8		reserved
9		reserved
10	$I_s$	s orbital ionization potential
11	$I_p$	p orbital ionization potential
12	$I_d$	d orbital ionization potential
13	A	Averaged one center, two electron integral used in calculating two center, two electron integral
14		reserved

### mindo3\_3.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1, 2	$Z_1, Z_2$	Atomic numbers defining pair of elements; negative value denotes capping atom
3	$\alpha_{AB}$	Bond parameter used in calculating core-core repulsion integral
4	$\beta_{AB}$	Bond parameter used in calculating resonance integral

### MNDO and MNDO/d Parameter Files

Files for MNDO and MNDO/d have the same format.

#### mindo\_1.abp or mndod\_1.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	$Z$	Atomic number
2	$U_{ss}$	One-center core-electron attraction, plus kinetic energy
3	$U_{pp}$	One-center core-electron attraction, plus kinetic energy
4	$U_{dd}$	One-center core-electron attraction, plus kinetic energy
5	$\beta_s$	s orbital bond parameter used in calculating resonance integral



Column	Symbol	Content
6	$\beta_p$	p orbital bond parameter used in calculating resonance integral
7	$\beta_d$	d orbital bond parameter used in calculating resonance integral
8	$\zeta_s$	s orbital Slater exponent
9	$\zeta_p$	p orbital Slater exponent
10	$\zeta_d$	d orbital Slater exponent
11	a	Parameter used in core-core repulsion integral
12	$E^{\text{atom}}$	Ground state electronic energy of isolated atom
13	$H^{\text{atom}}$	Heat of formation of free atom

## **mndo\_2.abp or mndod\_2.abp File Format**

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number
2	$G_{ss}$	One-center electron repulsion integral, $G_{ss} = (ss   ss)$
3	$G_{sp}$	One-center electron repulsion integral, $G_{sp} = (ss   pp)$
4	$G_{pp}$	One-center electron repulsion integral, $G_{pp} = (pp   pp)$

Column	Symbol	Content
5	$G_{p2}$	One-center electron repulsion integral, $G_{p2} = (pp   p'p')$
6	$H_{sp}$	One-center electron repulsion integral, $H_{sp} = (sp   sp)$
7		reserved
8		reserved
9		reserved
10	$D_1$	Distance of a dipole charge from its nucleus
11	$D_2$	Half the distance of a linear quadrupole charge from its nucleus
12	AM	Monopole-monopole interaction parameter
13	AD	Dipole-dipole interaction parameter
14	AQ	Quadrupole-quadrupole interaction parameter

## AM1 and PM3 Parameter Files

AM1 and PM3 parameter files share a common format, described below. The AM1 file names start with “am1\_” and the PM3 file names start with “pm3\_”.

### am1\_1.abp and pm3\_1.abp File Format

The description of each column of these parameter files is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number

Column	Symbol	Content
2	$U_{ss}$	One-center core-electron attraction, plus kinetic energy
3	$U_{pp}$	One-center core-electron attraction, plus kinetic energy
4	$U_{dd}$	One-center core-electron attraction, plus kinetic energy
5	$\beta_s$	s orbital bond parameter used in calculating resonance integral
6	$\beta_p$	p orbital bond parameter used in calculating resonance integral
7	$\beta_d$	d orbital bond parameter used in calculating resonance integral
8	$\zeta_s$	s orbital Slater exponent
9	$\zeta_p$	p orbital Slater exponent
10	$\zeta_d$	d orbital Slater exponent
11	$\alpha$	Parameter used in core-core repulsion integral
12	$E^{\text{atom}}$	Ground state electronic energy of isolated atom
13	$H^{\text{atom}}$	Heat of formation of free atom

**am1\_2.abp and pm3\_2.abp File Format**

The description of each column of these parameter files is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number
2	$G_{ss}$	One-center electron repulsion integral, $G_{ss} = (ss   ss)$
3	$G_{sp}$	One-center electron repulsion integral, $G_{sp} = (ss   pp)$
4	$G_{pp}$	One-center electron repulsion integral, $G_{pp} = (pp   pp)$
5	$G_{p2}$	One-center electron repulsion integral, $G_{p2} = (pp   p'p')$
6	$H_{sp}$	One-center electron repulsion integral, $H_{sp} = (sp   sp)$
7		reserved
8		reserved
9		reserved
10	$D_1$	Distance of a dipole charge from its nucleus
11	$D_2$	Half the distance of a linear quadrupole charge from its nucleus
12	AM	Monopole-monopole interaction parameter
13	AD	Dipole-dipole interaction parameter
14	AQ	Quadrupole-quadrupole interaction parameter

## am1\_3.abp and pm3\_3.abp File Format

In these files, the parameters in three successive (referred to as line 1-3 in the next table) lines apply only to one element. Each line contains 11 columns.

Column	Line	Symbol	Content
1	1-3	Z	Atomic number
2-11	1	$a_k$ ( $k = 1, 2 \dots 10$ )	Parameters used in calculating core-core interactions
	2	$b_k$ ( $k = 1, 2 \dots 10$ )	
	3	$c_k$ ( $k = 1, 2 \dots 10$ )	

## ZINDO/1 Parameter Files

### zindo1\_1.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponds to the capping atom)
2	$\zeta_s$	s-orbital Slater exponent for computing the overlap and coulomb integrals (a.u.)
3	$\zeta_p$	p-orbital Slater exponent for computing the overlap and coulomb integrals (a.u.)
4	$\zeta_d$	d-orbital Slater exponent for computing the coulomb integrals only (a.u.)
5	$I_s$	Ionization potential of s orbital (eV)
6	$I_p$	Ionization potential of p orbital (eV)

Column	Symbol	Content
7	$I_d$	ionization potential of d orbital (eV)
8	$\beta_s$	s-orbital bond parameter used in calculating resonance integral (eV)
9	$\beta_p$	p-orbital bond parameter used in calculating resonance integral (eV)
10	$\beta_d$	d-orbital bond parameter used in calculating resonance integral (eV)
11	$\zeta_{d1}$	d-orbital Slater exponent 1 for computing the overlap integrals (a.u.)
12	$C_1$	coefficient corresponding to the Slater exponent $\zeta_{d1}$
13	$\zeta_{d2}$	d-orbital Slater exponent 2 for computing the overlap integrals (a.u.)
14	$C_2$	coefficient corresponding to the Slater exponent $\zeta_{d2}$

### zindo1\_2.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponding to the capping atom)
2	$F^2$	Slater-Condon parameter $F^2(p,p)$ (a.u.)
3	$G^1$	Slater-Condon parameter $G^1(s,p)$ (a.u.)

Column	Symbol	Content
4	$F^2$	Slater-Condon parameter $F^2(p,d)$ (a.u.)
5	$F^2$	Slater-Condon parameter $F^2(d,d)$ (a.u.)
6	$F^4$	Slater-Condon parameter $F^4(d,d)$ (a.u.)
7	$G^1$	Slater-Condon parameter $G^1(p,d)$ (a.u.)
8	$G^2$	Slater-Condon parameter $G^2(s,d)$ (a.u.)
9	$G^3$	Slater-Condon parameter $G^3(p,d)$ (a.u.)
10	$E^{\text{atom}}$	Electronic energy of isolated atom (eV)
11	$H^{\text{atom}}$	Heat of formation of free atom (kcal/mol)

### zindo1\_3.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponding to the capping atom)
2	$I_s^a$	Second set of ionization potential of s-orbital, $3d^{n-1} 4s \rightarrow 3d^{n-1}$ in (eV)
3	$I_p^a$	Second set of ionization potential of p-orbital, $3d^{n-1} 4p \rightarrow 3d^{n-1}$ in (eV)
4	$I_d^a$	Second set of ionization potential of d-orbital, $3d^{n-1} 4s \rightarrow 3d^{n-2} 4s$ in (eV)
5		reserved
6		reserved

Column	Symbol	Content
7		reserved
8	$C_1^a$	Fractional contribution of the $d^{n-2}s^2$ configuration to the core integral
9	$C_2^a$	Fractional contribution of the $d^{n-1}s(p)$ configuration to the core integral
10	$C_3^a$	Fractional contribution of the $d^n$ configuration to the core integral

a. Applies only to the first row transition metals. Only one set of ionization potentials has been used in HyperChem for the second row transition metals. This set is obtained by averaging the ionization potentials from different configurations of the ground state of the corresponding atoms (see e.g., Anderson, W.P.; Cundari, T.R.; Zerner, M.C., *Int. J. Quantum Chem.* 39: 31-45, 1991).

## ZINDO/S Parameter Files

### zindos\_1.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number, negative value corresponding to the capping atom
2	$\zeta_s$	s-orbital Slater exponent for computing the overlap integrals (a.u.)
3	$\zeta_p$	p-orbital Slater exponent for computing the overlap integrals (a.u.)
4		reserved
5	$I_s$	Ionization potential of s orbital (eV)



Column	Symbol	Content
6	$I_p$	Ionization potential of p orbital (eV)
7	$I_d$	Ionization potential of d orbital (eV)
8	$\beta_s$	s-orbital bond parameter used in calculating resonance integral (eV)
9	$\beta_p$	p-orbital bond parameter used in calculating resonance integral (eV)
10	$\beta_d$	d-orbital bond parameter used in calculating resonance integral (eV)
11	$\zeta_{d1}$	d-orbital Slater exponent 1 for computing the overlap integrals (a.u.)
12	$C_1$	coefficient corresponding to the Slater exponent $\zeta_{d1}$
13	$\zeta_{d2}$	d-orbital Slater exponent 2 for computing the overlap integrals
14	$C_2$	coefficient corresponding to the Slater exponent $\zeta_{d2}$

### **zindos\_2.abp File Format**

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponding to the capping atom)
2	$F^2$	Slater-Condon parameter $F^2(p,p)$ (a.u.)
3	$G^1$	Slater-Condon parameter $G^1(s,p)$ (a.u.)
4	$F^2$	Slater-Condon parameter $F^2(p,d)$ (a.u.)

Column	Symbol	Content
5	$F^2$	Slater-Condon parameter $F^2(d,d)$ (a.u.)
6	$F^4$	Slater-Condon parameter $F^4(d,d)$ (a.u.)
7	$G^1$	Slater-Condon parameter $G^1(p,d)$ (a.u.)
8	$G^2$	Slater-Condon parameter $G^2(s,d)$ (a.u.)
9	$G^3$	Slater-Condon parameter $G^3(p,d)$ (a.u.)
10	$E^{\text{atom}}$	Electronic energy of isolated atom (eV)
11	$H^{\text{atom}}$	Heat of formation of free atom (kcal/mol)

### zindos\_3.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponding to the capping atom)
2	$I_s^a$	Second set of ionization potential of s-orbital, $3d^{n-1} 4s \rightarrow 3d^{n-1}$ in (eV)
3	$I_p^a$	Second set of ionization potential of p-orbital, $3d^{n-1} 4p \rightarrow 3d^{n-1}$ in (eV)
4	$I_d^a$	Second set of ionization potential of d-orbital, $3d^{n-1} 4s \rightarrow 3d^{n-2} 4s$ in (eV)
5		reserved
6		reserved
7		reserved

Column	Symbol	Content
8	$C_1$	Fractional contribution of the $d^{n-2}s^2$ configuration to the core integral
9	$C_2$	Fractional contribution of the $d^{n-1}s(p)$ configuration to the core integral
10	$C_3^a$	Fractional contribution of the $d^n$ configuration to the core integral

a. Applies only to the first row transition metals. Only one set of ionization potentials has been used in HyperChem for the second row transition metals. This set is obtained by averaging the ionization potentials from different configurations of the ground state of the corresponding atoms (see e.g., Anderson, W.P.; Cundari, T.R.; Zerner, M.C., *Int. J. Quantum Chem.* 39:31-45, 1991).

### zindos\_4.abp File Format

The description of each column of this parameter file is shown in the following table.

Column	Symbol	Content
1	Z	Atomic number (negative value corresponding to the capping atom)
2	$\gamma_{ss}$	Two-electron one-center coulomb integral, in eV
3	$\gamma_{\sigma\delta}$	Two-electron one-center coulomb integral, in eV
4	$\gamma_{dd}$	Two-electron one-center coulomb integral, in eV

## TNDO Parameter Files

TNDO and TNDO parameter files have a significantly different role in HyperChem than with the other semi-empirical methods. TNDO is designed to incorporate many different parameter sets each of which is designed to give accurate results relative to other semi-empirical methods but for a specific property. Thus one could derive a parameter set for predicting geometry, a parameter set for predicting UV spectra, a parameter set for predicting NMR coupling, and even a different parameter set for predicting NMR shielding than coupling.

### The Default TNDO Parameter Set

A TNDO parameter set is a single file that has exactly the same format as a CNDO/INDO parameter file except that the atomic number (first column) is replaced by the atom type. TNDO atom types are identical to Amber (amber2, i.e. “old amber”) atom types. There are expected to be many such files, one each for each property parameterized in the TNDO way. A file, default.abp, is included with HyperChem which when used with TNDO give exactly the CNDO or INDO results (TNDO/1 is CNDO and TNDO/2 is INDO).

### Improved Parameter Sets

The file default.abp is not a “good” TNDO parameter set since CNDO and INDO do not give accurate semi-empirical results compared to newer methods like AM1 and PM3. However, when default.abp is replaced by a parameter set which has been optimized for a property like molecular geometry, then TNDO will give even more accurate results than AM1 or PM3. As time goes on it can be expected that a variety of TNDO parameter sets will become available. A preliminary set tndo.abp has been partially optimized for molecular geometry. Parameter sets are available for NMR shielding and coupling in conjunction with the HyperNMR module.

The simplest approach to improving on default.abp is to separately scale the diagonal elements (columns 5,6, and 7 below) and off-diagonal elements (column 8 below) to improve the results for a particular property. HyperChem includes the capability of simply copying a parameter file to a new file name (see the Select Parame-

ter Set menu item) to assist in working with a variety of parameter sets.

**default.abp TNDO File Format**

Column	Symbol	Content
1	Type	Atom Type, e.g. HC; a negative type, e.g. -HC denotes a capping atom <sup>a</sup>
2	$\zeta_s$	s orbital Slater exponent (a.u.)
3	$\zeta_p$	p orbital Slater exponent (a.u.)
4	$\zeta_d$	d orbital Slater exponent (a.u.)
5	$-\frac{1}{2}(I_s + A_s)$	Negative of averaged ionization potential and electron affinity of s orbital (eV)
6	$-\frac{1}{2}(I_p + A_p)$	Negative of averaged ionization potential and electron affinity of p orbital (eV)
7	$-\frac{1}{2}(I_d + A_d)$	Negative of averaged ionization potential and electron affinity of d orbital (eV)
8	b	Bonding parameters (eV)
9	$F^2$	Slater-Condon parameters $F^2$ (p,p) (a.u.)
10	$G^1$	Slater-Condon parameters $G^1$ (s,p) (a.u.)
11	$E^{\text{atom}}$	Electron energy of isolated atom for CNDO <sup>b</sup> (eV)
12	$E^{\text{atom}}$	Electron energy of isolated atom for INDO <sup>b</sup> (eV)
13	$H^{\text{atom}}$	Heat of formation of free atom (kcal/mol)

- a. Parameterized fluorine atoms that represent atoms bonded to the atoms selected for the calculation (see *HyperChem Computational Chemistry*, Theory and Methods).
- b. Calculated from the parameters listed in this file for the corresponding ground states.

## Basis Set Files

The file name of a basis set file in HyperChem can be any name acceptable by MS-Windows that ends in '.BAS'.

### Basis Set File Format

HyperChem uses its own format to define a basis set. All the standard and commonly used basis sets can be fitted into this basis set format. Users can also define their own basis sets by using the HyperChem basis set format.

#### Basis Set

A basis function for a element includes the following items:

- Element symbol
- Number of shells
- Shell types
- Number of primitive Gaussians for each shell
- Scaling factor for each shell
- Exponent and contraction coefficients for each primitive Gaussian.

*Example:*

C:

;

Shells = 3

Shell1 =[ S 3 1.00 ]

1	0.172256E+03	0.617669E-01
2	0.259109E+02	0.358794E+00
3	0.553335E+01	0.700713E+00

Shell2 =[ SP 2 1.00 ]

```

1  0.366498E+01 -0.395897E+00  0.236460E+00
2  0.770545E+00  0.121584E+01  0.860619E+00
Shell3 =[ SP 1  1.00 ]
1  0.195857E+00  0.100000E+01  0.100000E+01

```

This example defines a 3-21G basis function for carbon. There are 3 shells. The first shell is a S type and contracted using 3 primitive Gaussians with a scaling factor of 1.0; the second shell is an SP type and contracted using 2 primitive Gaussians with a scaling factor of 1.0; and the third shell is a SP type and contracted using only 1 primitive Gaussian with a scaling factor of 1.0.

The types of shells are supported in HyperChem Release 5 are S, P, D, SP, and SPD.

## Syntax

This is the structure for defining a basis for an element:

```

<element-symbol>:
; <comment>
shells = <# number of shells for this element>
shell1 = [ <Ta> <##b> <sc> ]
<primitive 1> <exponent> <coefficientsd>
<primitive 2> <exponent> <coefficientsd>
... (repeat ## times for the primitives for this shell)
... (repeat # times for all the shells for this element)

```

- a. Shell type
- b. Number of primitive Gaussians for this shell
- c. Scaling factor for this shell
- d. Coefficients for this primitive Gaussian. The number of coefficients is one for a S, P, or D type of primitive Gaussian, two for a SP, type of primitive Gaussian, and three for a SPD type of primitive Gaussian



## Partial List of Basis Set Files

HyperChem includes many standard and commonly used basis set files. The following table is an overview of some of the more common of these files:

Filename	Contents	Reference
3-21g.bas	Definitions and contractions for 3-21G basis set	J. S. Binkley, J. A. Pople, and W. J. Hehre, <i>J. Am. Chem. Soc.</i> , <b>102</b> , 939 (1980). M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre, <i>J. Am. Chem. Soc.</i> , <b>104</b> , 2797 (1982).
3-21gs.bas	Definitions and contractions for 3-21G* basis set	W. J. Pietro, M. M. Francl, W. J. Hehre, D. J. Defrees, J. A. Pople, and J. S Binkley, <i>J. Am. Chem. Soc.</i> , <b>104</b> , 5039 (1982).
3-21gss.bas	Definitions and contractions for 3-21G** basis set	
4-21g.bas	Definitions and contractions for 4-21G basis set	W. J. Hehre, R. Ditchfield, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>56</b> , 2257 (1972). P. C. Hariharan and J. A. Pople, <i>Theor. Chim. Acta</i> , <b>28</b> , 213 (1973).
4-21gs.bas	Definitions and contractions for 4-21G* basis set	M. S. Gordon, <i>Chem. Phys. Lett.</i> , <b>76</b> , 163 (1980).
4-21gss.bas	Definitions and contractions for 4-21G** basis set	
4-31g.bas	Definitions and contractions for 4-31G basis set	W. J. Hehre, R. Ditchfield, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>56</b> , 2257 (1972). P. C. Hariharan and J. A. Pople, <i>Theor. Chim. Acta</i> , <b>28</b> , 213 (1973).
4-31gs.bas	Definitions and contractions for 4-31G* basis set	M. S. Gordon, <i>Chem. Phys. Lett.</i> , <b>76</b> , 163 (1980).
4-31gss.bas	Definitions and contractions for 4-31G** basis set	

Filename	Contents	Reference
5-31g.bas	Definitions and contractions for 5-31G basis set	W. J. Hehre, R. Ditchfield, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>56</b> , 2257 (1972). P. C. Hariharan and J. A. Pople, <i>Theor. Chim. Acta</i> , <b>28</b> , 213 (1973).
5-31gs.bas	Definitions and contractions for 5-31G* basis set	M. S. Gordon, <i>Chem. Phys. Lett.</i> , <b>76</b> , 163 (1980).
5-31gss.bas	Definitions and contractions for 5-31G** basis set	
6-21g.bas	Definitions and contractions for 6-21G basis set	J. S. Binkley, J. A. Pople, and W. J. Hehre, <i>J. Am. Chem. Soc.</i> , <b>102</b> , 939 (1980). M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre, <i>J. Am. Chem. Soc.</i> , <b>104</b> , 2797 (1982).
6-21gs.bas	Definitions and contractions for 6-21G* basis set	W. J. Pietro, M. M. Francl, W. J. Hehre, D. J. Defrees, J. A. Pople, and J. S. Binkley, <i>J. Am. Chem. Soc.</i> , <b>104</b> , 5039 (1982).
6-21gss.bas	Definitions and contractions for 6-21G** basis set	
6-31g.bas	Definitions and contractions for 6-31G basis set	W. J. Hehre, R. Ditchfield, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>56</b> , 2257 (1972). P. C. Hariharan and J. A. Pople, <i>Theor. Chim. Acta</i> , <b>28</b> , 213 (1973).
6-31gs.bas	Definitions and contractions for 6-31G* basis set	M. S. Gordon, <i>Chem. Phys. Lett.</i> , <b>76</b> , 163 (1980).
6-31gss.bas	Definitions and contractions for 6-31G** basis set	

Filename	Contents	Reference
6-311g.bas	Definitions and contractions for 6-311G basis set	R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>72</b> , 650 (1980). A. D. McLean and G. S. Chandler, <i>J. Chem. Phys.</i> , <b>72</b> , 5639 (1980).
6-311gs.bas	Definitions and contractions for 6-311G* basis set	
6-311gss.bas	Definitions and contractions for 6-311G** basis set	
sto-1g.bas	Definitions and contractions for STO-1G basis set	W. J. Hehre, R. F. Stewart, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>51</b> , 2657 (1969). J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>64</b> , 5142 (1976).
sto-1gs.bas	Definitions and contractions for STO-1G* basis set	R. F. Stewart, <i>J. Chem. Phys.</i> , <b>52</b> , 431 (1970).
sto-2g.bas	Definitions and contractions for STO-2G basis set	
sto-2gs.bas	Definitions and contractions for STO-2G* basis set	
sto-3g.bas	Definitions and contractions for STO-3G basis set	W. J. Hehre, R. F. Stewart, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>51</b> , 2657 (1969). J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>64</b> , 5142 (1976).
sto-3gs.bas	Definitions and contractions for STO-3G* basis set	R. F. Stewart, <i>J. Chem. Phys.</i> , <b>52</b> , 431 (1970).

Filename	Contents	Reference
sto-4g.bas	Definitions and contractions for STO-4G basis set	W. J. Hehre, R. F. Stewart, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>51</b> , 2657 (1969).
sto-4gs.bas	Definitions and contractions for STO-4G* basis set	J. B. Collins, P. V. Schleyer, J. S. Binkley, and J. A. Pople, <i>J. Chem. Phys.</i> , <b>64</b> , 5142 (1976). R. F. Stewart, <i>J. Chem. Phys.</i> , <b>52</b> , 431 (1970).
sto-5g.bas	Definitions and contractions for STO-5G basis set	
sto-5gs.bas	Definitions and contractions for STO-5G* basis set	
sto-6g.bas	Definitions and contractions for STO-6G basis set	
sto-6gs.bas	Definitions and contractions for STO-6G* basis set	
d95.bas	Definitions and contractions for D95 basis set	T.H. Dunning and P.J. Hay, in <i>Modern Theoretical Chemistry</i> , Plenum, New York, 1976.
d95s.bas	Definitions and contractions for D95* basis set	
d95ss.bas	Definitions and contractions for D95** basis set	

## Creating a New Basis Set

The procedure for creating a new basis set and adding to the list of basis sets in HyperChem is relatively straightforward.

To create a new basis set:

1. Create a file with the extension .BAS (e.g., MY.BAS) in the HyperChem Runfiles directory.
2. Edit this file with a text editor such as the Windows Notepad or Wordpad, and type in the information about the basis set: element symbols, number of shells for each element, shell types, number of primitives, scaling factors, and exponents and coefficients for each primitive.

For more information about the format for the basis set files, see the preceding sections.

3. Add an entry for your file in the section [BasisSet] in the Registry (for Windows 95 or Windows NT) or CHEM.INI (for Windows 3.1x). In the Registry, this entry will have a name (e.g., mybasis) and a value which is the name of your file. In CHEM.INI, the entry is a line of the form 'name=filename', e.g., mybasis=my.bas .

After you restart HyperChem, you can use this new basis set.

To use this new basis set:

1. Create or read in a molecular system.
2. Set up the *ab initio* calculation as usual.
3. In the Ab Initio Method dialog box, use Assign Other Basis Set to select your new basis set for the Other text box.

You can now apply this basis set to the selected atoms, or if no atoms are selected, to all atoms in the workspace.



# Appendix D

## HIN Files

HyperChem can store a molecular system from the workspace, plus information about this system and display conditions, in a HIN file. HyperChem can also store a molecular system in a PDB file, but HIN files offer more options and greater convenience.

HIN files are text files. You can edit these files with any text editor, like Windows Notepad or Write. If you use an editor that can impose its own format, keep the HIN file in its original text format. For example, with Windows Write, choose the No conversion option when you open a HIN text file.

A special feature of HIN files is that records can be added or potentially missing without affecting the operation of HyperChem. That is, some records are optional and are not required by HyperChem. Secondly, if HyperChem detects a record in the file with which it is not familiar then it will simply ignore that record. This allows extensions to HIN file formats while maintaining backward compatibility.

HyperChem contains a primitive drawing program associated with the concept of annotations. The HIN file in addition to storing molecular information may also store information associated with these annotations (text, line, ellipse, and rectangle).

### Structure of a HIN File

A HIN file contains several records, each on a separate line. Each line, except for a comment, begins with a keyword (for example, atom). Comment lines begin with a semicolon (;).

This is an example of a minimal HIN file for a methane molecule:

```
;This sample HIN file contains one methane molecule.  
forcefield amber
```

```

sys 0
view 40 1.03211 40 4.44992 1 0 0 0 1 0 0 0 1 0.0848873 -0.669469 -40
mol 1 "methane"
atom 1 - C CT - 0 -0.341802 0.309469 -6.05498e-007 4 2 s 3 s 4 s 5 s
atom 2 - H HC - 0 -0.341802 1.39947 -6.05498e-007 1 1 s
atom 3 - H HC - 0 0.685861 -0.0538608 -6.05498e-007 1 1 s
atom 4 - H HC - 0 -0.855625 -0.0538608 0.889987 1 1 s
atom 5 - H HC - 0 -0.855636 -0.0538687 -0.889978 1 1 s
endmol 1

```

For a description of the records in this example and other possible records, see the following sections.

## File Syntax

This is the complete syntax of a HIN file. <sup>o</sup> indicates an optional record. Each record must begin on a separate line.

```

; <comment>o
forcefield <force-field-name>
syso <temperature> <charge> <multiplicity>
seedo <random seed>
viewo <view-distance> <view-scale> <slab-distance> <slab-
thickness> <viewing-transform> . . . <viewing-transform>
boxo <x> <y> <z>
mol <mol#> <mol-name>
reso <res#> <res-name> <PDB#> <previous-res#> <next-res#>
atom <at#> <atom-name> <element> <type>
<flags> <at-charge> <x> <y> <z> <cn> <nbor# nbor-bond>1
  veo <at#> <x> <y> <z>
  masso <at#> <mass>
  basisseto <at#> <basis set>
  formalchargeo <at#> <formal charge>
atom . . .
  veo . . .
  masso . . .

```

1. This variable can be repeated up to twelve times.



```

basisset0 . . .
formalcharge0 . . .
atom . . .
  ve0 . . .
  mass0 . . .
  basisset0 . . .
  formalcharge0 . . .
. . .
endres0 <res#>
endmol <mol#>
selection0 <sel-name> <atoms>
selectrestraint0 <flag> <force-constant> <restr-value>
selectatom0 <at#> <mol#>
selectatom0 . . .
selectatom0 . . .
. . .
endselection0 <sel-name>
dynamics0 <snap-shot-file-name>
aromaticring0 <size> <mol# atom#>1
text0 <quoted text> <quoted font> <red> <green> <blue> <color
index> <point size> <centered flag> <rect pos> <weight> <italics>
<underline> <strikeout> <charset> <pitch>
line0 <red> <green> <blue> <line width> <line type> <rect pos>
circle0 <red> <green> <blue> <line width> <line type> <rect pos> <fill
flag>
rectangle0 <red> <green> <blue> <line width> <line type> <rect pos>
<fill flag>

```

## **; comment**

Each line of comments must have a semicolon at the beginning of the line. You can add and review comments from HyperChem by

1. This variable can be repeated up to 6 times.

using the Open File and Save File dialog boxes; you can also add and review comments by using any text editor.

Add comment lines anywhere in a HIN file. When you save the file in HyperChem, all comments move to the start of the file.

## forcefield

Syntax: *forcefield* <mm+, amber, **amber94**, bio+, **chamm22**, or *opls*>

HyperChem assigns a force field and the corresponding atom types to any molecular system in the workspace. You choose the force field from the Molecular Mechanics Force Field dialog box (choose Molecular Mechanics on the Setup menu).

HyperChem needs atom types for all molecular mechanics calculations, but not for semi-empirical quantum mechanics calculations.

When you draw a new molecule, HyperChem assigns \*\* as the “type” for each atom, indicating no atom type. When you choose Model Build or Calculate Types on the Build menu, or when you change force fields, HyperChem assigns appropriate atom types for the force field you chose. If you open a HIN file, HyperChem uses the force field and atom types that were in effect when you saved the file.

The variables for the *forcefield* keyword are mm+, amber, bio+, or opsl. If you do not assign a force field to a molecular system, HyperChem uses the last forcefield that you chose. However, atom type assignment is not automatic when you open the file.

## sys

Syntax: *sys* <temperature> <charge> <multiplicity>

The system record gives the temperature of the molecular system in Kelvin plus the total charge and multiplicity of the system.

**temperature** The default is 0 Kelvin. If you save the molecular system in a HIN file after a molecular dynamics calculation, this record stores the final temperature of the system corresponding to the atomic velocities.

- charge** This is the total net charge on the whole molecular system. It is the value used by quantum mechanical calculations to assign the total number of electrons. It corresponds to the script value `total-quantum-charge`. It has a default value of 0.
- multiplicity** This is the spin multiplicity of the system, i.e. it corresponds to singlet (1), doublet (2), triplet (3) or quartet (4). It is an option to setting up quantum mechanical calculations.

## seed

Syntax: `seed <random seed>`

The random seed used for Molecular Dynamics, Langevin Dynamics, and Monte Carlo. After any use of the random seed, a new random seed is generated.

**random seed** An integer in the range -32768 to 32767.

## view

Syntax: `view <view-distance> <view-scale> <slab-distance> <slab-thickness> <viewing-transform> . . . <viewing-transform>`

A view record gives instructions for transforming a 3D structure into 2D space.

- view-distance** The distance, in Ångstroms, from the viewer to the origin of the molecular system.
- view-scale** The magnification factor.
- slab-distance** The distance, in Ångstroms, from the viewer to the middle of the clipping slab.
- slab-thickness** The distance between the front and back clipping planes, in Ångstroms.
- viewing-transform** These 12 fields describe a 3 by 4 viewing transformation matrix, with four rows and three columns. The fields appear row by row. A fourth column, normally describing perspective, is not present.

## box

Syntax: *box* <x> <y> <z>

A box record reports the presence and x, y, and z dimensions of a periodic box, in Ångstroms, in the molecular system.

## mol

Syntax: *mol* <mol#> <mol-name>

A mol record starts the description of each molecule in the molecular system. An endmol record must end each molecule.

mol#	The number of the molecule in the molecular system, starting with 1. The number reflects the order the molecules were added to the system. Default = 1.
mol-name	Up to 64 characters describing a molecule. The name must be enclosed in double quotes. This name appears on the status line when you select the molecule. You can assign this name only by using the Name Molecule menu item.

## res

Syntax: *res* <res#> <res-name> <PDB#> <previous-res#> <next-res#>

A res record starts the description of each residue in a molecule. An endres record must end each residue.

res#	The number of the residue in the molecule, starting with 1 for the first residue added to a molecule.
res-name	The residue name (up to four characters) as it appears in the TPL file for residue-name.
PDB#	Preserves information found in a PDB file.
previous-res#	Preserves information found in a PDB file.
next-res#	Preserves information found in a PDB file.

## atom

Syntax: *atom* <at#> <atom-name> <element> <type>  
<flags> <at-charge> <x> <y> <z> <cn> <nbor# nbor-bond>

An atom record starts the description of each atom in a molecule.

<b>at#</b>	The number of an atom in a molecule, starting with 1. Atom numbers are continuous through a molecule. A combination of atom number and molecule number (see mol, above) uniquely identifies an atom in a molecular system.
<b>atom-name</b>	This name (up to four characters) usually comes from a Brookhaven PDB or HyperChem template file. It cannot contain spaces or quotes.
<b>element</b>	The chemical symbol (up to three characters) for the described atom.
<b>type</b>	The atom type (up to four characters) assigned by a force field to the described atom.
<b>flags</b>	Single letter codes that describe this atom (see the next section). “-” indicates no flag. Multiple flags appear as one “word,” without spaces. See the description of available flags in the next section.
<b>at-charge</b>	The formal charge on the atom, from a quantum mechanics calculation, from using Set Charge on the Build menu or from a TPL file. You can also add charges to an atom by editing the HIN file.
<b>x y z</b>	The coordinates of the atom in the Molecular Coordinate System.
<b>cn</b>	The number of atoms covalently bonded to this atom, or coordination number. cn is an integer from 0 to 12.
<b>nbor# nbor-bond</b>	This describes the bonded atoms and the type of bond for each. nbor# and nbor-bond occur as a pair of values. nbor# is the same as at# for bonded atoms. nbor-bond is a bond type: s, d, t, or a, for single, double, triple, or aromatic (conjugated) bonds. This set of values, nbor# nbor-bond, repeats to describe all bonded atoms (up to twelve times).

## Atom Flags

HyperChem uses these symbols for the flag variable in the atom record.

<b>h</b>	A heteroatom as defined in PDB files.
<b>i</b>	This atom is part of an improper torsion.
<b>x</b>	A united atom.
<b>s</b>	A selected atom.
<b>-</b>	No flag.

## vel

Syntax: *vel* <at#> <x> <y> <z>

If you save the molecular system with velocities (choose Velocities in the Save File dialog box), a vel record appears after each atom record, reporting the atomic velocities generated at the end of a molecular dynamics calculation. A velocity record applies to the preceding atom record and must follow immediately after that record.

<b>at#</b>	The number of an atom in a molecule, starting with 1. Atom numbers are continuous through each molecule. This is the same as “at#” that appears in the atom record.
<b>x y z</b>	The x, y, and z components of the atomic velocity, in Ångstroms/picosecond. The default is 0 0 0.

## mass

Syntax: *mass* <at#> <mass>

If you change the mass of an atom, a mass record is added following that atom’s record in the HIN file. This mass record applies to the preceding atom record.

<b>at#</b>	The number of an atom in a molecule, starting with 1. Atom numbers are continuous through each molecule. This is the same as “at#” that appears in the atom record.
------------	---

**mass**                    The mass of the atom, in atomic mass units. The default is the mass of the atom type as defined in the type file for the specified force field.

## basisset

Syntax: *basisset* <at#> <basis set>

If you apply a basis set to an atom for *ab initio* calculations, a basis set record is added following that atom's record in the HIN file. This basisset record applies to the preceding atom's record.

**at#**                    The number of an atom in a molecule, starting with 1. Atom numbers are continuous through each molecule. This is the same as "at#" that appears in the atom record.

**basis set**            The basis set applied to the atom, for example "STO-3G". The default is "None".

## formalcharge

Syntax: *formalcharge* <at#> <formal charge>

If you apply a formal charge to an atom for using the Formal Charge menu item, a formalcharge record is added following that atom's record in the HIN file. This formalcharge record applies to the preceding atom's record.

**at#**                    The number of an atom in a molecule, starting with 1. Atom numbers are continuous through each molecule. This is the same as "at#" that appears in the atom record.

**formal charge**      The value, e.g. -1, of the formal charge placed on this atom. The default is 0.

## endres

Syntax: *endres* <res#>

An endres record appears after the last atom in a residue.

**res#**            The number of the residue in the molecule, starting with 1 for the first residue added to a molecule.

## endmol

Syntax: *endmol* <mol#>

An endmol record appears after the last atom in a molecule.

**mol#**            The number of the molecule in the molecular system, starting with 1. The number reflects the order of adding molecules to the system. Default = 1.

## selection

Syntax: *selection* <sel-name> <atoms>

A selection record plus the following selectrestraint, selectatom, and endselection records appear if you save a molecular system containing a named selection (see “Name Selection . . .” on page 139).

**sel-name**        The name of a selection, either a predefined name, POINT, LINE, PLANE, or a name that you supply (Other), up to 19 characters long.

**atoms**            The number of atoms in a selection.

## selectrestraint

Syntax: *selectrestraint* <flag> <force-constant> <restr-value>

A selectrestraint record contains information needed to use the selection (see the previous section) as a restraint in an energy calculation (see “Restrains” on page 340). This record appears for every named selection, even though it might contain too many atoms for use as a restraint (more than four) and you did not apply the selection as a restraint. The flag value determines whether to use this named selection as a restraint (see the next section).

**flag**              This is a numeric field with a value from 0 to 15. A four-bit number (see the next section) determines the use, if any, of this restraint.

**force-constant** The force constant for a possible restraint. This can be computed or user defined.



**restr-value** Four numbers. The first is a length, angle, or torsion angle for a 2, 3 or 4-atom restraint. The last three define cartesian coordinates of a tether point for a 1-atom restraint.

### Setting Flag Bits for selectrestraint

The selectrestraint flag is the decimal value of seven bits:

bit	6	5	4	3	2	1	0
	64,0	32,0	16,0	8,0	4,0	2,0	1,0

**bit 0** 1 = restraint not chosen in Restraint Forces dialog box; 0 = restraint selected.

**bit 1** Reserved, always 2.

**bit 2** 4 = Computed (the default) for non-tether Value; 0 = Other (user defined) for non-tether Value.

**bit 3** 8 = Default for Force Constant; 0 = Other (user defined) for Force Constant.

**bit 4** 16 = Restraint is a one-atom tether; 0 = Restraint is not a tether.

**bit 5** Reserved, always 0.

**bit 6** 64 = Computed for tether Value; 0 = Other (user defined) for non-tether Value.

These are typical values for the selectrestraint flag:

**26** Tether restraint with Default for Force Constant and Other (user-defined) for Value.

**18** Tether restraint with Other (user-defined) for Value and Force Constant.

**14** Non-tether restraint with Computed for Value and Default for Force Constant.

**10** Non-tether restraint with Computed for Value and Other (user defined) for Force Constant.

**6** Non-tether restraint with Default for Force Constant and Other (user-defined) for Value.

2	Non-tether restraint with Other (user-defined) for Value and Force Constant.
Any odd	Named selection not used as a restraint.

## selectatom

Syntax: *selectatom* <at#> *molecule* <mol#>

Selectatom records define the atoms in a named selection. There is one record for each atom.

at#	The number of an atom in a molecule, starting with 1.
mol#	The number of the molecule, in creation order, in the molecular system, starting with 1. Default = molecule 1.

## endselection

syntax: *endselection* <sel-name>

The endselection record ends the description of each named selection.

sel-name	The name you gave a selection, either POINT, LINE, PLANE, or another name (Other).
----------	--

## dynamics

syntax: *dynamics* <snap-shot-file-name>

The dynamics record gives the path and name of the SNP file containing snapshot data (see “Snapshot Files” on page 623). If HyperChem cannot find the SNP file at the indicated path location, it searches for the filename in the current directory.

snap-shot-file-name	The snapshot filename includes the drive and path specification. The filename is often the same as the HIN filename, but with an <i>.snp</i> extension.
---------------------	---

## aromaticring

syntax: *aromaticring* <size> <mol# at#>

The aromaticring record gives a description of any aromatic rings that have been defined (usually by an L-double-click on the ring). An aromatic ring can be rendered with an interior circle to designate the aromaticity rather than rendering aromatic bonds with dotted lines. This optional rendering is only available if this record is included in the HIN file.

size	The number of atoms in the aromatic ring.
at#	The number of an atom in a molecule, starting with 1.
mol#	The number of the molecule, in creation order, in the molecular system, starting with 1. Default = molecule 1.

## text

syntax: *text* <quoted text> <quoted font> <red> <green> <blue> <color index> <point size> <centered flag> <rect pos> <weight> <italics> <underline> <strikeout> <charset> <pitch>

The text record defines a text annotation placed in the workspace. There is one record for each text annotation. These records do not require a molecule in the workspace, i.e. a HIN file can contain only annotation drawings.

quoted text	The text of the annotation (in double quotes).
quoted font	The text name of the font (in double quotes).
red	Intensity of red (0-255).
green	Intensity of green (0-255).
blue	Intensity of blue (0-255).
color index	Index for color choice in text edit dialog box.
point size	Size of the text in points.
centered flag	Flag (0 or 1) indicating text is formatted as centered.
rect pos	four integers describing rectangle position of text.
weight	Integer describing weight of text(bold, etc.).
italics	Flag (0 or 1) describing whether text is italicized.
underline	Flag (0 or 1) describing whether text is underlined.

<b>strikeout</b>	Flag (0 or 1) describing whether text has stikeout.
<b>charset</b>	Character set of text.
<b>pitch</b>	Pitch off text.

## line

syntax: *line* <red> <green> <blue> <line width> <line type> <rect pos>

The line record defines a drawn line placed in the workspace as an annotation. The line is not a drawn “bond”. There is one record for each line annotation. These records do not require a molecule in the workspace, i.e. a HIN file can contain only annotation drawings.

<b>red</b>	Intensity of red (0-255).
<b>green</b>	Intensity of green (0-255).
<b>blue</b>	Intensity of blue (0-255).
<b>line width</b>	Width of line (smallest value 1).
<b>line type</b>	Integer starting at 0 describing solid, dotted, etc.
<b>rect pos</b>	four integers describing rectangle position of line.

## circle

syntax: *circle* <red> <green> <blue> <line width> <line type> <rect pos> <fill flag>

The circle record defines a drawn ellipse or circle (degenerate ellipse) placed in the workspace as an annotation. There is one record for each circle annotation. These records do not require a molecule in the workspace, i.e. a HIN file can contain only annotation drawings.

<b>red</b>	Intensity of red (0-255).
<b>green</b>	Intensity of green (0-255).
<b>blue</b>	Intensity of blue (0-255).
<b>line width</b>	Width of line (smallest value 1) for circle.
<b>line type</b>	Integer starting at 0 describing solid, dotted, etc.

**rect pos**            four integers describing rectangle enclosing circle.  
**fill flag**            Flag (0 or 1) describing whether circle is filled.

## rectangle

syntax: rectangle *<red>* *<green>* *<blue>* *<line width>* *<line type>*  
*<rect pos>* *<fill flag>*

The rectangle record defines a drawn rectangle placed in the workspace as an annotation. There is one record for each rectangle annotation. These records do not require a molecule in the workspace, i.e. a HIN file can contain only annotation drawings.

**red**                    Intensity of red (0-255).  
**green**                Intensity of green (0-255).  
**blue**                 Intensity of blue (0-255).  
**line width**         Width of line (smallest value 1) for rectangle.  
**line type**            Integer starting at 0 describing solid, dotted, etc.  
**rect pos**             four integers describing rectangle.  
**fill flag**             Flag (0 or 1) describing whether rectangle is filled.



# Appendix E

## Residue Template Files

Template (TPL) files describe amino acid and nucleic acid residues that you can choose on the Databases menu to construct polymeric molecules. HyperChem uses these files when you construct a new polymer, when you open a HIN file that contains a polymer constructed this way, when you open a PDB file, or when you mutate one residue into another. The TPL files interpret standard residues by providing atom types, charges, connectivity information, and internal coordinates.

HyperChem supplies three TPL files:

<code>chem.tpl</code>	Contains internal (nonterminal) amino acid residues, nucleic acid residues, and residues for reading PDB files.
<code>chemct.tpl</code>	Contains C-terminal forms of all amino acid residues. These residues supply anionic carboxyl-terminal groups if you choose Make Zwitterion on the Databases menu.
<code>chemnt.tpl</code>	Contains N-terminal forms of all amino acid residues. These residues supply cationic amino-terminal groups if you choose Make Zwitterion on the Databases menu.

### Structure of a TPL File

You can examine the contents of the TPL files by using a word processor, like Windows Word, Notepad, or WordPad. Make sure that your editor does not convert the file to its own format nor add formatting such as page breaks. Most editors, such as Windows Notepad, allow this option.

TPL files describe a residue atom by atom. They can also provide information needed in molecular mechanics calculations.

This is part of a typical residue entry in a template file (the highlighted numbers, for example ⑧, are not part of the file):

```

; ALANINE①
[ALA]②A③
N:④      N⑥1⑦ (HN s -3 s CA s)⑧ \
          opls   N           -0.5700 imp -3 CA N HN \⑨
          amber  N           -0.4630 imp -3 CA N NH \⑨
          amber94 N         -0.41570 \⑨
          bio+   NH1         -0.3500 imp N -3 CA NH \⑨
          charmm22 NH1      -0.4700 \⑨
          internal -3 1.335 -2 116.60μ
HN④ H⑤ : H⑥ (N s) \
          opls   H           0.3700 \
          amber  H           0.2520 \
          amber94 H         0.27190 \
          bio+   H           0.2500 \
          charmm22 H         0.31 \
          int N   1.010
CA:④      C⑥ 2 ⑦ (N s HA s C s CB s) \
          opls   CH           0.29000 imp CB CA N C \
          amber  CT           0.0350 \
          amber94 CT         0.03370 \
          bio+   CH1E         0.1000 imp CA N C CB \
          charmm22 CT1       0.2520 \
          int N   1.449 -3    121.90
. . . .

```

① Comment record with the full name of a residue. Comment records must start with a semicolon. Write as many comments as you need.

② The residue name. The name should be from one to four characters long, in square brackets. The limit is 256 characters. This name appears in the appropriate Databases dialog box (for example, the Amino Acids dialog box), unless a \* follows the Residue type (see ③). Also, within these brackets, you can have one or more aliases (alternative names) for the residue, each separated by a



space. These can be up to four characters long. HyperChem uses aliases when it reads PDB files.

- ③ Residue type. This is A for amino acid, N for nucleic acid, or \* for some other type of residue. An \* in addition to A or N *prevents* HyperChem from showing this residue as a choice in the Amino Acids or Nucleic Acids dialog box.
- ④ An atom description starts here. This is the name of an atom in the residue (see below). These are the atom names used in Brookhaven PDB files (see page 503) and the names HyperChem uses for Name labels (choose Labels on the Display menu, then Atoms: Name). The atom name or name plus optional alias must end with a colon.
- ⑤ The optional aliases (alternative names) for an atom. There must be at least one space between the atom name and each of its aliases. The name and alias must end with a colon.
- ⑥ The chemical symbol for the element.
- ⑦ An optional atom tag (see below).
- ⑧ Bonding information for this atom (see below).
- ⑨ Force field information (see below). A back slash (\) at the end of a line indicates that this entry continues on the next line.
- ⑩ Atomic coordinates (see below).

## Template Syntax

This section contains a formal syntax for describing a residue in a template file. The text in angle brackets and italics represents variables that you provide. <sup>o</sup> means optional.

This describes the overall structure of a residue:

```
; <comment>o
[<->o <residue-name> <alias>o <alias>o . . .] <A|N|*><*>o
<atom-description>
<atom-description>
<atom-description>
<atom-description>
```

...

**;<comment>**

Each line of comment must begin with a semicolon. These lines can occur anywhere in the file, but cannot interrupt another line.

**[<residue-name>]**

[<-><sup>0</sup> <residue-name> A string containing no spaces or quotes. Use one to four characters. Enclose residue-name plus optional aliases in square brackets. The optional - indicates a complementary nucleic acid residue used to build in reverse order or double-stranded structures.

<alias><sup>0</sup>  
<alias><sup>0</sup> . . .] Any number of alternative names for a residue, up to four characters long. Each alias is separated from the others and from residue-name by a space. HyperChem uses aliases to recognize residues in PDB files, for example ASX for ASN, and DOD for HOH.

<A|N|\*> A for amino acid, N for nucleic acid, or an asterisk (\*) for unknown. This assigns the residue to the correct dialog box. \* after A or N (see the next item) overrides the assignment to a dialog box.

<\*><sup>0</sup> If it appears after A or N, it turns off this residue as a choice in the Amino Acids or Nucleic Acids dialog box.

**<atom-description>**

Each atom in a residue should have an atom-description in a single record. The record can occupy more than one line if each line ends with a \.

<atom-description> =

<atom-name> <aliases><sup>0</sup> : <element> <tag><sup>0</sup> (bonding-info) \

<force-field><sup>0</sup> <atom-type><sup>0</sup> <charge><sup>0</sup> <imp-torsion><sup>0</sup> \

<force-field><sup>0</sup> <atom-type><sup>0</sup> <charge><sup>0</sup> <imp-torsion><sup>0</sup> \

...

*internal*<sup>o</sup> <at1> <distance> <at2><sup>o</sup> <angle><sup>o</sup> <at3><sup>o</sup> <torsion><sup>o</sup>

**<atom-name> <aliases>**

Atom names follow standards defined by the Protein Data Bank for atoms in amino acid and nucleic acid residues. In an atom name, for example C3\*, an \* replaces the usual prime symbol (C3'). It cannot contain spaces or quotes.

*Note:* Do not use numbers for atom names to avoid confusion with atom tags (see next section).

For a complete list of atom names in amino acid side chains, nucleotide bases, and other molecules, see the Protein Data Bank document, *Atomic Coordinates and Bibliographic Entry Format Description*, July, 1989 (see page 503 for the source of this document).

**<aliases>**

An atom-name can have one or more optional names (aliases). Separate each with at least one space.

**<element>**

The chemical symbol for the described atom.

**<tag>**

An optional number describing the atom to neighboring residues.

Atom tags (numbers 1 to 9) describe connections to atoms in other residues. HyperChem assigns tags to specific terminal atoms in amino acid and nucleic acid residues. A minus sign preceding a tag number indicates the previous residue (lower sequence number).

Atom Name	Tag Number	Residue Type <sup>a</sup>
N	1	A
CA	2	A
C	3	A
O	4	A
P	4	N

Atom Name	Tag Number	Residue Type <sup>a</sup>
O5*	5	N
C5*	6	N
C4*	7	N
C3*	8	N
O3*	9	N

a. A = amino acid, N = nucleic acid.

### (bonding-info)

Bonding information indicates the presence and describes the type of covalent bonds between atoms. The bonding information consists of an atom name or tag, followed by the type of bond:

(Atom-name/tag bond-type Atom-name/tag bond-type . . .)

This sequence repeats until it describes each bond to an atom.

**Atom-name/tag** The name of a bonded atom in the same residue, or the tag of an atom if it is in an adjacent residue.

**bond-type** s, d, t, or a, for single, double, triple, or aromatic (conjugated) bonds.

**Caution:** Parentheses must enclose bonding information. Separate each entry within parentheses with white space. Make sure the entries occur in pairs: atom-name, then bond-type.

**Example:** (HN s -3 s CA s)

HN s: A single bond to hydrogen HN.

-3 s: A single bond to the atom with tag 3 in the previous residue.

CA s: A single bond to atom CA.

### <force-field>

Force field information describes the properties of an atom in terms of a molecular mechanics force field (see “Molecular Mechanics” on page 289). HyperChem can use these properties

during a molecular mechanics calculation. An atom description can contain from zero to four force field entries, or none. A force field description of "none" indicates a hydrogen that is part of a united atom representation.

Force field information includes this data, separated by spaces:

Force-field Atom-type Charge [imp At#1 At#2 At#3 At#4]<sup>1</sup>

Force-field <sup>o</sup>	mm+, opls, amber, amber94, bio+, or charmm22.
Atom-type <sup>o</sup>	Type of atom used during a molecular mechanics calculation. Atom types are specific to a force field.
Charge <sup>o</sup>	Charge on the atom.
imp <sup>o</sup>	An improper torsion involving the atom being described and three other atoms bonded to it. The order of the At#n (atom name or tag number) determines the improper torsion angle. This information is optional.

*Example:* bio+ HN1 -0.3500 imp N -3 CA NH

bio+: This information applies to the BIO+ force field.

HN1: The atom type for the described hydrogen.

-0.3500: The negative charge on this atom.

imp N -3 CA NH: An improper torsion involving this atom (N) and its three bonded neighbors: atom tagged -3 in the previous residue, atom CA, and atom NH.

## internal

HyperChem describes the position of an atom in terms of internal coordinates, or the relationship to its bonded neighbors. The description of atomic coordinates begins with the keyword *internal*, for internal coordinates. Atomic coordinates consist of names or tags for the neighboring atoms, each followed by a value for the relationship:

*internal* Atom-name/tag distance Atom-name/tag angle Atom-name/tag torsion-angle

1. Brackets indicate optional information.

<code>int<sup>o</sup></code>	The keyword introducing the atomic coordinates. Only the first two letters of this keyword are necessary.
<code>Atom-name/ tag<sup>o</sup></code>	The name of a bonded atom in the same residue, or the tag of an atom if it is in an adjacent residue.
<code>distance<sup>o</sup></code>	The length of the bond, in Ångstroms, to the neighbor.
<code>angle<sup>o</sup></code>	The bond angle formed by the described atom, Atom1, and Atom 2. Atom 2 and angle are optional in this record.
<code>torsion-angle<sup>o</sup></code>	The torsion angle formed by the described atom, Atom 1, Atom 2, and Atom 3. Atom 3 and torsion-angle are optional in this record.

The minimum internal coordinate record must contain Atom1 (atom-name/tag) and distance. If it does *not* contain Atom 3 and torsion-angle, the record must be last in an atom description.

*Example:* `int CA 1.525 N 111.10 -3`

Remember, atom tag “-3” represents an atom in the previous residue.

## Amino Acid Residues

Each amino acid residue requires an entry in each of the TPL files:

- `Chem.tpl` for internal amino acids.
- `Chemct.tpl` for the residue in the C-terminal position. HyperChem uses this residue for the C-terminal residue when you request a zwitterion. The terminal carbon in this residue has two bonded oxygens instead of one. The charge on the residue is -1.0.
- `Chemnt.tpl` for the residue in the N-terminal position. HyperChem uses this residue for the N-terminal residue when you request a zwitterion. The terminal nitrogen in this residue has three hydrogens instead of one. The charge on the residue is +1.0.

For this release of HyperChem, the template files contain templates for these residues:

Residue	Name/Alias	Residue	Name/Alias
Alanine	ALA	Phenylalanine	PHE
Glycine	GLY	Tyrosine	TYR
Serine	SER	Glutamic acid	GLU
Threonine	THR	Aspartic acid	ASP
Leucine	LEU	Lysine	LYS
Isoleucine	ILE	Proline	PRO
Valine	VAL	Cysteine	CYS,CYH, CSH
Asparagine	ASN, ASX	Cystine (S-S bridge)	CYX, CSS
Glutamine	GLN, GLX	Methionine	MET
Arginine	ARG	Acetyl beginning group	ACE
Histidine, delta H	HID	N-methylamine ending group	NME
Histidine, epsilon H	HIE	Amine ending group <sup>a</sup>	NH2
Histidine plus (protonated)	HIP	Oxygen ending group <sup>a</sup>	OME
Histidine generic (same as histidine protonated)	HIS	Dummy for building the first residue <sup>a</sup>	DUM
Tryptophan	TRP	Heme-histidine <sup>a</sup>	HEM, HEME
Methotrexate <sup>a</sup>	MTX	4-Hydroxyproline <sup>a</sup>	HYP
Gamma-amino butyric acid <sup>a</sup>	ABU	Ornithine <sup>a</sup>	ORN



Residue	Name/Alias	Residue	Name/Alias
Beta-alanine <sup>a</sup>	ALB	Pyrrolidone carboxylic acid (pyroglutamate) <sup>a</sup>	PCA
Betaine <sup>a</sup>	BET	Sarcosine <sup>a</sup>	SAR
Homoserine <sup>a</sup>	HSE	Taurine <sup>a</sup>	TAU
5-Hydroxylysine <sup>a</sup>	HYL	Thyroxine <sup>a</sup>	THY
Formyl beginning group <sup>a</sup>	FOR		

a. This residue does not appear in the Amino Acids dialog box; HyperChem uses the residue template to read residues from a PDB file.

## Nucleic Acid Residues

Each nucleic acid residue has a primary entry in the chem.tpl file and might have a secondary entry used to construct a complementary nucleotide residue during double-stranded synthesis or for backwards synthesis.

Secondary entries have residue names starting with a minus sign. For example, the residue dA pairs with -dT, and dT pairs with -dA. An \* immediately after the residue name suppresses display of the residue in dialog boxes.

For this release of HyperChem, the templates files contain templates for these residues:

Residue	Name/Alias	Residue	Name/Alias
Dummy for building the first residue <sup>a</sup>	DUP	Phosphate ending group, 3'	3CAP
Dummy for building the first residue, backward strand <sup>a</sup>	DUX	D-adenosine <sup>a, b</sup>	-DA
D-adenosine	DA	D-cytosine <sup>a, b</sup>	-DC

Residue	Name/Alias	Residue	Name/Alias
Thymidine	DT, T	d-guanosine <sup>a, b</sup>	-DG
D-cytosine	DC	Thymidine <sup>a, b</sup>	-DT
D-guanosine	DG	R-adenosine <sup>a, b</sup>	-RA
R-adenosine	RA, A, +A	Uracil <sup>a, b</sup>	-RU
Uracil	RU, U	R-guanosine <sup>a, b</sup>	-RG
R-guanosine	RG, G	R-cytosine <sup>a, b</sup>	-RC
R-cytosine	RC, C, +C	H beginning, 5' <sup>a, b</sup>	HB
Beginning group, 5'	5CAP	H end, 3' <sup>a</sup>	HE
Coenzyme A <sup>a, c</sup>	CoA	7-methylguanosine <sup>a, c</sup>	7MG
Flavin mononucleotide <sup>a, c</sup>	FMN	2'-O-methylguanosine <sup>a, c</sup>	OMG
Adenosine triphosphate <sup>a, c</sup>	(ATP)	Inosine <sup>a, c</sup>	I
1-methyladenosine <sup>a, c</sup>	1MA	Dihydrouridine <sup>a, c</sup>	H2U
5-methylcytosine <sup>a, c</sup>	5MC	5-methyluridine <sup>a, c</sup> (ribosylthymidine)	5MU
2'-O-methylcytidine <sup>a, c</sup>	OMC	Pseudouridine <sup>a, c</sup>	PSU
1-methylguanosine <sup>a, c</sup>	1MG	Base Y <sup>a, c</sup> (Wybutosine)	YG, Y
N(2)-methylguanosine <sup>a, c</sup>	2MG	Nicotinamide adenine dinucleotide <sup>a, c</sup>	NAD
N(2)-dimethylguanosine <sup>a, c</sup>	M2G	TIP3P water <sup>d</sup>	HOH, DOD, WAT

a. This residue does not appear in the Nucleic Acids dialog box. HyperChem uses the residue for constructing the complementary strand of a polynucleotide or for reading these residues from a PDB file.

b. For the negative strand or backwards synthesis.

c. This residue supplies atom bonding information only [*<atom-name> <aliases> : <element> <tag> (bonding-info)*], without force field, charge, or internal coordinate information.

d. For reading water molecules in a PDB file.

In this version of HyperChem, you cannot add TPL files; you can only add entries to the existing files.

## Adding New Residues

You can add new residues to the Amino Acids and Nucleic Acids dialog boxes, up to a total of 36 amino acids and 24 nucleic acids for each dialog box. HyperChem uses this information to interpret data files (HIN or PDB) and to construct the molecular system in the workspace.

All residues must be in the file chem.tpl, chemct.tpl, and chemnt.tpl (see the introduction to this chapter). Before you add residues to these files, make backup copies of the original files.

New residues must supply this minimal information:

```
[<residue-name>]<N|A|*>
```

```
<atom-name> : <element> (bonding-info)
```

```
<atom-name> : <element> (bonding-info)
```

```
...
```

To add a residue to a TPL file but suppress its appearance in the Amino Acids or Nucleic Acids dialog box, add an \* to the end of the residue-name record. HyperChem can still use this information to read files containing these residues.

New residues are available in the dialog boxes the next time you run HyperChem.

## General Requirements

To build a new residue, you must have its chemical structure (elements and bonding). It is also useful, but not necessary, to have bond lengths, bond angles, and torsion angles. You can get this information from published reports, by building a residue from atom and optimizing it with HyperChem, and by using parts of residues already in the TPL file.

To perform molecular mechanics calculations on molecules containing these residues, it is useful, but not necessary, to have atom types and charges for each atom. You can find this information in

published reports of a force field. You can also calculate charges by using a semi-empirical quantum mechanics calculations.

**Example:**

This example adds the ribonucleotide residue 5-methylcytidine (5MC) to the Nucleic Acids dialog box. This residue has a methyl group, not hydrogen, on the 5' carbon of the pyrimidine ring.

This residue is simple to construct, because the pyrimidine ring is the same as in deoxythymidine (dT).

Follow these steps:

1. Using a text editor, like Windows Notepad, or a word processor, like Windows Write, open the file chem.tpl. Be sure to choose the No conversion option in Windows Write to maintain the file in its original format.

2. Use the Find command to locate R-CYTOSINE.

**Note:** If you are using Windows Write, you can choose Repaginate on the File menu. This divides the file into pages so that you can move easily through it; the pagination does not interfere with HyperChem.

3. Copy the entire entry for R-CYTOSINE to another location. If you put the copy immediately after R-CYTOSINE, 5MC appears below rC in the dialog box.

4. Scroll up to the beginning of the pasted text. Then change the first three lines to this text:

```
    ; R-5-METHYL CYTIDINE
    ;
    [5MC]N
```

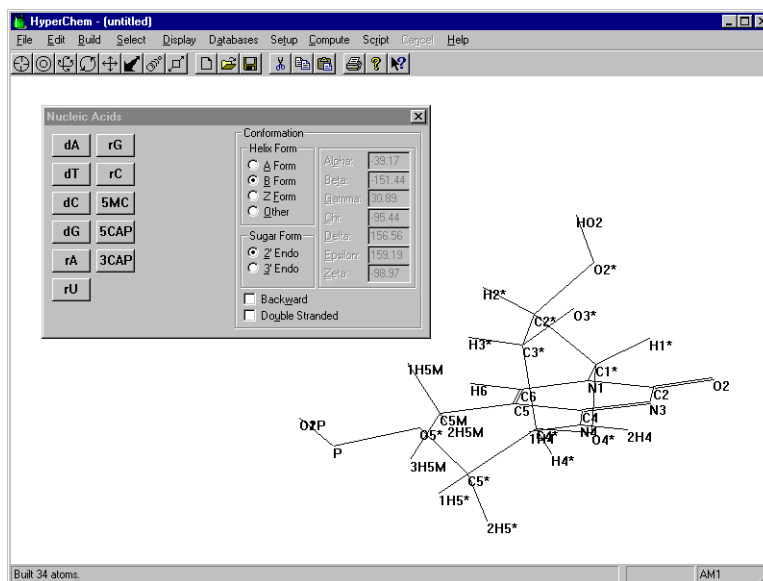
5. Use the Find command to locate [DT T]N.

6. Copy the four entries for the methyl group atoms. They start with these atom names:

```
C5M C7:
1H5M 1D5M H71 1H19:
2H5M 2D5M H72 2H19:
3H5M 3D5M H73 3H19:
```

7. Return to the entry for 5MC and paste these four atoms after atom H5 D5 H19.
8. Delete the entire entry for atom H5 D5 H19
9. Edit the first line for atom C5 to show the bond to atom C7 instead of H5:
 

```
C5:   C       (C6 d C7 s C4 s)  \
```
10. Save the file and leave the text editor.
11. Start HyperChem. If HyperChem is already open, choose Exit on the File menu and start it again.
12. Choose Nucleic Acids on the Databases menu. The dialog box now contains the residue 5MC.



**Note:** An alternative way to construct this residue is to change the deoxyribose sugar in dT to a ribose.



# Appendix F

## Default Settings in the Registry

The Registry contains all the default settings for HyperChem for Windows. HyperChem uses this information to remember your settings from the previous session. HyperChem reads the settings when you start a session.

During a HyperChem session you can change some of these settings from dialog boxes (for example, see “Preferences” on page 95). When you leave HyperChem, the Registry is updated to the recent settings. The Registry supplies the last choices you made in a HyperChem menu or the last values you entered in a dialog box. HyperChem uses these setting for each session until you change them.

Some settings are available only in the Registry. You can change these settings by typing `regedit` in the Open: field of Run... from the Windows Start button.

### Registry Structure

You can see and edit the contents of the Windows Registry by running the `regedit` Registry Editor. The HyperChem settings are in the section under,

`HKEY_CURRENT_USER/Software/Hypercube/HyperChem Pro/7.0`  
i.e., assuming your version of HyperChem is Release 7. The settings are as described below.

Default settings are in keys that contain multiple string values, binary values, or `DWORD` values. HyperChem generally uses string values even for values that are numerical or boolean (the

string is later converted to a numerical or boolean value). This example includes the first six text lines in the Ablnitio key:

### Ablnitio

Name	Type	Data
2ElectronBufferSize	REG_SZ	32000
2ElectronCutoff	REG_SZ	1.0e-10
CalculateGradient	REG_SZ	Yes
DirectSCF	REG_SZ	No
FiveOrbitals	REG_SZ	Yes

In the Registry, each key contains many String Values that have a Name (e.g. DirectSCF) a Type (i.e. REG\_SZ, meaning string value) and the Data (e.g. No). The Data can be Yes or No (Boolean), an integer (Int), a floating point number (Real), one of a limited set of possibilities (Enum) or a string such as the name of a file.

*Note:* Name and Data values in the Registry are *not* case-sensitive.

In the Registry, the keys for the different versions of the force fields are saved as subkeys under the main force field keys, and there are 12 main keys in all.

Ablnitio	Settings for <i>ab initio</i> calculations
amber	Settings for the AMBER force field
amber2	Settings for the amber2 parameter set (subkey in the Registry)
amber3	Settings for the amber3 parameter set (subkey in the Registry)
ambers	Settings for the ambers parameter set (subkey in the Registry)
amber94	Settings for the amber94 parameter set (subkey in the Registry)



<b>amber96</b>	Settings for the amber96 parameter set (subkey in the Registry)
<b>Basisset</b>	List of the basis set files (*.BAS) in the HyperChem Runfiles directory
<b>bio+</b>	Settings for the BIO+ force field
<b>bio83</b>	Settings for the bio83 parameter set (subkey in the Registry)
<b>bio85</b>	Settings for the bio85 parameter set (subkey in the Registry)
<b>charmm19</b>	Settings for the charmm19 parameter set (subkey in the Registry)
<b>charmm22</b>	Settings for the charmm22 parameter set (subkey in the Registry)
<b>charmm27</b>	Settings for the charmm27 parameter set (subkey in the Registry)
<b>Compute</b>	Settings for computational options
<b>Filter</b>	Information about input/output file filters
<b>mm+</b>	General settings for the MM+ force field
<b>mmplus</b>	Additional settings for the MM+ parameter set (subkey in the Registry)
<b>Network</b>	Settings for network licensing
<b>opls</b>	Settings for the OPLS force field
<b>opls, opls</b>	Settings for the OPLS parameter set (subkey in the Registry)
<b>Path</b>	Information about paths for parameter files, data storage, etc.
<b>Preferences</b>	Settings for keyboard operations, display options, etc.
<b>tndo</b>	Settings for the TNDO method
<b>default</b>	Settings for the TNDO parameter set that is simply CNDO/INDO (subkey in the Registry)

## Registration

The registration data for your copy of HyperChem is contained in a different section of the registry. You can see and edit the contents of the Windows Registry by running the regedit Registry Editor by typing regedit to Run... under the Windows Start button. The Registration settings are in the key under,

HKEY\_LOCAL\_MACHINE/Software/Hypercube/HyperChem Pro/7.0

The entries include the information you entered when you installed and licensed your copy of HyperChem.

## Ab Initio Settings

*Note:* These AbInitio settings also apply to Density Functional Calculations (DFT).

### AbInitio

Keyword	Value Type	Value	How to Set	Use
CalculateGradient	Boolean	Yes, No	Ab Initio or Density Functional Advanced Options dialog box	Specify whether or not the gradient calculation is needed. This option applies only to single-point calculations.
DirectSCF	Boolean	Yes, No	Ab Initio or Density Functional Advanced Options dialog box	Specify whether or not to use direct SCF calculation.
2ElectronCutoff	Real	> 0	Ab Initio or Density Functional Advanced Options dialog box	Eliminate the two-electron integrals with values less than this cutoff.
GaussSCFConvergence	Real	0.0-100	Ab Initio or Density Functional Options dialog box	This is the limit which the energy difference between two consecutive iterations in an SCF calculations should reach.
2ElectronBuffer-Size	Int	> 50	Ab Initio or Density Functional Advanced Options dialog box	Specify the buffer size to store the two-electron integrals before sending them to a hard disk.

Keyword	Value Type	Value	How to Set	Use
FiveDOrbitals	Boolean	Yes, No	Ab Initio or Density Functional Advanced Options dialog box	Specify the type of d orbitals. Yes uses 6 d orbitals (i.e., $d_{xx}$ , $d_{yy}$ , $d_{zz}$ , $d_{xy}$ , $d_{xz}$ , $d_{yz}$ ). No uses 5 d orbitals (i.e., $d_0$ , $d_{+1}$ , $d_{-1}$ , $d_{+2}$ , $d_{-2}$ ).
MOInitialGuess	enum	Core-Hamiltonian, Projected-Huckel, Projected-CNDO, Projected-INDO	Ab Initio or Density Functional Advanced Options dialog box	Specify a method to produce an initial guess of MO coefficients.
IntegralFormat	enum	Regular or Raffenetti	Ab Initio or Density Functional Advanced Options dialog box	Specify the format to save the two-electron integrals. Density Functional choice forces this value to Regular.
MP2Correlation-Energy	Boolean	Yes, No	Ab initio Advanced Options dialog box	Specify whether MP2 correlation energy should be calculated with an <i>ab initio</i> or Density Functional single-point calculation. Meaning for DFT is dubious.
MP2FrozenCore	Boolean	Yes, No	This file	Specify whether MP2 correlation calculation should use a frozen core
IntegralPath	string		File/Preferences/Paths property sheet	Specify the hard disk (drive) and the path to save the two-electron integrals
UseGhostAtom-Selection	Boolean	Yes, No	Ab initio Advanced Options dialog box	Specify whether or not to use a ghost-atom selection.

# AMBER Settings

## amber

Keyword	Value Type	Value	How to Set	Use
CurrentFiles	Int	0 to 2	Select "Force field" Parameter Set dialog box	The index of the AMBER parameter set to use. Default = 0 for first CustomName; 1 = second, and so on. Default = 1
Amber94CurrentFiles	Int	3 to (number of CustomNames - 1)	Select "Force field" Parameter Set dialog box	The index of the AMBER94 parameter set to use. Default = 3 for first Amber94 CustomName; 4 = second, and so on. Default = 3
AtomTypeMass	Enum	Filename: *typ.txt *typ.dbf or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types for this force field. Default = ambertyp.txt
Amber94AtomTypeMass	Enum	Filename: *typ.txt or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types for this force field. Default = amber94typ.txt
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type of file containing AtomTypeMass parameters. Default = Text
CustomNames	Enum	Names separated by commas	Registry <sup>a</sup>	The names of parameter sets associated with the AMBER and AMBER94 force field. Default = amber2, amber3, ambers, amber94, amber96
MMVersionIditem	Int	0-(number of parameter sets -1)	Select "Force field" Parameter Set dialog box	The index of the parameter set to use.

a. Use direct editing of Registry to change this item

**amber, CustomName**

These are the subkeys for the parameter sets with the *Custom Names* amber2, amber3, ambers, amber94, and amber96:

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type for the following files. Default = Text
Dielectric	Enum	Constant or DistanceDepend	Force Field Options dialog box	Sets the dielectric permittivity for calculating electrostatic interactions as either constant or distance dependent (see DielectricScale, below). Default = DistanceDepend
DielectricScale	Real	1–1000	Force Field Options dialog box	Sets the scale factor for dielectric permittivity (see previous entry). Default = 1
1–4Electrostatic-Scale	Real	0–1	Force Field Options dialog box	Sets the scale factor for calculating charge interactions between atoms separated by three bonds. Default = 0.5
1–4vanderWaals-Scale	Real	0–1	Force Field Options dialog box	Sets the scale factor for calculating van der Waals interactions between atoms separated by three bonds. Default = 0.5
6–12AtomVDW-Format	Enum	RStarEpsilon SigmaEpsilon SlaterKirkwood	Registry <sup>a</sup>	Sets the method for calculating atom-based van der Waals interactions from data in the file specified in 6–12AtomVDW (see below). <sup>b</sup> Default = RStarEpsilon

Keyword	Value Type	Value	How to Set	Use
6-12PairVDW-Format	Enum	RStarEpsilon SigmaEpsilon AijBij	Registry <sup>a</sup>	Sets the method for calculating pair-based van der Waals interactions from data in the file specified in 6-12PairVDW (see below). <sup>b</sup> Default = RStarEpsilon
QuadraticStretch	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond stretch parameters for this parameter set. Default = am89astr.txt
QuadraticBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle parameters for this parameter set. Default = am89aben.txt
FourierTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing torsion parameters for this parameter set. Default = am89ator.txt
FourierImpTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing improper torsion parameters for this parameter set. Default = am89aimp.txt
6-12AtomVDW	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for individual atom types. Default = am89anbd.txt
6-12PairVDW (optional)	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for specific pairs of atom types. <sup>b</sup> Default = none
10-12PairHBond (optional)	Enum	Filename	This file <sup>a</sup>	Directs HyperChem to the file containing the hydrogen bonding parameters. <sup>b</sup> Default = am89ahbd.txt

a. Use direct editing of Registry to change this item.

- b. See *HyperChem Computational Chemistry, Theory and Methods*.



# Basis Set Settings

## BasisSet

This registry key defines the basis set file that you use when you select a basis set in the graphical user interface (GUI). That is, the names that appear in the *ab initio* or density functional dialog boxes when you wish to select a basis set are defined here. These names are tied here to a specific basis set file.

The entries (String Names) in this section define files ending with .BAS in the HyperChem Runfiles directory. Thus, you can modify the basis set files and create your own basis set files. To use a new basis set you will need to include it in this section of the Registry.

The following contains most of the standard and commonly used basis sets. All these basis sets are included in HyperChem distribution disks. This section may look different if Hypercube, Inc. or you deleted some .BAS files or added some other .BAS files to the HyperChem Runfiles directory.

Keyword	Value Type	Value	How to Set	Use
3-21G	enum	3-21g.bas	Registry	Define 3-21G basis set.
3-21G*	enum	3-21gs.bas	Registry	Define 3-21G* basis set.
3-21G**	enum	3-21gss.bas	Registry	Define 3-21G** basis set.
4-21G	enum	4-21g.bas	Registry	Define 4-21G basis set.
4-21G*	enum	4-21gs.bas	Registry	Define 4-21G* basis set.
4-21G**	enum	4-21gss.bas	Registry	Define 4-21G** basis set.
4-31G	enum	4-31g.bas	Registry	Define 4-31G basis set.
4-31G*	enum	4-31gs.bas	Registry	Define 4-31G* basis set.
4-31G**	enum	4-31gss.bas	Registry	Define 4-31G** basis set.
5-31G	enum	5-31g.bas	Registry	Define 5-31G basis set.

Keyword	Value Type	Value	How to Set	Use
5-31G*	enum	5-31gs.bas	Registry	Define 5-31G* basis set.
5-31G**	enum	5-31gss.bas	Registry	Define 5-31G** basis set.
6-21G	enum	6-21g.bas	Registry	Define 6-21G basis set.
6-21G*	enum	6-21gs.bas	Registry	Define 6-21G* basis set.
6-21G**	enum	6-21gss.bas	Registry	Define 6-21G** basis set.
6-31G	enum	6-31g.bas	Registry	Define 6-31G basis set.
6-31G*	enum	6-31gs.bas	Registry	Define 6-31G* basis set.
6-31G**	enum	6-31gss.bas	Registry	Define 6-31G** basis set.
6-311G	enum	6-311g.bas	Registry	Define 6-311G basis set.
6-311G*	enum	6-311gs.bas	Registry	Define 6-311G* basis set.
6-311G**	enum	6-311gss.bas	Registry	Define 6-311G** basis set.
STO-1G	enum	sto-1g.bas	Registry	Define STO-1G basis set.
STO-1G*	enum	sto-1gs.bas	Registry	Define STO-1G* basis set.
STO-2G	enum	sto-2g.bas	Registry	Define STO-2G basis set.
STO-2G*	enum	sto-2gs.bas	Registry	Define STO-2G* basis set.
STO-3G	enum	sto-3g.bas	Registry	Define STO-3G basis set.
STO-3G*	enum	sto-3gs.bas	Registry	Define STO-3G* basis set.
STO-4G	enum	sto-4g.bas	Registry	Define STO-4G basis set.
STO-4G*	enum	sto-4gs.bas	Registry	Define STO-4G* basis set.
STO-5G	enum	sto-5g.bas	Registry	Define STO-5G basis set.
STO-5G*	enum	sto-5gs.bas	Registry	Define STO-5G* basis set.

Keyword	Value Type	Value	How to Set	Use
STO-6G	enum	sto-6g.bas	Registry	Define STO-6G basis set.
STO-6G*	enum	sto-6gs.bas	Registry	Define STO-6G* basis set.
D95	enum	d95.bas	Registry	Define D95 basis set.
D95*	enum	d95s.bas	Registry	Define D95* basis set.
D95**	enum	g95ss.bas	Registry	Define D95** basis set.

## BIO+ Settings

### bio+

Keyword	Value Type	Value	How to Set	Use
CurrentFiles	Int	0 to 2	Select "Force field" Parameter Set dialog box	The index of the BIO+ parameter set to use. 0 = first CustomName; 1 = second, and so on. Default = 0
Charmm22CurrentFiles	Int	3 to (CustomNames-1)	Select "Force field" Parameter Set dialog box	The index of the Charmm22 parameter set to use. 3 = first Charmm22 CustomName; 4 = second, and so on. Default = 3
AtomTypeMass	Enum	Filename: *typ.txt *typ.dbf or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types for this force field. Default = biotyp.txt
Charmm22AtomTypeMass	Enum	Filename: *typ.txt or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types for this force field. Default = charmmtyp.txt

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type of file containing AtomTypeMass parameters. Default = Text
CustomNames	Enum	Names separated by commas	Registry <sup>a</sup>	The names of parameter sets associated with the BIO+ and CHARMM22 force field. Default = bio83,bio85,charmm19,charmm22,charmm27
MMVersionIditem	Int	0-(number of parameter sets -1)	Registry <sup>a</sup>	The index of the parameter set to use.

a. Use a text editor, like Wordpad, to change this record in chem.ini.

## bio+, CustomName

These are the subkeys for the parameter sets with the *Custom Names* bio83, bio85, charmm19, charmm22, charmm27:

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type for the following files. Default = Text
Dielectric	Enum	Constant or DistanceDepend	Force Field Options dialog box	Sets the dielectric permittivity for calculating electrostatic interactions as either constant or distance dependent (see DielectricScale, below). Default = DistanceDepend for 85, charmm*, Constant for 83
DielectricScale	Real	1-1000	Force Field Options dialog box	Sets the scale factor for dielectric permittivity (see previous entry). Default = 1 for 85, charmm*, 2.5 for 83

Keyword	Value Type	Value	How to Set	Use
1-4Electrostatic-Scale	Real	0-1	Force Field Options dialog box	Sets the scale factor for calculating charge interactions between atoms separated by three bonds. Default = 1
6-12AtomVDW-Format	Enum	RStarEpsilon SigmaEpsilon SlaterKirkwood	Registry <sup>a</sup>	Sets the method for calculating atom-based van der Waals interactions from data in the file specified in 6-12AtomVDW (see below). <sup>b</sup> Default = RStarEpsilon for 85, charmm*, SlaterKirkwood for 83
6-12PairVDF-Format	Enum	RStarEpsilon SigmaEpsilon AijBij	Registry <sup>a</sup>	Sets the method for calculating pair-based van der Waals interactions from data in the file specified in 6-12PairVDW (see below). <sup>b</sup> Default = RStarEpsilon
QuadraticStretch	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond stretch parameters for this parameter set. Default = biostr.txt for 85, bio83str.txt for 83, charmm19str.txt for charmm19, etc.
QuadraticBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle parameters for this parameter set. Default = bio85ben.txt for 85, bio83ben.txt for 83, charmm19ben.txt for charmm19, etc.
FourierTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing torsion parameters for this parameter set. Default = bio85tor.txt for 85, bio83tor.txt for 83, charmm19tor.txt for charmm19, etc.

Keyword	Value Type	Value	How to Set	Use
QuadraticImp-Torsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing improper torsion parameters for this parameter set. Default = bio85imp.txt for 85, bio83imp.txt for 83, charmm19imp.txt for charmm19, etc.
6-12AtomVDW	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for individual atom types. Default = bio85nbd.txt for 85, bio83nbd.txt for 83, charmm19nbd.txt for charmm19, etc.
6-12PairVDW (optional)	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for specific pairs of atom types. <sup>b</sup> Default = bio85npr.txt for 85, none for 83 charmm19npr.txt for charmm19, etc.
6-12Atom14VDW (optional)	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing atom-based van der Waals parameters for atoms separated by three bonds. <sup>b</sup> Default = none

a. Use direct editing of Registry to change this item.

b. See *HyperChem Computational Chemistry, Theory and Methods*.

# Computational Settings

## Compute

Keyword	Value Type	Value	How to Set	Use
Optimizer	Enum	SteepestDescent FletcherReeves PolakRibiere NewtonRaphson, EigenvectorFollowing, ConjugateDirections	Molecular Mechanics, Ab Initio, Density Functional, or Semi-empirical Optimization dialog box	Sets the algorithm for geometry optimization. Default = PolakRibiere
RMSGradient-Criterion	Real	0-1e+10	Molecular Mechanics, Ab Initio, Density Functional, or Semi-empirical Optimization dialog box	Sets the RMS gradient termination condition for a geometry optimization calculation. Default = 0.1
HeatTime	Real	0-1e+10	Molecular Dynamics Options dialog box	Sets the time, in picoseconds, to change the temperature from Starting to Simulation. Default = 0
RunTime	Real	0-1e+10	Molecular Dynamics Options dialog box	Sets the length of the period, in picoseconds, when the temperature is not changed. Default = 1
CoolTime	Real	0-1e+10	Molecular Dynamics Options dialog box	Sets the time, in picoseconds, to change the temperature from Simulation to Final. Default = 0

Keyword	Value Type	Value	How to Set	Use
TimeStep	Real	0–1e+10	Molecular Dynamics Options dialog box	Sets the Step size, in picoseconds, between evaluations of the total energy and forces for a molecular system. Default = 0.001
Starting-Temperature	Real	0–1e+10	Molecular Dynamics Options dialog box	Sets the temperature of the system at the beginning of Heat Time. Default = 0 Kelvin
Simulation-Temperature	Real	0–1e+10	Molecular Dynamics Options dialog box	Sets the temperature of the system at the beginning of Run Time. Default = 300 Kelvin
FinalTemperature	Real	0–1e+10	Molecular Dynamics Options dialog box	Sets the temperature of the system at the end of Cool Time. Default = 0 Kelvin
TemperatureStep	Real	0–1e+10	Molecular Dynamics Options dialog box	Sets the increments of temperature change, if any, during the calculation. Default = 0 Kelvin
Constant-Temperature	Bool.	Yes, No	Molecular Dynamics Options dialog box	Turns on or off Constant temperature during a molecular dynamics simulation. Default = Yes
Temperature-RelaxationTime	Real	0–1e+10 (> TimeStep)	Molecular Dynamics Options dialog box	Sets the Bath relaxation time constant, in picoseconds, for constant temperature regulation. Default = 0.1
Restart	Bool.	Yes or No	Molecular Dynamics Options dialog box	Sets the initial velocities of atoms to previously determined values. Default = No



Keyword	Value Type	Value	How to Set	Use
DataFrequency	Int	1–max. number of time steps or 32767	Molecular Dynamics Options dialog box	Sets the interval, in time steps, for the molecular dynamics calculation to send new values to HyperChem. Same as Data collection period. Default = 1
RefreshFrequency	Int	1–max. number of data steps or 32767	Molecular Dynamics Options dialog box	This sets the interval, in data steps (see DataFrequency) for updating the display (same as Screen refresh period). Default = 1
SnapShot-Frequency	Int	1–max. number of data steps or 32767	Molecular Dynamics Snapshots dialog box	Sets the interval, in data steps, between snapshots of a molecular dynamics simulation. Same as Snapshot period. Default = 1
AveragesFrequency	Int	1–max. number of data steps or 32767	Molecular Dynamics Averages dialog box	Sets the interval, in data steps, between averaging values in a molecular dynamics simulation. Same as Average/graph period. Default = 1
SCFConvergence	Real	0.0–100.0	Semi-empirical or Ab Initio and DFT Options dialog box	This is the limit to which the difference of SCF calculations should be calculated.
SCFIteration	int	1–32767	Semi-empirical or Ab Initio and DFT Options dialog box	The maximum number of iterations to extend an SCF calculation.
Opt	Enum	Orbital, OrbitalSquared	Orbitals dialog box (orbital squared checkbox)	This controls the appearance of the orbital plots.

Keyword	Value Type	Value	How to Set	Use
Orbital	Enum	Homo, LUMO, Number	Orbitals dialog box	This controls the type of information displayed in the orbital number field when an orbital is selected.
HorizontalGrid	int	2–32767	Grid dialog box	Number of grid points in the x dimension.
VerticalGrid	Int	2–32767	Grid dialog box	Number of grid points in the y dimension
GraphContours	Int	1–32767	Grid Dialog Box	Determines the number of contour level in a contour plot.
UHF	Boolean	Yes, No	Semi-empirical or Ab Initio and DFT Options dialog Box	Selects whether a semi-empirical calculation will use the Unrestricted or restricted Hartree-Fock approximations.
ExcitedState	Boolean	Yes, No	Semi-empirical or Ab Initio and DFT Options dialog Box	Determines whether the current configuration No is of the ground state or the first excited state.
Accelerate Convergence	Boolean	Yes, No	Semi-empirical or Ab Initio and DFT Options dialog Box	If Yes, then uses DIIS procedure to accelerate
HuckelConstant	Real	0–10.0	Extended Hückel Options dialog box	The proportionality factor to use in an Extended Hückel calculation. Usually 1.75
HuckelScaling Factor	Real	0–100000	Extended Hückel Options dialog box	Scaling factor for interaction between classical and quantum portions of a system when using mixed mode calculation.

Keyword	Value Type	Value	How to Set	Use
Weight	Boolean	Yes, No	Extended Hückel Options dialog box	A toggle for using a simple Hückel constant (No) or a constant including a weighting factor for orbital diffuseness (Yes).
DOrbitalsSecond Row	Boolean	Yes, No	Extended Hückel Options dialog box	Yes to include d orbitals from the basis set for the second long row of atoms, (Si to Cl), or no to ignore d orbitals (and shorten the calculation)
BetaOrbitals	Boolean	Yes, No	Orbital Options dialog box	When using the UHF approximation, toggles the plot of alpha (no) or Beta (yes) orbitals on a contour plot.
Configuration Interaction	enum	NoCI, SinglyExcited, Microstate	Configuration Interaction dialog box	Determines which type of configuration interaction is performed.
QuantumLevel Shift	int	0–32767	Registry <sup>a</sup>	This is the number of SCF iterations to perform a level shift, which separates the HOMO and LUMO levels during an SCF calculation in order to accelerate convergence.
CICriterion	enum	Orbital, Energy	Configuration Interaction dialog box	This determines whether the criterion for the number of orbitals to include in a configuration interaction is based upon the lowest and highest orbitals, or the maximum excitation energy of an electronic transition.
CIUnoccupied	int	0–32767	Configuration Interaction dialog box	This is the number of unoccupied orbitals to include in a CI calculation.
CIOccupied	int	0–32767	Configuration Interaction dialog box	This is the number of occupied orbitals to include in a CI calculation.

Keyword	Value Type	Value	How to Set	Use
CIExcitation Energy	Real	0–10000.0	Configuration Interaction dialog box	This is the maximum excitation energy to use as a limit for the number of configurations to use in a CI calculation.
QuantumMultiplicity	Int	1–6	Extended Hückel Options dialog box	The spin multiplicity of the system in the workspace
Molecular-Mechanics-Method	Enum	mm+ amber bio+ opls	Molecular Mechanics-Force Field dialog box	Sets the force field for a molecular mechanics calculation. The force field stored in a HIN file changes this setting. Default = mm+
SemiEmpirical-Method	Enum	Extended-Huckel CNDO INDO MINDO3 MNDO AM1 PM3 ZINDO1 ZINDOS	Semi-empirical Method dialog box	Sets the method for a semi-empirical calculation. Default = ExtendedHuckel
CalculationMethod	Enum	Molecular-Mechanics SemiEmpirical Ablnitio	Choose on the Setup menu	Sets the type of calculation method. Default = MolecularMechanics
MechanicsPrint Level	Int	0–9	Start Log dialog box	The level of information output to a log file for molecular mechanics calculations.
QuantumPrint Level	Int	0–9	Start Log dialog box	The level of information output to a log file for quantum mechanics calculations.

Keyword	Value Type	Value	How to Set	Use
WarningType	Enum	None, Log, Message	Registry <sup>a</sup>	Determines the destination of warning messages for calculations: None = no warnings, Log = messages to log file only, Message = messages to screen and log file. Default = Message
DynamicsGraph-Onset	Int	3–max. number of averaging steps or 32767	Registry <sup>a</sup>	Sets the number of averaging steps (see AveragesFrequency) to compute before plotting instantaneous values. Default = 3
Animate Vibrations	Boolean	Yes, No	Vibrational Spectrum Dialog Box	Determines whether a vibrational normal mode will be displayed as vectors or as a molecular animation.
IRAnimate Amplitude	Real	0.0–10.0	Vibrational Spectrum Dialog Box	The amplitude of the vibrations of a normal mode.
IRAnimateSteps	Int	3–32767	Vibrational Spectrum Dialog Box	The number of steps during each half cycle of a vibration animation.
TndoVersion	Int	0-1	Semi-empirical Options Dialog Box	The version of TNDO used (0=CNDO-based, 1=INDO-based). Default = 1
ExchangeFunctional	Enum	None Hartree-Fock Slater Becke88 PW91 Gill96 PBE96 HCTH98 B3-LYP B3-PW91 EDF1 Becke97	Exchange Correlation Potential Dialog Box	The type of Exchange Functional to use in a Density Functional (DFT) calculation. The last four combination functionals are specified by a particular value of the this variable (ignoring the correlation functional below. Default = None.

Keyword	Value Type	Value	How to Set	Use
CorrelationFunctional	Enum	None Perdew86 VWN LYP PZ81 PW91 PBE96 HCTH98	Exchange Correlation Potential Dialog Box	The type of Correlation Functional to use in a Density Functional (DFT) calculation. This value is ignored if the exchange functional above is a combination functional (one of the last four). Default = None.
ReadHCTHCoefficients	Boolean	Yes ,No	Registry <sup>a</sup>	If Yes, then 15 coefficients are read from file hcth.coef rather than use internal HCTH values. The file has five rows of the three coefficients, $C_X$ , $C_{C\sigma\sigma}$ , $C_{C\alpha\beta}$ for the exchange, parallel and anti-parallel correlation.
UseSmallDensityCutoff	Boolean	Yes, No	DFT Integration Grids Dialog Box	If Yes, then all integration points involving a density smaller than SmallDensityCutoff (see below) are ignored. If No, then the computation is performed for all integration points. Default = No.
SmallDensityCutoff	Real	0 to .001	DFT Integration Grids Dialog Box	The smallest density to ignore if UseSmallDensityCutoff (see above) is Yes. Default = 1.0e-6.
StateToOptimize	Enum	SCF MP2 CI	Variable To Optimize Dialog Box	The energy to optimize for Conjugate Directions Optimizer. For SCF, Ab Initio SCF or DFT ground states are optimized. For MP2, the Ab Initio SCF energy plus the MP2 energy is optimized. For CI, the configuration interaction excited state, CIStateToOptimize, is optimized. Default = SCF.

Keyword	Value Type	Value	How to Set	Use
CIStateToOptimize	Int	0 to number of CI excited States	Variable To Optimize Dialog Box	Integer specifying which of the excited states coming from a Configuration Interaction calculation to optimize. The value 0 specifies the ground state, the value 1 specifies the first excited state, etc. Default = 0.
MechanicsComponentFlags	Int	Depends on which flags, starting at low bit, are set.	Components of Force Field Included Dialog Box	A set of bit flags which, if set, request that a component of the force field be included. Bit 0 is Bond terms, Bit 1 is Angle terms, etc. Default is -1 (all bits set).
UseMullikanApproximationtoDensity	Boolean	Yes, No	Density Functional Method Dialog Box	New Feature (Mullikan Approximation) - may not be available.
UseAuxilliaryBasis	Boolean	Yes, No	Density Functional Method Dialog Box	New Feature (Orbital plus Auxilliary Basis Sets) - may not be available.
AuxilliaryBasis	Int	0 to (number of auxilliary basis sets available -1)	Density Functional Dialog Box	New Feature (Orbital plus Auxilliary Basis Sets) - may not be available.

a. Use direct setting of Registry to change this item.

## Filters Settings

This key describes the file filters in HyperChem.

## Filters

Keyword	Value Type	Value	How to Set	Use
MDL MOL	String	<description> <filter> <ddlfilename> <file type>	Registry	Directs HyperChem to the .DLL file that converts between .HIN files and MDL .MOL files. Default=*.mol molhin.dll MOL
MOPAC Z-Matrix	String	<description> <filter> <ddlfilename> <file type>	Registry	Directs HyperChem to the .DLL file that converts between .HIN files and MOPAC Z-Matrix files. Default=*.zmt zmathin.dll ZMT
Tripos MOL2	String	<description> <filter> <ddlfilename> <file type>	Registry	Directs HyperChem to the .DLL file that converts between .HIN files and Tripos MOL2 files. Default=*.ml2 ml2hin.dll ML2
ChemDraw	String	<description> <filter> <ddlfilename> <file type>	Registry	Directs HyperChem to the .DLL file that converts between .HIN files and ChemDraw files. Default=*.chm chemdrw.dll CHM



# MM+ Settings

## mm+

Keyword	Value Type	Value	How to Set	Use
CurrentFiles	Int	0 to (number of Custom-Names-1)	Select "Force field" Parameter Set dialog box	The index of the MM+ parameter set to use. 0 for first Custom-Name, 1= second, and so on. Default = 0
AtomTypeMass	Enum	Filename: *typ.txt *typ.dbf or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types and masses for this force field. Default = mmptyp.txt
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type of file containing MM+ AtomTypeMass parameters. Default = Text
CustomNames	Enum	Names separated by commas	Registry <sup>a</sup>	The names of parameter sets associated with the MM+ force field. Default = mmplus

a. Use direct setting of Registry to change this item.

## mm+, *CustomName*

These are subkeys that define the settings for the MM+ parameter set with the *CustomName* mmplus:

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type for the following files. Default = Text
MMPElectrostatics	Enum	BondDipoles, AtomicCharges	MM+ Options dialog box	Specifies the form of electrostatic interactions used for MM+ calculations. Default = BondDipoles

Keyword	Value Type	Value	How to Set	Use
MMPStretch	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond stretch parameters for this parameter set. Default = mmpstr.txt
MMPBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle bending parameters for this parameter set. Default = mmpben.txt
MMPBend4	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle bending parameters for four-membered rings, for this parameter set. Default = mmpben4.txt
MMPBend3	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle bending parameters for three-membered rings, for this parameter set. Default = mmpben3.txt
MMPOOPBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing the out-of-plane bending parameters for this parameter set. Default = mm poop.txt
MMPStretchBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing parameters coupling bond stretching and bond angle bending for this parameter set. In this release, stretch-bend constants are internal to HyperChem and not available in a parameter file. Default = none
MMPTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond torsion parameters for this parameter set. Default = mmptor.txt

Keyword	Value Type	Value	How to Set	Use
MMPTorsion4	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond torsion parameters for four-membered rings, for this parameter set. Default = mmptor4.txt
MMPAtomVDW	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing atom-based van der Waals interaction parameters for this parameter set. Default = mmpnbd.txt
MMPPairVDW	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing pair-based van der Waals interaction values for this parameter set. In this release, pair-based VDW constants are internal to HyperChem and not available in a parameter file. Default = none
MMPCubicStretch	Real	Any negative real number	Registry <sup>a</sup>	The MM+ cubic stretch coefficient for calculating bond stretch energies. Default = -2

a. Use direct setting of Registry to change this item.

# Network Settings

## Network

With this key you can specify options for running your calculations on different machines on your network with HyperChem's remote calculation modules.

Keyword	Value Type	Value	How to Set	Use
Networking	Boolean	Yes, No	Registry <sup>a</sup>	Whether you have one or more backend modules (Yes) or not (No)
BackendCommunications	Enum	Local, Remote	Network Options dialog box	Whether calculations should be run remotely or on the local machine
BackendEthernet-Server	String	Any string	Network Options dialog box	Specifies the name of the remote machine that the backend will run on
BackendUserId	String	Any string	Network Options dialog box	Specifies the login name that should be used on the remote machine
BackendNodes	Int	1	Network Options dialog box	Specifies the number of parallel processes to run. Ignored in the current version of HyperChem, which will use 1

a. Use direct editing of Registry to change this item.

# OPLS Settings

## opls

Keyword	Value Type	Value	How to Set	Use
CurrentFiles	Int	0 to (number of CustomNames -1)	Select "Force field" Parameter Set dialog box	The index of the OPLS parameter set to use. Default = 0 for first CustomName; 1= second, and so on. Default = 0
AtomTypeMass	Enum	Filename: *typ.txt *typ.dbf or any other	Registry <sup>a</sup>	Directs HyperChem to the file containing the atom types for this force field. Default = oplstyp.txt
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type of file containing OPLS AtomTypeMass parameters. Default = Text
CustomNames	Enum	Names separated by commas	Registry <sup>a</sup>	The names of parameter sets associated with the OPLS force field. Default = opls

a. Use direct editing of Registry, to change this item.

## opls, CustomName

These are the subkeys that define the settings for the parameter set with the *Custom Name* opls:

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	Text Dbf	Registry <sup>a</sup>	Sets the type for the following files. Default = Text

Keyword	Value Type	Value	How to Set	Use
Dielectric	Enum	Constant	Force Field Options dialog box	Sets the dielectric permittivity for calculating electrostatic interactions to Constant. There are no other choices. Default = Constant
DielectricScale	Real	1–1000	Force Field Options dialog box	Sets the scale factor for dielectric permittivity (see previous entry). Default = 1
1–4Electrostatic-Scale	Real	0–1	Force Field Options dialog box	Sets the scale factor for calculating charge interactions between atoms separated by three bonds. Default = 0.5
1–4vanderWaals-Scale	Real	0–1	Force Field Options dialog box	Sets the scale factor for calculating charge interactions between atoms separated by three bonds. Default = 0.125
6–12AtomVDW-Format	Enum	RStarEpsilon SigmaEpsilon SlaterKirkwood	Registry <sup>a</sup>	Sets the method for calculating atom-based van der Waals interactions from data in the file specified in 6–12AtomVDW (see below). <sup>b</sup> Default = SigmaEpsilon
6–12PairVDW-Format	Enum	RStarEpsilon SigmaEpsilon AijBij	Registry <sup>a</sup>	Sets the method for calculating pair-based van der Waals interactions from data in the file specified in 6–12PairVDW (see below). <sup>b</sup> Default = SigmaEpsilon
QuadraticStretch	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond stretch parameters for this force field. Default = oplsstr.txt

Keyword	Value Type	Value	How to Set	Use
QuadraticBend	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing bond angle parameters for this parameter set. Default = oplshen.txt
FourierTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing torsion parameters for this parameter set. Default = oplstor.txt
FourierImpTorsion	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing improper torsion parameters for this parameter set. Default = opsimp.txt
6-12AtomVDW	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for individual atom types. <sup>b</sup> Default = oplshbd.txt
6-12PairVDW (optional)	Enum	Filename	Registry <sup>a</sup>	Directs HyperChem to the file containing van der Waals parameters for specific pairs of atom types. <sup>b</sup> Default = none

a. Use direct editing of Registry, to change this item.

b. See *HyperChem Computational Chemistry, Theory and Methods*.

# Path Settings

## Path

With this key you can specify where HyperChem should find its program files, parameter files, etc.

Keyword	Value Type	Value	How to Set	Use
ChemExePath	string	Any path name	File/Preferences/Paths property sheet	The directory which contains the HyperChem program files
ChemParmPath	string	Any path name	File/Preferences/Paths property sheet	The directory which contains the parameter files for molecular mechanics and semi-empirical calculations.
ChemLogPath	string	Any path name	File/Preferences/Paths property sheet	The directory that log files will be stored in
ChemScriptPath	string	Any path name	File/Preferences/Paths property sheet	The directory that script files are read from



# Preference Settings

## Preferences

Keyword	Value Type	Value	How to Set	Use
ButtonRotation-Angle	Real	0–3600°	Key Preferences dialog box	The maximum step size for XY rotation using the [3]/[4]/[2]/[1] keys. Default = 7.5 degrees
ButtonZRotation-Angle	Real	0–3600°	Key Preferences dialog box	The maximum step size for Z rotation using the [Home]/[End] keys. Default = 7.5 degrees
DraggingRotation-Angle	Real	0–3600°	Tool Preferences dialog box	The maximum XY rotation by moving the XY rotation tool across the screen once. Default = 360 degrees
DraggingZRotation-Angle	Real	0–3600°	Tool Preferences dialog box	The maximum Z rotation by moving the Z rotation tool across the screen once. Default = 360 degrees
ButtonTranslate-Step	Int	0–1000Å	Key Preferences dialog box	The maximum step size for XY translation using the [Shift] + [3]/[4]/[2]/[1] keys. Default = 1 Ångstrom
ButtonZTranslate-Step	Int	0–1000Å	Key Preferences dialog box	The maximum step size for Z translation using the [Shift] + [Home]/[End] keys. Default = 1 Ångstrom
DraggingZ-TranslateStep	Int	0–1000Å	Tool Preferences dialog box	The maximum Z translation by moving the Z translation tool across the screen once. Default = 10 Ångstroms

Keyword	Value Type	Value	How to Set	Use
ButtonClipping-Step	Int	0–1000Å	Key Preferences dialog box	The maximum clipping plane movement using the [Shift]/[Ctrl] + [PgDn]/[PgUp] keys. Default=1 Ångstrom
DraggingClipping-Step	Int	0–1000Å	Tool Preferences dialog box	The maximum clipping plane movement by moving the Z clipping tool across screen once. Default=10 Ångstroms
ButtonZoomStep	Int	>1–1000Å	Key Preferences dialog box	The maximum step size for magnification using the [PgDn]/[PgUp] keys. Default= 2
DraggingZoom-Step	Int	>1–1000	Tool Preferences dialog box	The maximum magnification by moving the Zoom tool across screen once. Default=10
MouseMode	Enum	Drawing Selecting Rotating ZRotating Translating ZTranslating Zooming Clipping	Choose a Tool with the mouse	Sets the default tool and cursor. Default = Drawing
MultipleSelections	Bool.	Yes, No	Select menu	Sets Multiple Selections on or off. Default = No
SelectSphere	Bool.	Yes, No	Select menu	Sets Select Sphere on or off. Default = No
SelectLevel	Enum	Atoms Residues Molecules	Select menu	Sets the selection level. Default = Atoms
FastTranslate	Bool.	Yes, No	Tool Preferences dialog box	Sets Fast translation on or off. Default = Yes

Keyword	Value Type	Value	How to Set	Use
TranslateMolecules	Bool.	Yes, No	Tool Preferences dialog box	Sets Whole molecule translation on or off. Default = Yes
DefaultAtomic-Number	Int	0–103	Element Table dialog box	Sets the default element for drawing. 0 = lone pair. Default = 6
ExplicitHydrogens	Bool.	Yes, No	Build menu	Sets Explicit Hydrogens to on or off. Default = No
AllowIons	Bool.	Yes, No	Build menu	Sets Allow Ions to on or off. Default = No
ConstrainDrawing	Bool	Yes, No	Build Menu	Sets drawing mode to constrained bonds and angles on or off. Default = No
HydrogenBond-Distance	Real	Positive real number	Registry <sup>a</sup>	Sets the maximum length for an H bond. Default = 3.2 Ångstroms
HydrogenBond-Angle	Real	0–180°	Registry <sup>a</sup>	Sets the minimum angle for an H bond. Default = 150 degrees
Perspective	Bool.	Yes, No	Display menu	Sets Perspective View on or off. Default = No
HydrogenShown	Bool.	Yes, No	Display menu	Set the display of hydrogens on or off. Default = Yes
MultipleBond-Shown	Bool.	Yes, No	Display menu	Sets the display of multiple bonds on or off. Default = Yes
AromRingShown	Bool	Yes, No	Display menu	Sets the display of aromatic rings as circles on or off. Default = No
HBondShown	Bool.	Yes, No	Display menu	Sets the display of hydrogen bonds on or off. Does not calculate bonds. Default = Yes

Keyword	Value Type	Value	How to Set	Use
IsosurfaceType	Enum	wire-mesh, Jorgensen-Salem, lines, flat-surface, shaded-surface, Gouraud-shaded-surface, translucent-surface	Isosurface Options dialog box	Specify the rendering mode for isosurfaces
StickPixelWidth	int	0 to 25	Sticks property sheet	Specify width of bonds drawn as sticks, in units of pixels
BallsRadius	float	0 to 1	Balls property sheet	Size of the balls relative to the maximum value, in balls rendering
BallsCylindersRadius	float	0 to 1	Balls property sheet	Size of the balls relative to the maximum value, in balls and cylinders rendering
CylindersRadius	float	0 to 1	Cylinders property sheet	Size of cylinders relative to the maximum value, in balls and cylinders rendering
BackgroundColor	Enum	<i>Color</i> <sup>b</sup>	Preferences dialog box	Sets the color of the HyperChem workspace (window). Default = Black
AtomColor	Enum	ByElement or <i>color</i> <sup>b</sup>	Preferences dialog box	Sets the color of all bonds. Default = ByElement
SelectionColor	Enum	ThickLine or <i>color</i> <sup>b</sup>	Preferences dialog box	Sets the display for selected atoms and bonds. Default = Green

Keyword	Value Type	Value	How to Set	Use
DefaultAtomLabels	Enum	None AtomSymbol AtomName AtomNumber AtomType BasisSet Mass Charge Chirality	Labels dialog box	Sets the type of label for an atom. Default = None
DefaultResidue-Labels	Enum	None ResidueName Sequence Name-Sequence	Labels dialog box	Sets the type of label for a residue. Default = none
CPKNumberOf-Rings	Int	0–32767	This file <sup>a</sup>	Sets the number of concentric disks used in a Spheres rendering. Higher numbers improve the appearance but increase computing time. Default = 20
CPKMaxDouble-BufferAtoms	Int	0–number of atoms	This file <sup>a</sup>	If DoubleBuffered is on (Yes) and there is enough memory, HyperChem displays a complete rendering containing up to this number of atoms. More atoms or lack of memory turns off double buffering and HyperChem shows the rendering atom-by-atom. Default = 50
DotSurfaceAngle	Real	–90 to +90°	This file <sup>a</sup>	This is the angle of dot planes relative to the XZ plane of the Viewer's Coordinate System. A value of 0 sets the dot planes perpendicular to the screen. Default = 25 degrees

Keyword	Value Type	Value	How to Set	Use
BondSpacing-DisplayRatio	Real	0-0.1	This file <sup>a</sup>	This setting determines the line spacing of double, triple, and aromatic bonds for printing Sticks renderings. The spacing is the product of this value and the smallest dimension of a rectangle (width or height) that encloses the molecular system. Default = 0.01
WallEyedStereo	Bool.	Yes or No	This file <sup>a</sup>	This setting turns on either walleyed stereo (Yes) or cross-eyed stereo (No). Default = Yes
FillPDBRecords	Bool.	Yes or No	This file <sup>a</sup>	Adds blank spaces to each record to 70 characters. Some programs reading PDB files might require this. Default = No
WritePDB-Hydrogens	Bool.	Yes or No	Save File dialog box	Includes ATOM or HETATM records for hydrogen atoms in PDB files. Default = No
WritePDBConnect	Bool.	Yes or No	Save File dialog box	Includes all CONECT records, with bonding information, in a PDB file. Default = No
SaveVelocities	Bool.	Yes or No	Save File dialog box	Includes Vel records in a HIN file, with the velocity for each atom. Default = No
PrinterBackgroundWhite	Boolean	Yes, No	This file <sup>a</sup>	Controls whether the printer background color is white
ToolHelpPopup	Bool.	Yes, No	This file <sup>a</sup>	Enables popup help messages when cursor pauses over tool icons. Default = Yes

Keyword	Value Type	Value	How to Set	Use
FileFormat	Enum	HIN SKC PDB ML2 ZMT	Save File dialog box	Sets the type of file for storing a molecular system. Default = HIN
UseCustomMenu	Bool.	Yes, No	This file <sup>a</sup>	Specifies whether a custom menu should be used (Yes) or the default menu used (No)
CustomMenuFile	File-name	Any filename	This file <sup>a</sup>	Specifies the name of the file containing the custom menu
ShadedBalls	Bool.	Yes, No	Balls property sheet	Specifies if shading should be used in Balls rendering
ShadedCPK	Bool.	Yes, No	Spheres property sheet	Specifies if shading should be used in Spheres rendering
HighlightedBalls	Bool.	Yes, No	Balls property sheet	Specifies if highlighting should be used in Balls rendering
HighlightedCPK	Bool.	Yes, No	Spheres property sheet	Specifies if highlighting should be used in Spheres rendering
ColorCylindersByElement	Bool.	Yes, No	Cylinders property sheet	Specifies if cylinders should be colored according to the atoms that they join (Yes) or drawn in gray (No)
NetLicense	Int	0 to 15	Network Preferences property sheet	Specifies which networks to request a network license from. The setting is the sum of the following values: 8 if TCP/IP is enabled, 0 otherwise; 4 if IPX/SPX is enabled, 0 otherwise; 2 if NETBIOS/NETBEUI/NETAP is enabled, 0 otherwise; 1 if Local access is enabled, 0 otherwise

Keyword	Value Type	Value	How to Set	Use
NetTimeoutLicense	Int	0 to 32767	Network Preferences property sheet	Specifies the time in minutes without the user activating a menu item before a network license will time out.
TCPNetMask	String	Four integers (each 0 to 255) separated by periods	Network Preferences property sheet	Specifies the mask used for TCP/IP addressing
TCPTimeout	Int	0 to 32767	Network Preferences property sheet	Specifies the maximum time in seconds to allow for a response from the TCP/IP network request
TCPRetry	Int	0 to 32767	Network Preferences property sheet	Specifies the maximum number of times to try to get a response to a TCP/IP network request
UseSockets	Bool	Yes, No	Registry <sup>a</sup>	Specifies that Sockets are to be used for (front end - back end) communication rather than DDE. Default = No.
UseAsyncTrans	Bool	Yes, No	Registry <sup>a</sup>	Use an asynchronous version of DDE rather than a synchronous version. Default = No.
PlotInterpPoints	Int	0 to 10	Registry <sup>a</sup>	Number of interpolated points added to potential plots to smooth out data. Default = 3.
PlotOutliersEliminated	Int	0 or 1	Registry <sup>a</sup>	Whether or not to eliminate outliers in potential plots that would distort plot. 0 is No, 1 is Yes. Default = 0.
AnnotationColor	String	Three numbers separated by space; e.g. 255 255 255 is white	Annotation/Color dialog box	Color used for lines, circles, and rectangles. Not used for text annotations.



Keyword	Value Type	Value	How to Set	Use
AnnotationLayerHidden	Bool.	Yes, No	Annotations/Options menu	Whether or not to hide the annotation layer
AnnotationLayerInFront	Bool.	Yes, No	Annotations/Options menu	Whether or not to have the annotation layer in front of the molecular layer
AnnotationFilled	Bool.	Yes, No	Annotations/Options menu	Whether or not to fill circle and rectangle annotations

- a. Use direct editing of Registry to change this item.  
 b. *Color*: black, blue, green, cyan, red, violet, yellow, white

## Registration Settings

### Registration

This key stores information about the registered user. It is in HKEY\_LOCAL\_MACHINE rather than HKEY\_CURRENT\_USER..

Keyword	Value Type	Value	How to Set	Use
User	string	Any string	Installation	The name of the registered user
Organization	string	Any string	Installation	The company or organization of the registered user
SerialNo	string	Valid serial number	Installation	The serial number of the product
Dealer	string	Any string	Installation	The name of the product dealer



# Credits and Acknowledgments

HyperChem uses public domain force fields and semi-empirical methods with author approval. Hypercube would like to thank the following authors for their valuable input and assistance:

Dr. Norman Allinger, University of Georgia; author of the MM2 force field.  
Dr. Peter Kollman, University of California at San Francisco; author of the AMBER force field.  
Dr. William Jorgensen, Yale University; author of the OPLS force field.  
Dr. Martin Karplus, Harvard University; author of the CHARMm force field.

Dr. John A. Pople, Carnegie-Mellon University; author of the CNDO and INDO semi-empirical methods and father of *ab initio* methods.  
Dr. Michael J.S. Dewar, The University of Texas at Austin; author of the MINDO, MNDO, and AM1 semi-empirical methods.  
Dr. James J.P. Stewart, Frank J. Seiler Research Laboratory, United States Air Force Academy; author of the PM3 semi-empirical method.  
Dr. Michael C. Zerner, University of Florida; author of the ZINDO/1 and ZINDO/S semi-empirical methods.

HyperChem uses sample data files from Brookhaven National Laboratories, Protein Data Bank. Additional Protein Data Bank files are used with author approval. Hypercube would like to thank the following authors for allowing us to redistribute their file structures:

Professor Roberto J. Poljack, Institute Pasteur; author of 1F19.ent.  
Professor Dr. R. Huber, MPI f. Biochemie; author of 2PTC.ent.  
Dr. Larry C. Sieker, University of Washington; author of 2EST.ent.  
Professor T.L. Blundell, Birbeck College; author of 1GCN.ent and 1XY1.ent.  
Professor S.H. Kim, University of California at Berkeley; author of 6TNA.ent.  
Professor Martha M. Teeter, Boston College; author of 1CRN.ent.

Hypercube would like to thank Eric Drexler, President, Foresight Institute and visiting scholar at Stanford University, and Ralph Merkle of Xerox Parc for use of the bearing1.hin and bearing2.hin files.

---



# Glossary

In this Glossary, terms in bold have their own entries.

**Ab initio.** A form of **quantum mechanics** calculation in which each orbital is computed as the sum of a set of primitive geometric functions (the basis set). By computing the interaction of many orbital wave functions, *ab initio* calculations can give very accurate results, at the cost of requiring large amounts of storage space and relatively long computation times.

**AM1.** Austin Model 1. A semi-empirical, **SCF** method for chemical calculations. One of the MOPAC methods [Stewart, J.J.P., *J. Comp.-Aided Molec. Design* 4(1):1-108, 1990]. An improvement of the **MNDO** method. Useful for organic molecules containing elements from long rows 1 and 2 of the periodic table, but not transition metals. Generally the most accurate method included in HyperChem. Calculates electronic properties, optimized geometries, total energy, and heat of formation. Reference: Dewar, M.J.S., *et al.*, *J. Am. Chem. Soc.* 107:3902, 1985.

**All atom.** A molecular mechanics calculation that represents each atom as an individual particle. Contrast to **united atom**.

**AMBER.** Assisted Model Building and Energy Refinement. A molecular mechanics method, or **force field**, for chemical calculations. Developed for proteins and nucleic acids. This force field provides both an **all atom** and a **united atom** representation. Reference: Weiner, S.J., *et al.*, *J. Comp. Chem.* 7:230-252, 1986.

**Atom type.** Atom types represent classes of chemical environments assigned to each atom used in a force field calculation. The characteristics of an environment include hybridization, formal charge, and immediate bonded neighbors. For example, the AMBER atom type C represents an  $sp^2$  carbonyl carbon, CT represents a tetrahedral carbon, and CH is a *united atom* representation of an  $sp^3$  carbon including one hydrogen. Each force field has a

different set of atom types. HyperChem can assign atom types to a molecular system, using rules in chem.rul.

**Backbone.** The continuous covalent path through a molecule. For polypeptides, the backbone includes all peptide bonds, alternating with alpha carbons, plus disulphide bonds. For polynucleotides, the backbone includes all phosphodiester bonds, alternating with the C<sub>3'</sub>-C<sub>4'</sub> carbons in the sugar (ribose or deoxyribose).

**Back end.** HyperChem computational software. HyperChem is constructed in modules, with **front-end** software running on a PC and back-end software running on the same PC or another computer. Back-end software includes HyperNewton (AMBER, BIO+, and OPLS), HyperMM+ (MM+), HyperGauss (*ab initio*), HyperEHT (Extended Hückel), and HyperNDO (CNDO, INDO, MINDO/3, MNDO, AM1, and PM3).

**BIO+.** An implementation of the CHARMM (Chemistry at HARvard Macromolecular Mechanics) force field, developed in the group of Martin Karplus at Harvard University. A molecular mechanics method, or **force field**, for chemical calculations. Developed for proteins. References: Brooks, B.R., *et al.*, *J. Comp. Chem.* 4:187-217, 1983; Reiher, W.E., II, *Theoretical Studies of Hydrogen Bonding*, Ph.D. thesis, Harvard University, 1985.

**Bitmap.** A screen image composed of individual pixels (dots). A bitmapped image has the same resolution as the monitor.

**Brookhaven Protein Data Bank (PDB).** A database of protein and nucleic acid structures. Information for each molecular structure includes atomic coordinates, atomic connectivity, and references. HyperChem can read in and correctly interpret all PDB files with atomic coordinate or connectivity information. PDB files have the extension *.ent*.

**Charge density.** Distribution of electronic charge density for a molecule. The shape of a molecule is often associated with a surface of constant charge density. HyperChem calculates charge density as a sum of molecular orbital densities, each the square of the orbital wave function.

**Clipboard.** The temporary storage area for data and graphics provided by Windows. Cut and Copy in HyperChem place a text description of atoms onto the Clipboard. Copy Image in HyperChem places graphic images (both metafile and bitmapped) onto

the Clipboard. You can view the Clipboard contents directly using the Clipboard program in the Windows Main Applications group.

**Click.** Press and release a mouse button quickly.

**Clipping.** Displaying a section through a molecular system in the HyperChem **workspace**. Hides unwanted foreground and background atoms so you can focus on an area of interest. The section is called a **clipping slab**. This is a rectangular box with adjustable depth, and with height and width equal to the height and width of the HyperChem workspace. See **clipping planes**.

**Clipping planes.** The front and back boundaries of a **clipping slab**, individually adjustable from 0 Ångstroms (the viewer's position) to a number of Ångstroms depending on the size and magnification (Zoom factor) of the molecular system.

**Clipping slab.** A rectangular section through a molecular system bounded by two adjustable **clipping planes**.

**Closed-shell.** A molecule with no unpaired electrons.

**CNDO.** Complete Neglect of Differential Overlap (see **NDO**). This is the simplest of the SCF methods for semi-empirical quantum mechanics calculations. It is useful for calculating ground-state electronic properties of open- and closed-shell systems, geometry optimization, and total energy. HyperChem uses CNDO/2. Reference: Pople, J. A. and Beveridge, D.L., *Approximate Molecular Orbital Theory*, McGraw-Hill, 1970.

**Connectivity.** Information on the covalent bonding of an atom to each of its neighbors. PDB files can include this information in CONECT records. HIN files have this information at the end of Atom records, in <nbor# nbor-bonds> statements.

**Constraint.** Constraints for the **Model Builder** fix values of structural entities when an approximate 3D structure is being built. These constraints are a high priority for the Model Builder. However, adding excessive constraints can result in some being dropped.

**Convergence.** The criterion for ending a chemical calculation. A semi-empirical **SCF** mechanics calculation converges (ends) when the difference in energy after two consecutive iterations is less than a convergence limit (default of 0.01 kcal/mol) or the calculation reaches an iteration limit (default of 50). For geometry optimization calculations, the convergence limit is either a number of calculation cycles (the default number is 15 times the number of

atoms) or an RMS gradient for the molecular system (default of 0.1 kcal/Ångstrom mol).

**Coordinate system.** A three-dimensional (3D) space containing x, y, and z Cartesian coordinates for each atom in a molecular system. HyperChem has a Viewer's Coordinate System and a Molecular Coordinate System. From the Viewer's Coordinate System, you can translate and rotate your viewpoint relative to a fixed molecular system, giving the appearance of a moving molecular system. Rotation or translation in the Molecular Coordinate System changes the positions of individual atoms.

**Correlation Energy.** The energy that the Hartree-Fock method leaves out that describes the energy coming from the explicit correlation of the positions of electrons, rather than from their average positions. Ab Initio calculations can compute correlation energies via second-order Moller-Plesset theory (MP2). Density functional theory (DFT) gives correlation energies in a self-consistent-field theory with equations (the Kohn-Sham equations) that resemble the Hartree-Fock equations. The Kohn-Sham equations require an exchange functional and a correlation functional. They reduce to Hartree-Fock theory when the exchange functional is the Hartree-Fock functional and the correlation functional is omitted.

**Correlation Functional.** The dependence of the correlation energy upon the electron density. A formula for relating the electron density to the correlation energy. Density functional theory proposes methods for computing the exchange and correlation energy from the electron density.

**CPK.** Acronym for **Corey-Pauling-Koltun**; refers to a popular set of atomic 'radii' used for depicting space-filling pictures or building plastic models.

**Cursor keys.** The [1] [2] [3] [4] [Home] [End] keys. Use these keys alone, with [Shift], or with [Ctrl]+[Shift] to rotate or translate molecules in the **workspace**.

**Cursor.** The pointer or tool on screen that you move using the mouse. Use a cursor to choose items from menus, to draw and move molecules, to make choices in dialog boxes, and to edit text.

**Cycle.** During a **geometry optimization** calculation, the number of cycles indicates the number of times a calculation takes a new direction on the potential surface to locate a minimum. Each cycle includes one or more points, which are calculations to evaluate the



energy and gradient in this direction. HyperChem reports the cycle number and number of points on the **status line**.

**DDE.** Dynamic Data Exchange. A system for exchanging data between HyperChem and other Windows applications. HyperChem acts as a server: it supplies information only when a client application asks for specific data.

**Default element.** The element the HyperChem Drawing tool uses to construct molecules. There can be only one default element at a time, but you can change it as needed.

**Density functional.** The dependence of the energy upon a function (the electron density). Different functionals describe a different dependence upon the electron density. Density functional theory describes a method of calculating the energy by focussing on the electron density rather than the wavefunction.

**Desktop.** The screen display that Microsoft Windows uses to present windows and icons.

**Dialog box.** A type of Window for accepting information or displaying a message. The presence of most dialog boxes blocks other HyperChem functions. You must close these dialog boxes to reactivate HyperChem. Persistent dialog boxes (for example, the Amino Acids dialog box) do not interfere with HyperChem functions.

**Dielectric constant.** A factor (epsilon) that modifies charge-charge interactions during molecular mechanics calculations. In the absence of solvent molecules, a distance-dependent dielectric constant or a “constant” dielectric constant greater than 1 simulates the presence of solvent. In the presence of solvent, as in a hydrated periodic box, the dielectric constant should be Constant, rather than Distance dependent. For a constant dielectric,  $\epsilon = (\text{permittivity of free space}) \times (\text{scale factor})$ . For a distance-dependent dielectric,  $\epsilon = (\text{permittivity of free space}) \times (\text{scale factor}) \times (\text{interatomic separation})$ . See **electrostatic potential**.

**Drag.** Move an object on screen by holding down a mouse button while moving the mouse. Objects can be molecules or highlight bars in a menu. When you complete the movement, release the mouse button. For example, you can move a selected molecule by dragging a translation tool across the HyperChem workspace while holding the right mouse button. You can hold the left mouse button and move the mouse pointer through a menu to highlight a menu item. Release the mouse button to choose the item.

**Dynamic Data Exchange.** See DDE.

**Electrostatic potential.** The potential energy at every point in space arising from a distribution of charged atoms. The electrostatic potential ( $U$ ) at a distance  $r$  from a point charge is given by Coulomb's law:  $U = C_q/(\epsilon r)$ , where  $C$  is a constant (dependent on units) and  $\epsilon$  is the **dielectric constant** (or epsilon). For multiple charges in 3D space (both nuclear charges and electrons in molecular orbitals), this equation becomes appropriately complex. HyperChem calculates electrostatic potential as the potential energy of a unit positive charge interacting with the molecular system. In calculating the interaction with electrons in molecular orbitals, HyperChem neglects components due to diatomic differential overlap (see NDO). HyperChem calculates and plots electrostatic potential from the results of single point, semi-empirical quantum mechanics calculations. You can use these results to predict initial attack positions of protons and other ions during reactions.

**Exchange Functional.** The dependence of the exchange energy upon the electron density. A formula for relating the electron density to the exchange energy. Density functional theory proposes methods for computing the exchange and correlation energy from the electron density.

**Explicit hydrogens.** When Explicit hydrogens is active during drawing, HyperChem expects all hydrogen atoms to be drawn. Atoms with only one bond are drawn as hydrogen, rather than as the **default element**. This is in contrast to hydrogen atoms added by the Model Builder after drawing with only heavy atoms.

**Force constant.** A term in each equation for a **force field** that transforms data on atomic coordinates into a potential energy measurement. For example, for bond stretching energy:

$$E_{bond} = \sum_{bonds} K_r (r - r_0)^2$$
, where  $E$  is potential energy,  $r$  is bond length,  $r_0$  is equilibrium ("strain-free") bond length, and  $K_r$  is the force constant. For each force field, one or more **parameter sets** define the force constants and equilibrium geometries (for example,  $r_0$ ).

**Force field.** A molecular mechanics method for chemical calculations. These calculations can be single point, geometry optimization, or molecular dynamics. HyperChem provides four force fields: MM+, AMBER, BIO+, and OPLS. A force field has three components: equations defining the potential energy of a molecular

system as a function of atomic coordinates, atom types, and parameter sets that fit the equations to experimental data. See the separate entries for **atom type** and **parameter set**.

**Front end.** See **back end**.

**Geometry optimization.** A calculation, using either a molecular mechanics or quantum-mechanical method, to find a minimum energy (stable) configuration for a molecular system. The calculation adjusts atomic coordinates in steps to find a configuration in which net forces on each atom are reduced to zero. This is usually a **local minimum** on the potential surface, which may not necessarily be the **global minimum**.

**Global minimum.** The configuration of a molecular system with the lowest potential energy.

**Gradient.** The rate of change (first derivative) of the energy of a molecular system as a function of atomic positions. A gradient of zero indicates a configuration with minimum energy a (**local** or **global minimum** or a transition state). When HyperChem reports a single value for Gradient, this is the **RMS gradient** of the forces on all atoms.

**Ground state.** The lowest energy electronic state. See **lowest state**.

**Group selection.** A selection of a subset of atoms, residues, or molecules in a molecular system. The choice of Atoms, Residues, or Molecules on the Select menu regulates the level of group selection. With the Selection tool active, you group-select by dragging a boundary around the molecular structures, with both left and right mouse buttons pressed (press left before right). Select Sphere on the Select menu toggles group selection between a 2D and 3D process. 2D selection involves all atoms projecting onto a 2D selection rectangle. 3D selection involves all atoms within a sphere centered on an atom.

**HIN.** HyperChem Input file. This is the principal file type HyperChem uses to store molecular systems. The files have the extension *.hin*.

**HOMO.** Highest Occupied Molecular Orbital. This is used as a reference point for describing the position of any molecular orbital as being at or below this orbital. Molecular orbitals are a linear combination of atomic orbitals (LCAO).

**Hückel method.** A simple and approximate method for semi-empirical quantum mechanics calculations. The Extended Hückel method used in HyperChem is useful only for **single point** calculations, not for geometry optimization or molecular dynamic calculations. Extended Hückel calculations produce qualitative or semi-quantitative descriptions of molecular orbitals and electronic properties (for example, net atomic charges and spin distributions). This is not a Self-Consistent Field (**SCF**) method. Reference: Hoffman, R., *J. Chem. Phys.* 39(6):1397-1412, 1963.

**Icon.** A small graphic symbol representing a tool or application program.

**INDO.** Intermediate Neglect of Differential Overlap (see **NDO**). An SCF method for semi-empirical quantum mechanics calculations. It improves on **CNDO** by accounting for certain one-center repulsions between electrons on the same atom. Useful for calculating ground-state electronic properties of open- and closed-shell systems, geometry optimization, and total energy. Reference: Pople, J. A. and Beveridge, D.L., *Approximate Molecular Orbital Theory*, McGraw-Hill, 1970.

**Improper torsion.** A torsion (dihedral) angle not formed by a sequence of three bonds. Some molecular mechanics force fields use such angles in energy terms to maintain planarity or to prevent inversion of tetrahedral carbons that have one hydrogen in a **united atom** representation.

**Inertial axes.** The three axes (primary, secondary, and tertiary) associated with moments of inertia for a molecular system or selected subsystem. The axes are functions of the location and mass of atoms. They belong to the system and are independent of the Viewer's or Molecular Coordinate Systems. The primary inertial axis usually marks the longest distance through a molecular system, and the tertiary axis usually marks the shortest distance.

**Isosurface.** A three-dimensional surface which shows where in three-dimensional space the specified property has a given value; it is like the skin of a solid object. For example, for the mathematical function  $s = x^2 + y^2 + z^2$ , an isosurface plot for  $s = 1$  would give a spherical surface of radius 1; isosurface plots for other values of  $s$  would give spherical surfaces of other sizes.

**Iteration.** In an **SCF** quantum mechanics method, a calculation of molecular orbitals starts with a guess about the form of the orbitals. Each trial (or iteration) refines the guess for the next trial.

When the difference between two successive iterations is less than a convergence limit (that you can set), the SCF calculation has reached **convergence**.

**Label.** Information about the components of a molecular system HyperChem can display with the system. Labels apply either to atoms or residues. Atom labels are Symbol (chemical), Name (according to **PDB** file conventions), Number (sequence), Type (**atom type**), Basis Set, Mass, Charge, or Chirality (**R** or **S**). Residue labels are Name, Sequence, or Name+Sequence. HyperChem applies labels to the current selection.

**LINE.** A predefined, **named selection** identifying atoms in the selection, plus the primary inertial axis of selected atoms. (The primary inertial axis corresponds to the smallest eigenvalue of the moment of inertia matrix.) A molecular system can have only one LINE selection at a time.

**Local minimum.** A minimum on the energy surface for a molecular system representing a low potential energy conformation. Molecular energy surfaces can have many minima, the lowest being the **global minimum**. **Geometry optimization** calculations can result in a structure at a local minimum.

**Log.** A file containing the results of a chemical calculation and comments. HyperChem automatically collects this information if you choose Start Log on the File menu (see **Start Log**). The default filename is chem.log and the file usually has the extension *.log*.

**Lowest state.** The lowest electronic state of a particular spin multiplicity; not necessarily the **ground state**.

**LUMO.** Lowest Unoccupied Molecular Orbital. This is used as a reference point for describing the position of any molecular orbital as being at or above this orbital. Molecular orbitals are a linear combination of atomic orbitals (LCAO).

**Metafile.** An image composed of drawing commands. The resolution of a metafile image does not depend on the monitor, but is the resolution of the output device. For example, if you print a metafile on a laser printer, the resolution may be 300 dpi.

**MINDO/3.** Modified Intermediate Neglect of Differential Overlap, version 3. This is an SCF method for semi-empirical quantum mechanics calculations. An extension of INDO, MINDO/3 uses parameters fit to experimental results, instead of accurate calculations. Useful for large organic molecules, cations, and

polynitro compounds. Calculates electronic properties, geometry optimizations, and total energy. Reference: Dewar, M.J.S. *et al.*, *J. Am. Chem. Soc.* 97:1285, 1975.

**MM+.** The most general method for molecular mechanics calculations, developed for organic molecules. This is an **all atom** force field. MM+ is based on the MM2 (77) functional form, authored by Dr. Norman Allinger of the University of Georgia. HyperChem assigns atom types and parameters not normally available to MM2, extending the range of chemical compounds that this force field can accommodate. MM+ extends MM2 to allow options such as periodic boundary conditions and molecular dynamics common to other force fields. Reference: Allinger, N.L., *J. Am. Chem. Soc.* 99:8127-8134, 1977.

**MNDO.** Modified Neglect of Diatomic Overlap. This is an SCF method for semi-empirical quantum mechanics calculations. Useful for various organic molecules containing elements from long rows 1 and 2 of the periodic table, but not transition metals. Eliminates some errors in MINDO/3. Calculates electronic properties, optimized geometries, total energy, and heat of formation. Reference: Dewar, M.J.S. and Theil, W., *J. Am. Chem. Soc.* 99:4899, 1977.

**Model Builder.** The facility in HyperChem that constructs 3D models of chemical structures from 2D drawings. HyperChem uses built-in rules to assign standard bond lengths, bond angles, torsion angles, and stereochemistry. The Model Builder also assigns atom types for the currently active **force field**, and can automatically add hydrogens to satisfy valence requirements. These are approximate structures and might require refinement by **geometry optimization**.

**Molecular dynamics.** Calculations simulating the motion of each atom in a molecular system at a fixed energy, fixed temperature, or with controlled temperature changes. The result of molecular dynamics calculation is called a **trajectory**. HyperChem can use either a molecular mechanics or semi-empirical quantum mechanics method for a molecular dynamics trajectory. You can use this calculation to derive a large number of structural and thermodynamic properties, including alternative **local minima**, energy differences between different configurations, and reaction mechanisms and pathways.

**Molecular mechanics.** Various methods for calculating the potential energy of molecular systems as a function of the coordinates of their atomic nuclei, neglecting explicit treatment of elec-

trons. Electronic effects are implicit in the analytic functional forms and parameterization. All methods use empirical data to determine individual force constants (for example, bond stretch constants) and equilibrium (“strain-free”) values for each geometric quantity (for example, bond lengths). HyperChem provides these methods: **MM+**, **AMBER**, **BIO+**, and **OPLS**.

**Multiplicity.** The total spin multiplicity is  $2S+1$ , where  $S$  is the total electron spin of the system. Each unpaired electron counts for a spin of  $+1/2$ . Paired electrons have a net spin of zero ( $+1/2 - 1/2 = 0$ ). A closed-shell system has a multiplicity of 1 (singlet). An open-shell system can have a multiplicity of 2, 3, 4 (doublet, triplet, quartet, respectively), or higher. Use integers from 1 to 6 in HyperChem.

**Named selection.** A set of atoms that have an assigned name, either predefined (**POINT**, **LINE**, or **PLANE**) or any other name up to 19 characters long. You can use named selections to repeat a selection or to carry out specific operations (for example, inversion, reflection, rotation, averaging and graphing during molecular dynamics, or setting restraints during a calculation).

**NDO.** Neglect of Differential Overlap. For semi-empirical quantum mechanics calculations, this is a simplification eliminating many integrals describing the interactions of electrons. The CNDO method neglects the calculation of all differential overlap integrals. Other NDO methods (INDO, MNDO/3, MNDO, and AM1) neglect some integrals.

**Open shell.** A molecular system containing unpaired electrons.

**OPLS.** Optimized Potentials for Liquid Simulations. A molecular mechanics method, or **force field**, for chemical calculations. Developed for proteins and nucleic acids. It is similar to **AMBER** but treats nonbonded interactions more accurately. This force field provides only a **united atom** representation.

**Orbital.** The function describing the spatial distribution of an electron. Atomic orbitals describe the electrons in atoms. Molecular orbitals, derived as a linear combination of atomic orbitals (LCAO), describe electrons in molecules.

**Parameter set.** For molecular mechanics force fields, empirically or theoretically derived force constants and structural values applying to specific bonded combinations of atoms (each described by an **atom type**). Each force field might have more than one parameter set, but you can use only one set at a time. For

MM+, HyperChem supplies parameters when they are not available in the MM2 parameter set.

For semi-empirical calculations, parameter files (with the extension *.abp*) contain empirical values for the integrals in the calculation.

**PDB.** See **Brookhaven Protein Data Bank**.

**Periodic box/ boundary conditions.** A periodic box is a cube surrounding a molecular system. HyperChem adds standard water molecules (TIP3P, equilibrated at 300K and one atmosphere) to the box, providing a constant-density solvent. In molecular mechanics calculations, the box is treated as if it were surrounded by 26 identical boxes (images), each containing the identical molecules. Atoms in each image experience the same forces (exhibit the same motions) as the molecules in the central box. If, during a calculation, molecules move out of the box, an identical particle “enters” from the neighboring image. These periodic boundary conditions simulate a continuous system with a constant density of molecules. You can remove the water molecules, but retain the periodic box and boundary conditions, simulating a crystal lattice.

With the HyperChem periodic boundary conditions, an atom interacts only with one nearest image of other atoms and never with an image of itself in molecular mechanics calculations.

**PLANE.** A predefined, **named selection** identifying the atoms in the selection, plus the plane formed by the primary and secondary inertial axes of the selected atoms. A molecular system can have only one PLANE selection at a time.

**POINT.** A predefined, **named selection** identifying the atoms in the selection, plus the center of mass of the selected atoms. A molecular system can have only one POINT selection at a time.

**Potential (energy) surface.** A multidimensional plot of the potential energy of a molecular system as a function of all variables (degrees of freedom) in the system (for example, bond angles). The potential surface for a nonlinear molecule with  $N$  atoms (greater than 2) has  $3N-6$  independent degrees of freedom (6 = three degrees reserved for rotation and three reserved for translation of the whole molecular system). Potential energy surfaces are usually diagrammed in only one or two dimensions. A 1D potential surface might show a bond energy versus length. A 2D potential surface might show a contour plot of energy as a function of two torsion angles.



Important features on a potential surface are minima—which HyperChem can locate with geometry optimization and molecular dynamics calculations—and saddle points, which represent structural transition states.

**Quantum mechanics.** The theory that energy is absorbed or radiated in discrete packets, known as quanta. A consequence of this is that electrons travel in a limited number of orbitals around an atomic nucleus, and that each orbital is characterized by a specific energy. Moving electrons have the properties of particles and waves. The Schrödinger equation and its derivatives describe completely the behavior of electrons and nuclei, and hence the behavior of chemical compounds.

Semi-empirical calculations in HyperChem assume fixed nuclei and use approximate solutions of the Schrödinger equation, plus empirical data (parameters), to predict electronic properties of molecular systems.

HyperChem's *ab initio* calculations solve the Roothaan equations (see the *Computational Chemistry* reference manual) without any further approximation apart from the use of a specific finite basis set. They involve a more fundamental approach to solving the Schrödinger equation than the semi-empirical methods do.

**Rendering.** The visual representation of a molecular system. This version of HyperChem provides five types of rendering: Sticks, Disks, Spheres (CPK-type models), Dots, and Sticks & Dots. Space-filling renderings use standard CPK radii for the atoms.

**RHF.** Restricted Hartree-Fock theory. The representation of a wave function with spin-paired electrons (up-spin and down-spin) restricted to occupy the same spatial molecular orbitals. Each spatial orbital can also contain single (spin-up or spin-down) electrons.

**RMS gradient.** Total energy gradient calculated as a root mean square value. The gradient (G) is the rate of change (first derivative) of total energy (E) with respect to displacement of each atom in the x, y, and z directions. For atoms 1 to *n*,

$$G_{RMS} = \sqrt{\frac{1}{3n} \sum_{i=1}^n \left( \left( \frac{\partial E}{\partial x_i} \right)^2 + \left( \frac{\partial E}{\partial y_i} \right)^2 + \left( \frac{\partial E}{\partial z_i} \right)^2 \right)}$$

The units are kcal/mol Ångstrom. HyperChem reports this value for **geometry optimization** and **single point** calculations. An RMS gradient of zero means the structure is at a **local minimum**

or saddle point in the potential energy surface, not necessarily at the structure and state of lowest energy (**global minimum**).

**Restraint.** Restraints for energetic calculations add restoring forces to selected atoms. Restraints apply to atomic distances (one or two atoms), bond angles (three atoms), and torsion angles (four atoms).

**Scale to fit.** An item on the Display menu that scales the visible atoms or molecules to fit into the **workspace**. If there is a selection, the selected atoms fill the workspace. Without a selection, HyperChem scales the whole molecular system to fit the workspace.

**SCF.** Self-Consistent Field. This is an iterative method, used in all **NDO**, **semi-empirical** quantum mechanics calculations. It accounts for electron repulsion energies in solutions of the Schrödinger equation. The SCF or Hartree-Fock method simplifies this task by assuming the movements of electrons are independent and that an electron interacts with the mean field of other electrons in a molecule. In HyperChem, all semi-empirical quantum mechanics methods, except Extended Hückel, are SCF methods.

An SCF calculation begins by estimating the wave function describing the electron orbitals and electron repulsions. Based on this guess, HyperChem calculates a new wave function and compares the result with the estimate. HyperChem repeats this process until the estimate and the result converge or become “self-consistent.” Each iteration improves the orbitals until the result reaches a **convergence** limit.

**Script.** A set of instructions, stored in a file with a *.scr* extension, to run HyperChem. Similar to a macro program in other applications. HyperChem scripts set or read variables, give commands, and activate menu items. Use a text editor to create a script file, then use the Script menu in HyperChem to run a script.

**Self-consistent field method.** See **SCF**.

**Semi-empirical.** A type of quantum mechanics chemical calculation that uses parameters derived from experiments to simplify the calculation process.

**Single point.** A calculation that determines the total energy (in kcal/mol) and **gradient** of a molecular system or of selected atoms. With a quantum-mechanical method, a single point calculation also determines the electron (charge) distribution in the system.

The calculation represents only the present molecular configuration, a single point on the energy surface for the molecular system.

**Snapshot.** A record of a single time point in a **molecular dynamics** simulation. You can instruct HyperChem to take snapshots at regular intervals during the simulation. Coordinates and velocities for the molecular system are stored in SNP files. You can later play back the snapshots to view the changes in configuration of the system, plot data values, and obtain averages of data values. Data for chosen variables (like total energy or a bond torsion) are stored in a CSV file.

**Spin density.** The probability of finding more spin-up electrons than spin-down electrons at a point in space. For a closed-shell system, spin density is zero at every point. HyperChem can plot total spin density for a molecular system after it performs a single point, semi-empirical or *ab initio* quantum mechanics calculation. You can use plots of spin density at the nucleus to predict the results of electron spin resonance (ESR) spectra. See **spin pairing** and **multiplicity**.

**Spin multiplicity.** See **multiplicity**

**Spin pairing.** Two electrons occupying the same orbital must be paired and have opposite spins, one up spin and one down spin. Closed-shell systems contain only paired electrons. Open-shell systems contain unpaired electrons.

**Status line.** The line at the bottom of a HyperChem window displaying messages and data.

**3D surface.** See **isosurface**.

**Tool.** A cursor for performing tasks, like drawing or translation, in the HyperChem **workspace**. Tool icons are in a column at the left side of the workspace. L-clicking on a tool icon converts the cursor into a tool.

**Trajectory.** The history of atomic motion over time. This term is often applied to a **molecular dynamics** simulation. It can be saved in a **snapshot** file.

**Translation.** Movement of a molecule without rotation.

**Type rules.** Rules for each **force field**, found in the file chem.rul (or its binary equivalent, typerule.bin), for assigning atom types to individual atoms or atom groups.

**UHF.** Unrestricted Hartree-Fock method. A representation of a wave function providing separate spatial orbitals for spin-up ( $\alpha$ ) and spin-down ( $\beta$ ) electrons. Each  $\alpha$  or  $\beta$  orbital can accommodate zero or one electron. You can use UHF calculations for calculating dissociation reactions of closed-shell systems.

**United atom.** A simplification of a **force field** that treats certain atom groups (heavy atoms like C or N atoms and their bonded hydrogens) as a single atom. United atom treatment requires special atom types incorporating the properties of the constituent atoms and type rules for assigning these types. United atom representation is optional with AMBER and MM+, automatically assigned to many atoms with OPLS, and is the present default for BIO+. Another term for united atom is extended atom. The alternative to united atom is "all atom," which represents each atom separately, with a distinct **atom type**.

**Wave function.** See **orbital**.

**Window.** A working area on the desktop containing a software program (like HyperChem). A window has a title bar at the top, containing the name of the program. Only one window can be active at a time. The active window receives input from the mouse and keyboard. A dialog box is a type of window.

**Workspace.** The region of a Window that contains data; for HyperChem, a molecular system.

**Z Clipping.** See **clipping**.

**Zoom.** The HyperChem function that magnifies or reduces the size of the molecular system in the workspace.

**Zwitterion.** A molecule with regions of formal positive and negative charge. This is the usual state of polypeptides at neutral pH in solution, with  $-\text{NH}_3^+$  and  $-\text{COO}^-$  terminal groups. A polypeptide constructed from a HyperChem database has incomplete, uncharged termini ( $-\text{N}-\text{H}$  and  $-\text{C}-\text{O}$ ), unless terminal groups were used.

# Index

## Symbols

- \* and \*\* in basis set name CC-271
- \*ben parameter file RM-660
- \*imp parameter file RM-660
- \*nbd parameter file RM-660
- \*tor parameter files RM-660
- \*typ parameter files RM-660
- ? — Product support GS-20
- “Default Settings in Registry” on page 673 RM-28
- \*\*\* atom type CC-215

## Numerics

- 10–12 potential CC-26
- 1–4 electrostatic interactions CC-108
- 1–4 interactions CC-190, CC-204
- 1–4 van der Waals interactions CC-108
- 24-bit color RM-100
- 2D Contours RM-393
- 2-electron integral storage RM-103
- 3D Isosurface RM-393
- 3D isosurface RM-398
- 4-31G scale factors, table CC-270
- 6–12 function CC-26
- 6–12 potential CC-184

## A

- A form, polynucleotide strand conformation

- RM-160
- Ab Initio
  - dialog box RM-14
- Ab initio
  - basis set files RM-703–RM-709
- ab initio calculations CDK-195, CDK-229
- ab initio* calculations HM-213
- Ab initio calculations, HyperChem tutorial GS-219
- ab initio d orbitals RM-326
- ab initio ghost atoms RM-326
- ab initio* integral storage RM-103
- ab initio method CC-261–CC-278
- Ab Initio options RM-318
- ab initio quantum mechanics CC-31
- Abort warning message RM-359
- About HyperChem, Help menu RM-116
- absolute energy CC-22
- absorption, infrared CC-369
- Accelerate convergence HM-182
- Accelerate convergence, Semi-empirical Options dialog box RM-306
- acceleration, convergence CC-49, CC-116, CC-240
- acceptance ratio CC-102
- ACCR, Selection, Molecular Dynamics Averages dialog box RM-378, RM-379

accumulate selections, Selection tool RM-227  
 accuracy  
   atomic charges CC-143  
   dipole moments CC-140  
   heats of formation CC-136  
   proton affinities CC-137  
   rotational barriers CC-139  
   UV-visible spectra CC-152  
   vibrational frequencies CC-149  
 Accuracy of results HM-242  
 Acetyl (Ace), amino acid residues RM-143  
 Activation energies CC-339  
 activation energy CC-17, CC-139  
 Add  
   Counter Ions RM-162, RM-499  
     Databases menu RM-140  
   Hydrogens RM-123  
     Build menu RM-117  
   Restraint Forces dialog box RM-341  
 Add H & Model Build, changes to Model Build  
   RM-479  
 adding atoms/bonds RM-472  
 adding comments to Log file RM-617  
 adding hydrogen atoms RM-30, RM-477  
 adding new residues RM-739–RM-741  
 adding unusual residues RM-496, RM-501  
 affinities, proton CC-137  
 Alanine (Ala), amino acid residues RM-142  
 alanine zwitterion  
   creating isolated structure GS-148–  
     GS-157  
   minimized structure GS-163  
   optimizing solvated structure GS-  
     162  
   solvating GS-158  
 Algorithm, Compute menu RM-364  
 align CDK-215  
 Align Molecules  
   dialog box RM-216–RM-218  
   Edit menu RM-202  
 Align Molecules on Edit menu GS-180  
 Align Viewer RM-19  
   dialog box RM-215–RM-216  
   Edit menu RM-202  
 All Atom force fields CC-28  
 All Atoms RM-129  
   Build menu RM-118  
 all atoms CC-178, CC-197, CC-202  
 Allow Ions ??–RM-126  
   Element Table dialog box RM-122  
 Allow ions option GS-47  
 Alpha  
   polynucleotide torsion angle RM-160  
 alpha electrons CC-38  
 alpha helix  
   creating GS-99  
   displaying GS-118–GS-119  
 Alpha helix, residues conformation RM-144  
 alpha spins CC-46  
 Alpha, Orbitals dialog box RM-402  
 am\*hbd parameter file RM-661  
 AM1 CC-123, CC-132, CC-156, CC-249, CC-  
   305–CC-306  
   elements treated by RM-302  
   parameter files RM-690–RM-693  
   Semi-empirical Method dialog box  
     RM-299  
 AMBER CC-105, CC-110, CC-176, CC-197  
   file structure RM-744  
   force field GS-129, GS-130–GS-131  
   Molecular Mechanics Force Field  
     dialog box RM-290–RM-291  
   settings RM-749–RM-751  
 Amber CDK-102  
 amber,amber2, file structure RM-744, RM-  
   750–RM-751  
 amber,amber3, file structure RM-750–RM-751  
 amber,ambers, file structure RM-750–RM-751  
 amino acid CDK-223  
 Amino Acids

dialog box RM-14  
 amino acids  
   building peptides from GS-92  
   conformation changes RM-496  
   mutation RM-496  
   for polypeptide construction RM-141–RM-142  
 Amino Acids dialog box GS-91–GS-92, RM-140, RM-142–RM-145, RM-504  
   residues RM-141  
 amino acids, adding RM-739  
 Amino Acids, Databases menu RM-139–RM-141, RM-494  
   residues RM-142–RM-145, RM-735–RM-737  
 amount of Z Translation RM-45  
 analyzing HyperChem results RM-618  
 Anderson, W.P., Edwards, W.D., Zerner, M.C. and ZINDO/1 RM-680  
 angle bending CC-182, CC-193, CC-198, CC-203, CC-221  
 angle bending, sextic CC-193  
 angle, atomic coordinates for TPL file RM-734–RM-735  
 Angle, Rotate dialog box RM-210  
 anharmonicity CC-149  
 animate CDK-230  
 Animations, *See* AVI  
 annealing time, molecular dynamics CC-92  
 annealing, simulated CC-81, CC-360  
 annealing, trajectory CC-346  
 anti-parallel spins CC-288  
 application  
   DDE CDK-108  
 applications RM-604–RM-605  
   DDE RM-604–RM-605  
 applying restraints RM-343  
 Approximation, weak coupling HM-183, HM-250  
 Architecture CDK-8  
 architecture, HyperChem CC-161  
 Arginine (Arg), amino acid residues RM-143  
 argument CDK-30, CDK-31, CDK-51, CDK-98  
   command line RM-3  
 arguments, for variables and commands RM-516–RM-517  
 aromatic rings, torsion angles and multiple RM-131  
 array RM-581  
 array HSV CDK-33  
 Arrow keys HM-89  
 Asparagine (Asn), amino acid residues RM-143  
 Aspartic acid (Asp), amino acid residues RM-143  
 Assistance, product GS-20  
 asterisk in basis set name CC-271  
 atom coordinates CDK-165, CDK-187, CDK-192, CDK-199, CDK-220  
 Atom labels HM-96, HM-168  
 Atom number, Select menu RM-224  
 atom numbering CDK-30, CDK-59, CDK-100, CDK-195  
 Atom rendering HM-96  
 Atom rendering options HM-168  
 Atom rendering, colors HM-188, HM-192  
 Atom rendering, radii HM-189, HM-193  
 Atom type files HM-186, HM-194  
 atom types CC-176, CC-177  
   displaying GS-155  
 Atom types, calculate HM-155  
 atom types, new or changed CC-180  
 Atom types, setting HM-105  
 Atom typing rules HM-194  
 atom velocities CC-76  
 atom, HIN file syntax RM-717  
 Atom, Numbers, Select dialog box RM-236  
 ATOM, record in PDB-type files RM-505  
 atomic charges CC-143, CC-192  
   assigning GS-151–GS-152  
   calculating GS-202  
   displaying GS-182  
 Atomic charges, MM+ Options dialog box RM-

- atomic coordinates RM-734–RM-735
    - See also* files, syntax for HIN; Molecular Coordinate System
  - atomic mass CC-178
  - atomic movement, Monte Carlo RM-371
  - atomic orbital electron populations
    - in log file RM-632, RM-633
  - Atom-name# (TPL file)
    - atomic coordinate RM-735
    - bonding information RM-733
  - atoms RM-30, RM-223–RM-225
    - adding hydrogen RM-478
    - boundary GS-213
    - characteristics GS-84–GS-85
    - deleting/deselecting GS-54
    - drawing GS-46–GS-47
    - editing GS-82
    - group selection GS-53–GS-55
    - individuals edited GS-64–GS-65
    - labeling GS-63–GS-64
      - by number GS-81
      - by symbol GS-83
    - name, Select menu RM-223
    - names for TPL file RM-732
    - numbering RM-237
    - pseudo-atoms GS-213–GS-214
    - selected by name RM-236
    - selecting GS-51–GS-53
    - selecting by number RM-236–??
    - selecting in 3D rectangle GS-107
    - Selection level RM-32
    - spherical selection GS-108–GS-110
    - tag numbers in TPL file RM-732–RM-733
    - types RM-654
    - typing rules RM-655
  - Atoms, available HM-81
  - atoms, fixed CC-86
  - Atoms, Select menu RM-221
  - atoms, unknown type CC-215
  - Atom-type
    - in TPL file RM-734
    - Select menu RM-224
  - Aufbau principle CC-43, CC-51
  - Austin Model 1 *See* AM1
  - automatic operation with scripts RM-509
  - Average/graph period, Molecular Dynamics
    - Averages dialog box RM-380
  - Averages
    - averages files. *See also* Molecular Dynamics
    - Molecular Dynamics Averages dialog box RM-380
    - Molecular Dynamics Options dialog box RM-369–RM-374
  - averages, molecular dynamics results CC-88
  - averaging, dynamics data CC-351
  - averaging, dynamics, deviations CC-354
  - Avg. & Graph, Molecular Dynamics Averages
    - dialog box RM-379
  - AVI Movie RM-376
  - axial hydrogens GS-141
    - measuring GS-139
- ## B
- B form, polynucleotide strand conformation
    - RM-160
  - back end CDK-8, CDK-11, CDK-16, CDK-137, CDK-187, CDK-191, CDK-206, CDK-222, CC-161
    - remote CDK-18, CDK-222
  - backbone
    - coloring GS-104
    - displaying GS-104, GS-117
    - selecting GS-102–GS-103
  - background color RM-95
  - Backward, polynucleotide strand formation
    - RM-161–RM-162



Bacon, A.D. and Zerner, M.C.  
     and ZINDO/1 RM-680  
 Bar chart, coupling constants HM-138  
 Bar chart, peak information HM-129  
 Bar chart, plotting HM-155  
 Bar chart, shielding constants HM-138  
 Bar chart, spectral data HM-139  
 Bar chart, table data HM-170  
 barrier, rotational CC-139  
 barriers, potential CC-84  
 barriers, reaction CC-33  
 BAS file CDK-154  
 base pairs, unusual RM-500  
 basics, HyperChem RM-17–RM-25  
 Basis function, extra RM-316  
 basis functions, extra RM-328  
 basis functions, non-nuclear CC-120  
 basis set CDK-197, CDK-229, CC-45  
 Basis set assignment RM-316, RM-318  
 basis set superposition effects CC-120  
 basis set superposition error (BSSE) CC-278  
 basis set, creating RM-709  
 basis set, HIN file syntax RM-719  
 basis set, minimal CC-265  
 basis set, split-valence, table of KLM values  
     CC-268  
 basis sets  
     references CC-114  
     selection CC-113  
 basis sets, references CC-226  
 basis sets, split-valence CC-268  
 bath relaxation constant CC-74  
 Bath relaxation, Molecular Dynamics Options  
     dialog box RM-372  
 Beta  
     polynucleotide torsion angle RM-160  
     sheet, amino acid residue confor-  
         mation RM-144  
 beta electrons CC-38  
 beta spins CC-46  
 Beta, Orbitals dialog box RM-403  
 Bibliography HM-252  
 bibliography, computational chemistry CC-2  
 binary communication CDK-139, CDK-141  
 binary message CDK-141  
 binding energy CC-135  
     in log file RM-633  
 BIO+ CC-105, CC-176, CC-202  
     force field RM-291  
     Molecular Mechanics Force Field  
         dialog box RM-291  
 biphenyl RM-131, RM-654  
 Bipyramidal, Constrain Geometry dialog box  
     RM-135  
 biradicals CC-47  
 bitmap RM-613  
     graphics RM-631  
     use RM-615  
 Bitmap file format, Setup Image RM-100  
 bitmap preferences RM-99  
 black box — not CC-164  
 black-and-white output RM-105, RM-395  
 block CDK-146  
 block diagonal method CC-62  
 block diagonal Newton-Raphson optimization  
     CC-338  
 block diagonal Newton-Raphson optimization  
     method RM-366  
 BMP. *See* bitmap  
 boat cyclohexane  
     geometry optimization GS-140  
 Boltzmann distribution CC-19, CC-95, CC-  
     343, RM-384  
 bond CDK-57, CDK-217, CDK-219, CDK-  
     220, CDK-221  
 bond angle bending CC-183  
 bond angles CC-23  
     measuring GS-86–GS-87  
 bond angles and bending energy  
     in log file RM-638  
 bond breaking CC-47  
 bond breaking/formation, simulation CC-93  
 Bond Color RM-96  
 bond dipoles CC-107, CC-192

Bond dipoles, MM+ Options dialog box RM-292  
 bond dissociation CC-38  
 bond distances  
     measuring GS-85  
 bond formation and breaking CC-22  
 bond length and stretching energy  
     in log file RM-638  
 bond order  
     changing GS-60–GS-62  
 Bond rendering HM-99  
 Bond rendering options HM-168  
 bond rotation RM-39  
 bond stretch CC-182  
 bond stretching CC-183, CC-191, CC-198, CC-202, CC-218  
 bond torsions and torsional energy  
     in log file RM-638  
 bond-breaking CDK-56  
 bond-centered basis functions, see ghost atoms  
 bond-centered orbitals CC-278  
 bonded neighbors CC-177  
 bonding information. *See* HyperChem Input files; template files  
 bonding interactions CC-182  
 bonds CC-23  
     angle in parameter file RM-666–RM-668  
     changing the order RM-475–RM-476  
     hydrogen GS-89, GS-122–GS-123  
     length in parameter file RM-665–RM-666  
     measuring distances GS-85  
     *See also* Constrain entries  
     torsion constraint, rebuilding molecule GS-142  
 bonds, automatic generation RM-82  
 bond-type, bonding information for TPL file RM-733  
 books CDK-160, CDK-174, CDK-195  
     Tcl CDK-96  
     books on Tcl CDK-23  
     Boolean arguments CDK-30, CDK-51, RM-517  
     Borland CDK-3, CDK-149, CDK-150  
     Born-Oppenheimer approximation CC-11, CC-33, CC-168  
     boundary atoms GS-213  
     boundary conditions, periodic, theory CC-210  
     bovine pancreatic trypsin inhibitor GS-209–GS-210  
     box, HIN file syntax RM-716  
     box, removing periodic RM-339  
     BPTI *See* bovine pancreatic trypsin inhibitor  
     breaking bonds, simulation CC-93  
     Brillouin's Theorem CC-41, RM-310  
     Broadband decoupled spectrum HM-107  
     Brookhaven PDB *See* Brookhaven Protein Data Bank  
     Brookhaven Protein Data Bank GS-113, RM-80, RM-84, RM-503–RM-507  
         format GS-33  
         opening files from GS-114–GS-115  
         reading files from GS-210–GS-211  
         Save File dialog box RM-83  
     BSSE CC-120, CC-278  
     bug or error reporting GS-20  
     Build menu RM-117–RM-119  
         Calculate Types GS-152  
         Constrain Bond Torsion GS-201  
         Constrain Geometry GS-85  
         Default Element RM-29  
         Explicit Hydrogens RM-30  
         script messages RM-529–RM-532  
         Set Atom Type GS-85  
         Set Charges GS-85, GS-182  
     building  
         polypeptides GS-92  
     building 3D structure GS-83–GS-84, GS-153–GS-155  
     building chair cyclohexane GS-132–GS-133  
     building molecules

- menu items RM-117–RM-192
  - from residues RM-492–RM-501
- butadiene RM-654
- butadiene, modeling CC-220
- button CDK-98, CDK-104, CDK-197
- button code CDK-106
- Button, control-menu HM-143
- Button, maximize HM-143
- Button, minimize HM-143
- Button, restore HM-144
- By name, Select dialog box RM-233
- bypassing a dialog box CDK-51
- C**
- <sup>13</sup>C HM-81, HM-82, HM-107, HM-111, HM-119, HM-178, HM-183, HM-185
- C CDK-95, CDK-137, CDK-138, CDK-160
- C++ CDK-95, CDK-138, CDK-149, CDK-159
- C60 CDK-58, CDK-114
- Calculate atom types HM-155
- Calculate Types RM-129
- Build menu RM-118
- Calculate Types on Build menu GS-152
- calculation methods
  - molecular mechanics RM-289
- Calculation of NMR spectra HM-118
- Calculation options HM-105
- Calculation options, SCF HM-112
- Calculation options, setting up HM-136
- Calculation options, shielding and coupling HM-157
- calculation types CC-16
- Calculation, chemical shifts (theory) HM-242
- Calculation, starting HM-114, HM-137, HM-158
- calculations
  - mixed GS-211
- calculations, ab initio RM-318
- Calculations, detailed results HM-210
- calculations, HyperChem tutorial GS-127–GS-128
- calculations, mixed mode CC-256, RM-334
- Calculations, mixed-mode HM-82
- callback CDK-131, CDK-156, CDK-157, CDK-165
- CambridgeSoft ChemDraw Files
  - open RM-80
- Cancel RM-425–RM-469
  - script messages RM-573
- cancel CDK-207, CDK-212
- Cancel button CDK-46
- Cancel menu CDK-46
- cancel-notify CDK-30
- canonical ensemble CC-74
- Capping atom, quantum/classical boundary HM-82, HM-132
- capping atoms CC-256, CC-258–CC-260
- caps for residues RM-158, RM-159
- caption CDK-212
- carboxyl group solvation GS-203–GS-204
- Cartesian Gaussians CC-272
- case
  - preferences RM-15
  - sensitivity RM-511, RM-656
- CDK CDK-1
- Components CDK-2
- Cells, replace table HM-154, HM-167
- center of mass
  - in log file RM-638
- centering molecules GS-74
- center-of-mass CDK-55
- chair to boat cyclohexane transformation GS-137–GS-138
- change-user-menuitem CDK-37
- changing chirality RM-485
- changing molecular conformation RM-481–RM-486
- changing/choosing drawing elements RM-29
- Charge
  - on status line RM-224
  - in TPL file RM-734
- charge CDK-220, CDK-221, CDK-222
- Charge density RM-393
- charge density CC-140
- Charge on molecule HM-181

charge, formal CC-177  
 charge-charge interaction CC-201  
 charges, atomic CC-143, CC-192  
 charges, keep atom RM-545  
 charges, point CC-107  
 CHARMM CC-105, CC-110, CC-202  
   file structure RM-745  
   force field RM-662  
   settings RM-755–RM-758  
 charmm, charmm85, file structure RM-745  
 charmnp parameter file RM-661  
 Chart table data HM-170  
 Chart, coupling constants HM-138  
 Chart, peak information HM-129  
 Chart, plotting bar HM-155  
 Chart, shielding constants HM-138  
 Chart, spectral data HM-139  
 check a box, using keyboard alternatives RM-10  
 chem.exe, program file RM-621  
 chem.ini/Registry, default settings in RM-743–RM-785  
 chem.rul CC-178  
 chem.rul, atom typing file RM-653  
 CHEM.SCR CDK-53  
 chem.tpl, HyperChem TPL file RM-727  
 chemct.tpl, HyperChem TPL file RM-727  
 ChemDraw files  
   open RM-80  
 ChemDraw files, reading RM-81  
 chemical calculations RM-287–RM-469  
 chemical environment CC-177  
 chemical equilibrium CC-137  
 chemical reactions CDK-56, CC-33, CC-360–CC-363  
 chemical reactivity CC-144  
 Chemical shift view HM-145  
 Chemical shifts, calculation (theory) HM-242  
 chemnt.tpl, HyperChem TPL file RM-727  
 ChemPlus CDK-16, CDK-54, CDK-117  
 Chi, polynucleotide torsion angle RM-160  
 chirality CDK-192, RM-29, RM-207, RM-208, RM-482  
   changes RM-485  
   evaluated with Model Builder RM-487  
 chirality, changing GS-235  
 choice RM-30  
 choosing a dialog box option, using keyboard alternatives RM-10  
 choosing a menu item, using keyboard alternatives RM-9  
 choosing conformations, amino acid residues RM-495  
 CI CC-243, CC-245  
 CI energy  
   in log file RM-633  
 CI Method, Configuration Interaction dialog box RM-309  
 CI singles calculations RM-300, RM-412, RM-415  
 Cis, Constrain Bond Torsion dialog box RM-139  
 clamped-nuclei approximation CC-168  
 classical mechanics CC-172  
 classical perturbation CC-277  
 classical trajectory analysis CC-360  
 Classical/quantum calculations HM-82  
 classical-quantum boundary GS-213–GS-214  
 Classification of Hcl Commands CDK-205  
 Clear  
   removes atoms from workspace RM-201, RM-203  
 Clear on Edit menu GS-55  
 Clear workspace HM-153  
 click definition RM-31  
 client CDK-108, CDK-213  
 client-server CDK-8, CDK-16  
 clip increment RM-97, RM-98  
 Clipboard RM-101  
   *See also* Cut  
   choosing image destination RM-613  
   destination, Setup Image RM-101  
   graphics stored RM-101, RM-203–RM-

205, RM-613–RM-614  
 use RM-613–RM-614  
 clipboard CDK-218  
 copying to GS-56  
 Clipboard, copying results to HM-103, HM-137  
 clipping CDK-208  
 clipping slab RM-47  
 changing GS-76–GS-79  
 default GS-76  
 close a dialog box, using keyboard alternatives RM-10  
 close a menu, using keyboard alternatives RM-9  
 closed-shell singlet ground states CC-39  
 Closing HyperNMR HM-104, HM-154  
 CNDO CC-123, CC-132, CC-154, CC-249, CC-252, CC-284–CC-288  
 elements treated by RM-301  
 parameter files RM-679, RM-684–RM-685  
 CNDO computation method GS-182  
 CNDO mixed model CC-287  
 coefficients, MO, initial guess RM-325  
 cold link CDK-122  
 collision  
 reactive CDK-55  
 color CDK-217  
 Color preferences HM-165  
 color, bond RM-96  
 color, selection RM-96  
 color, window RM-95  
 colors  
 default RM-293  
 saving images RM-614  
 Colors, atom rendering HM-188, HM-192  
 colors, isosurface settings RM-102  
 Colors, window and object HM-154  
 command line argument RM-3  
 command substitution in Tcl CDK-97  
 Commands  
 Help menu RM-115  
 script message syntax RM-516  
 comma-separated values file CC-355  
 Comments  
 Log Comments dialog box RM-92  
 Open File dialog box RM-80  
 Run Script dialog box RM-106  
 Save File dialog box RM-83, RM-85  
 communicating CDK-137  
 communication CDK-108, CDK-222  
 communication channel CDK-122  
 Communication files HM-186, HM-199  
 Comparison with experimental results HM-242  
 Compile Type Rules RM-130  
 Build menu RM-118  
 compiled scripts CDK-54  
 Complement Selection RM-232  
 Select menu RM-221  
 Selection tool RM-34  
 Complement Selection on Select menu GS-106, GS-115, GS-166  
 complementing a subset GS-106  
 Complete Neglect of Differential Overlap (CNDO) CC-249  
 Complete Neglect of Differential Overlap. *See* CNDO  
 computational modules CC-161  
 Compute Menu HM-158  
 Compute menu RM-357–RM-416  
 Geometry Optimization GS-135, GS-140, GS-143, GS-156, GS-174  
 Molecular Dynamics GS-168  
 script messages RM-556–RM-568  
 Single Point GS-134  
 Compute shielding and coupling HM-114  
 Computed  
 Restraint Forces dialog box RM-341, RM-342  
 Computed values  
 Constrain Bond Angle dialog box

RM-138  
 Constrain Bond Length dialog box  
 RM-137  
 Constrain Bond Torsion dialog box  
 RM-139  
 Constrain Geometry dialog box RM-  
 134  
**CONNECT**  
 connectivity records RM-85  
 Creating PDB-type files RM-506  
 configuration interaction CDK-230, RM-327  
 Configuration Interaction (CI) CC-39, CC-243,  
 CC-245  
 configuration interaction (CI)  
 practical applications RM-310  
 Configuration Interaction dialog box RM-308–  
 RM-310  
 Configuration interaction, Semi-empirical Op-  
 tions dialog box RM-307  
 Configuration, WIN.INI HM-78  
 configurations  
 number of in log file RM-633  
 configure CDK-104  
 conformation, Beta sheet GS-92  
 conformational searching CC-80, CC-90  
 conformations for residues RM-495, RM-499  
 conjugate gradient CC-61  
 conjugate gradient optimization CC-126, CC-  
 336  
 connected to expressions, type rules syntax  
 RM-657  
**Connectivity**  
 CONNECT records RM-85  
 Creating PDB-type files RM-506  
 conservation, energy CC-73, CC-128  
 console applications CDK-10, CDK-155  
 console programs CDK-160  
 Console window CDK-191  
 constant energy CC-89  
 constant temperature CC-74, CC-90  
 Constant temperature, Molecular Dynamics  
 Options dialog box RM-372  
 constant temperature/energy CC-80  
 Constrain Bond Angle RM-137  
 Build menu RM-119  
 dialog box RM-138  
 Constrain Bond Length RM-136  
 Build menu RM-119  
 dialog box RM-137  
 Constrain Bond Torsion RM-138–RM-139  
 Build menu RM-119  
 dialog box RM-139  
 Constrain Bond Torsion on Build menu GS-201  
 Constrain Geometry RM-133–RM-136  
 Build menu RM-119  
 dialog box RM-134–RM-136  
 Constrain Geometry on Build menu GS-85  
 constraint CDK-220  
 constraints CDK-219  
 Model Builder RM-482  
 constructing molecules RM-471–RM-501  
 Continue warning message RM-359  
 contour CDK-232  
 Contour Grid RM-395  
 Contour Grid property sheet RM-395–RM-396  
 Contour Level, isosurface RM-397  
 Contour levels RM-395  
 contour plots CC-253  
 contours GS-185  
 contours, 2D RM-393  
 contracted Gaussian functions CC-263, CC-264  
 Control menu button GS-29  
 control structures CDK-55, CDK-64  
 Control-menu box, HyperChem windows RM-  
 5  
 Control-menu button HM-87, HM-143  
 controls  
 VB CDK-120  
 Conventional semi-empirical methods HM-226  
 Converged, status line, geometry optimization  
 RM-363  
 convergence CDK-227, CDK-229, CC-239  
 Convergence acceleration HM-182

convergence acceleration CC-49, CC-116  
 convergence criteria CC-63  
 convergence limit CC-116  
 Convergence limit, SCF HM-181  
 Convergence limit, Semi-empirical Options dialog box RM-306  
 convergence, SCF CC-49  
 converting 2D structures to 3D RM-487  
 Cool steps, Monte Carlo RM-371  
 Cool time, Molecular Dynamics Options dialog box RM-371  
 cooling time, molecular dynamics CC-92  
 cooling, dynamics CC-75  
 cooling, trajectory CC-346  
 coordinate data RM-623  
     HIN file RM-717  
     SNP (snapshot) files RM-627–RM-629  
     TPL file RM-734  
 Coordinate systems HM-152  
 coordinate systems RM-18–RM-19  
 coordinated ring, orientation restraint RM-343  
 coordinates  
     in log file RM-632, RM-633  
 coordinates of an atom CDK-104  
 coordination CDK-221  
 Copy Image RM-206, RM-614  
     for the Clipboard RM-206  
     CPK (spheres) rendering settings in chem.ini/Registry RM-781  
     Edit menu RM-201  
     for a file RM-206  
 Copy structure or picture HM-154  
 Copy structures  
     onto Clipboard RM-203, RM-480  
     Edit menu RM-201  
     onto Clipboard in ISIS format RM-204  
 Copying an image HM-103  
 core Hamiltonian matrix  
     in log file RM-632  
 core Hamiltonian MO coefficients CC-276  
 core orbitals CC-45  
 core-core interaction  
     in log file RM-633  
 correlation energy CC-39  
 coulomb field, external CC-256  
 Coupling constants, calculating HM-113  
 Coupling constants, option HM-106  
 Coupling constants, plotting HM-138  
 Coupling constants, viewing HM-117, HM-138, HM-155  
 Coupling, nuclear spin HM-222  
 covalent radii CC-219  
 Crambin, sample PDB file RM-506  
 create-atom CDK-59, CDK-219  
 creating  
     2D sketch GS-81  
     alpha helix GS-99  
     isolated alanine zwitterion GS-148–GS-158  
     polypeptides GS-91–GS-98  
     small molecules, 2D/3D GS-57–GS-67  
     water molecule GS-179–GS-180  
     zwitterion GS-93–GS-94  
 creating PDB-type files RM-504–RM-506  
 creating unusual base pairs RM-500  
 creating zwitterions RM-495  
 cross-eyed stereo viewing  
     setting in chem.ini/Registry RM-782  
 cross-linking with Cyx RM-495  
 cross-references to scripts messages RM-584  
 CSV file CC-355  
 cubic stretch CC-176, CC-191  
 cubic stretch constant  
     in log file RM-638  
 current-file RM-519  
 cursor CDK-218  
 Cursor, Setup Image RM-102  
 cursors CDK-208  
 custom menus CDK-6, CDK-9, CDK-21,

CDK-24, CDK-37, CDK-38  
customizing HyperChem CDK-1, CDK-4,  
CDK-8, CDK-207  
custom-title CDK-47  
cut  
    cutting and pasting atoms RM-479  
    moves atoms from workspace to  
    Clipboard RM-203, RM-479–  
    RM-481  
Cut, Edit menu RM-201  
cutoff CDK-223, CDK-229  
cutoff, two-electron integral CC-275  
Cutoffs  
    Force Field Options dialog box RM-  
    295  
    MM+ Options dialog box RM-292  
cutoffs with periodic boundary conditions RM-  
337  
cutoffs, nonbonded CC-66, CC-108, CC-189  
cutoffs, nonbonded potential CC-29  
cutting and pasting atoms RM-479–RM-481  
Cycles, geometry optimization  
    status line report RM-363  
    Termination condition RM-366  
cycles, optimization CC-63  
Cycles, transition state search  
    Termination condition RM-388  
cyclohexane  
    building chair form GS-132  
    building twist boat form GS-141  
    chair to boat transformation GS-137–  
    GS-138  
    geometry optimization GS-129  
    structural properties of chair form  
    GS-133  
Cysteine (Cys), amino acid residue RM-143  
Cystine (Cyx)  
    amino acid residue RM-143  
    cross-linking RM-495  
Cytochrome C (3CYT), sample PDB files RM-

506

## D

D EKIN, Selection, Molecular Dynamics Aver-  
ages dialog box RM-378  
D EPOT, Selection, Molecular Dynamics Aver-  
ages dialog box RM-379  
D ETOT, Selection, Molecular Dynamics Av-  
erages dialog box RM-379  
D *Named* selections, RMS deviation, Molecular  
Dynamics Averages dialog box RM-379  
d orbital sets RM-326  
d orbitals, ab initio CC-119  
d Orbitals, Hückel method CC-122  
d orbitals, Semi-empirical Options (Hückel) di-  
alog box RM-305  
D TEMP, Selection, Molecular Dynamics Av-  
erages dialog box RM-379  
dA, nucleotide residue RM-159  
dashed lines RM-105, RM-395  
Data collection period, Molecular Dynamics  
Options dialog box RM-373  
Data file, starting HM-113  
Data file, stopping HM-122  
Data input HM-75, HM-81, HM-89  
data variable messages RM-573–RM-577  
Databases menu RM-139–RM-141, RM-493–  
RM-496  
    Amino Acids RM-493–RM-496  
    Nucleic Acids RM-497–RM-499  
    script messages RM-542–RM-545  
DBF files, parameters CC-206  
dbf force field parameter files RM-653  
dC, nucleotide residue RM-159  
DDE CDK-7, CDK-9, CDK-12, CDK-34,  
CDK-107, CDK-119, CDK-127,  
CDK-137, CDK-154, RM-509–RM-  
609  
    Network CDK-17  
DDE communication CDK-156  
DDE conversations CDK-108, CDK-115  
DDE server CDK-108  
DDE. *See also* Dynamic Data Exchange



DDE\_ADVICE CDK-109  
 DDE\_EXECUTE CDK-109  
 DDE\_REQUEST CDK-109  
 DDE\_ADVICE, data update request RM-605  
 DDE\_EXECUTE CDK-111, CDK-117  
 DDE\_EXECUTE, executes HyperChem RM-604  
 DDE\_INITIATE CDK-109  
 DDE\_INITIATE, starts HyperChem RM-604  
 DDE\_REQUEST CDK-117  
 DDE\_REQUEST, information from HyperChem RM-605  
 DDE\_TERMINATE, ends conversation with HyperChem RM-605  
 DDE\_UNADVISE, data value to HyperChem RM-605  
 Deamino-oxytocin, sample PDB file RM-507  
 declarations CDK-214  
 Decoupled NMR HM-106  
 Default  
   colors RM-293  
   Restraint Forces dialog box RM-340–RM-342  
   settings, in chem.ini/Registry RM-743–RM-785  
   Starting value, Grid Control dialog box RM-395  
 Default Element RM-120–RM-122  
   on Build menu RM-30, RM-117  
   changes RM-474  
   dialog box RM-14, RM-122  
     Allow ions, Build menu RM-121  
     Element Table dialog box GS-179, RM-30  
     Explicit Hydrogens RM-30  
 Default force field RM-131  
 default force field CC-216  
 default menus CDK-44  
 default MM+ force field CC-214  
 default orbital exponents, table CC-280  
 default scheme CC-190  
 Default settings, HyperNMR HM-186, HM-187  
 Degeneracy, Electronic Spectrum dialog box RM-414  
 degenerate state CC-40  
 deleting  
   atoms GS-55  
   bonds GS-55  
 Delta, polynucleotide torsion angle RM-160  
 density matrices  
   in log file RM-633, RM-634  
 density, charge CC-140  
 density, electron CC-54  
 density, spin CC-55  
 deselect a group of atoms, Selection tool RM-33  
 deselect backbone atoms, Selection tool RM-33  
 deselecting  
   atoms GS-54  
 Deselecting atoms HM-102  
 Destination, Setup Image dialog box RM-101  
 Detailed results of calculations HM-210  
 Deuterium HM-82, HM-111, HM-178  
 development CDK-159, CDK-173  
 dG, nucleotide residue RM-159  
 Dialog Box  
   Open Log RM-90  
 dialog box CDK-51, CDK-196  
 dialog boxes  
   HyperChem RM-13–RM-14  
   persistent RM-14  
   removal RM-13  
 dielectric constant CC-27, CC-87, CC-188  
   in log file RM-638  
 dielectric constant, distance dependent CC-188  
 dielectric functions CC-107  
 dielectric options GS-161–GS-162  
 dielectric permittivity CC-188  
 diff, status line RM-363, RM-369  
 diffusion constant, solvent CC-95  
 diffusion limited aggregation CDK-204  
 dihedral angle CC-195, CC-198, CC-203

dihedral angle rotation CC-182  
 dihedral angles CC-183, CC-221  
 dihedral potential CC-25  
 dihedrals, improper CC-184  
 DIIS convergence acceleration HM-182  
 DIIS convergence accelerator CC-116, CC-240  
 dipole CDK-231, RM-692  
 dipole interaction energy  
     in log file RM-638  
 dipole moment CDK-102, CDK-211, CC-54,  
     CC-140, CC-250, CC-287  
     in log file RM-638  
 Dipole moment, importing RM-92  
 Dipole moment, storing RM-93  
 dipole, oscillating electric CC-369  
 dipole, transition CC-369  
 dipole-dipole interactions CC-187  
 dipole-moment CDK-28, CDK-49, CDK-216,  
     CDK-221  
 dipoles  
     in log file RM-633  
 dipoles, bond CC-107, CC-192  
 direct command CDK-49, CDK-66, CDK-140  
 direct commands CDK-205  
 Direct Inversion in the Iterative Subspace (DI-  
     IS) CC-49, CC-240  
 Direct Inversion of Iterative Subspace (DIIS)  
     convergence acceleration HM-182  
 direct SCF calculation CC-119, CC-275  
 directories, changing GS-35  
 Directory  
     Open File dialog box RM-79–RM-82  
     Save File dialog box RM-83–RM-85  
 Directory, change HM-91  
 Disks rendering HM-97  
 dispersion forces CC-182  
 Display color preferences HM-165  
 Display menu GS-37  
     Overlay GS-167  
     Recompute H Bonds GS-206  
     Scale to Fit GS-74  
     script messages RM-536–RM-541  
     Show All GS-105  
     Show Hydrogens GS-66  
     Show Selection Only GS-117  
 Display options HM-167  
 Display preferences HM-154  
 display settings  
     using GS-37–GS-43  
 Display, rendering options HM-155  
 displaying  
     alpha helix GS-118–GS-119  
     atom types GS-155  
     atomic charges GS-182  
     backbone GS-104  
     disulfide bridges GS-119–GS-121  
     hydrogen bonds GS-122–GS-123  
     PDBs GS-210–GS-211  
     polypeptides GS-92  
     primary structure GS-116  
     ring structures GS-121  
     side chains GS-105, GS-117  
     solvated system GS-160–GS-162  
     subsets GS-99–GS-112  
 displays, setting conditions RM-95–RM-97,  
     RM-611  
 dissociation, bond CC-38, CC-47  
 Distance dependent dielectric, Force Field Op-  
     tions dialog box RM-293  
 distance, atomic coordinate in TPL file RM-735  
 distribution function CC-100  
 disulfide bridges  
     displaying GS-119–GS-121  
 DLL CDK-119, CDK-129, CDK-139, CDK-  
     149, CDK-151, CDK-176  
 DNA B form, sample PDB file RM-507  
 DNA/RNA construction RM-497–RM-501  
 docking molecules CC-86, RM-82  
 Done, Molecular Dynamics Results dialog box  
     RM-380–RM-382  
 d-orbitals on 2nd row, option HM-172  
 dot surface CDK-216

double buffering, in chem.ini/Registry RM-781  
Double Strand, polynucleotide strand conformation RM-161  
double zeta CC-270  
double-click definition RM-9  
doublet CC-116, CC-229, CC-242  
doublet open-shell ground states CC-39  
doubly occupied orbitals  
    in log file RM-632, RM-633  
drag definition RM-8  
drawing  
    2D sketch GS-58  
    atoms GS-46–GS-47  
    basic techniques GS-45–GS-56  
    bonds GS-48–GS-51  
drawing mode GS-47, GS-81  
drawing molecules RM-29, RM-471–RM-481  
Drawing tool GS-58, RM-29–RM-30, RM-471  
    changing RM-473–RM-475  
    icon, mouse clicks RM-29  
    molecules creation RM-18  
Dreiding force field CC-216  
Drive names  
    Open File dialog box RM-79–RM-81  
    Run Script dialog box RM-106–RM-107  
    Save File dialog box RM-83–RM-86  
dT, nucleotide residue RM-159  
d-type functions, number CC-272  
dummy atoms RM-343  
Dynamic Data Exchange CDK-107, CDK-127, RM-509–RM-609  
    integrating HyperChem RM-509  
Dynamic Link Library CDK-8, CDK-119, CDK-149, CDK-176  
dynamics data, averages CC-349  
dynamics results file CC-355  
dynamics sampling interval CC-82  
dynamics, molecular CC-71  
dynamics, periods CC-75  
dynamics, quenched CC-80

## E

Edit Menu HM-154  
Edit menu RM-201–RM-220  
    Align molecules GS-180  
    Clear GS-55  
    Copy GS-55  
    Reflect GS-138  
    script messages RM-526–RM-528  
editing  
    2D sketch GS-82  
    alanine zwitterion structure GS-149–GS-150  
    atoms GS-64–GS-65, GS-82  
    basic techniques GS-45–GS-56  
editing text RM-85, RM-91–RM-92  
EHT CC-153, CC-278–CC-284  
eigenvalues  
    in log file RM-632, RM-633, RM-634  
eigenvector CDK-228  
eigenvector follow GS-233  
eigenvector following CC-68, CC-127, CC-340  
Eigenvector following, transition state search RM-387, RM-389, RM-390, RM-391  
Eigenvector-Follow method RM-365, RM-366  
eigenvectors (wavefunctions)  
    in log file RM-632, RM-633  
eigenvectors of CI secular equation  
    in log file RM-634  
EKIN, Selection, Molecular Dynamics Averages dialog box RM-378  
elastase-TFAP, sample PDB file RM-506  
electron density CC-54, CC-251  
electron lone pairs CC-200  
electron spin resonance (ESR) CC-253  
electron-electron repulsion CC-262  
electronic density CC-125  
electronic energy  
    in log file RM-633  
electronic properties of proteins GS-209–GS-217

electronic spectra CDK-231, CC-40  
 electronic spectroscopy CC-245  
 Electronic Spectrum  
     **Compute Menu** RM-411–RM-416  
     **Compute menu** RM-358  
 electronic spectrum CC-129  
 Electronic Spectrum dialog box RM-412–RM-414  
 Electronic state HM-181  
 electronic states CC-242  
 electrons  
     **total number in log file** RM-632, RM-633  
 electrons, alpha and beta CC-38  
 electrostatic interactions CC-107, CC-182, CC-187, CC-199, CC-201, CC-205, CC-222  
 Electrostatic potential RM-392  
 electrostatic potential CC-27, CC-55, CC-125, CC-141, CC-254–CC-255  
 electrostatic potential, plotting GS-184  
 Electrostatic scale factor, Force Field Options dialog box RM-294  
 electrostatics CDK-223  
 Electrostatics, MM+ Options dialog box RM-292  
 Element name on status line RM-224  
 Element Properties RM-123, RM-475  
 Element Table dialog box RM-30  
     **changes** RM-474  
     **Explicit Hydrogens** RM-478  
 Element, displaying labels HM-96  
 Elements, available HM-81  
 elements, available RM-300  
 E-mail support GS-21  
 embeddable CDK-95  
 Empty workspace HM-153  
 enable menu CDK-46  
 Enantiomers. *See* Chirality  
 End time, Molecular Dynamics Playback dialog box RM-376  
 END, record in PDB-type files RM-506  
 Ending HyperNMR HM-104, HM-154  
 endmol, HIN file syntax RM-720  
 endres, HIN file syntax RM-719  
 endselection, file syntax RM-722, RM-723, RM-724, RM-725  
 Energy  
     **Geometry Optimization** RM-363  
     **Single Point** RM-360  
 energy CDK-223, CDK-226, CDK-228, CDK-229, CDK-230, CDK-231  
     **CI** RM-633  
     **excitation**  
         **in log file** RM-633  
 energy barriers CC-84  
 energy components  
     **in log file** RM-638  
 energy conservation CC-73, CC-128  
 Energy Criterion  
     **Configuration Interaction dialog box** RM-309  
     **practical application** RM-310  
 energy minimization GS-129–GS-145  
 energy, activation CC-17, CC-139  
 energy, binding CC-135  
 energy, constant CC-80, CC-89  
 Energy, Orbitals dialog box RM-403  
 Enlarge structure HM-101, HM-157, HM-161  
 ensemble averaging CC-95  
 ensemble averaging, dynamics CC-343  
 ensemble, isokinetic vs. isothermal CC-74  
 ENT file CDK-50  
 ENT. *See* Brookhaven Protein Data Bank  
 enthalpies CC-22  
 enthalpy of formation CC-135  
 entry CDK-100, CDK-197  
 entry widget CDK-100  
 enum CDK-28, CDK-30, CDK-52  
 enum, script argument RM-517  
 Envelope HM-149  
 Envelope line widths HM-125  
 Envelope view, spectra HM-123  
 Envelope width, fine control HM-127, HM-191

Envelope widths HM-156  
Envelope, spectrum HM-156  
environment CDK-149, CDK-161, CDK-188  
environment, chemical CC-177  
EPOT, Selection, Molecular Dynamics Averages dialog box RM-378  
Epsilon  
    Dielectric RM-294  
Epsilon, polynucleotide torsion angle RM-161  
Equations, zillions HM-213–HM-252  
equilibration time, molecular dynamics CC-91  
equilibration, molecular dynamics CC-76  
equilibration, trajectory CC-346  
equilibrium bond length  
    in log file RM-638  
equilibrium properties CC-19, RM-384  
equilibrium, chemical CC-137  
Equipment requirements HM-77  
Equivalent atoms HM-83, HM-110  
Equivalent atoms, selecting HM-134, HM-173, HM-176  
equivalent-atoms-# selection in HyperChem HM-111, HM-135, HM-178  
Erase workspace HM-153  
Ergodic Hypothesis CC-343  
error and bug reporting GS-20  
errors CDK-131, CDK-132, CDK-137, CDK-148, CDK-205, CDK-213  
ESR CC-253  
ETOT, Selection, Molecular Dynamics Averages dialog box RM-378  
event-driven applications CDK-155  
Excel macros CDK-116  
Excel. *See* Microsoft  
exchange integrals, one-center CC-290  
exchange phenomena CC-288  
exchange repulsion CC-38, CC-182  
excitation energy  
    in log file RM-633  
excited singlet states CC-39  
excited state CC-242  
Exclusion Principle, Pauli CC-35  
executable file path RM-103  
EXECUTE, Excel message RM-606  
Exit CDK-198, RM-105  
    on File menu RM-78  
exiting HyperChem GS-43  
Exiting HyperNMR HM-104, HM-154  
Expand peaks HM-126  
Experiment, comparison with HM-242  
Expert options HM-107  
Explicit Hydrogens RM-30, RM-119–RM-120  
    Build menu RM-117  
    Element Table dialog box RM-122  
explicit hydrogens CDK-58  
Explicit Hydrogens option GS-47  
exploring 3D structure GS-153–GS-155  
exponents, entering RM-15  
Export orbital data RM-78, RM-93  
Exporting pictures HM-103  
expr CDK-97  
Extend Ring RM-242–RM-243  
    Select menu RM-222  
Extend Side Chain RM-243–RM-244  
    Select menu RM-222  
Extend to sp<sup>3</sup> CC-256, RM-244–RM-245  
    Select menu RM-222  
Extended Hückel RM-298  
    elements treated by RM-301  
    parameter file RM-681  
Extended Hückel calculations, log file information RM-632  
Extended Hückel method CC-35, CC-130  
Extended Hückel method, options CC-121  
Extended Hückel Theory (EHT) CC-153, CC-229, CC-278–CC-284  
extended hydrogen (EH) RM-128  
    in atom typing RM-658  
external coulomb field CC-256  
external Tcl/Tk CDK-9, CDK-132, CDK-139  
extra basis functions RM-328

**F**  
<sup>19</sup>F HM-81, HM-82, HM-111, HM-119, HM-178, HM-183, HM-185

- factory CDK-213
- factory settings CDK-207
- Fast translation RM-98
- fast translation RM-42
- Fermi spin-spin coupling constants, option HM-106
- Ferredoxin, sample PDB file RM-507
- file CDK-210, CDK-218
  - choosing image destination RM-613–RM-614
  - destination, Setup Image RM-101
  - format
    - images RM-613
    - Setup Image RM-100, RM-101
- file extension CDK-52
- file formats RM-79, RM-84
- File Menu HM-153
- File menu GS-33, RM-77–RM-105
  - New GS-56, GS-126
  - Preferences GS-52
  - Save GS-59
  - saving GS-58–GS-60
  - script messages RM-519–RM-525
  - Start Log GS-130
  - Stop Log GS-144
- File open HM-89, HM-153, HM-163
- file operations CDK-210
- file paths RM-103
- File save HM-118, HM-153, HM-164
- File type, changing HM-91
- File, atom type HM-186
- file, CSV CC-355
- file, dynamics results CC-355
- File, structure HM-186
- filename RM-79
  - arguments RM-516
  - case preferences RM-15
  - Open File dialog box RM-79
  - Save File dialog box RM-83, RM-83–RM-86

- files
  - AM1 parameter RM-690–RM-693
  - averages RM-629
  - bond angle parameter RM-666–RM-668
  - bond length parameter RM-665–RM-666
  - Brookhaven PDB format GS-113
  - chem.ini/Registry RM-743–RM-785
  - CNDO/INDO parameter RM-684–RM-685
  - Copy Image RM-206
  - extended Hückel parameter RM-681
  - force field parameters RM-659–RM-661
  - format RM-613
  - graphics RM-630
  - Help text RM-622
  - HIN structure RM-711–RM-712
  - hydrogen bond parameter RM-677
  - HyperChem Input RM-19, RM-711–RM-722, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-725, ??–RM-725, ??–RM-725
  - HyperChem program RM-621
  - improper torsion parameter RM-670–RM-672
  - Log GS-134, RM-616–RM-617, RM-621
  - Log on File menu GS-130
  - management RM-77–RM-116
  - MNDO parameter RM-688–RM-690
  - nonbonded parameter RM-673–RM-677
  - opening a HIN file GS-33

out-of-plane bending parameter RM-673  
 parameter file compile RM-353  
 PDB GS-114–GS-115  
 plot RM-629  
 program RM-621–RM-622  
 Protein Data Bank RM-503–RM-507  
 reading PDB RM-503  
 residue template RM-727–RM-741  
 residues RM-739  
 Save File dialog box RM-83–RM-86  
 saving in HIN format GS-59–GS-60  
 script RM-510, RM-622  
 selecting GS-35–GS-37  
 Semi-empirical parameters RM-679–RM-693  
 snapshot RM-623–RM-629  
 stopping Log File GS-144  
 structure  
     chem.ini/Registry RM-743–??  
     chem.tpl RM-727–RM-735  
 syntax for HIN RM-712–RM-722, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-725, ??–RM-725, ??–RM-725  
 template (TPL) structure RM-727–RM-741  
 torsion parameter RM-669–RM-670  
 type parameter RM-665  
 typing rules RM-655  
 water RM-621  
 Files, communication HM-186  
 Files, parameter HM-251  
 files, snapshot CC-357–CC-359  
 Files, structure HM-81  
 Final temperature, Molecular Dynamics Options dialog box RM-372  
 finite state machine CDK-33  
 Fitting structure to workspace HM-101  
 fixed atoms CC-86  
 Flat surface isosurface RM-398  
 Fletcher-Reeves CC-61  
 Fletcher-Reeves conjugate gradient method RM-365  
 Fletcher-Reeves optimization CC-337  
 float CDK-52  
 float, script arguments RM-517  
 Fock matrices  
     in log file RM-634  
 force constant  
     in log file RM-638  
 Force Constant, Restraint Forces dialog box RM-342  
 force constants, units CC-191  
 force field  
     AMBER GS-129, GS-130  
     choosing GS-130–GS-131, GS-152–GS-153  
     files RM-653–RM-677  
     MM+ GS-130  
     Parameters dialog box RM-352, RM-659  
     status line report RM-224  
 Force Field Options dialog box RM-293–RM-295  
 force field options, set up CC-208  
 force field parameters CC-176  
 force field selection CC-105  
 force fields CC-21  
 force fields, distinguishing features CC-105  
 force fields, limitations CC-175  
 forcefield coordinates  
     in log file RM-638  
 form CDK-120  
 formal charge CC-177  
 formation, heat of CC-135

forming bonds, simulation CC-93  
Fortran CDK-3, CDK-10, CDK-95, CDK-119,  
CDK-137, CDK-150, CDK-174,  
CDK-187, CDK-191  
frame CDK-98, CDK-103  
free radicals CC-242  
frequencies  
    in log file RM-634  
frequency CDK-231  
frequency, dynamics sampling CC-82  
Frequency, spectrometer HM-183  
Friction coefficient, Langevin dynamics RM-  
373  
friction coefficient, solvent CC-95  
front end CDK-8, CDK-11, CDK-17, CDK-  
195, CC-161  
Frontier molecular orbitals CC-146  
Frontier Orbital theory CC-145  
Frontier orbitals CC-44  
frozen atoms CC-86  
Full screen  
    setting, choosing extent of Copy  
        Image RM-612  
    Setup Image RM-100

## G

Gamma, polynucleotide torsion angle RM-160  
Gauche+, Constrain Bond Torsion dialog box  
RM-139  
Gauche-, Constrain Bond Torsion dialog box  
RM-139  
Gauge-Invariant Atomic Orbitals HM-218  
Gaussian basis set CC-263  
Gaussian functions CC-263  
Gaussian peak function HM-124, HM-157  
Gaussian Type Orbitals (GTO) CC-45, CC-263  
Gaussians, Cartesian CC-272  
Gaussians, Hermite CC-272  
geminal interactions CC-182  
General principles, NMR HM-214  
geometric restraints CC-83  
geometry optimization GS-129–GS-145  
    alanine zwitterion GS-156

    analyzing results GS-143  
    boat cyclohexane GS-140  
    molecular mechanics GS-130  
    of a subset GS-205  
Geometry Optimization on Compute menu GS-  
135, GS-140, GS-143, GS-156, GS-  
174  
Geometry Optimization, Compute Menu RM-  
357, RM-362–RM-367  
geometry optimizations CC-16, CC-59, CC-  
138  
geometry, molecular CC-43  
Get HSV  
    statements RM-511  
    types of input messages RM-510  
    variables, messages syntax RM-511  
Get HyperChem HM-153  
getting started GS-27–GS-44  
ghost atoms RM-326  
ghost-atoms CC-120, CC-278  
GIAO HM-218  
global minima CC-359  
global minimum CC-333  
    definition GS-130  
glossary, Help menu RM-115  
Glucagon, sample PDB files RM-507  
Glutamic acid (Glu), amino acid residue RM-  
143  
Glutamine (Gln), amino acid residue RM-143  
Glycine (Gly), amino acid residue RM-142  
Gouraud shaded surface RM-398  
Gradient  
    Single Point RM-360  
    status line report RM-362–RM-363  
gradient CDK-227, CDK-229  
    definition GS-129  
gradient of reference conformation  
    in log file RM-633  
gradient, RMS CC-63  
graph CDK-57, CDK-59, CDK-231, CDK-233  
Graph table data HM-170



Graph, coupling constants HM-138  
 Graph, peak information HM-129  
 Graph, plotting bar HM-155  
 Graph, shielding constants HM-138  
 Graph, spectral data HM-139  
 graphical user interface CC-161  
 graphics  
     bitmap RM-99, RM-100, RM-631  
     files RM-100, RM-630  
     metafile RM-101, RM-630  
     RGB bitmap RM-100  
     stored on Windows Clipboard RM-101, RM-206, RM-613–RM-615  
     transfers RM-612–RM-615  
     using Windows Applications RM-615  
 graphics settings RM-99  
 gray menu items RM-13  
 grid CDK-233  
 Grid Mesh Size, isosurface RM-394  
 Grid, 2D contour RM-395  
 grid, isosurface RM-393  
 ground state CC-242  
 group rotation RM-39  
 group selection defined RM-227–RM-228  
 GTO CC-45, CC-263  
 GTO primitives CC-263  
 GUI CDK-17, CDK-95, CDK-119, CDK-138, CDK-187, CDK-191, CDK-196

**H**

<sup>1</sup>H HM-81, HM-82, HM-111, HM-119, HM-178, HM-183, HM-185  
<sup>2</sup>H HM-82, HM-111, HM-178  
 Half-cystine (Cyx), amino acid residue RM-143  
 half-electron excited singlet states CC-39  
 half-electron method CC-243  
 half-electron technique CC-48  
 Hamiltonian CC-169, CC-227  
     electronic CC-170  
     exact CC-169  
     nuclear CC-170  
     Hamiltonian matrix elements CC-281  
     HAPI CDK-9, CDK-10, CDK-107, CDK-119, CDK-128, CDK-149, CDK-154, CDK-195, CDK-203  
     HAPI calls CDK-107, CDK-137, CDK-138  
     HAPI library CDK-140  
     hard nucleophiles and electrophiles CC-146  
     Hardware requirements HM-77  
     hard-wired menus CDK-38  
     harmonic approximation CC-149  
     harmonic frequencies CC-128  
     harmonic oscillator CC-22  
     harmonic potential CC-24  
     Hartree-Fock CC-38, CC-234  
     Hartree-Fock approximation CC-131, CC-261  
     Hartree-Fock method HM-217  
     hcAlloc CDK-147, CDK-290  
     hcConnect CDK-130, CDK-140, CDK-190, CDK-242  
     hcCopy CDK-130  
     hcDisconnect CDK-130, CDK-140, CDK-190, CDK-244  
     hcExec CDK-24, CDK-46, CDK-99, CDK-130, CDK-197, CDK-198  
     hcExecBin CDK-141, CDK-248  
     hcExecTxt CDK-140, CDK-190, CDK-245  
     hcExit CDK-140, CDK-245  
     hcFree CDK-148, CDK-291  
     hcGetBlock CDK-146, CDK-267  
     hcGetErrorAction CDK-132, CDK-148, CDK-295  
     hcGetInt CDK-142, CDK-251  
     hcGetIntArr CDK-142, CDK-254  
     hcGetIntArrElm CDK-142, CDK-255  
     hcGetIntVec CDK-142, CDK-253  
     hcGetIntVecElm CDK-142, CDK-258  
     hcGetNotifyData CDK-131, CDK-147, CDK-288  
     hcGetReal CDK-143, CDK-252  
     hcGetRealArr CDK-143, CDK-257  
     hcGetRealArrElm CDK-143, CDK-260  
     hcGetRealArrXYZ CDK-143, CDK-261  
     hcGetRealVec CDK-143, CDK-256

hcGetRealVecElm CDK-143, CDK-259  
 hcGetRealVecXYZ CDK-143, CDK-262  
 hcGetStr CDK-144, CDK-263  
 hcGetStrArrElm CDK-144, CDK-265  
 hcGetStrVecElm CDK-144, CDK-264  
 hcInitAPI CDK-140, CDK-241  
 Hcl CDK-6, CDK-22, CDK-99, CDK-205  
 Hcl command CDK-49  
 Hcl script CDK-2, CDK-9, CDK-29, CDK-38,  
     CDK-64, CDK-99, CDK-196  
 Hcl script command CDK-130  
 Hcl text string CDK-141  
 hcLastError CDK-131, CDK-148, CDK-294  
 hcNotifyDataAvail CDK-147, CDK-287  
 hcNotifySetup CDK-147, CDK-286  
 hcNotifyStart CDK-131, CDK-147, CDK-284  
 hcNotifyStop CDK-131, CDK-147, CDK-285  
 hcQuery CDK-45, CDK-99, CDK-130  
 hcQueryBin CDK-141, CDK-148, CDK-249  
 hcQueryTxt CDK-140, CDK-148, CDK-190,  
     CDK-246  
 hcSetArrElm CDK-146  
 hcSetBlock CDK-147, CDK-283  
 hcSetErrorAction CDK-131, CDK-148, CDK-  
     296  
 hcSetInt CDK-144, CDK-268  
 hcSetIntArr CDK-144, CDK-271  
 hcSetIntArrElm CDK-145, CDK-272  
 hcSetIntVec CDK-144, CDK-270  
 hcSetIntVecElm CDK-145, CDK-275  
 hcSetReal CDK-145, CDK-269  
 hcSetRealArr CDK-145, CDK-274  
 hcSetRealArrElm CDK-145, CDK-277  
 hcSetRealArrXYZ CDK-146, CDK-278  
 hcSetRealVec CDK-145, CDK-203, CDK-273  
 hcSetRealVecElm CDK-145, CDK-276  
 hcSetRealVecXYZ CDK-145, CDK-279  
 hcSetStr CDK-146, CDK-280  
 hcSetStrArrElm CDK-282  
 hcSetStrVecElm CDK-146, CDK-281  
 hcSetTimeouts CDK-131, CDK-148, CDK-293  
 hcShowMessage CDK-148, CDK-292  
 header files CDK-138, CDK-149  
 headers, Snapshot files RM-623–RM-626  
 heat of formation CC-135  
     in log file RM-633  
 Heat steps, Monte Carlo RM-371  
 Heat time RM-370  
 heating time, molecular dynamics CC-91  
 heating, dynamics CC-75  
 heating, trajectory CC-346  
 heat-of-formation CDK-221  
 Helix Form, polynucleotide strand conforma-  
     tion RM-159  
 Hellman-Feynman theorem HM-219  
 Help HM-87  
 help  
     on-line GS-24, GS-29  
 Help Menu HM-159  
 Help menu GS-29, RM-115  
     script messages RM-578  
 Help text HM-186, HM-199  
 Help text, file RM-621, RM-622  
 Help, product GS-20  
 Hermite Gaussians CC-272  
 Hessian CC-67, CC-340  
 Hessian matrix CC-128  
 HETATOM, creating PDB-type files RM-505  
 hfExecTxt CDK-195  
 hfGetRealArr CDK-201  
 hfSetRealArr CDK-195  
 hide-toolbar CDK-47  
 highest occupied molecular orbital GS-189  
 highlighting menus RM-12  
 HIN file HM-81, HM-89, HM-90, HM-153,  
     HM-163, HM-186, HM-200  
 HIN *See* HyperChem input file  
 HIN *See* HyperChem input file  
 Histidine (Hid), amino acid residue RM-143  
 Histidine (Hie), amino acid residue RM-143  
 Histidine (Hip His), amino acid residue RM-  
     143  
 HOMO CC-146, CC-231  
 HOMO *See* highest occupied molecular orbital  
 HOMO-, Orbitals dialog box RM-403  
 Hooke's law CC-22

Horizontal grid points RM-395  
 hot link CDK-109, CDK-123  
 HSV CDK-3, CDK-8, CDK-25, CDK-27,  
 CDK-66, CDK-101, CDK-108, CDK-  
 122, CDK-130, CDK-140, CDK-198,  
 CDK-205  
   arguments CDK-30  
   array CDK-33  
   environment CDK-58  
   read CDK-29, CDK-49, CDK-140  
   scalar CDK-31  
   vector CDK-32  
   write CDK-29, CDK-49, CDK-140  
 HSV. *See* HyperChem State Variables  
 hsv.h  
   generating CDK-142  
 huckel CDK-228  
 Hückel constant CC-121, CC-282  
 Hückel Constant, Semi-empirical Options dia-  
 log box RM-304  
 hybridization CC-177, CC-216  
 hydrogen CDK-222  
   bond parameter file RM-677  
   Explicit Hydrogens RM-30, RM-119–  
   RM-120  
   PDB file storage RM-85  
 hydrogen bond potential CC-26  
 hydrogen bonding CC-182, CC-188, CC-200  
 hydrogen bonding term CC-205  
 hydrogen bonds GS-89, CDK-217  
   calculating GS-122–GS-123  
   displaying GS-122–GS-123, GS-206  
   recomputing GS-206  
 hydrogens CDK-217, CDK-219  
 hydrogens, axial GS-141  
 Hydrogens, creating PDB-type files RM-506  
 HyperChem  
   About dialog box RM-116  
   files RM-619–RM-631  
   program files RM-620, RM-621–RM-  
   622  
   settings RM-747–RM-765  
 HyperChem API CDK-2, CDK-137, CDK-148,  
 CDK-149, CDK-154  
 HyperChem Application Programming Inter-  
 face CDK-2, CDK-107, CDK-119,  
 CDK-137  
 HyperChem architecture CC-161  
 HyperChem Command Language CDK-6,  
 CDK-22, CDK-49, CDK-95, CDK-  
 123, CDK-127, CDK-138  
 HyperChem input file (HIN) GS-33–GS-34  
 HyperChem Input files RM-79, RM-619, RM-  
 711–RM-722, ??–RM-723, ??–RM-  
 723, ??–RM-723, ??–RM-723, ??–  
 RM-724, ??–RM-724, ??–RM-724,  
 ??–RM-724, ??–RM-724, ??–RM-  
 724, ??–RM-725, ??–RM-725, ??–  
 RM-725  
   atoms RM-717–RM-718  
   bonding information RM-733  
   force field information RM-714  
   option RM-84  
   residues RM-716  
   save RM-84  
   saving molecular system RM-19  
   selections RM-720–RM-722, ??–RM-  
   723, ??–RM-723, ??–RM-723,  
   ??–RM-723, ??–RM-724, ??–  
   RM-724, ??–RM-724, ??–RM-  
   724, ??–RM-724, ??–RM-724,  
   ??–RM-725, ??–RM-725, ??–  
   RM-725  
   view RM-715  
 HyperChem OS CDK-138  
 HyperChem State Variables RM-510, RM-  
 518–RM-584  
   Get statements RM-511, RM-517  
   Set statements RM-511  
 HyperChem state variables CDK-3, CDK-27

HyperChem Tools RM-27–RM-50, RM-61–??  
 moving atoms with mouse RM-20,  
 RM-27–RM-50, RM-61–??

HyperChem Windows  
 Control menu box RM-5  
 Preferences dialog box RM-95–RM-  
 105  
 Setup Image RM-102

HyperChem, importing structure from HM-94,  
 HM-136, HM-153

HyperChem, interaction with HM-79, HM-81

HyperChem, selection of equivalent atoms  
 HM-111, HM-135, HM-178

HyperChem, selection of labelled atoms HM-  
 112, HM-180

HyperChem, selection of NMR atoms HM-109,  
 HM-134, HM-176

HyperChem, selection of quantum atoms HM-  
 108, HM-133, HM-175

HyperChem, use with HM-75

hyperconjugation, mixed mode calculations  
 CC-259

HyperEHT CDK-12

HyperGauss CDK-12, CDK-195

HyperMM+ CDK-12

HyperMM+, HyperNewton, HyperEHT, Hy-  
 perNDO, HyperGauss CC-162

HyperNDO CDK-12, CDK-191

HyperNewton CDK-12, CDK-191

HyperNMR CDK-16

**I**

Icon, backend HM-114, HM-121

Icon, HyperNMR HM-85, HM-142

icons RM-27

IGG, sample PDB file RM-506

Ignore warning message RM-359–RM-360

image setup preferences RM-99

Image, copying HM-103

images CDK-218  
 choosing  
 destination RM-613–??

extent RM-612

color RM-614

file formats RM-613

imp parameter file RM-660, RM-670–RM-672

imp, force field information in TPL file RM-  
 734

Import orbital data RM-78, RM-92

Importing structure from HyperChem HM-94,  
 HM-153

Importing structures HM-89

improper dihedrals CC-184, CC-198, CC-203

improper torsion parameter file RM-670–RM-  
 672

IMSG CDK-15, CDK-34

IMSGs. *See* input messages

In vacuo  
 geometry optimization, Compute  
 menu RM-367, RM-372

include file CDK-139

Increment for contour plot, Grid Control dialog  
 box RM-396

incremental model building RM-487–RM-488

increments, rotation and translation RM-97,  
 RM-98

independent-electron approximation CC-229

independent-electron method CC-279

Index, Help menu RM-115

individual molecular orbitals, plotting GS-189–  
 GS-195

INDO CC-123, CC-132, CC-155, CC-249, CC-  
 252, CC-288–CC-292  
 elements treated by RM-301  
 parameter files RM-684–RM-685  
 Semi-empirical Method dialog box  
 RM-298

INDO mixed model CC-287

inertia, moments RM-539

inertial axes CDK-216, CDK-222, RM-21–  
 RM-25

info CDK-213

Information table, peak HM-129

infra red CDK-231

infrared absorption CC-369  
 Infrared frequencies, characteristic CC-150  
 infrared spectroscopy CC-149  
 infrared spectrum, negative frequencies CC-149  
 inhibit redisplay CDK-215  
 initial conditions, dynamics CC-343  
 initial guess, MO coefficients CC-119, CC-276  
 Initialization script RM-4  
 initialization script CDK-53  
 INITIATE, Excel message RM-606  
 Inner radius, MM+ Options dialog box RM-293  
 inorganic structures, building RM-489  
 Input files HM-186  
 input messages RM-510–RM-517  
   int (internal) atomic coordinates,  
     TPL file RM-734–RM-735  
   using HyperChem messages RM-510–RM-517  
 Input, data HM-81  
 Input, structure HM-75  
 Installation, software HM-77  
 integer CDK-30  
 integer, arguments CDK-52, RM-517  
 integral cutoff, two-electron CC-117, CC-275  
 integral format  
   Raffenetti CC-274  
   regular CC-273  
 integral format, Raffenetti RM-322  
 integral storage path RM-103  
 Integrated Development Environment CDK-189  
 integrating HyperChem, Dynamic Data Exchange RM-509  
 Integration, peak HM-127  
 Intensities of transitions HM-249  
 Intensity, level for inclusion of peak HM-190  
 interactions, bonding CC-182  
 interactions, electrostatic CC-182  
 interactions, nonbonding CC-182  
 INTERFACE CDK-152  
 interface to HyperChem CDK-1, CDK-6,  
   CDK-137  
 Intermediate Neglect of Differential Overlap (INDO) CC-249  
 intermediates CC-33  
 intermediates, potential CC-137  
 intermediates, reactive CC-339  
 interpreter CDK-95, CDK-127  
 interval, dynamics sampling CC-82  
 Introduction to HyperNMR HM-81  
 Invert RM-206–RM-207  
   on Edit menu RM-202, RM-486  
 ionic reactivity CC-146  
 ionization energy CC-139  
 ionization potential CC-43  
 IPX/SPX RM-104  
 IR spectra  
   in log file RM-633  
 IR spectrum, importing RM-92  
 IR spectrum, storing RM-93  
 ISIS Sketch files  
   open RM-80  
   save RM-84  
 ISIS/Draw  
   clipboard format RM-204  
 isokinetic energy ensemble CC-74  
 isolated atomic energy  
   in log file RM-633  
 Isoleucine (Ile), amino acid residue RM-143  
 Isomer, residue conformation RM-144  
 Isosurface RM-796  
 isosurface GS-186, CDK-217, CDK-232, CDK-233  
 Isosurface Colors RM-102  
 Isosurface contour level RM-397  
 Isosurface Grid RM-393  
 isosurface grid RM-394  
 Isosurface Rendering RM-396  
 isosurface transparency level RM-398  
 Isosurface, 3D RM-393  
 Isosurface, mapped 3D RM-393  
 isosurfaces CC-253  
 isothermal ensemble CC-74

Isotopes HM-82, HM-111, HM-173, HM-178  
ITEM CDK-44  
item

DDE CDK-108

iter, status line report RM-363, RM-369

Iteration limit GS-183

iteration limit CC-116

Iteration limit, Semi-empirical Options dialog  
box RM-306

Iterations, maximum SCF HM-182

## J

Jorgensen-Salem isosurface RM-397

## K

keep-atom-charges RM-545

keyboard

alternatives

HyperChem RM-10–RM-12

Windows RM-9–RM-10

alternatives to mouse GS-31–GS-32

equivalents for tools RM-49–RM-50

shortcuts GS-32

keyboard accelerator CDK-39

Keyboard alternatives HM-88

Keyboard equivalents, mouse actions HM-162

Keyboard rotation and translation RM-98

keys to operate HyperChem tools RM-49–RM-50

kinetic energy integrals CC-262

KLM values, table, split-valence basis set CC-268

Klopman-Salem equation CC-144

Koopmans' theorem CC-139

## L

label CDK-103, CDK-197

labeling GS-38–GS-39

atoms GS-63–GS-64

by number GS-81

by symbol GS-83

removing labels GS-38

subsets GS-100–GS-102

Labelled atom selection HM-178

Labelled atoms HM-82, HM-111, HM-173,  
HM-178

labelled-atoms selection in HyperChem HM-112, HM-180

labels CDK-98, CDK-218, CDK-220

Labels dialog box GS-38, GS-210

Labels, atom HM-96, HM-168

lambda CC-69

Lambda, reaction map coefficient RM-354

Langevin Dynamics RM-383

Compute menu RM-357

Langevin dynamics GS-174, CC-18, CC-95

Large molecules, calculating HM-107, HM-130

LCAO HM-218, CC-35, CC-232

LCAO-MO CC-44

leap-frog algorithm CC-72, CC-344

legacy applications CDK-155

legacy code CDK-119

legacy programs CDK-10

Lennard-Jones function CC-26

Lennard-Jones potential CC-184, CC-196

Leucine (Leu), amino acid residue RM-143

levels of molecular organization RM-17–RM-18

library CDK-137, CDK-149, CDK-151, CDK-190

License Timeout Interval RM-104

license, network RM-104

Limit, SCF convergence HM-181

LINE

Align Molecules dialog box RM-218

Align Viewer dialog box RM-216

definition RM-22

Name Selection dialog box RM-238

rotate about, Rotate dialog box RM-210

Line view, spectra HM-122

Line width function HM-124

Line width scroll bar HM-148

Line width, fine control HM-127, HM-191  
 Line widths HM-156  
 Line widths, peak HM-125  
 Linear  
     Constrain Bond Angle dialog box  
         RM-138  
     Constrain Geometry dialog box RM-  
         135  
 Linear Combination of Atomic Orbitals HM-  
     218  
 linear combination of atomic orbitals CC-35  
 Linear Combination of Atomic Orbitals-Molec-  
     ular Orbital CC-44  
 LineDown CDK-114  
 Lines, isosurface RM-397  
 LineUp CDK-114  
 LinkExecute CDK-123  
 LinkItem CDK-122, CDK-123  
 LinkMode CDK-122, CDK-123  
 LinkRequest CDK-122  
 LinkTopic CDK-122  
 load command CDK-129  
 LoadHAPI CDK-139, CDK-151, CDK-181  
 loading CDK-124  
 load-time CDK-149, CDK-151  
 load-user-menu CDK-38  
 local minima CC-333  
     definition GS-130  
 Log Comments  
     on File menu RM-78  
 log file  
     entering comments in GS-134  
     keeping information in GS-130  
     stopping GS-144  
 log file information RM-631–RM-638  
     for UV visible spectra computation  
         RM-415  
 log file path RM-103  
 Log file print level HM-189  
 Log file, starting HM-113, HM-154, HM-165  
 Log file, stopping HM-122, HM-154  
 Log files HM-186, HM-210, RM-621  
     HyperChem files RM-621  
     results stored RM-616–RM-617  
 log files CDK-214  
 Log on File menu GS-130  
 Log P RM-456  
 log, open RM-90  
 logging CDK-214  
 London dispersion forces CC-40  
 Lone pair (LP)  
     in atom typing RM-658  
 lone pairs CC-200  
 Lorentzian peak function HM-124, HM-157  
 lowest excited electronic state of ethylene GS-  
     209–GS-217  
 lowest state CC-242  
 Lowest state, Semi-empirical Options dialog  
     box RM-306  
 lowest unoccupied molecular orbital GS-189  
 LUMO CC-146, CC-231  
 LUMO *See* lowest unoccupied molecular orbit-  
     al  
 LUMO+, Orbitals dialog box RM-403  
 Lysine (Lys), amino acid residue RM-143  
**M**  
 MacroButton CDK-112  
 macromolecules  
     working with GS-113–GS-126  
 macros CDK-110, CDK-113, CDK-117  
 Magnetic shielding HM-216  
 mailing list GS-21  
 MAIN subroutine CDK-113  
 Make Zwitterion RM-145–RM-146  
     Databases menu RM-140  
 makefile CDK-150, CDK-151  
 Map, reaction map list box RM-355  
 Mapped Isosurface RM-393  
 mass, atomic CC-178  
 mass, HIN file syntax RM-718  
 master CDK-13  
 master - slave Architecture CDK-13  
 Max delta, Monte Carlo RM-371

Maximize button HM-143  
 maximize button GS-29  
 maximize button, HyperChem Windows RM-5  
 Maximize/Minimize buttons HM-88  
 Maximum Excitation, Configuration Interaction dialog box RM-309  
 Maximum iterations, SCF HM-182  
 Maximum number of water molecules, Periodic Box Options dialog box RM-338  
 MDL MOL files  
   open RM-80  
   save RM-84  
 measurements  
   atom identification, with Selection tool RM-35  
   bond angles, with Selection tool RM-36  
   bond lengths, with Selection tool RM-36  
   distances between atoms, with Selection tool RM-36  
   improper angles, with Selection tool RM-36  
   improper torsion angles, with Selection tool RM-36  
   molecule identification, with Selection tool RM-36  
   residue identification, with Selection tool RM-35  
   with Selection tool RM-36  
   torsion angles, with Selection tool RM-36  
 measuring  
   axial hydrogens GS-139  
   nonbonded distances GS-88  
   properties of minimized system GS-136–GS-137  
   structural properties GS-81–GS-90  
   torsion angles GS-87–GS-88  
  
 Mechanics Print Level RM-90  
 MechanicsPrintLevel, in chem.ini RM-631  
 mechanisms, reaction CC-137  
 memory allocation CDK-147, CDK-156  
 MENU CDK-44  
 menu CDK-207, CDK-212  
   adding CDK-37  
   enable CDK-46  
 menu activation CDK-49, CDK-50, CDK-67  
 Menu Activator RM-511  
   types of input messages RM-511  
 Menu bar HM-87, HM-144  
 menu bar GS-29  
 menu bar, HyperChem Windows RM-5  
 menu caption CDK-212  
 menu file CDK-24, CDK-38  
 Menu Files CDK-2  
 menu invocation CDK-140  
 menu item CDK-39, CDK-50  
 Menu item, choosing HM-89  
 menu items  
   Semi-empirical GS-206, GS-215  
   with ellipsis GS-33–GS-34  
 menu structure CDK-21  
 Menu, close HM-88  
 Menu, open HM-88  
 menu-help RM-515  
 MENUITEM CDK-44  
 menuitem CDK-212  
 menus  
   Build GS-85, GS-152, GS-182, GS-201, RM-117–RM-139  
   close using keyboard alternatives RM-9  
   Compute GS-134, GS-135, GS-140, GS-143, GS-156, GS-168, GS-174, RM-357–RM-469  
   Control GS-29  
   Databases RM-139–RM-192, RM-493–RM-500  
   Display GS-37, GS-41, GS-42, GS-74,



GS-105, GS-116, GS-167, RM-21, RM-216  
 Edit GS-138, GS-180, RM-201–RM-220, RM-486  
 File GS-33, GS-58–GS-60, GS-126, GS-130, RM-77–RM-105, RM-616  
 gray items RM-13  
 Help GS-29, RM-115–RM-116  
 HyperChem RM-12–RM-13  
 keyboard alternatives GS-31–GS-32  
 Script RM-105–RM-107  
 Select GS-106, GS-117, GS-166, RM-34–RM-35, RM-220–RM-246, RM-496  
 Setup GS-182, GS-201, GS-205, GS-206, RM-287–??  
 Merge  
   adds molecules RM-82  
   on File menu RM-77  
 merging two structures GS-203–GS-204  
 mesh isosurface RM-397  
 mesh, isosurface grid RM-394  
 message CDK-24, CDK-49, CDK-122, CDK-213  
   status CDK-215  
 message box CDK-148  
 message passing CDK-108  
 messages CDK-7, CDK-14, CDK-33, CDK-108, CDK-155, CDK-157, CDK-207, RM-518–RM-584  
   cross references of RM-579  
 messages for scripts RM-518  
   arguments RM-516  
   commands RM-516  
   cross-references RM-584–RM-602  
   input RM-510–RM-517  
   online Help RM-115  
   script file RM-509  
   syntax RM-511–RM-517  
   using with HyperChem RM-510–RM-517  
   variable RM-511–RM-516  
   warning RM-359–RM-360  
 Messages, HyperNMR HM-87, HM-144  
 messaging CDK-107  
 Metafile RM-99, RM-613, RM-615  
   graphics RM-630  
   Setup Image dialog box RM-101  
 metafile CDK-218  
 metafile preferences RM-99  
 methane CDK-57  
 Methionine (Met), amino acid residue RM-143  
 Metropolis method CC-100  
 MFC CDK-10, CDK-159, CDK-173  
 microcanonical ensemble CC-74  
 Microsoft Developer Studio CDK-151  
 Microsoft Excel CDK-2, CDK-17, CDK-107, CDK-115, RM-606–RM-609  
 Microsoft Foundation Classes CDK-10, CDK-159, CDK-173  
 Microsoft Windows CDK-107, RM-4–RM-6  
   controls printer RM-94  
 Microsoft Windows API CDK-148  
 Microsoft Word CDK-2, CDK-107, CDK-109  
 microstate CI CC-40  
 Microstate CI Method CC-40  
   Configuration Interaction dialog box RM-309  
   practical application RM-310  
 microstates CC-244  
 MINDO/3 CC-123, CC-132, CC-155, CC-249, CC-252, CC-292–CC-296  
 MINDO3  
   elements treated by RM-301  
   extension of INDO RM-298  
   parameter files RM-679, RM-685–RM-688  
   Semi-empirical Method dialog box RM-298  
 minima, global CC-359

minima, local CC-333  
 minimal basis set CC-265  
 minimizations, geometry CC-59  
 Minimize button HM-88, HM-143  
     HyperChem Windows RM-5  
     Molecular Dynamics Results dialog  
     box RM-382  
 Minimum distance between solvent and solute  
     atoms, Periodic Box Options dialog  
     box RM-338  
 minimum, global CC-333  
 Minor, Align Molecules dialog box RM-218  
 miscellaneous variable messages RM-580–  
     RM-582  
 mixed calculations  
     choosing an area of interest GS-211–  
     GS-212  
 Mixed mode calculations RM-311, RM-334  
 mixed mode calculations CC-256, CC-277  
 mixed model, CNDO and INDO CC-287  
 mixed model, MINDO/3 CC-295  
 mixed model, MNDO, AM1, and PM3 CC-304  
 Mixed-mode calculations HM-82, HM-107,  
     HM-130  
 MM+ CC-105, CC-176, CC-190  
     file structure RM-745  
     force field GS-130–GS-131  
     Molecular Mechanics Force Field  
     dialog box RM-290  
     parameters RM-680  
     settings RM-769–RM-771  
 MM+ Options dialog box RM-291–RM-293  
 mm+,mm+ file structure RM-745  
 MM2 CC-110, CC-190  
 MM2(1977) CC-106  
 mmpben4 parameter file RM-660  
 mmplus.exe program file RM-621  
 mmpoop parameter file RM-660  
 mmptor4 parameter file RM-660  
 MNDO CC-123, CC-132, CC-155, CC-156,  
     CC-249, CC-296–CC-304  
     elements treated by RM-302  
     parameter files RM-688–RM-690  
     Semi-empirical Method dialog box  
     RM-299  
 mnu file CDK-37  
 MO coefficients, initial guess CC-119, CC-276,  
     RM-325  
 mode  
     drawing GS-47, GS-81  
 Model Build, changed from Add H & Model  
     Build RM-479  
 Model Build/Add H & Model Build RM-124–  
     RM-125  
     Build menu RM-117  
 Model Builder RM-124–RM-125  
     2D drawings conversions RM-117  
     adds hydrogen atoms RM-30  
     assigns atom types RM-129  
     creating 3D structures GS-83–GS-85  
     evaluates chirality of atoms RM-487  
     incremental model building RM-487  
     invoking GS-65–GS-66  
 model builder CDK-58, CDK-218  
 Modified Intermediate Neglect of Differential  
     Overlap *See* MINDO/3  
 modifying parameters CC-207  
 mol, HIN file syntax RM-716  
 MO-LCAO approximation CC-231  
 Molecular Coordinate System HM-152, RM-  
     18–RM-20  
 Molecular Dynamics GS-147–GS-170  
     Averages dialog box RM-377–RM-  
     380  
     Named Selections use RM-242  
     averages files RM-629  
     averaging GS-171–GS-173  
     calculations RM-368–RM-382  
     Compute Menu RM-368–RM-383  
     Compute menu RM-357  
     in water GS-147–GS-177

- on Compute menu GS-168
- Options dialog box RM-369–RM-374
- Playback dialog box RM-375–RM-376
- playback dynamics GS-170
- results GS-173
- Results dialog box RM-380–RM-382
- simulated annealing GS-168–GS-170
- Simulation
  - storing results RM-617–RM-618
  - time periods RM-382–RM-383
- molecular dynamics CDK-55, CDK-225, CC-18, CC-71, CC-127, CC-342–CC-359
- Molecular dynamics calculations CC-88
- molecular dynamics equilibration CC-76
- Molecular Dynamics Options dialog box GS-168, GS-173
- molecular geometry CC-43
- molecular graph CDK-59
- Molecular Mechanics
  - calculations RM-289–??
  - Force Field dialog box RM-290–RM-291
  - method use RM-361
  - on Setup menu RM-287
  - Optimization dialog box RM-364–RM-367
- molecular mechanics CDK-102, CDK-222, CC-21, CC-175
- molecular mechanics calculations
  - log file information RM-638
- Molecular Mechanics on Setup menu GS-201, GS-205
- Molecular Mechanics Optimization
  - dialog box GS-156
- molecular mechanics, type selection CC-105
- molecular orbital calculations GS-179–GS-196
- molecular orbitals CC-126, RM-404
  - practical guidelines RM-405
  - total number saved in log file RM-632
- molecular organization, levels RM-17–RM-18
- Molecular properties, plotting RM-392
- molecular renderings
  - changing GS-39–GS-43
- molecular system RM-18–RM-20
  - printing RM-94, RM-616
- Molecular system view HM-144, HM-155
- Molecule, charge HM-181
- Molecule, copying HM-103
- molecules RM-226
  - built from residues RM-492–RM-500
  - changes after drawing RM-475
  - creating GS-57–GS-67
  - drawing RM-471–RM-486
  - in Microsoft Excel macro RM-607, RM-609
  - NMA GS-197–GS-199
  - numbering RM-237
  - propene
    - creating GS-57–GS-65
  - reflecting one end GS-138
  - residues RM-17
  - saving GS-196
  - selected by name RM-236
  - Selection level RM-32
  - translating/rotating/scaling GS-69–GS-80
  - unlabelling GS-92
  - water GS-204
- Molecules, Apply to, Rotate dialog box RM-210
- Molecules, Select menu RM-221
- Møller-Plesset perturbation RM-321
- Møller-Plesset second-order perturbation (MP2) CC-41, CC-117, CC-246–CC-248, CC-261
- moments-of-inertia RM-539
- monitor HyperChem CDK-132
- monopole RM-692

Monte Carlo GS-174, CDK-225, CC-19, CC-95, RM-384  
     Compute menu RM-358  
 MOPAC Z-Matrix file HM-81, HM-89, HM-153, HM-163, HM-186, HM-200  
 MOPAC Z-Matrix files  
     open RM-80  
     save RM-84  
 Morse function CC-24  
 mouse CDK-208  
     in changing clipping slab GS-79  
     keyboard alternatives GS-31–GS-32  
     pointer, HyperChem Windows RM-5  
     right button functions RM-20  
     use RM-7–RM-9  
     using GS-30–GS-31  
 Mouse operations HM-88, HM-161  
 Move structure HM-161  
 movement, atomic, Monte Carlo RM-371  
 Movie, AVI RM-376  
 Moving a molecule HM-89  
 moving in a dialog box, using keyboard alternatives RM-10  
 Moving structures HM-100, HM-157  
 moving through a list, using keyboard alternatives RM-10  
 MP2 CC-117, CC-261  
 mp2 CDK-229  
 MP2 calculations RM-321  
 MP2 Correlation Energy CC-246–CC-248  
 Mulliken approximation CC-282  
 Mulliken atomic charges CC-143  
 Multiple Selections RM-227  
     Select menu RM-221  
 Multiple Selections on Select menu GS-51, GS-54  
 multiple windows use RM-6  
 multiplicity CDK-221, CC-229, CC-242  
 Multiplicity, spin HM-181  
 multiplicity, spin CC-46, CC-116  
 mutagenesis, site-specific GS-94–GS-97, GS-123  
 Mutate  
     Databases menu RM-140  
     dialog box RM-190–RM-192  
 mutating  
     amino acid residues RM-496  
     nucleic acid residues RM-499–RM-501  
**N**  
<sup>15</sup>N HM-81, HM-82, HM-111, HM-119, HM-178, HM-183, HM-185  
 Name Selection  
     dialog box RM-237–RM-239  
     Molecular Dynamics use RM-241  
     for Restraints RM-241  
     Select menu RM-222  
 Named *selections*, Molecular Dynamics Averages  
     dialog box RM-378–RM-379  
 names, assignment for selections RM-236  
 naming selections RM-240–RM-241  
 NDDO HM-228, CC-132, CC-249  
 NDO HM-226, CC-35, CC-248–CC-250  
 NDO calculations  
     log file information RM-633  
 NDO methods CC-131  
 NDO methods, options CC-122  
 ndo.exe program file RM-621, RM-622  
 negative frequencies CC-149  
 Neglect of Diatomic Differential Overlap (NDDO) HM-228, CC-132, CC-249  
 Neglect of Differential Overlap (NDO) HM-226, CC-35, CC-131, CC-248–CC-250  
 neighbors CDK-219  
     maximum handled by Model Builder RM-489  
 neighbors, bonded CC-177  
 net charges  
     in log file RM-632

log file RM-633  
 NETBIOS, NETBEUI, NETAP RM-104  
 network CDK-17  
 Network License RM-104  
 network license timeout RM-104  
 Network options RM-346  
 New  
   on File menu RM-77  
   prepares HyperChem for new molecules RM-78  
   residues RM-739–RM-741  
 New on File menu GS-56, GS-126  
 new version of Tcl/Tk CDK-128  
 New workspace HM-153  
 newton.exe program file RM-621  
 Newton-Raphson CC-62  
 Newton-Raphson optimization CC-338  
 Newton-Raphson Optimization Method RM-366  
 next lowest state CC-242  
 Next lowest state, Semi-empirical Options dialog box RM-306  
 NMA *See* N-methylacetamide  
 NMAKE CDK-161  
 N-methyl amino (Nme), amino acid residue RM-143  
 N-methylacetamide  
   changing from cis- to trans- form GS-199–GS-201  
   creating GS-197–GS-199  
   geometry optimization GS-201  
 NMR atoms HM-82  
 NMR atoms, selecting HM-108, HM-133, HM-173, HM-175  
 NMR atoms, selection by type HM-180  
 NMR files, format HM-201  
 NMR, general principles HM-214  
 nmr.exe HM-186  
 nmr.hlp HM-186, HM-199  
 nmr.ini HM-186, HM-187  
 nmr\_cupl.ab HM-198  
 nmr\_cupl.abp HM-186, HM-252  
 nmr\_shft.abp HM-186, HM-198, HM-252  
 nmr-atoms selection in HyperChem HM-109, HM-134, HM-176  
 nmrcalc.exe HM-186, HM-187  
 nmrparam.in HM-186, HM-199  
 nmrparam.out HM-186, HM-199  
 nmrspect.in HM-186, HM-199  
 nmrspect.out HM-186, HM-199  
 nmrtype.rul HM-186, HM-194  
 nmrtype.txt HM-186, HM-194  
 NOE data CC-84  
 nonbonded cutoffs CC-66, CC-108  
 nonbonded distances  
   measuring GS-88  
 non-bonded interactions CC-26  
 nonbonded interactions CC-182  
 nonbonded parameter files RM-673–RM-677  
 nonbonded potential CC-29  
 non-nuclear basis functions CC-278  
 normal coordinate analysis CC-366  
 normal modes  
   in log file RM-634  
 normal modes, vibration CC-128  
 notation, basis sets CC-271  
 notification CDK-131, CDK-132, CDK-147, CDK-161  
 notification agent CDK-156  
 notifications CDK-29, CDK-128, CDK-155, CDK-212  
 notify-on-update CDK-30  
 Nuclear Magnetic Resonance *see* NMR  
 Nuclear Overhauser Effect (NOE) CC-84  
 Nuclear spin coupling HM-222  
 nuclear-electron attraction CC-262  
 nucleic acid CDK-224  
 Nucleic Acids RM-157–RM-158, RM-497–RM-500  
   on Databases menu RM-140, RM-497  
   dialog box RM-14, RM-159–RM-162, RM-192, RM-497  
   mutating residues RM-192, RM-499–RM-501

residues RM-737–RM-741  
nucleic acids, adding RM-739  
number of d orbitals CC-119  
Number of residues selected, Select menu RM-226  
Number, Orbitals dialog box RM-403  
numbering atoms, residue molecules RM-237  
numbering of atoms CDK-30  
Numbers, Select dialog box RM-234

## O

<sup>15</sup>O HM-81, HM-82, HM-111, HM-119, HM-178, HM-183, HM-185  
occupancies  
    in log file RM-632, RM-633  
occupancy, orbital CC-43  
occupied alpha orbitals  
    in log file RM-633  
occupied beta orbitals in UHF  
    in log file RM-633  
Occupied Orbital Criterion  
    Configuration Interaction dialog  
        box RM-309  
OCR file CDK-54  
Octahedral, Constrain Geometry dialog box  
    RM-135  
OLE CDK-108, CDK-137  
Omega, amino acid residue torsion angle RM-144  
OMSG CDK-15, CDK-29, CDK-34, CDK-211, CDK-212  
OMSG, using HyperChem messages RM-510  
one-center exchange integrals CC-290  
on-line help GS-24  
opaque isosurface RM-397  
Open  
    File dialog box RM-78–RM-82  
    on File menu RM-77  
    menu, using keyboard alternatives  
        RM-9  
    Run Script dialog box RM-106  
    Script, on Script Menu RM-106

Start Log dialog box RM-90  
open architecture CDK-13  
Open file HM-89, HM-153, HM-163  
Open File dialog box GS-34  
Opening HyperNMR HM-85, HM-142  
open-shell CDK-56  
Operating system HM-77  
operating system CDK-21  
OPLS CC-105, CC-110, CC-176, CC-200  
    file structure RM-745  
    Molecular Mechanics Force Field  
        dialog box RM-291  
    settings RM-773–RM-775  
opls, opls, file structure RM-745  
optimization CDK-226  
optimization cycles CC-63  
optimization methods CC-335–CC-339  
optimization termination CC-63  
optimization variables GS-135  
optimizations, structural CC-59, CC-138  
Options  
    Molecular Dynamics Options dia-  
        log box RM-369–RM-374  
Options, calculation, setting up HM-136  
Options, expert HM-107  
options, force field, set up CC-208  
Options, SCF HM-112  
Options, shielding & coupling HM-105  
Options, shielding and coupling HM-171  
Options, spectral, setting up HM-118, HM-136  
Orbital  
    described RM-404  
orbital CDK-227, CDK-228, CDK-230  
Orbital Criterion CC-41  
    Configuration Interaction dialog  
        box RM-309  
    practical application RM-310  
Orbital data, export RM-93  
orbital data, export RM-78  
Orbital data, import RM-78, RM-92  
orbital eigenvalues (energies)

in log file RM-632, RM-633  
orbital energy diagrams CC-230  
orbital exponents, default, table CC-280  
orbital occupancy CC-43  
orbital plots CC-253  
Orbital squared  
Orbitals dialog box RM-404  
Orbital, Orbitals dialog box RM-402  
Orbitals  
    Compute Menu RM-401–RM-405  
    Compute menu RM-358  
orbitals CDK-195, CDK-198, CDK-202, CDK-204  
    total number in log file RM-633  
Orbitals dialog box RM-401–RM-404  
Orbitals on 2nd row, d, option HM-172  
Orbitals, importing RM-92  
Orbitals, storing RM-93  
orientation, structure RM-82  
orienting 3D structure GS-153  
Origin, translate to, Edit menu RM-211  
oscillating electric dipole CC-369  
Oscillator Strength  
    Electronic Spectrum dialog box RM-414  
oscillator strength CDK-231  
Other  
    Restraint Forces dialog box RM-342  
Ousterhout, John CDK-95  
Outer radius, MM+ Options dialog box RM-293  
out-of-plane bending CC-195  
out-of-plane bending parameter file RM-673  
out-of-plane bends CC-221  
overlap integrals CC-262  
overlap matrix  
    in log file RM-632, RM-634  
overlap weighting factors, ZINDO CC-307  
Overlay on Display menu GS-167  
oxytocin, sample PDB file RM-507

## P

<sup>31</sup>P HM-81, HM-82, HM-111, HM-119, HM-178, HM-183, HM-185  
pack CDK-24, CDK-99, CDK-101, CDK-106, CDK-197  
package CDK-129  
Pan  
    Electronic Spectrum dialog box RM-414  
    Orbitals dialog box RM-402  
Pancreatic trypsin inhibitor complex, sample PDB file RM-507  
par parameter files RM-653, RM-661  
parallel spins CC-288  
parameter file path RM-103  
Parameter files HM-251  
parameter files RM-660–RM-661  
    compilation RM-353  
parameter set CDK-223  
parameter sets CC-205, RM-661–RM-664  
parameter sets, compiling CC-208  
parameter sets, new CC-207  
Parameterization, TNDO HM-240  
parameters, changed molecular mechanics CC-181  
parameters, default CC-190  
parameters, force field CC-176  
parameters, missing CC-215  
parameters, modifying CC-207  
Parameters, semi-empirical HM-186, HM-198, HM-251  
parameters, wild-card approach CC-215  
Part of a system, calculating HM-82, HM-107, HM-130  
Paste  
    copies from Clipboard to workspace RM-204–RM-205  
    cutting and pasting atoms RM-479–RM-481  
    Edit menu RM-201  
Paths RM-103

Pauli Exclusion Principle CC-35  
 pdb CDK-210  
 PDB Files  
     reading non-standard RM-81  
 PDB files, non-standard RM-504  
 PDB *See* Brookhaven Protein Data Bank  
 Peak information table HM-129  
 Peak information, plotting HM-129  
 Peak integration HM-127  
 Peak intensity, minimum level for inclusion  
     HM-190  
 Peak line intensity HM-149  
 Peak line location HM-149  
 Peak line widths HM-125  
 Peak selection HM-127  
 Peak width function HM-124, HM-156  
 Peak width, fine control HM-127, HM-191  
 Peaks HM-149  
 Peaks, spectrum HM-156  
 Peaks, zoom in HM-126  
 peptide  
     exploring alpha-helical nature of  
         GS-102  
 period, dynamics averaging CC-351  
 period, dynamics data collection CC-351  
 period, dynamics sampling CC-82  
 period, dynamics screen refresh CC-351  
 period, dynamics snapshot CC-351  
 periodic CDK-207, CDK-217  
 Periodic boundary conditions RM-336  
 periodic boundary conditions GS-159–GS-160,  
     CC-65, CC-210  
     cutoffs/dielectric adjustments GS-  
         161–GS-162  
 Periodic boundary conditions, dynamics RM-  
     372  
 periodic boundary conditions, theory CC-210  
 Periodic Box RM-335–RM-339  
     options, dialog box RM-337–RM-338  
     size, Periodic Box Options dialog  
         box RM-337  
 Periodic Box Options dialog box GS-158–GS-  
     160  
 periodic box, non-aqueous RM-339  
 periodic box, removing RM-339  
 periodic-box-size RM-539  
 permittivity, dielectric CC-188  
 perspective CDK-216  
 Perspective View  
     for Z translation RM-43  
 Perspective View on Display menu GS-72  
 Perturbation theory, self-consistent HM-237  
 Perturbation theory, TNDO HM-233, HM-236  
 perturbation, classical CC-277  
 perturbation, Møller-Plesset CC-41  
 perturbation, second-order RM-321  
 Petzold, Charles CDK-160  
 phase space CC-100  
 phases, trajectory CC-346  
 Phenylalanine (Phe), amino acid residue RM-  
     143  
 Phi, amino acid residue torsion angle RM-144  
 picture settings RM-99  
 Picture, copying HM-103, HM-154  
 pictures  
     files RM-100  
 planar molecules CC-334  
 PLANE  
     definition RM-22  
     Name Selection dialog box RM-238  
     Reflect through, Edit menu RM-208  
 Plane offset, 2D contour grid RM-395  
 playback CDK-226  
 playback dynamics GS-170  
 Playback, Options, Compute menu RM-373–  
     RM-374  
 Plot files RM-629  
     HyperChem files RM-620  
 Plot Molecular Properties RM-392  
     Compute Menu RM-391–RM-399,  
         RM-399–RM-400  
     Compute menu RM-358  
 Plot table data HM-170  
 Plotting coupling constants HM-138



Plotting peak information HM-129  
 Plotting results RM-399–RM-400  
 Plotting shielding constants HM-138  
 Plotting spectrum HM-139  
 PM3 CC-123, CC-132, CC-156, CC-249, CC-305–CC-306  
     background RM-302  
     parameter files RM-680  
     Semi-empirical Method dialog box RM-299  
 POINT CDK-55  
     definition RM-22  
     Edit menu RM-212  
     Invert through, Edit menu RM-206–RM-207  
     Name Selection dialog box RM-238  
     Translate to, Edit menu RM-210  
 Point  
     Restraint Forces dialog box RM-342  
 point charges CC-107, CC-256  
 Pointer rotate HM-157  
 Pointer select HM-157  
 Pointer Zoom & Translate HM-157  
 points, grid RM-393  
 points, status line report, geometry optimization RM-363  
 Polak-Ribiere CC-61  
 Polak-Ribiere conjugate gradient method RM-365  
 Polak-Ribiere optimization CC-338  
 Polarizability RM-458  
 polarizability CC-187  
 polarization functions CC-271  
 polarized basis sets CC-271  
 polynucleotide construction RM-497–RM-498  
 polypeptide creation GS-91–GS-98  
 polypeptide/polynucleotide backbone  
     Selection tool RM-245  
 pop CDK-211, CDK-213  
 Pople-Nesbet equations CC-237  
 Pop-up help for tool icons RM-27  
 potential  
     angle CC-23  
     electrostatic CC-27, CC-55  
     harmonic CC-24  
     Morse CC-24  
     non-bonded CC-26  
     nonbonded CC-29  
     torsional CC-25  
 potential energy surfaces CC-11, CC-12, CC-164  
 potential intermediates CC-137  
 potential switching function CC-29  
 potential, electrostatic CC-141  
 potential, shifting CC-30  
 Power Station CDK-3, CDK-152  
 Preferences RM-95–RM-105  
     dialog box RM-95–RM-105  
     on File menu RM-78  
     settings, using multiple windows RM-6  
 Preferences dialog box GS-52  
 Preferences on File menu GS-52  
 Preferences, color HM-165  
 Preferences, display HM-154  
 Primary axis, Align Molecules dialog box RM-217  
 primary inertial axis RM-21  
 primary structure  
     displaying GS-116  
 primitives, GTO CC-263  
 principal moments of inertia  
     in log file RM-638  
 Print  
     on File menu RM-78  
 print CDK-207, CDK-214, RM-523  
 Print level, log files HM-189, HM-210  
 Print Levels, Mechanics and Quantum RM-90  
 Printers icons and Control Panel RM-94  
 printing  
     molecular system RM-616  
     results RM-362

print-variable-list CDK-67, CDK-215  
problems, product support for GS-20  
procedures in Tcl CDK-98  
**PRODUCT**  
    Name Selection dialog box RM-238,  
        RM-239  
Product, reaction map list box RM-355  
product, transition state search RM-387  
products, reaction CC-144  
program file path RM-103  
program files RM-621–RM-622  
Program files, HyperNMR HM-186  
programming CDK-95, CDK-100, CDK-149,  
    CDK-155, CDK-159, CDK-173  
Projected CNDO/INDO MO coefficients CC-  
    277  
Projected Hückel MO coefficients CC-277  
Proline (Pro), amino acid residue RM-143  
propene  
    creating GS-57–GS-65  
Properties  
    Element Table dialog box RM-122  
properties, element RM-123, RM-475  
Protein Data Bank CDK-50  
Protein Data Bank files RM-18, RM-503–RM-  
    507  
    creation RM-504–RM-507  
    open RM-80  
    options RM-85  
    samples RM-506–RM-507  
    save RM-84  
proteins, construction RM-493–RM-495  
protocol CDK-13, CDK-108, CDK-137  
protocol, network RM-104  
proton affinities CC-137  
protonated nitrogens GS-214  
protonation of water GS-221–GS-226  
Pseudo-atoms HM-82, HM-132  
pseudo-atoms GS-213–GS-214, RM-343  
pseudo-halogens CC-258–CC-260  
Psi, amino acid residue torsion angle RM-144  
push CDK-211, CDK-213

## Q

QCPE CDK-191  
quadratic stretch CC-176, CC-191  
quadruple zeta CC-271  
quadrupole RM-692  
Quantum atoms HM-82  
Quantum atoms, selecting HM-107, HM-130,  
    HM-132, HM-173  
quantum mechanics CDK-227  
    ab initio CC-31  
    semi-empirical CC-32  
Quantum Print Level RM-90  
quantum-atoms selection in HyperChem HM-  
    108, HM-133, HM-175  
QuantumPrintLevel, in chem.ini RM-415, RM-  
    631  
quartet CC-242  
quasi-Newton CC-341  
quenched dynamics CC-80  
query-response-has-tag CDK-29, CDK-45,  
    CDK-101  
query-value CDK-29, CDK-50, CDK-99,  
    CDK-212  
Quitting HyperNMR HM-104, HM-154

## R

rA, nucleotide residue RM-159  
radicals, free CC-242  
Radii, atom rendering HM-189, HM-193  
Raffenetti integral format CC-118, CC-274,  
    RM-322  
Random seed, Monte Carlo and Langevin RM-  
    373  
random velocities CC-345  
Rayleigh-Schrödinger many-body perturbation  
    (RSPT) CC-41, CC-246  
rC, nucleotide residue RM-159  
**REACTANT**  
    Name Selection dialog box RM-238  
Reactant, reaction map list box RM-354  
reactant, transition state search RM-387  
reaction barriers CC-33  
reaction coordinate CC-14, CC-33, CC-144

reaction coordinate parameter (lambda) CC-69  
 Reaction Map CC-69, RM-353  
 reaction mechanisms CC-137, CC-339  
 reaction products CC-144  
 reactions, chemical CC-360–CC-363  
 reactivity, chemical CC-144  
 reactivity, ionic CC-146  
 read CDK-155  
 Read file HM-153, HM-163  
 read/write CDK-28, CDK-66  
 reading HSV CDK-29  
 reading PDB files RM-503  
 read-script CDK-44, CDK-211  
 read-tcl-script CDK-25, CDK-44, CDK-127,  
 CDK-211  
 real number CDK-52  
 real number, script arguments for scripts RM-  
 517  
 Recompute H Bonds, Display menu GS-206  
 rectangular selection GS-107  
 recursive scripts CDK-54  
 Reference shielding HM-119, HM-158, HM-  
 184, HM-192  
 Reflect RM-208  
     on Edit menu RM-202, RM-486  
 Reflect on Edit menu GS-138  
 reflection plane  
     defining GS-137  
 Refractivity RM-457  
 registering of HSV CDK-27  
 relaxation constant, bath CC-74  
 relief ridge on label CDK-103  
 REMARK, creating PDB-type files RM-505  
 remeasuring axial hydrogens GS-141  
 Remove Ionic Ends RM-146  
     Databases menu RM-140  
 Remove, Restraint Forces dialog box RM-341  
 removing atoms RM-473  
 render CDK-216, CDK-232  
 Rendering  
     dialog box GS-39–GS-43  
 Rendering menu HM-155  
 Rendering options HM-94, HM-144, HM-167  
 Rendering options, atom HM-168  
 Rendering options, bond HM-168  
 rendering options, isosurface RM-396  
 Rendering, atom colors HM-188, HM-192  
 Rendering, atom radii HM-189, HM-193  
 Rendering, atoms HM-96  
 Rendering, bonds HM-99  
 Reopen HyperChem GS-57  
 reopening  
     HIN file GS-69  
 Replace selection HM-167  
 Replace table cells HM-154  
 repulsion, exchange CC-182  
 request to HyperChem CDK-35  
 REQUEST, Excel message RM-606  
 res, HIN file syntax RM-716  
 Rescale view HM-157  
 Rescale, Molecular Dynamics Results dialog  
     box RM-382  
 residue CDK-224  
 Residue name and number, Select menu RM-  
 226  
 residue template files RM-619, RM-727–RM-  
 741  
 residues RM-225–RM-226  
     amino acids RM-735–RM-737  
     bonded atoms RM-17  
     conformation RM-144  
     files RM-739  
     molecules built from RM-492–RM-  
     501  
     mutating nucleic acids RM-499–RM-  
     501  
     nucleic acids RM-737–RM-738  
     numbering RM-237  
     replacing GS-94–GS-98  
     Selection level RM-32  
     template files RM-727–RM-741  
     unusual RM-496, RM-501  
 Residues, on Select menu RM-221, RM-496  
 Resolution, spectrometer HM-183

resonance integrals CC-307  
 restart velocities CC-345  
 Restart, Molecular Dynamics Options dialog box RM-374  
 Restore button HM-144  
 restore button, HyperChem Windows RM-5  
 Restrained Value  
     Restraint Forces dialog box RM-341  
 restraint CDK-227  
 Restraint Forces dialog box RM-340–RM-343  
     named selections RM-241  
 restraint, orientation of coordinated system RM-343  
 Restraints RM-340–RM-343  
 restraints  
     calculations RM-341–RM-343  
     definition RM-340  
     Restraint Forces dialog box RM-340  
 restraints, structural CC-83  
 restraints, theory CC-212  
 Restricted Hartree-Fock (RHF) CC-38, CC-116, CC-240–CC-244  
 Restricted Hartree-Fock calculation GS-183  
 Restricted Open-shell Hartree-Fock (ROHF) CC-116  
 results  
     averages (CSV) file RM-629  
     from HyperChem RM-611–RM-618  
     stored in Log file RM-616–RM-617  
 Results, copying to Clipboard HM-137  
 Results, detailed output HM-210  
 Results, log file print level HM-189  
 Results, saving HM-118  
 rG, nucleotide residue RM-159  
 RGB Bitmap file format, Setup Image RM-100  
 RHF CDK-56, CDK-203, CC-38, CC-46, CC-116, CC-237, CC-240–CC-244  
 RHF half-electron technique CC-48  
 RHF *See* Restricted Hartree-Fock  
 RHF, Spin pairing, Semi-empirical Options dialog box RM-307  
 rhinovirus, sample PDB file RM-507  
 ribbons CDK-217  
 Ridley, J.E. and Zerner, M.C.  
     and ZINDO/S RM-680  
 ring structures  
     finding and selecting GS-121  
 rings  
     making aromatic GS-62, GS-82  
     selection GS-111  
 rings, atom typing syntax RM-659  
 rings, multiple aromatic RM-131  
 RMS Gradient CC-332  
 RMS gradient CC-63  
 RMS gradient, Termination condition, geometry optimization RM-366  
 RMS gradient, Termination condition, transition state search RM-387  
 RNA/DNA construction RM-497–RM-498  
 ROHF CC-116  
 Roothaan equations CC-235, CC-261  
 root-mean-square gradient CC-63  
 Rotate  
     dialog box RM-209–RM-210  
     Edit menu RM-202  
 rotate CDK-215  
 rotate example CDK-43  
 rotating a side chain GS-124–GS-126  
 rotating molecules GS-69–GS-80  
 Rotating structures HM-89, HM-101, HM-157, HM-161  
 rotation CDK-208  
 rotation speed RM-97, RM-98  
 rotation, group RM-39  
 rotation, side chain RM-39  
 rotational barriers CC-139  
 Rotational equivalence HM-83, HM-110, HM-173, HM-176  
 RSPT CC-41, CC-246  
 rU, nucleotide residue RM-159  
 rules, atom typing CC-178  
 Run Script dialog box RM-106  
 Run steps, Monte Carlo RM-371  
 Run time, Molecular Dynamics Options dialog

box RM-370  
Running HyperNMR HM-85, HM-142  
running the calculation GS-136  
running, trajectory CC-346  
runtime CDK-149, CDK-151

## S

saddle point CC-67  
    definition GS-129  
saddle points CC-17, CC-33  
sample PDB files RM-506  
sampling interval CC-82  
save a structure GS-97  
Save As  
    on File menu RM-77  
    fill PDB records, setting in  
        chem.ini/Registry RM-782  
    stores molecular system in file RM-  
        83  
Save file HM-153, HM-164  
Save File dialog box GS-59, RM-83–RM-85  
Save on File menu GS-59  
Save, File menu GS-58–GS-60  
Save, on File menu RM-77  
saving  
    color images RM-614  
    files in HIN format GS-59–GS-60  
    molecular system in HIN file RM-19  
    recalling selections, Selection tool  
        RM-34  
    results RM-362  
    structures GS-67  
Saving molecules HM-118  
scalar HSV CDK-31  
Scale factor, Force Field Options dialog box  
    RM-294  
Scale to fit HM-157  
Scale to Fit on Display menu GS-74  
scaling molecules GS-69–GS-80  
Scaling structure HM-101  
SCF CC-39, CC-234

SCF calculation options HM-112, HM-173,  
    HM-181  
SCF calculation, direct CC-275  
SCF convergence CC-49  
SCF Technique CC-45  
SCF. *See* Self-Consistent Field  
Schrödinger equation CC-11, CC-171, CC-227  
scr file CDK-22, CDK-50  
Screen refresh period, Molecular Dynamics  
    Options dialog box RM-367, RM-373  
script CDK-33  
    arguments CDK-30  
    initialization CDK-53  
    instantiation CDK-54  
    recursive CDK-54  
script argument CDK-30  
script command CDK-205  
script commands CDK-4, CDK-7, CDK-22,  
    CDK-49, CDK-67  
script editor CDK-54  
script file CDK-22, CDK-52  
script file path RM-103  
script menu CDK-37  
Script menu, enabling menu items RM-107  
Script One . . . Script Ten RM-106  
    Script Menu RM-105  
Script, startup RM-4  
scripts RM-509–RM-584, RM-602–RM-609  
    arguments RM-516–RM-517  
    commands RM-516  
    cross reference of RM-579  
    input messages RM-510–RM-511  
    menu RM-105–RM-107  
    menu activators RM-517  
    online Help RM-115  
    syntax RM-511–RM-517  
    variables RM-510, RM-511–RM-516  
Scroll bar, line width HM-148  
scrolling, list of files GS-34  
SDK CDK-10, CDK-159, CDK-173  
second order optimizer CC-62

Secondary axis RM-21  
   Align Molecules dialog box RM-217  
 second-order perturbation RM-321  
 seed, HIN file syntax RM-715  
 Seed, random, Monte Carlo and Langevin RM-373  
 select RM-232–RM-237  
   a group of atoms, Selection tool RM-34  
   a polypeptide/polynucleotide backbone, Selection tool RM-35  
   a rotatable side chain, Selection tool RM-35  
   rings, Selection tool RM-34  
   side chains, Selection tool RM-34  
 Select All RM-231  
   atoms to the sp<sup>3</sup>-sp<sup>3</sup> bonds, Selection tool RM-35  
   Select menu RM-221  
   structures, Selection tool RM-35  
 select atom  
   HIN file syntax RM-722  
   molecules by name/number, Selection tool RM-34  
 Select Backbone RM-245  
   in any molecule, Selection tool RM-33  
   in residue polymer RM-35, RM-246  
   Select menu RM-222  
 Select Backbone on Select menu GS-117  
 Select dialog box RM-233–RM-237  
 Select menu RM-220–RM-246  
   Complement Selection GS-106, GS-115, GS-166  
   Multiple Selections GS-54  
   Select GS-51  
   Select Backbone GS-102, GS-117  
   use RM-34–RM-35  
 Select on Select menu GS-51  
 Select Parameter Set RM-351–RM-352  
 Select Parameter Set on Setup menu GS-202  
 Select Sphere RM-227–RM-230  
   Select menu RM-221  
 select then operate CDK-59  
 selecting CDK-66  
   a file GS-35–GS-37  
   atoms GS-51–GS-53  
   backbone GS-102–GS-103  
   groups of atoms GS-53–GS-55  
   ring GS-111  
   ring structures GS-121  
   side chains GS-111–GS-112  
   subsets GS-99–GS-112  
 Selecting atoms HM-102, HM-157, HM-161  
 selecting by name/number, atoms RM-236–??  
 Selection  
   HIN file syntax RM-720  
   icon, mouse clicks RM-31  
   Molecular Dynamics Averages dialog box RM-377–RM-379  
   Restraint Forces dialog box RM-341  
 selection CDK-55, CDK-205, CDK-209, CDK-218  
 Selection Color RM-96  
 selection level  
   setting GS-51  
 Selection rules HM-249  
 selection scripts CDK-209  
 Selection tool RM-30–RM-31  
   measuring RM-35–RM-36  
 Selection, peak HM-127  
 selectrestraint  
   HIN file syntax RM-720–RM-722  
   setting flag for RM-721–RM-722  
 Self Consistent Field options HM-173, HM-181  
 Self-Consistent Field  
   on Setup menu RM-298  
 Self-Consistent Field (SCF) CC-39, CC-234,

CC-261  
 Self-consistent perturbation theory HM-237  
 Semi-empirical  
   Options dialog box RM-305–RM-307  
 semi-empirical  
   for quantum mechanical calculations RM-296–??, RM-296–RM-297  
   method RM-361  
   mixed mode RM-311  
   parameter files RM-679–RM-699  
 Semi-empirical dialog box RM-298–RM-300  
 Semi-empirical method HM-171  
 semi-empirical method, selection CC-134, CC-153  
 semi-empirical methods CDK-228  
 Semi-empirical methods, conventional HM-226  
 Semi-empirical methods, typed HM-228  
 Semi-empirical on Setup menu GS-182, GS-206, GS-215  
 Semi-empirical Options (Hückel) dialog box RM-304–RM-305  
 Semi-empirical parameters HM-186, HM-198  
 semi-empirical quantum mechanics CC-32  
 sensitivity, rotation and translation RM-97, RM-98  
 serial number RM-115  
 Serine (Ser), amino acid residue RM-142  
 server CDK-16, CDK-35, CDK-108  
 set CDK-97  
 Set Atom Type RM-130–RM-131  
   Build menu RM-119  
   dialog box RM-131  
 Set Atom Type on Build menu GS-85  
 Set Bond  
   Angle dialog box RM-219–RM-220  
   Length dialog box RM-218–RM-219  
   Torsion dialog box RM-220  
 Set Bond Angle, Edit menu RM-202  
 Set Bond Length, Edit menu RM-202  
 Set Bond Torsion, Edit menu RM-202  
 Set Charge RM-132–RM-133  
   Build menu RM-119  
   dialog box RM-133  
 Set Charges on Build menu GS-85, GS-182  
 Set HSV  
   types of input messages RM-510  
   variables, script messages syntax RM-511–RM-512  
 Set Mass RM-132  
 Set Velocity dialog RM-344  
 setting up graphics transfer RM-612–??  
 setting up restraints RM-342  
 settings  
   AMBER RM-749–RM-751  
   CHARMM RM-755–RM-758  
   display conditions RM-611  
   MM+ RM-769–RM-771  
   OPLS RM-773–RM-775  
   Registry or chem.ini default RM-743–RM-785  
 Settings, HyperNMR default HM-186, HM-187  
 Settings, WIN.INI HM-78  
 Setup Image preferences RM-99  
 Setup Menu HM-157  
 Ab initio  
   on Setup menu RM-288  
 Compile Parameter File  
   on Setup menu RM-289  
 Network  
   on Setup menu RM-288  
 Periodic Box  
   on Setup menu RM-288  
 Reaction Map  
   on Setup menu RM-289  
 Restraints  
   on Setup menu RM-288  
 Select Parameter Set  
   on Setup menu RM-289

Semi-empirical  
     on Setup menu RM-287  
 Set Velocity  
     on Setup menu RM-288  
 Setup menu RM-287-??  
     Molecular Mechanics GS-201, GS-205  
     script messages RM-545–RM-555  
     Select Parameter Set GS-202  
     Semi-empirical GS-182, GS-206, GS-215  
 Setup, calculation HM-105  
 Setup, software HM-77  
 sextic angle bending CC-193  
 Shaded isosurface RM-398  
 Shielding & coupling calculation options HM-157, HM-171  
 Shielding and coupling calculation options HM-105  
 Shielding constant view HM-145  
 Shielding Constants, calculating HM-113  
 Shielding constants, calculation option HM-106  
 Shielding constants, plotting HM-138  
 Shielding constants, viewing HM-115, HM-137, HM-155  
 Shielding tensor display HM-168  
 Shielding tensor, calculate HM-191  
 Shielding tensor, displaying HM-156  
 Shielding tensors HM-116  
 Shielding, magnetic HM-216  
 Shielding, reference HM-119, HM-158, HM-184, HM-192  
 Shifted cutoffs, MM+ Options dialog box RM-293  
 shifted function CC-108  
 shifting function CC-189  
 shifting potential CC-30  
 Shortcut  
     clear workspace HM-159  
     coupling constants HM-115, HM-117, HM-160  
     file open HM-159  
     file save HM-118, HM-160  
     get structure HM-94, HM-136, HM-160  
     rotation HM-102, HM-160  
     selection HM-103, HM-160  
     shielding constants HM-115, HM-116, HM-160  
     translate and zoom HM-101, HM-160  
     view spectrum HM-122, HM-123, HM-160  
     view structure HM-160  
 Show All on Display menu GS-105  
 Show Axes on Display menu RM-21, RM-216  
 Show Hydrogens on Display menu GS-66  
 Show Perspective RM-512  
 Show Selection Only on Display menu GS-117  
 Show Stereo  
     setting in chem.ini/Registry RM-782  
     WallEyed Stereo RM-782  
 side chain rotation RM-39  
 side chain selection GS-212–GS-213  
 side chains  
     displaying GS-105, GS-117  
     rotating GS-124–GS-126  
     selecting GS-111–GS-112  
 side groups syntax, atom typing RM-659  
 simulated annealing GS-168–GS-170, CC-81, CC-360  
 Simulation temperature, Molecular Dynamics Options dialog box RM-372  
 Single Point  
     calculation RM-362  
     calculations GS-134, GS-155  
     Compute Menu RM-360–RM-362  
     Compute menu RM-357  
     on Compute menu GS-134  
     single point calculation CC-16, CC-53, CC-331  
     single point calculations CDK-206, CC-125  
     singlet CC-116, CC-229, CC-242



singlet ground states CC-39  
 singlet molecules CC-38  
 singlet-triplet splittings CC-40  
 singly excited CI CC-40  
 Singly Excited CI Method  
     Configuration Interaction dialog  
         box RM-309  
         practical application RM-310  
 singly occupied orbitals  
     in log file RM-632, RM-633  
 site-specific mutagenesis GS-94–GS-98, GS-123  
 skills  
     creating a polypeptide GS-91–GS-98  
     creating small molecules GS-57  
     drawing/editing techniques GS-45  
     electronic properties of proteins GS-209–GS-217  
     geometry optimization GS-129  
     getting started GS-27  
     lowest excited electronic state of ethylene GS-209–GS-217  
     measuring structural properties GS-81  
     molecular dynamics in water GS-147  
     molecular orbital calculations GS-179–GS-196  
     protonation of water GS-221–GS-226  
     translating/rotating/scaling molecules GS-69–GS-80  
     vibrations and transition states of ammonia GS-227–GS-239  
     water–N-methylacetamide interaction GS-197  
     working with macromolecules GS-113  
     working with subsets GS-99  
 Slab, Z Clip dialog box RM-214  
 Slater determinants (microstates)  
     in log file RM-634  
 Slater exponents for shielding, option HM-107, HM-172  
 Slater Type Orbitals (STO) CC-45, CC-263  
 Slater-Condon parameters CC-290  
 slave CDK-13  
 slice of molecule *See* clipping slab  
 Slow three-center integrals, option HM-172  
 Smallest box enclosing solute, Periodic Box Options dialog box RM-337  
 smooth shaded isosurface RM-398  
 snapshot  
     files RM-623  
         HyperChem files RM-620  
         Molecular Dynamics Playback dialog box RM-376  
         SNP files RM-623–RM-629  
 Snapshot dialog box RM-375  
 snapshot files CC-357–CC-359  
 snapshot, dynamics period CC-351  
 snapshots GS-171  
 Sodium, counter ions RM-499  
 soft electrophiles and nucleophiles CC-146  
 solid color isosurface RM-398  
 solute temperature CC-77  
 solute-solute interactions CC-65  
 solute-solute interactions GS-159  
 solvating  
     alanine zwitterion GS-158  
     carboxyl group GS-203–GS-204  
     molecule optimization GS-162–GS-163  
 solvation CDK-207, CC-65  
 Solvent effects HM-185  
 solvent effects CC-65, CC-210  
 solvent friction and diffusion CC-95  
 solvent temperature CC-77  
 solvents, non-aqueous RM-339  
 source command CDK-133  
 Space bar HM-101, HM-157  
 spacebar

for Rescale, Molecular Dynamics  
     Results dialog box RM-382  
 Special Slater exponents for shielding HM-107  
 Specifying atoms HM-102  
 spectra CDK-231  
 Spectra, calculation HM-118  
 Spectra, envelope view HM-123  
 Spectra, line view HM-122  
 Spectra, plotting HM-139  
 Spectra, types HM-81, HM-183  
 spectra, UV-visible, saving RM-412  
 spectra, vibrational, saving RM-386, RM-408  
 Spectra, viewing HM-122, HM-139, HM-148,  
     HM-156  
 Spectral options HM-158  
 Spectral options, setting up HM-136, HM-182  
 Spectral values HM-129, HM-139  
 Spectral values, displaying HM-156, HM-169  
 Spectrometer frequency HM-183  
 speed, translation and rotation RM-97, RM-98  
 Spheres rendering GS-40–GS-41, HM-97  
 Spheres, Rendering dialog box  
     settings (CPK) in chem.ini/Registry  
         RM-781  
 spherical selection GS-108–GS-110  
 Spin coupling, nuclear HM-222  
 spin couplings CC-40  
 Spin density RM-392  
 spin density CC-55, CC-125, CC-253  
 spin density, plotting GS-189  
 spin interactions CC-291  
 spin multiplicities  
     in log file RM-633  
 Spin multiplicity HM-181  
     Semi-empirical Options (Hückel)  
         dialog box RM-304  
     Semi-empirical Options dialog box  
         RM-305  
 spin multiplicity CC-46, CC-116  
 Spin multiplicity, Electronic Spectrum dialog  
     box RM-414  
 spin pairing CC-236  
 Spin pairing, Semi-empirical Options dialog  
     box RM-306  
 spin state CC-229, CC-242  
 Spin wave functions HM-247  
 spins, parallel or anti-parallel CC-288  
 spins, unpaired CC-38  
 Spin-spin coupling constants, option HM-106  
 splittings, singlet-triplet CC-40  
 split-valence basis set, table of KLM values  
     CC-268  
 split-valence basis sets CC-268  
 spreadsheet CDK-107, CDK-115  
 spreadsheet program, Microsoft Excel RM-606  
 SPX/IPX RM-104  
 Square Planar  
     Constrain Geometry dialog box RM-  
         135  
 star in basis set name CC-271  
 Start Log  
     on File menu RM-78  
     stores chemical calculations RM-90  
 Start log file HM-154, HM-165  
 Start time, Molecular Dynamics Playback dia-  
     log box RM-376  
 starting HyperChem GS-27–GS-28, RM-3–  
     RM-4  
 Starting HyperNMR HM-85, HM-142  
 Starting temperature, Temperature, Compute  
     menu RM-371  
 Starting value, 2D contour grid RM-396  
 Startup script RM-4  
 state energies  
     in log file RM-633  
 State, electronic HM-181  
 state, excited CC-242  
 state, lowest CC-242  
 State, Semi-empirical Options dialog box RM-  
     306  
 static field, classical CC-256, CC-277  
 static linkage CDK-150  
 stationary point  
     definition GS-129

stationary points CC-17  
 Stationary-state wave functions HM-248  
 statistical mechanical averages, dynamics CC-345  
 Status bar HM-144  
 Status line HM-87  
 status line GS-29  
 status line report RM-363  
 status line, HyperChem Windows RM-6  
   atom report RM-224  
   molecule report RM-227  
   residue report RM-226  
 steepest descent CC-60  
 steepest descent on potential energy surface RM-364  
 steepest descent optimization CC-126, CC-335  
 Step count, Molecular Dynamics Playback dialog box RM-376  
 Step size  
   Molecular Dynamics Options dialog box RM-371  
   Molecular Dynamics Playback dialog box RM-376  
 step size, molecular dynamics CC-92  
 stereo CDK-216, CDK-217  
 stereochemistry CDK-219, RM-482  
   displaying stereobonds with wedges RM-482–RM-485  
   *See also* chirality  
 stereochemistry, changing GS-235  
 Stewart, J.J.P.  
   and PM3 RM-680  
 STO CC-45, CC-263  
 STO-3G exponents, table CC-267  
 STO-NG method CC-265  
 Stop Log RM-91, RM-616  
   on File menu RM-78  
 Stop log file HM-154  
 Stopping HyperNMR HM-104, HM-154  
 storing results  
   in Log file RM-616–RM-617

## Molecular Dynamics Simulation RM-617–RM-618

str parameter files RM-660  
 stretch-bend cross term CC-106  
 stretch-bend interactions CC-194, CC-221  
 stretching parameters  
   in log file RM-638  
 string CDK-30, CDK-51, CDK-97, CDK-146  
 string arguments, script messages RM-517  
 structural optimizations CC-138  
 structural properties  
   comparing GS-136–GS-137  
   measuring GS-133, GS-154  
 structural restraints CC-83  
 structure  
   chem.tpl RM-727–RM-741  
   HIN file RM-711  
   TPL file RM-727–RM-735  
 structure alignment GS-180–GS-181  
 Structure copy HM-154  
 Structure input HM-75, HM-81, HM-89  
 Structure rendering options HM-167  
 Structure rotation HM-161  
 Structure selection HM-161  
 Structure translation HM-161  
 structures, saving GS-67  
 Subset, calculating HM-107, HM-130  
 subsets, complementing GS-106  
 Sugar Form, polynucleotide strand conformation RM-161  
 superposing  
   molecules GS-163–GS-168  
 Support GS-20, GS-21  
 Switched cutoffs, MM+ Options dialog box RM-293  
 switching function CC-108, CC-189  
 switching function, potential CC-29  
 Symbols, displaying atomic HM-96  
 symmetrical systems CC-41  
 Symmetry equivalence HM-83, HM-110, HM-173, HM-176  
 Symmetry, Orbitals dialog box RM-403

synchronous transit GS-234, CC-69, CC-127, CC-341

Synchronous transit, transition state search RM-387

syntax

- atom typing RM-656–RM-659
- HIN files RM-712–RM-722, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-723, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-724, ??–RM-725, ??–RM-725, ??–RM-725
- script messages RM-511
- TPL files RM-729–RM-735

sys, HIN file syntax RM-714

System requirements HM-77

System, Z Clip dialog box RM-214

## T

Table cells, replace HM-154, HM-167

Table data, chart HM-170

Table, peak information HM-129

tags in TPL files RM-732–RM-733

tags, atomic coordinate RM-734

Tcl CDK-23, CDK-55, CDK-95

- procedures CDK-98

Tcl books CDK-96

Tcl commands CDK-97

Tcl file CDK-24, CDK-127, CDK-133

Tcl resources CDK-96

Tcl script CDK-99, CDK-102, CDK-132, CDK-196

Tcl/Tk CDK-6, CDK-95, CDK-127

Tcl/Tk interpreter CDK-127, CDK-149

Tcl/Tk script CDK-2, CDK-6, CDK-9, CDK-22, CDK-45, CDK-157

TclOnly CDK-24, CDK-45, CDK-98

TCP/IP RM-104

TEMP, Selection, Molecular Dynamics Averages dialog box RM-378

temperature CDK-225

temperature regulation CC-73

Temperature step, Molecular Dynamics Options dialog box RM-372

temperature, constant CC-74, CC-80, CC-90

temperature, constant dynamics CC-347, CC-349

temperature, dynamics CC-343

Temperature, Molecular Dynamics Options dialog box RM-371

temperature, reaction simulation CC-362

temperature, solvent CC-77

template CDK-115

template file CDK-110

template files RM-727–RM-741

- bonding information RM-733
- syntax RM-729–RM-735

Tensor, display shielding HM-156, HM-168

Tensor, shielding, calculate HM-191

Tensors, shielding HM-116

TERMINATE, Excel message RM-606

Termination condition, geometry optimization RM-366

termination condition, optimization CC-63

tertiary axis RM-21

- Align Molecules dialog box RM-217

Tether to

- Restraint Forces dialog box RM-342

Tetrahedral

- Constrain Bond Angle dialog box RM-138
- Constrain Geometry dialog box RM-135

text communication CDK-140

text object in VB CDK-122

Text, Help HM-186

THAPI CDK-128, CDK-129, CDK-130

Theory HM-213–HM-252

Three-center integrals, include option HM-172

three-dimensional group selection RM-230

Threonine (Thr), amino acid residue RM-143

Time

- Compute menu RM-370

on status line RM-373  
 time periods, Molecular Dynamics simulation  
     RM-382–RM-469  
 time step, trajectory CC-346  
 timeout CDK-131, CDK-148  
 TIP3P water model CC-65  
 TIP3P water molecules RM-336  
 title CDK-207  
     of window CDK-47  
 Title bar HM-87, HM-143  
 title bar GS-28  
 title bar, HyperChem Windows RM-5  
 Tk CDK-6, CDK-23, CDK-55, CDK-129  
 Tk dialog box CDK-47, CDK-201  
 TK window CDK-134  
 TNDO HM-225  
 TNDO parameterization HM-240  
 TNDO perturbation theory HM-233, HM-236  
 TNDO/1 HM-171, HM-214, HM-229  
 TNDO/2 HM-172, HM-214, HM-230  
 Tool bar HM-87, HM-144  
 tool bar GS-29  
 Tool Buttons HM-159  
 tool command language CDK-6  
 Tool Menu HM-157  
 Tool preferences RM-97  
 Toolbar  
     clear workspace HM-159  
     coupling constants HM-115, HM-117,  
         HM-160  
     file open HM-159  
     file save HM-118, HM-160  
     get structure HM-94, HM-136, HM-160  
     rotation HM-102, HM-160  
     selection HM-103, HM-160  
     shielding constants HM-115, HM-116,  
         HM-160  
     translate and zoom HM-101, HM-160  
     view spectrum HM-122, HM-123, HM-  
         160  
     view structure HM-160  
 toolbar CDK-47, CDK-207  
 Toolkit CDK-9, CDK-95  
 tools RM-27–RM-50, RM-61–??  
     Help menu RM-115  
     icons  
         HyperChem Windows RM-5  
         mouse clicks RM-27  
     popup help RM-27  
     XY rotation GS-75  
     XY translation GS-70–GS-71  
     Z translation GS-72  
     Zoom GS-73  
 Top level setting, choosing extent of Copy Im-  
     age RM-612  
 Top level window, Setup Image RM-100  
 topic  
     DDE CDK-108  
 torsion  
     bond constraint GS-141  
 torsion angles  
     measuring GS-87–GS-88  
 torsion angles and multiple aromatic rings RM-  
     131  
 torsion parameter file RM-669–RM-670  
 torsional angle rotation CC-182  
 torsional angles CC-183  
 torsional energy CC-195, CC-198, CC-203,  
     CC-221  
 torsional potential CC-25  
 torsion-angle, atomic coordinate in TPL file  
     RM-735  
 torsions, improper CC-184  
 Total charge  
     density RM-393  
     Semi-empirical Options (Hückel)  
         dialog box RM-304–RM-305  
     Semi-empirical Options dialog box  
         RM-305  
 total charge density, plotting GS-188  
 total density matrix

in log file RM-632  
total electron density CC-54  
total energy  
    in log file RM-632, RM-633  
Total Energy, on status line RM-373  
Total spin density RM-392  
total spin density CC-55  
total spin density, plotting GS-189  
total-energy CDK-206  
TPL. *See* template files  
trajectory phases CC-346  
trajectory, classical CC-360  
Trans, Constrain Bond Torsion dialog box RM-139  
transfer RNA, sample PDB file RM-507  
transferring graphics RM-612-??  
transforming from chair to boat cyclohexane GS-137-GS-138  
transit, synchronous, transition state search RM-387  
transition dipole CC-369  
Transition intensities HM-249  
Transition State RM-386  
    **Compute** menu RM-358  
transition state search CC-17, CC-67, CC-127, CC-340  
transition state searching GS-233, GS-234  
transition state setup RM-353  
transition states CC-33, CC-138  
Transition Structures CC-339  
Transition, Electronic Spectrum dialog box RM-414  
transitions, selecting in Electronic Spectrum dialog box RM-414  
Translate  
    dialog box RM-210-RM-212  
    **Edit** menu RM-202  
translate CDK-216  
Translate dialog box GS-71-GS-72  
Translate Selection, Translate dialog box RM-211  
Translate Viewer, Translate dialog box RM-

211  
Translating a molecule HM-89  
translating molecules GS-69-GS-80  
Translating structure HM-161  
Translating structures HM-100, HM-157  
translation CDK-208  
translation speed RM-97, RM-98  
translation, fast RM-98  
Translucent surface RM-398  
Transparency Level, isosurface RM-398  
Transparent isosurface RM-398  
Trigonal  
    **Constrain Bond Angle** dialog box RM-138  
    **Constrain Geometry** dialog box RM-135  
triple zeta CC-271  
triplet CC-116, CC-229, CC-242  
triplet ground states CC-47  
triplet open-shell ground states CC-39  
Tripos MOL2 Files  
    save RM-84  
Tripos MOL2 Files  
    open RM-80  
trypsin inhibitor complex, sample PDB file RM-507  
Trypsin, sample PDB file RM-507  
Tryptophan (Trp), amino acid residue RM-143  
tutorial  
    **Ab Initio** calculations GS-219  
    **HyperChem** calculations GS-127-GS-128  
Tutorials HM-85-HM-140  
twist boat cyclohexane GS-141  
    **optimization** GS-143  
twist boat structure minimization GS-143  
two-dimension selection RM-227-RM-228  
two-electron integral cutoff CC-117  
two-electron integrals CC-272  
    format CC-118  
    **Raffenetti** format CC-118

txt files, parameters CC-206  
 txt force field parameter files RM-653  
 Type  
     Parameter file RM-665  
 type 1 CDK-9, CDK-22, CDK-49  
 type 2 CDK-9, CDK-22, CDK-95  
 Type files, atom HM-194  
 Type, atom HM-186  
 Typed Neglect of Differential Overlap (TNDO)  
     HM-214, HM-225  
 Typed semi-empirical methods HM-228  
 typerule.bin force field file RM-653  
 types, atom CC-176, CC-177  
 Types, atom, setting HM-105  
 Types, calculate atom HM-155  
 Types, displaying atomic HM-96  
 typing rules for atoms RM-655  
 Typing rules, atom HM-194  
 typing rules, atom CC-178  
 Tyrosine (Tyr), amino acid residue RM-143  
**U**  
 UHF CDK-56, CDK-227, CC-38, CC-46, CC-116, CC-237, CC-240–CC-244  
 UHF, Spin pairing, Semi-empirical Options dialog box RM-306  
 ultra violet CDK-231  
 UMSG CDK-15, CDK-33  
 United Atom force fields CC-28  
 United Atoms RM-128–RM-129  
     Build menu RM-118  
 united atoms CC-178, CC-197, CC-201, CC-202  
 units, force constants CC-191  
 UNIX CDK-1, CDK-17, CDK-18, CDK-107, CDK-129, CDK-155, CDK-188  
 unloading CDK-124  
 Unoccupied Orbital Criterion  
     Configuration Interaction dialog box RM-309  
 unpaired spins CC-38  
 Unrestricted Hartree-Fock (UHF) CC-38, CC-116, CC-240–CC-244  
 Unweighted constant, Semi-empirical Options (Hückel) dialog box RM-304  
 update CDK-45, CDK-47  
 Urey-Bradley term CC-106  
 user interface CC-161  
 user interface, HyperChem GS-25, GS-28–GS-35  
 users list GS-21  
 Using Help, Help menu RM-115  
 using HyperChem script messages RM-510–RM-517  
 utilities RM-77–RM-78  
 UV spectrum, importing RM-92  
 UV spectrum, storing RM-93  
 UV visible spectra  
     and ZINDO/S RM-300  
     computing  
         practical applications RM-414–RM-415  
         saving information in log file RM-415  
     in log file RM-633  
 UV-visible spectra CC-40, CC-129  
     computing CC-363–CC-364  
     theory CC-363–CC-364  
 UV-visible spectra, characteristic CC-153  
 UV-visible spectra, saving RM-412  
**V**  
 valence atomic orbitals CC-279  
 Valence state ionization energy (VSIE) CC-281  
 Valine (Val), amino acid residue RM-143  
 Values, display spectral HM-169  
 Values, spectral HM-129  
 Values, spectral, displaying HM-156  
 van der Waals CC-182, CC-184, CC-196, CC-199, CC-201, CC-204, CC-222  
 van der Waals energy  
     in log file RM-638  
 van der Waals interactions CC-26  
 van der Waals scale factor, Force Field Options dialog box RM-295

variable CDK-101  
 variables in Tcl CDK-97  
 variables, script messages RM-510–RM-511  
 vector HSV CDK-32, CDK-101  
 vector variable CDK-115  
 vel, HIN file syntax RM-718  
 Velocities  
     Molecular Dynamics Snapshots dialog box RM-375  
     storage RM-85  
 velocities CDK-56, CDK-210, CDK-221  
 velocities, distribution of atom CC-76  
 velocities, random distribution CC-345  
 velocities, rescaling CC-347  
 velocities, restart CC-345  
 velocity, set atom RM-344  
 Verlet algorithm CC-97  
 version number RM-115  
 Vertical grid points RM-395  
 vertical ordering CDK-99  
 vibration, transition state search RM-387, RM-389, RM-390, RM-391  
 vibrational analysis CDK-207, CDK-217, CC-128, CC-149  
     and IR spectroscopy CC-365–CC-371  
 vibrational spectra, saving RM-386, RM-408  
 Vibrational Spectrum  
     Compute Menu RM-406–RM-410  
     Compute menu RM-358  
 vibrational spectrum, importing RM-92  
 vibrational spectrum, storing RM-93  
 Vibrations  
     Compute Menu RM-385–RM-386  
     Compute menu RM-358  
 vibrations and transition states of ammonia GS-227–GS-239  
 vicinal interactions CC-182, CC-190  
 View Menu HM-155  
 View molecular system HM-144  
 View spectrum HM-148  
 view, HIN file syntax RM-715  
 Viewer's Coordinate System RM-19  
 Viewer, Apply to, Rotate dialog box RM-211  
 Viewer's Coordinate System HM-152  
 virtual orbitals CC-43, CC-46  
 visible atoms RM-222  
 Visual Basic CDK-2, CDK-9, CDK-107, CDK-119, CDK-139, CDK-149, CDK-150, CDK-154, CDK-204  
 Visual Basic controls CDK-120  
 Visual C++ CDK-3, CDK-149, CDK-159, CDK-173  
 VMSG CDK-15, CDK-34  
 Volume RM-456  
 VSIE CC-281  
 VSIE values, table CC-281  
**W**  
 walleyed stereo  
     setting in chem.ini/Registry RM-782  
 Walsh's rules CC-43  
 warning CDK-214  
 warning messages RM-359–RM-360  
     destination, in chem.ini/Registry RM-765  
 water  
     molecular dynamics GS-147–GS-177  
     molecules  
         creating GS-179–GS-180  
 water box CC-211  
 water model, TIP3P CC-65  
 water molecules RM-501  
 water template files RM-619–RM-621  
 water–N-methylacetamide interaction GS-197–GS-207  
 wave function CC-37  
 wave function calculation GS-182–GS-183, GS-206, GS-214–GS-216  
 Wave functions, spin HM-247  
 Wave functions, stationary-state HM-248  
 Wavelength, Electronic Spectrum dialog box RM-413, RM-414  
 Weak coupling approximation HM-183, HM-250  
 wedges



- displaying stereobonds with RM-482
- explained RM-484
- types of RM-482
- Weighted diffuseness, Semi-empirical Options (Hückel) dialog box RM-304
- widget CDK-46, CDK-98, CDK-102, CDK-120, CDK-129
- Width, peak, fine control HM-191
- Widths, line HM-156
- wild-card approach, parameters CC-215
- WIN.INI settings HM-78
- window
  - border, HyperChem Windows RM-6
  - changing size of GS-30
  - parts of HyperChem window GS-28–GS-30
  - Setup Image RM-100
- Window Color RM-95
- window title CDK-47
- window, expanding to full screen GS-45
- Window, HyperNMR HM-86, HM-142
- window\_color CDK-141
- window-color CDK-22, CDK-27, CDK-116, CDK-122, CDK-141, CDK-179, CDK-218
- Windows
  - HyperChem RM-4–RM-6
  - keyboard alternatives RM-9–RM-12
- Windows API CDK-10, CDK-159
- Windows Applications
  - graphics from WMF RM-615
  - using DDE RM-604–RM-605
- Windows Clipboard, graphics stored RM-613–RM-614
- Windows Program Manager, Main icon RM-94
- windows, keyboard alternatives GS-31
- Windows, Microsoft HM-77
- WinMain CDK-156, CDK-165
- WINSOCK CDK-18, CDK-149
- Wire mesh isosurface RM-397
- wish CDK-128, CDK-150
- Wizard CDK-174, CDK-176, CDK-178
- WMF. *See* metafiles
- WndProc CDK-165
- Woodward-Hoffmann rules CC-43
- Word Basic CDK-113
- Word Macros CDK-110
- word processor CDK-107
- work, saving GS-58–GS-60
- working in dialog boxes RM-14
- Workspace
  - HyperChem Windows RM-5
  - setting, choosing extent of Copy Image RM-612
  - window, Setup Image RM-100
- workspace GS-29
  - clearing GS-56
- Workspace, HyperNMR HM-86, HM-142
- World Wide Web CDK-23, CDK-96, CDK-117
- write CDK-155
- Write file HM-153, HM-164
- writing HSV CDK-29
- X**
- X Axis
  - Align Molecules dialog box RM-217
  - Align Viewer dialog box RM-215
  - Rotate About, Rotate dialog box RM-210
- XLM files CDK-117
- XY rotation HM-101
  - right button functions RM-20
  - tool RM-37–RM-38
- XY rotation tool GS-75
- XY translation HM-100
  - right button functions RM-20
  - step size setting RM-43
  - tool RM-40–RM-41
- XY translation tool GS-70–GS-72
- Y**
- Y Axis
  - Align Molecules dialog box RM-217

Align Viewer dialog box RM-215  
Rotate About, Rotate dialog box  
RM-210

## Z

### Z Axis

Align Molecules dialog box RM-217  
Align Viewer dialog box RM-216  
Rotate About, Rotate dialog box  
RM-210

### Z Clip

dialog box RM-213–RM-215  
Edit menu RM-202

### Z Clip dialog box GS-76, GS-77

### Z clipping tool GS-76–GS-79, RM-47–RM-49

### Z form, polynucleotide strand conformation RM-160

### Z rotation

right button functions RM-20  
setting amount RM-40  
tool RM-39–RM-40

### Z rotation tool GS-76

### Z translation RM-43–RM-45

right button functions RM-20  
tool RM-43–RM-45

### Z Translation tool GS-72

### ZDO CC-284

### Zerner, Michael

and ZINDO/1 RM-680

### zero differential overlap (ZDO) CC-284

### zeta, double CC-270

### Zeta, polynucleotide torsion angle RM-161

### zeta, triple and quadruple CC-271

### zindo CDK-228

### ZINDO/1 CC-123, CC-132, CC-157, CC-249, CC-252, CC-306–CC-308

background RM-303

parameter files RM-680

Semi-empirical Method dialog box  
RM-299

### ZINDO/S CC-123, CC-132, CC-157, CC-249, CC-252, CC-307–CC-309

and UV visible spectra RM-300

background RM-303

parameter files RM-680, RM-696, RM-  
700

Semi-empirical Method dialog box  
RM-300

### Z-Matrix file HM-81, HM-89, HM-153, HM- 163, HM-186, HM-200

### Z-Matrix files

input filter RM-81

open RM-80

save RM-84

### Zoom

dialog box RM-212

Edit menu RM-202

sensitivity setting RM-46

tool RM-45–RM-46

### zoom CDK-208, CDK-216

### Zoom Factor, Zoom dialog box RM-212

### Zoom in, spectrum HM-126

### zoom speed RM-97, RM-98

### Zoom structure HM-101, HM-161

### Zoom tool GS-73–GS-74

### Zoom, Electronic Spectrum dialog box RM-413 zwitterion

creating GS-93–GS-94

zwitterions creation RM-495

# HyperChem<sup>®</sup> for Windows

What's New in Release 7.5

# Copyright © 2003 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperGauss, HyperChemOS, HyperNMR. Pocket HyperChem and ChemPlus are trademarks of Hypercube, Inc.

## Third Party Trademarks

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines, Inc.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Table of Contents

<b>I. Introduction</b>	<b>1</b>
Release 7.5 . . . . .	1
New Features . . . . .	1
New OpenGL Graphics . . . . .	1
New User Colors . . . . .	2
Different Rendering for Different Atoms . . . . .	3
New Protein Capabilities . . . . .	3
New Secondary Structure Renderings . . . . .	4
Minor Modifications . . . . .	4
Limitations . . . . .	5
<b>II. OpenGL 3D Graphics</b>	<b>7</b>
Introduction . . . . .	7
<b>III. User Colors</b>	<b>9</b>
Introduction . . . . .	9
Choosing User Colors . . . . .	9
User-Color Script Variables . . . . .	10
<b>IV. Differential Renderings for Atoms</b>	<b>11</b>
Introduction . . . . .	11
Choosing an Atom Rendering . . . . .	12

<b>V. Protein Capabilities</b>	<b>15</b>
Introduction . . . . .	15
Protein Data Bank Files . . . . .	16
Selecting secondary structures . . . . .	17
Secondary Structures (Atoms) not Secondary Structure Renderings Can Be Selected . . . . .	18
New Peptide Builder Choices . . . . .	18
Rapid Electron Densities and Electrostatic Potentials . . . . .	19
<b>VI. Rendering of Secondary Structure</b>	<b>23</b>
Introduction . . . . .	23
<b>VII. Minor Additions and Deletions</b>	<b>27</b>
Introduction . . . . .	27
Additions . . . . .	27
Double and Triple Bonds . . . . .	27
New Script Variables and Commands . . . . .	29
HyperChem State Variables . . . . .	29
New Commands . . . . .	30
Other Script Changes . . . . .	30
Deletions . . . . .	30

# Chapter 1

## Introduction

### Release 7.5

This manual describes only what is new in Release 7.5 of HyperChem. The accompanying Release 7 manuals are still current for Release 7.5 and this new manual describes only the differences and new features that were introduced into HyperChem in going from Release 7 (or Release 7.1) to Release 7.5 of HyperChem for Windows.

### New Features

The new features of HyperChem Release 7.5 are divided into 5 major types:

- New OpenGL 3D Graphics
- New User Colors
- Different Renderings for Different Atoms
- New Protein Capabilities
- New Secondary Structure Renderings
- Minor Modifications

### New OpenGL Graphics

Hypercube, Inc. was incorporated in 1985 and the first versions of HyperChem for Windows started appearing in the late 1980s. These early PC versions has no access to libraries for 3D graphics and hence Hypercube developed 3D renderings of molecules, orbitals, etc. based upon Microsoft 2D "Win API" graphics. Until Release 7.5 all the 3D rendering in HyperChem used these in-

house algorithms based on simple 2D primitives provided by the Windows operating systems. This allowed Hypercube to port software to Pocket HyperChem very rapidly, for example (Pocket PCs do not yet support standard 3D graphics libraries). However, it is appropriate to move on with better rendering opportunities in future versions of HyperChem.

In the very early 1990s, Hypercube developed a Unix version of HyperChem for the Silicon Graphics, Inc. (SGI) hardware platforms. These products used a 3D graphics library from SGI referred to as GL. The quality of the graphics on these SGI platforms was generally better than the graphics that HyperChem has used on PCs until now. The graphics in HyperChem for Windows has been perfectly adequate and quite sophisticated given that it is based on 2D primitives. However, the time has come for HyperChem to move on to full 3D commercial libraries such as Direct 3D or OpenGL (the evolution of GL).

Beginning with Release 7.5, HyperChem now uses OpenGL libraries for all its 3D graphics. This not only enhances the quality of existing graphical renderings but allows Hypercube to rapidly develop new renderings such as the secondary structure protein renderings included in Release 7.5

## New User Colors

Again, since HyperChem was developed back in the 1980s for the first version of Windows, it was restricted to a total of 8 standard colors. Since many shades of these 8 are available for shaded models of molecules and orbitals, it has not been a debilitating limitation. However, it does not allow pastels and specific colors that a particular user might desire. With Release 7.5 the limitation to 8 colors is relaxed to essentially permit you to use any of the 16 million (24 bit) colors available with most graphics cards.

Rather than forcing a user to continually decide on these 24 bits, Release 7.5 introduces the concept of "User Colors" that you can set to any of the 16 million choices and then reuse easily. This is done by adding to the standard 8 Windows colors, 4 user-definable colors referred to as user1, user2, user3, and user4. HyperChem 7.5 provides an easy way for you to define and retain, and remember these colors.



## Different Rendering for Different Atoms

While Hypercube, Inc.'s version of HyperChem for Unix (SGI) allowed this, until now the Windows version of HyperChem required you to use the same rendering for every atom in the system. With Release 7.5, you can now choose to mix renderings and any of the various atom renderings can be used with any atom. A new tube rendering is now available so that any atom in the system can now be rendered as:

- Sticks
- Balls and Sticks
- Balls and Cylinders
- Overlapping CPK Spheres
- Tubes

In addition, any atom can have a dot surface added to its rendering. The radii of all balls, spheres, cylinders, and tubes can be varied.

## New Protein Capabilities

Release 7.5 adds significant new capability for dealing with peptides and proteins, particularly as related to Protein Data Bank (PDB) files. The new features are:

- Peptide builder knows further secondary structure, such as 3-10 helix, for example.
- Protein secondary structure is part of each residues definition. Each residue can be helix, sheet, turn or random coil.
- Protein secondary structure is automatically extracted from PDB files and automatically retained in corresponding HIN files.
- You can make a selection based upon protein secondary structure.
- Secondary structures, such as helices and sheets, can be rendered in a variety of new ways. There are now two main rendering motifs - rendering of atoms and rendering of secondary structures.

- Selection can be used to arbitrarily color secondary structure renderings.

## New Secondary Structure Renderings

HyperChem has always contained a relative primitive ribbon rendering of proteins. That is, the earlier rendering consisted of parallel lines along the backbone. With Release 7.5 and the introduction of OpenGL into HyperChem, a number of other more sophisticated solid ribbon-like renderings are now available. These consist of:

- Ribbon lines.
- Thin solid ribbons.
- Thick solid ribbons.
- Cylinders - normally representing the position of head and tail of helices.
- A tubular coil rendering - normally representing random coil secondary structures.

The properties of these protein secondary structure renderings, such as thickness (or radius) can be manipulated to suite your personal taste. They can be colored in arbitrary ways. A default secondary structure rendering is based of the actual secondary structure (such as cylinders for helices). However, you can customize the secondary structure rendering (for example, using a tubular coil rendering for helices rather than a cylinder).

## Minor Modifications

In addition to the above, a few minor additions or changes are available in Release 7.5 These include:

- New script commands and HyperChem State Variables (HSVs). Scripting has been extended to access the new features.
- A registry setting is available to allow you to revert to the legacy graphics model if some particular older feature is now unavailable with Open GL. As HyperChem goes forward this may be eliminated.

- Double and triple bonds are now visible with renderings other than sticks.
- Menu items are available to quickly turn on or off either of the two renderings - atoms or protein secondary structures.

## Limitations

With the wholesale addition of OpenGL rendering to replace HyperChem's legacy Windows API rendering, it is likely that certain features of earlier HyperChem might get lost in the shuffle. All efforts were made to avoid this and in a very few situations older Windows renderings were retained just to avoid this problem. For example, the view of a molecule from above so as to be able to see and manipulate the clipping planes is still a legacy rendering. HyperChem has not really lost any "legacy features" in replacing its legacy rendering. However, there are a few situations that do require clarification.

OpenGL has no concept of a Windows Metafile and hence metafiles are not available in Release 7.5 This affects, for example, using non-shaded renderings to create metafiles for printing that have the resolution of the printer.

Printing, in general is not as flexible, and may require you to copy an image to the clipboard and into Windows paint for printing.

Temporarily at least, the circle inside an aromatic ring denoting an alternative to the rendering of aromatic bonds is unavailable.



# Chapter 2

## OpenGL 3D Graphics

### Introduction

This chapter contains a few essential details of the new OpenGL graphics model.

Little needs to be said about the new graphics since in most situations it simply replaces previous renderings with a newer, higher quality rendering without any user involvement. However, HyperChem 7.5 contains a HyperChem registry setting under Preferences that allows one to essentially obtain a legacy graphics copy of HyperChem for the unforeseen situations in which this is necessary. In general this setting never need be set to a form different from its default setting shown below. The registry item is:

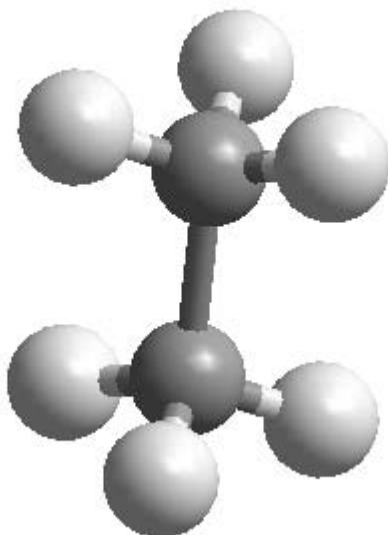
```
RenderUsingOpenGL    Yes
```

This entry appears under Preferences in,

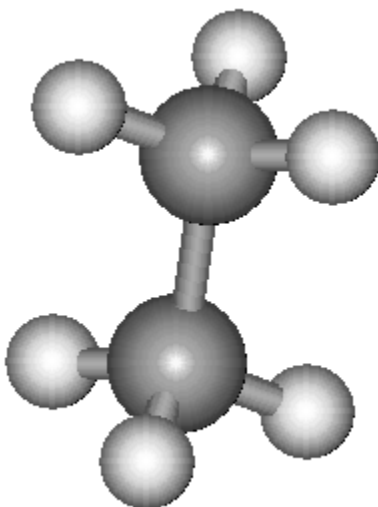
```
HKEY_CURRENT_USER\SOFTWARE\HYPERCUBE\...
```

If a value of No is used then HyperChem will not use the new OpenGL renderings and in many situations may even give ambiguous results, etc. Only use a value other than Yes in very specific understood circumstances. This setting will eventually be eliminated.

The difference between the old and new renderings is illustrated by the following two "Balls and Cylinder" renderings of ethane.



NEW



LEGACY

# Chapter 3

## User Colors

### Introduction

This chapter describes the addition of new colors in Release 7.5. Up until now HyperChem has used the 8 standard colors described by 3 bits (red=0 or 1, green= 0 or 1, blue=0 or 1). These 3 bits lead to the 8 colors black, blue, green ,cyan, red, violet, yellow, and white. Only these 8 colors have been used in HyperChem for Windows until now.

A simple standard available for in Windows is to use 8 bits to describe red, green, or blue rather than just one bit each. Such 24-bit color selection, however, would require remembering the 24-bits or their equivalent in order to be able to replicate the color later - a somewhat awkward scheme. Instead, HyperChem 7.5 introduces the concept of a small number of user colors that can be named and set for easy replication at any time. Possibly the original 8 colors ought now to be user definable or there ought to be a larger set, say 50, user definable colors. For the time being however, HyperChem 7.5 allows you to define and name 4 additional colors that then become immediately available as alternatives to the original 8 colors. Each of these 4 "user colors" can be any of the 16 million colors available from 24 bits.

### Choosing User Colors

The menu item, User Colors..., brings up the following dialog box:



This dialog box allows you to observe and set each of the four user colors and their name. These colors will then be immediately available by name in other parts of HyperChem and will be saved in the registry so that they are available each time you execute HyperChem.

**To define a user color:**

1. Use the option buttons to select one of the four colors User1 to User4.
2. Click on Define User Color to bring up the standard color selection dialog box, select a color, and click on OK to return to the User Color dialog box.
3. Type in a name for your new color in the text-entry field.
4. Repeat for other user colors or click on OK to complete your definition of new colors.

## User-Color Script Variables

The new user colors are definable via scripting. For example, to define a new color for User2 that is "Olive" the two appropriate setting of HyperChem Script Variables (HSVs) are:

```
user2-color "128 128 64"
```

```
user2-text "Olive"
```

Having defined any of the four user colors, they should appear any time it is appropriate to make a choice of color.



# Chapter 4

## Differential Renderings for Atoms

### Introduction

This chapter describes the various changes to atom rendering in Release 7.5. Until Release 7.5 the atom rendering was a global property of the molecular system. Now the atom rendering is an explicit property of each atom in the system and any atom can be rendered from any of the available renderings. The set of atom renderings has also grown with Release 7.5. A new tube rendering is now available.

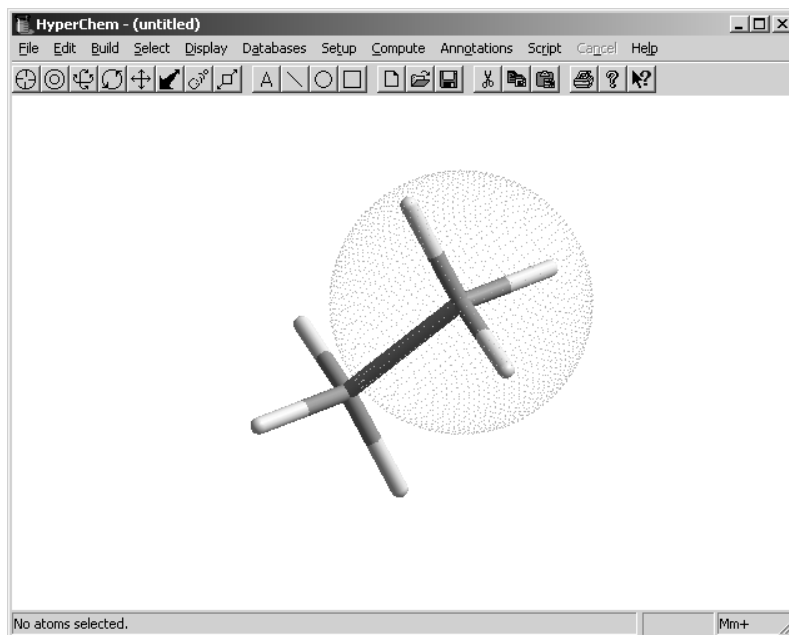
The set of atom renderings is:

- Sticks
- Balls (and Sticks)
- Balls and Cylinders
- CPK Overlapping Spheres
- Tubes

The thickness of sticks is variable as is the radius of all Balls, Spheres, Tubes, and Cylinders.

A dot surface rendering can be added to any of the above primary renderings.

The new tube rendering is shown below with a dot surface added to one of the carbon atoms.



## Choosing an Atom Rendering

With Release 7.5, renderings of atoms and molecules have been expanded to include two fundamentally different types of renderings:

- Atom renderings
- Secondary structure renderings

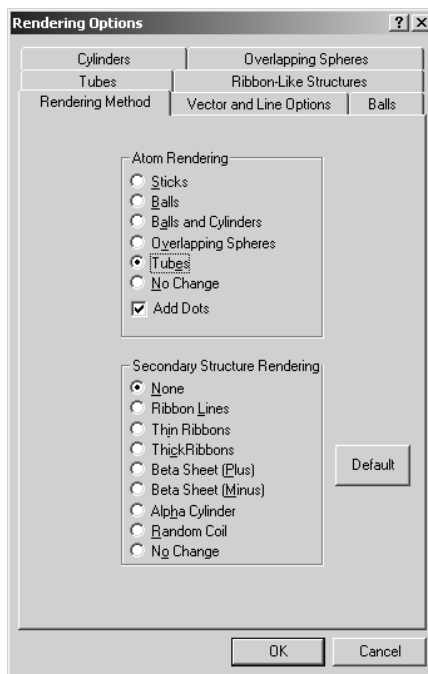
These two sets of rendering are controlled by the same Rendering Dialog box and hence each now includes a No Change option so that you can, for example, change a secondary structure rendering without affecting the atom rendering.

### To choose an atom rendering:

1. Use ordinary selection to select a subset of the atoms in the system to which you wish to apply a specific rendering. A null set

is equivalent to a complete set, i.e. if you make no selection the new rendering will apply to all atoms.

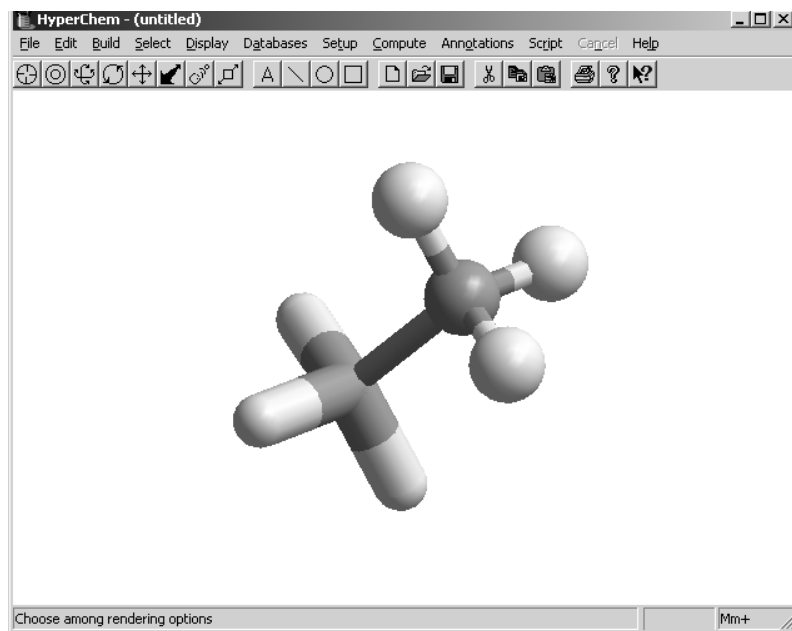
- Use the menu item, Display/Rendering, to bring up the following Dialog box,



- Select the desired rendering for the atoms as above. If you have no protein secondary structure, the associated bottom part of the dialog box can be ignored.
- Click on OK to close the dialog box (or use the other tabs above to set specific properties of Balls, Cylinders, etc. prior to clicking on OK).

To apply different renderings to different portions of a molecule repeat the above process with a different atom selection. For example, a rendering of ethane that mixes Balls and Cylinders with Tubes is shown below.

## Choosing an Atom Rendering



# Chapter 5

## Protein Capabilities

### Introduction

HyperChem Release 7.5 adds certain new capabilities that apply to proteins and peptide structures. These new features are:

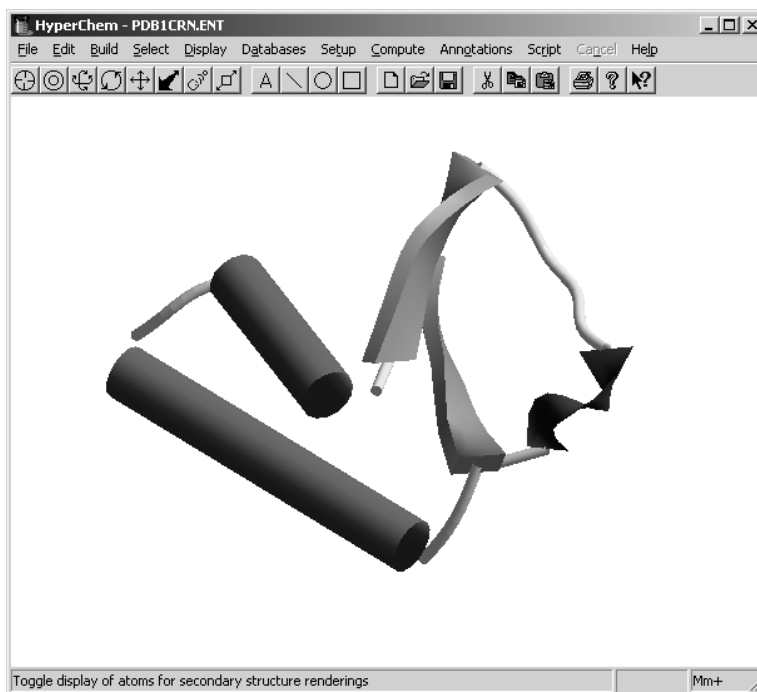
- Ability to extract secondary structure information from a Protein Data Bank file
- Addition of a new secondary structure property to all amino acid residues. The secondary structure can be one of helix, sheet, turn, or random coil.
- Inclusion of the above secondary structure information in HIN files.
- New renderings of secondary structure. These secondary structure renderings can be tied to the actual explicit protein secondary structure or chosen independently. They can be colored and manipulated in a variety of ways. They are fully described in the next chapter.
- New options for the peptide builder that extend the previous options of alpha helix and extended beta sheet to include left-handed alpha helix, 3-10 helix,  $\pi$ -helix, parallel and anti-parallel pleated-sheets plus beta turns.
- The ability to plot the electron density and electrostatic potential of a protein. This option is not restricted to proteins but a new algorithm was developed with proteins explicitly in mind. This option is similar to previous quantum mechanical options but has been made so fast that it can be applied to large molecules such as proteins. The option is made available when using molecular mechanics to distinguish it from conventional quantum mechanical computations of electron density and electrostatic potential.

- Normal selection has been extended to allow residues to be selected based upon their secondary structure property. For example, one can select all residues that have helical structure.

## Protein Data Bank Files

Protein Data Bank files describe secondary structures as either *helix*, *sheet*, or *turn*. By default HyperChem now assigns a *random-coil* structure to the residues in the PDB file that have not been explicitly assigned to one of the other three secondary structures.

Thus, for example, when one reads the small protein Crambin for a PDB file, one obtains the following secondary structures:



Residues labelled in the PDB file as part of a helix are described by default in HyperChem by a red cylinder with an axis equivalent to the helix axis. Sheets are describe by thick green ribbons, turns by thin blue ribbons, and all other residues by a yellow tubular coil. This information is transferred to any HIN file subsequently writ-

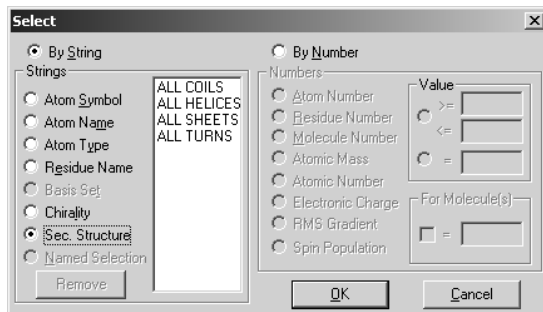
ten. For example, the residue record in a HIN file might now appear as follows:

```
res 1 THR 1 - - sheet
```

where this residue is described as part of a beta sheet. The empty two fields above (required dashes) means that this information (prev and next residue) is missing in the PDB file. The form of the last field describing secondary structure can be one of <helix, sheet, turn> with a default to random coil when the field is empty (missing). The above default rendering of a PDB file can be modified in various ways described in the next chapter.

## Selecting secondary structures

As has been the case since the first version of HyperChem, it is possible to select residues in various ways - by number, by type, etc. Thus one can, for example, quickly select residues 21-47. With Release 7.5 it is now possible to select residues by their secondary structure. If your molecule has protein secondary structure, then clicking on the Select/Select menu item brings up a dialog box resembling the following:



By selecting, All Helices, for example, a selection will be made of all residues having the helix property.

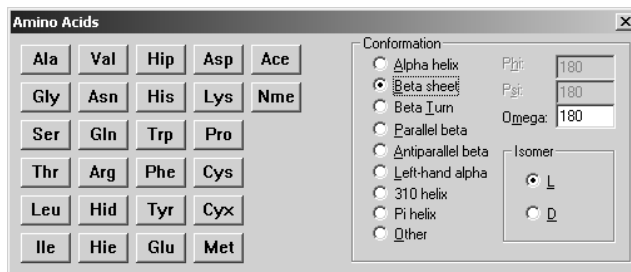
## Secondary Structures (Atoms) not Secondary Structure Renderings Can Be Selected

An important point to note is that only atoms can be selected in HyperChem. One might select all the atoms of a residue or all the atoms of a molecule at once depending upon the unit of selection at the top of the Select menu but one is still selecting only atoms not secondary structure renderings. It might potentially be useful to be able to select atoms and get visual feedback via changes in the color of the associated secondary structure rendering. This is not currently possible.

The consequence of this is significant when one is visualizing only the secondary structure as in the above rendering. One gets no immediate visual feedback of what is selected unless one is showing the associated atoms. This may be a problem if for clarity of display you have turned off the atom visualization and have only secondary structure visualization. The atoms can still be selected and de-selected as if they were visible but one obtains no visual feedback. Thus, it may often be useful to use the toggle, Don't Show Atoms in the Display menu. That is, show them when you need to make a selection and don't show them when viewing the secondary structure.

## New Peptide Builder Choices

The peptide builder has been extended to be consistent with the Sequence Editor module. That is, it can now build more than simple alpha helices and extended beta sheets. The new options are shown below:





An additional result of changes to the peptide builder is that visualization of the associated secondary structure is immediately available as per Protein Data Bank Files discussed above. As with information extracted from PDB files, this secondary structure information is now stored in the residue record of a HIN file.

## Rapid Electron Densities and Electrostatic Potentials

The variety of engines available to calculate energies and derivatives of the energy for molecules divide into two main camps - molecular mechanics and quantum mechanics. For molecules the size of proteins a calculation of a quantum mechanical wavefunction is a long, time-consuming process and hence molecular mechanics methods are in routine use for such large molecules. While molecular mechanics leads to energies it does not normally allow computation of electron densities nor electrostatic potentials arising from the electron density.

HyperChem Release 7.5 includes a new rapid method for evaluating these quantities that is referred to here as the "Spherical Atom Approximation." It uses only atom charges that are a feature of all molecular mechanics methods.

The electron density derived from a Hartree-Fock wave function is described by a density matrix,  $P_{\mu\nu}$ ,

$$\rho(\vec{r}) = \sum_{\mu\nu} P_{\mu\nu} \phi_{\mu}(\vec{r}) \phi_{\nu}(\vec{r})$$

and depends upon the value of the basis function orbitals,  $\phi_{\mu}$ , at a point  $\mathbf{r}$  in space. For a zero differential overlap wavefunction, only the diagonal terms are retained and this reduces to,

$$\rho(\vec{r}) = \sum_{\mu} P_{\mu\mu} |\phi_{\mu}(\vec{r})|^2$$

The sum here runs over all basis functions. Dividing this into atomic contributions gives,

$$\rho(\vec{r}) = \sum_A \sum_{\mu} P_{\mu\mu} |\phi_{\mu}(\vec{r})|^2$$

If one makes the assumption that  $\phi_s^2 = \phi_x^2 = \phi_y^2 = \phi_z^2$ , which we call the spherical atom approximation, then,

$$\rho(\vec{r}) = \sum_A q_A |\phi_s^A(\vec{r})|^2$$

and the electron density can be calculated from the total electrons,  $q_A$ ,

$$q_A = \sum_{\mu} P_{\mu\mu}$$

assigned to the atom and the average value of  $\phi^2$  for the valence orbitals (taken to be equivalent to the value for the valence s-orbital). The value of  $q_A$ , for example, for a neutral Carbon atom is 4.0 the number of valence electrons. The value is modified by

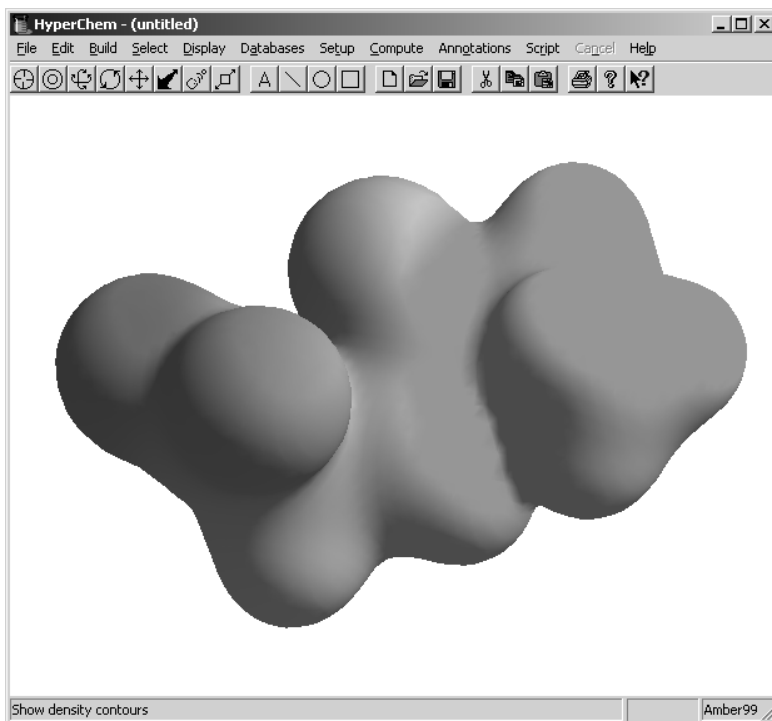
the net atomic charge which is a property of force fields such as Amber and Charmm.

The electrostatic potential is derivable in an identical way with one-center electrostatic integrals replacing  $\phi^2$ .

**To display the electrostatic potential mapped onto the electron density using non wave function methods:**

1. Create a molecule of any kind.
2. Use Setup/Molecular Mechanics to choose a molecular mechanics method such as Amber, Charmm, etc. Generally the MM+ method will not have other than zero net atom charges unless they have first been generated by a Gasteiger-like QSAR properties calculation or by a quantum mechanical calculation.
3. Use Compute/Plot Molecular Graphs to choose a property to plot along with the various parameters for the plot such as contour value, isosurface grid, type of isosurface rendering, etc. In this case, one would choose electrostatic potential and 3D mapped isosurface. The Mapped Function Options allow more than one kind of color mapping for the electrostatic potential.
4. Choose OK to render the plot.

For example, the electrostatic potential, for the di-GLY zwitterion looks as follows,



# Chapter 6

## Rendering of Secondary Structure

### Introduction

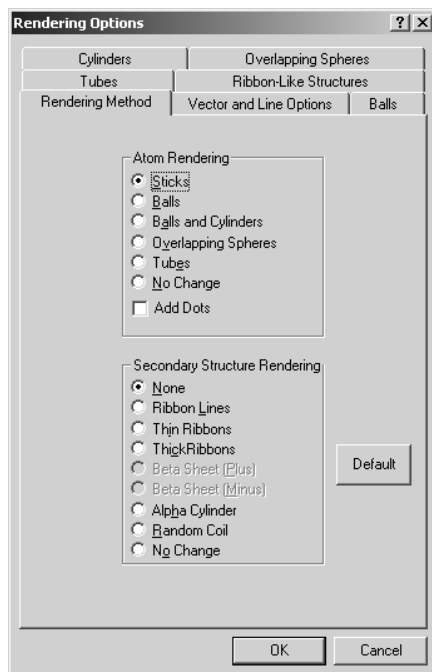
With the use of OpenGL in HyperChem Release 7.5 it becomes possible to introduce a number of new renderings for protein secondary structure that were not possible with legacy Windows graphics. Prior to Release 7.5 only a ribbon lines rendering was available. It was restricted to describing the whole protein backbone without differentiation.

The rendering of protein secondary structure can now use any of the following renderings:

- Parallel lines for ribbons
- A thin solid ribbon
- A thick solid ribbon
- A solid cylinder indicating the starting and ending position of a sequence of residues
- A tubular coil

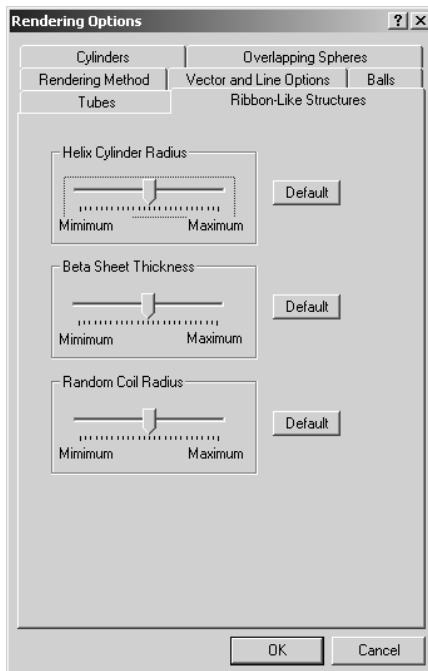
The default rendering for helices is the cylinder, for sheets it is the thick ribbon, for turns it is the thin ribbon, and for random coils

it is the tubular coil. the dialog box for choosing secondary structure renderings is the same one used to choose atom renderings,



One can, of course, choose any rendering for any secondary structure. The Default button, however, returns to the default render-

ings just described. The parameters of these renderings can be modified with the, Ribbon-like Structures tab,



The radius of the cylinder, the thickness of the thick ribbon, and the radius of the tubular coil are modifiable as shown above. Each has its default parameter which is recovered via the Default button.





# Chapter 7

## Minor Additions and Deletions

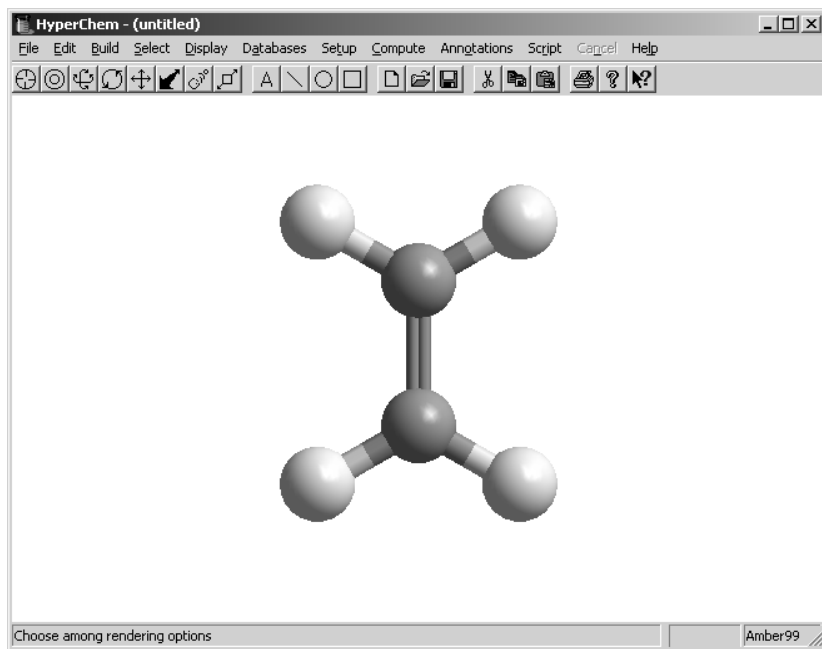
### Introduction

HyperChem Release 7.5 includes a few minor additions described here.

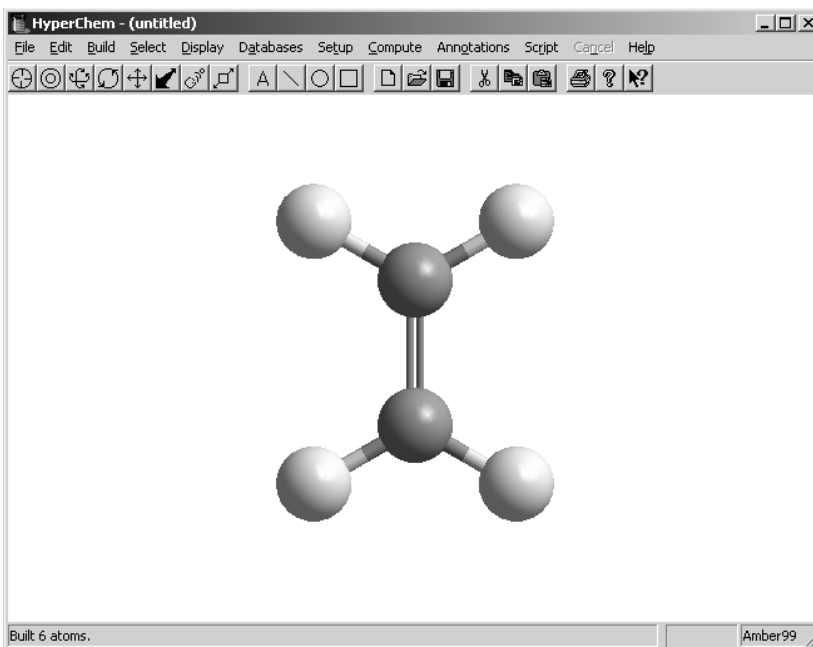
### Additions

#### Double and Triple Bonds

HyperChem now displays double and triple bonds in renderings other than simple sticks. For example, the double bond in ethylene with a Balls and Cylinders rendering is show below:



A further separation of the two bonds in a double bond can be affected by changing the width of cylinders. Thus, an alternative rendering is:



This distinguishes the two bonds more clearly and may or may not be more desirable than the first rendering.

## New Script Variables and Commands

### HyperChem State Variables

New or modified HSVs are:

**render-method - enum**

- the method for rendering selected atoms . The values are sticks, balls, balls-and-cylinders, spheres, tubes, and nochange.

**secondary-structure-method - enum**

- the method for rendering a secondary structure . The values are none, lines, thin, thick, cylinder, coil, and nochange.

**dont-show-atom-structure** - boolean

- when true turns off display of atoms.

**show-secondary-structure** - boolean

- when true turns on display of protein secondary structure.

**temporary-path** - string

- a path for temporary storage.

## New Commands

New script commands are:

**color-secondary-selection** string1

String1 names a color, such as Blue or User2, for the secondary structure rendering associated with the current atom selection.

**select-secondary-structure** string1

String1 names secondary structures, such as All Helices, that are to be selected by this command.

**un-select-secondary-structure** string1

String1 names secondary structures, such as All Helices, that are to be de-selected by this command.

## Other Script Changes

In addition to the above HSVs, the variables user1-color, user1-text, user2-color, ... described in Chapter 3 are now available. Finally, any previous HSV or Command that used an enum of the 8 colors Black ... White, now also take User1, User2, User3, or User4 as arguments.

## Deletions

Because of the switch to OpenGL a few features have been lost, possibly temporarily.

Transparency or translucent orbitals, densities, etc. is no longer available. This will be corrected in future versions of HyperChem.

These older renderings are still available via the registry switch back to legacy graphics (RenderUsingOpenGL).

# **HyperChem<sup>®</sup> for Windows**

## **What's New in Release 8.0**

# Copyright © 2007 Hypercube, Inc.

All rights reserved

The contents of this manual and the associated software are the property of Hypercube, Inc. and are copyrighted. This publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

**HYPERCUBE, INC. PROVIDES MATERIALS “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL HYPERCUBE, INC. BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF PURCHASE OR USE OF THESE MATERIALS, EVEN IF HYPERCUBE, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THESE MATERIALS. THE SOLE AND EXCLUSIVE LIABILITY TO HYPERCUBE, INC., REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THE MATERIALS DESCRIBED HEREIN.**

Hypercube, Inc. reserves the right to revise and improve its products as it sees fit.

## Hypercube Trademarks

HyperChem is a registered trademark of Hypercube, Inc. HyperMM+, HyperNewton, HyperEHT, HyperNDO, HyperGauss, HyperChemOS, HyperNMR. Pocket HyperChem and ChemPlus are trademarks of Hypercube, Inc.

## Third Party Trademarks

Microsoft, MS-DOS, and Excel are registered trademarks, and Windows is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines, Inc.

All other brand and product names are trademarks or registered trademarks of their respective holders.

# Chapter 1 1

## Introduction 1

Release 8.0 1
New Features 1
Microsoft Vista Compatibility 2
New Third-Party Interfaces 3
Introduction 3
Current Status 3
What is an Interface? 4
History of These Interfaces. 4
New Batch Capabilities 4
The EXT File 5
The Past and the Future 5
Universal Use of Double Precision 6
Undo and Redo Capabilities for Model Building 6
Easy Access to Molecules via a Recent File List 6
Geometric Measurement Involving Points, Lines, and Planes 7
A Chemical Substituent Operation 7
Revised Toolbar with Easier Access to Model Building 8
Calculation of Entropies and Free Energies 8
Calculation of Heat Capacities 8
Calculation of Zero-Point Energies 9
Computation of Rate Constants 9
Computation of Equilibrium Constants 10
Further Capabilities for MP2 Perturbation Energies 10
Separation of Configuration Interaction from Single Points 10
Display of Line Width Envelopes for IR and UV Spectra 11
Separation of MM-QM Capabilities from Current Selection 11
Separation of Fixed Atoms from Current Selection 12
Vibrational Analysis for Molecular Mechanics 12
Applied Electric Fields for Molecular Mechanics 13
Ability to Explore "Particle-in-a Box" Wave Functions 13
Procedure 14
Ability to Select Desired Energy Units (kcal, kJ, or Hartree) 14



# Chapter 1

## Introduction

### Release 8.0

This manual describes only what is new in Release 8.0 of HyperChem. The accompanying Release 7 manuals are still current for Release 8.0 as is the "What's New in Release 7.5 Manual." This new manual describes only the differences and new features that were introduced into HyperChem in going from Release 7.5 to Release 8.0 of HyperChem for Windows.

### New Features

The new features of HyperChem Release 8.05 include the following:

- Microsoft Vista Compatibility
- New Third-Party Interfaces
- New Batch Capabilities
- Universal Use of Double Precision
- Undo and Redo Capabilities for Model Building
- Easy Access to Molecules via a Recent File List
- Geometric Measurement Involving Points, Lines, and Planes
- A Chemical Substituent Operation
- Revised Toolbar with Easier Access to Model Building
- Calculation of Entropies and Free Energies
- Calculation of Heat Capacities
- Calculation of Zero-Point Energies

- Computation of Rate Constants
- Computation of Equilibrium Constants
- Further Capabilities for MP2 Perturbation Energies
- Separation of Configuration Interaction from Single Points
- Rendering of POINT, LINE, and PLANE
- Integration of Temperature as a System Variable
- Display of Line Width Envelopes for IR and UV Spectra
- Separation of MM-QM Capabilities from Current Selection
- Separation of Fixed Atoms from Current Selection
- Ability to Fix Atoms in Quantum Mechanical Computations
- Vibrational Analysis for Molecular Mechanics
- Applied Electric Fields for Molecular Mechanics
- Ability to Explore "Particle-in-a Box" Wave Functions
- Ability to Select Desired Energy Units (kcal, kJ, or Hartree)
- Miscellaneous Bug Fixes and Small Improvements

## Microsoft Vista Compatibility

Microsoft Vista involves extensive new and modified features. HyperChem 8 has been prepared to provide reliable computation in this new environment. In some cases, however, manufacturers' graphics drivers (OpenGL drivers) have not proved to yet be up to the earlier standards of Windows XP. Where problems have been seen, installation of a new driver from the graphics card manufacturer has often eliminated any problem.

To use HyperChem effectively, one should have a good graphics card and a compatible newer driver. Any graphics hardware should be sufficient to run HyperChem. However, inexpensive machines will often have graphics hardware, included on the motherboard, that results in relatively slow manipulation of large molecules, compared to that with a modern 3rd-party graphics card.

# New Third-Party Interfaces

## Introduction

HyperChem 8 has the capability of being a graphics and GUI provider (including the displaying of results) for a number of third-party software packages. These packages may include other ab initio and semi-empirical packages such as GAMESS and MOPAC2007. Included with HyperChem 8 is the copyrighted source code for a number of these interfaces.

A standard menu item in HyperChem, "Third-Party Interfaces" executes a standard HyperChem Command Language (HCL) script installed with the product. This script has the pre-defined name, `thirdparty.scr`. This script can be modified by users to add other third-party applications but comes included with a variety of evolving interfaces.

These interfaces use the elegant "open architecture" features of HyperChem that allow software outside HyperChem to interact and read/write information to/from HyperChem. The interface software is copyrighted "Open Source" software that any HyperChem user can modify to improve the interface or to create a new interface to his/her own software.

The interfaces generally allow a user, for example, to run a GAMESS job from within HyperChem, and get back the results for display. The first level of interface, that is included to begin with in HyperChem 8, allows a user to display optimized structures, vibrational spectra and animations of normal modes plus 3D renderings of orbitals, electron densities, and electrostatic potentials. It is to be expected that these interfaces evolve to become richer as users and members of Hypercube, Inc. contribute to their capability.

## Current Status

The first version of these interfaces are relatively primitive. They could even be considered "bad". They are not up to the standards of HyperChem and the source code is not offered as an example of elegant coding practice. They, principally, indicate the possibilities that HyperChem offers of being a universal "front end" for a large number of other applications. Users are invited to participate

in creating interfaces that do become elegant examples of coding practice.

## What is an Interface?

The interfaces are all C++ code compilable and compiled with Visual Studio 2005. While Fortran interfaces are possible, the intent here is to stick to C++. An interface does not require other than a binary version of the third party application. An interface creates an input file for a third-party application, spawns that application as a child process and captures the output. The output is parsed by the interface and the results are passed back to HyperChem for internal storage and possible display

## History of These Interfaces.

These interfaces were initiated at Hypercube Europe a number of years ago with the involvement of Mirek Sopek and Wojciech Planka. The goal at that point was to interface HyperChem to the source code of QChem. Later Neil Ostlund at Hypercube extended that idea to interface a binary product, in this case Parallel Quantum System's ab initio product. Hypercube, Inc. at one time sold these third-party applications but that is not a current goal. The interfaces are offered only as a possibly useful way to increase the utility of HyperChem. The current interfaces (March 2007) are informal, not well documented or polished, source code that potentially offers the many features of HyperChem to arbitrary third-party software packages.

The interfaces are of widely disparate quality and each differs in some way from the other although they could, in principal all be combined into one code base. At this stage they are each separate Visual Studio Projects. Some provide only the preparation of input but most allow at least the importing of orbitals, geometries, and IR spectra. None of the interfaces are well tested at this stage, although that will improve as time goes on. It is expected that regular updates will be offered over the web.

## New Batch Capabilities

HyperChem has traditionally operated in a purely interactive mode where a calculation (a back end, e.g. HyperNewton) is con-

nected in a "live fashion" to HyperChem (the front end). This has been the case even when the back end resided on another machine on the network. This is not a terrible limitation since one can run many HyperChem front ends simultaneously if one likes.

With HyperChem 8, however, one now has the choice of running a calculation interactively or in Batch mode. In Batch mode the computation is severed and carries on by itself while HyperChem is free to read in a new molecule or continue on in any way it pleases, including spawning more batched jobs. The back ends have been instructed by the front end that they are indeed batch processes and instead of sending their results live to the front end, they create a \*.ext file that can be read at any later time into the HyperChem front end to display the results just as if the calculation was run interactively.

## **The EXT File**

HyperChem has always used EXT files for import and export of results but these files have taken on extra capability and meaning with Release 8. These files are structured text files that contain HyperChem State Variables (HSV's) so that they are easily incorporated back into HyperChem as if they were resident information. They must match the current molecule in the work space or match a HIN file that is read prior to reading an EXT file. Batch processes create both a HIN file and an EXT file that should go together.

The Third-Party Interface capability also uses EXT files. They are created by the interface from results parsed out of the output of the third party application.

The EXT file now contain thermodynamic information such as Energy, the coordinates of atoms, and Electronic, Vibrational, and Rotational spectra. These are now also used in the computation of rate constants and equilibrium constants.

## **The Past and the Future**

The third-party interfaces of HyperChem 8 allow interactive or batch use or a third "compute-mode" that generates input files only. When running in this third mode only an input file is created but no attempt is made to run the third-party application. HyperChem does not operate in this third mode for its native back ends since they are intimately associated with HyperChem and, at

this stage, only need to get their input from HyperChem rather than a file. These capabilities, however, lend themselves to a future version of HyperChem that is more oriented to large-scale parallel computation on remote machines. The introduction of batch operations and third-party back ends into HyperChem is

## **Universal Use of Double Precision**

Because of Hypercube, Inc. long history, there has been a legacy history of using single precision floating point to save time when double precision was not absolutely necessary. Thus, until now coordinates of molecules at the front end (i.e. HyperChem) were single precision while coordinates at the back end (e.g. HyperGauss) were double precision where they needed to be.

This saved time for graphical and communication operations where precision was not critical. However, small discrepancies between the front end and back end coordinates were occasionally noticeable such as when restarting an optimization. The optimization lost some precision when results were sent back to the front end. With faster graphics and faster machines, HyperChem is now universally double precision.

## **Undo and Redo Capabilities for Model Building**

A feature that has occasionally been adamantly requested in HyperChem is now available. This is an "undo" operation for molecular manipulations. Thus one can delete atoms, draw atoms, etc. and decide that these changes are undesirable and Undo them. Along with undo is a "redo" when the undo is decided as undesirable! These features are common in many programs but not in molecular modeling. They are now available in Release 8.

Naturally one cannot undo everything so this capability is restricted to changes in the molecular structure of the molecules in the workspace.

## **Easy Access to Molecules via a Recent File List**

Another feature that is common in many Windows programs that has been missing in HyperChem is the ability to see a list of recent

file openings and closing so that one doesn't have to go searching again through the file system to read in a molecule that one recent read. This "File List" is now available in HyperChem and is four files long.

## **Geometric Measurement Involving Points, Lines, and Planes**

HyperChem has long had the idea of a POINT, LINE, or PLANE. If one selects a set of atoms (all or any subset of the workspace) one can define POINT, LINE, or PLANE. The POINT is the center of mass of the selection. The LINE is the principal axis of the selection and the PLANE is a plane through the center of mass perpendicular to the tertiary axis. Thus PLANE is, for example, the plane of a Benzene molecule.

With HyperChem one can now visualize these geometric features and make associated measurements. Thus, one can ask about the variables associated with the selection that previously created POINT, LINE and PLANE and the current selection. Thus measurements are available of the distance between two POINTs, the distance of a POINT to a LINE or a PLANE, the angle between two LINES or a LINE and a PLANE or the angle between two PLANES.

## **A Chemical Substituent Operation**

One of the most elegant features of HyperChem is the ability to create a three-dimensional molecular structure by just drawing it and applying the model builder. This remains true.

However, with HyperChem 8, another rapid drawing capability is available. This involves the usual chemical idea of chemical substituents, R. In HyperChem these substituents replace any selected Hydrogen atom. Thus H->R has become a standard operation for a variety of common R-groups, including Phenyl (Ph). It is expected that a near term release of HyperChem will even allow users to define their own R groups. In any event it is now easier and faster to create molecules from standard components. Starting with H<sub>2</sub> or CH<sub>4</sub>, for example, one could create any organic molecule with a few clicks rather than having to draw the whole molecule.

This new operation substitutes any currently selected Hydrogen and ignores the selection of atoms that are not Hydrogen.

## Revised Toolbar with Easier Access to Model Building

Model building in HyperChem normally involves the selection of an element, often Carbon, to draw a skeleton of the molecule which is then possibly modified with the addition of some Nitrogens, Oxygens or more rarely other elements, and then performing a model build without "Explicit Hydrogen". This adds the requisite Hydrogens automatically. This operation is now easier as the selection of Carbon, Nitrogen, Oxygen, and the subsequent Model Building have been added to the toolbar. It is still possible to do everything as before. The new toolbar elements just make common operations faster.

## Calculation of Entropies and Free Energies

All previous versions of HyperChem really only dealt with Energy (Enthalpy?) rather than Free Energy. This is in some sense a historical association with Quantum Chemistry which has traditionally focussed on these to the detriment of Entropy and other thermodynamic variables. Calculating entropies, of course, requires more effort than just the "simple" energy. However, with the computation of vibrational and rotational spectra comes the possibility of computing the energy(E), entropy(S), and Helmholtz free energy ( $A=E-TS$ ). These calculations are now available in HyperChem 8 as a function of Temperature. Temperature is now a more fundamental quantity in HyperChem than before as are the thermodynamic quantities that depend on it.

## Calculation of Heat Capacities

As with Energy, Entropy, and Free Energy, it is now possible to calculate Heat Capacities. These are now routinely computed along with the other thermodynamic quantities that depend upon the temperature.



## Calculation of Zero-Point Energies

At zero degrees Kelvin, the energy is the dominant quantity of interest but does not only have an electronic component. Until now vibrational analysis has not reported the zero-point energy of vibration. These now are a part of any vibrational analysis.

## Computation of Rate Constants

Computational Chemistry and been better at Structure and Thermodynamics than at Kinetics. Many molecular modeling programs have little to say about rate constants which are obviously an important quantity in chemistry. With Release 8, HyperChem makes a start at making reactivity a mainstream molecular modeling activity. While only computing rate constants using the simplest Transition State Theory it is a beginning towards being a fundamental component of the whole of chemistry rather than only what computational chemists are best at.

HyperChem 8 computes partition functions for reactants A and B (in bimolecular reactions) or just A (in unimolecular reactions) and then computes the partition function for the Transition State. The input to these calculations are the structure of each of these species (created in HyperChem and then stored in HIN files) as well as the energy, and vibrational and rotational spectra of the species (created in HyperChem and then stored in EXT files).

These quantities can come from external third party packages as well (as described in the Third Party Interface Section above. The partition functions simply require the vibrational spectra (frequencies only) and rotational spectra (moments of inertia only) from an EXT file created by HyperChem or elsewhere.

A calculation of the rate constant as a function of temperature is then made and becomes available as a simple plot for placing into Power Point, etc.). In addition, the Arrhenius parameters can be extracted from the variation of the rate constant as a function of temperature.

If desired, and the the corresponding energies are available for the products (not just the reactants and transition state), a plot of the energy of reactants, transition state, and products is available.

## Computation of Equilibrium Constants

Since free energies are now available in HyperChem, a similar simple capability for calculating equilibrium constants as a function of temperature to that described for rate constants above is now available. The Helmholtz free energy  $A$  as a function of temperature is calculated from the electronic, vibrational, rotational, and translational components of the energy and entropy. The equilibrium constant for the reaction is then just the appropriate  $\exp(-\Delta A/kT)$ .

## Further Capabilities for MP2 Perturbation Energies

HyperChem has had available the computation of second-order correlation energies via the MP2 method. With Release 8 these are given a more prominent position in that any single point energy used, for example, by optimization, by potential plots, by rate constants, by molecular dynamics, etc., can now include the MP2 energy as well as the SCF energy. Previously, the check box for MP2 only showed that correlation energy as a property of the SCF calculation. Now that check box will use SCF+MP2 results as the energy for subsequent computations. The MP2 result is considerably more reliable in many circumstances than SCF Hartree-Fock result and with advances in desktop computation speeds it seems appropriate to give MP2 a more prominent role.

The MP2 gradients, unfortunately are still computed numerically rather than analytically so these calculations are certainly not as fast as pure SCF calculations. One also should be conscious that the check box for MP2 will be used universally and slow down what previously might have been only SCF computations.

## Separation of Configuration Interaction from Single Points

In a corresponding move to that for MP2 replacing SCF, a more prominent role for Configuration Interaction (CI) is expected in the future. In addition, CI was somewhat hidden in nested dialog boxes for Single Point calculations so that it was not always clear that CI was turned on. This option has now been made explicit

with a Single Point CI menu item for clarity and future additions to this capability.

## Display of Line Width Envelopes for IR and UV Spectra

HyperChem has performed IR and UV computations for many years. These spectra are displayed as stick drawings with individual intensities shown on the plot. The similar display of NMR spectra over the years has had "line width" capability of assigning a line width to each spectral line (the same line width for each frequency) and then summing them up to obtain an envelope that simulates what the experimental spectrum might look like. No line widths are computed - only a slider is made available to simulate increasing global line widths.

Release 8 makes this same facility available for IR and UV spectra that has been available for NMR spectra. The line width is initially set to zero but a simple slider changes the appearance of the spectra to the satisfaction of the user.

## Separation of MM-QM Capabilities from Current Selection

HyperChem for many years (the first wide spread implementation) had the capability of performing MM-QM calculations, i.e. calculations that on a large system treat part of the molecule with quantum mechanics (QM) and the remaining part of the molecule with molecular mechanics (MM). This capability operated via the current selection. If a subset selection was invoked at the time a quantum calculation was requested, the selected portion of the molecule was treated via quantum mechanics and the remaining portion via molecular mechanics. That is, the charges of the MM part were included in the core Hamiltonian of the quantum part.

While convenient, this use of "current selection" has proved limiting in that "current selection" meant something different during pure MM calculations. There it meant atoms that were allowed to move rather than remain fixed in space. This also made it impossible to fix atoms in space during quantum calculations.

With release 8 a "named selection" is used to distinguish QM atoms from MM atoms in a MM-QM computation. The new pre-defined named selection is called "MECHANICAL ATOMS". Such a selection of atoms will become the MM atoms of any quantum calculation.

## Separation of Fixed Atoms from Current Selection

This new option related the one just described above. Atoms subset selected in a pure MM calculation were previously the only ones allowed to move. While convenient, this was limiting. Now a new pre-named selection, "FIXED ATOMS" is used to specify that any atoms in either a MM or QM calculation are fixed in Cartesian space and not allowed to move in optimizations, molecular dynamics, Monte Carlo, etc. This capability is not new in MM but does allow a whole new capability in Quantum Mechanical calculation.

These fixed atoms need not have their gradients calculated in any computation and as such this new option ought to speed up calculations. In addition there may be situations where one wants to leave a portion of a molecule fixed and only allow optimization, dynamics, etc. on a portion of the molecule. As with MECHANICAL ATOMS, one ought to be careful now that FIXED ATOMS are not set when you don't want that option. These two options are now less visible than when they involved the current selection.

## Vibrational Analysis for Molecular Mechanics

Until Release 8, vibrational analysis was restricted to one of the quantum mechanical methods. Now it is available across the board with any of the "Energy Engines" available in HyperChem. With vibrational analysis and rotational moments of inertia, it is now possible to calculate Entropies and Free Energies across the board as well.

It may still be possible to spend lots of computational time performing vibrational analysis, particularly for large molecules since second derivatives are still not computed analytically for any of the methods. It is a goal for HyperChem in the future to speed up those methods that depend upon second derivatives of the energy

such as vibrational analysis. This ought to be, in principal, relatively easy for molecular mechanics.

## Applied Electric Fields for Molecular Mechanics

Electric fields are now available in the workspace for any of the "Compute Engines". Previously, the ability to apply an electric field was restricted to quantum mechanical methods. In molecular mechanics, the electric field interacts with the atom charges on each of the atoms. For MM+, which has options for either atom charges or bond dipoles, the electric field interacts only with the atomic charges.

## Ability to Explore "Particle-in-a Box" Wave Functions

A new capability that might be thought to be somewhat disjoint from the rest of HyperChem is its new ability to perform calculations of the energies and wavefunctions for relatively arbitrary shaped "Particles in a Box". This capability is likely to be extended later to include hydrogenic wavefunctions (spherical harmonics) as well as anharmonic and harmonic wavefunctions simple diatomics. These capabilities are part of a standard Physical Chemistry curriculum and are targets for HyperChem to enhance its applicability in the educational experience.

The particle in a box functionality is implemented as an Annotation to keep it separate from the normal molecular modeling functions. The box is a box with infinite walls. Later it may be useful to extend this capability to finite boxes involving continuum (scattering) functions. A dialog box allows user to choose the width and scale of the box and the number (n) of partitions of the box into either constant or linear potentials. The bottom of the box is drawn by the user in a very constrained way until the whole bottom, for the n partitions, is complete. At that point, the wave functions and energies levels are presented. These are simply annotations and can be treated exactly as any other annotation. The various components of the box can be deleted, as with other annotations. Selecting the dialog box again draws the walls for a new box to be drawn. Selecting New eliminates the box along with all other elements of the workspace.

## Procedure

1. Click on the Particle-in-a-Box menu to bring up the dialog box for choosing the parameters of the box.
2. Drag with the mouse from the far left to the right (and horizontal or slightly up or down) to draw the bottom of the first partition.
3. Continue for each partition.
4. When you reach the right hand side of the box, the program will compute the energy levels of the box.
5. Click on any of the energy levels to see the wavefunction.
6. Select New or select individually each of the elements of the box to delete them as annotations, or alternatively,
7. Click on the menu item again to choose a new box and new parameters for a different box.

## Ability to Select Desired Energy Units (kcal, kJ, or Hartree)

The default units for HyperChem are kcal/mol for energies and Angstroms for distances. Unless specifically indicated otherwise, this is basis of the units for the quantities presented. An exception to this is Electric and Magnetic fields where the unit of the field is atomic units. An electron one Bohr away from a point exerts an electric field of 1.0 a.u. Also dipole moments are routinely represented by Debye and orbital energies by Electron Volts.

With HyperChem 8 an attempt is made to rationalize the presentation of units. This is not yet completely unambiguous since units are not presented everywhere. The direction is to

(1) Indicate the units everywhere and allow a button to change the value from one unit to another unit, possibly more relevant or desired by the user.

(2) Allow the user to select the specific a set of units that are presented everywhere. This is partially implemented with regard to the status line. A Preferences setting allows the user to choose between kcal/mol, kJ/mol, and atomic units (a.u.). This generally affects the presentation on the status line but may not be uni-

**versal. Log files commonly give both kcal/mol and a.u. but do not yet use kJ/mol anywhere. The presentation of units will be further rationalized in future releases.**

*Ability to Select Desired Energy Units (kcal, kJ, or Hartree)*