
Q-Chem User's Guide

Version 2.0
January 2001

Version 2.0

January 2001

Q-Chem is a trademark of Q-Chem, Inc. All rights reserved.

The information in this document applies to version 2.0 of Q-Chem.

January 2001 User's Guide

This edition authored by:

Martin Head-Gordon
Peter Gill (chapter 8)

It is a revised and expanded version of the previous (1.2) edition, written by:

Dr. Jeremy P. Dombroski

Published by:

Q-Chem, Inc.
Four Triangle Lane
Suite 160
Export, PA 15632-9255

Customer Support:

Telephone: (724) 325-9969
Facsimile: (724) 325-9560
email: support@q-chem.com
website: <http://www.q-chem.com>

© Copyright 2000 Q-Chem, Inc. This document is protected under the U.S. Copyright Act of 1976 and state trade secret laws. Unauthorized disclosure, reproduction, distribution, or use is prohibited and may violate federal and state laws.

TABLE OF CONTENTS

TABLE OF CONTENTS	i
-------------------------	---

CHAPTER 1 INTRODUCTION 1

1.1	ABOUT THIS MANUAL	1
1.2	CHAPTER SUMMARIES	1
1.3	CONTACT INFORMATION	2
1.3.1	<i>General Inquiries and Sales</i>	2
1.3.2	<i>Customer Support</i>	2
1.4	Q-CHEM, INC.....	2
1.5	COMPANY MISSION	3
1.6	Q-CHEM FEATURES.....	3
1.6.1	<i>What's New in Q-Chem 2.0?</i>	3
1.6.2	<i>Summary of Methods and Features</i>	4
1.7	HIGHLIGHTED FEATURES.....	5
1.7.1	<i>Theoretical Advancements</i>	5
1.7.2	<i>Graphical User Interface (GUI)</i>	6
1.8	CURRENT DEVELOPMENT AND FUTURE RELEASES.....	6
1.9	CITING Q-CHEM	7

CHAPTER 2 INSTALLATION 9

2.1	Q-CHEM INSTALLATION REQUIREMENTS	9
2.1.1	<i>Execution Environment</i>	9
2.1.2	<i>Hardware Platforms and Operating Systems</i>	9
2.1.3	<i>Memory and Hard Disk</i>	9
2.2	INSTALLING Q-CHEM.....	10
2.3	LICENSE REQUIREMENTS.....	10
2.4	ENVIRONMENT VARIABLES	10
2.5	USER ACCOUNT ADJUSTMENTS	11
2.5.1	<i>Example Login File Modifications</i>	11
2.6	THE QCHEM.SETUP FILE	12
2.7	RUNNING Q-CHEM	12
2.8	TESTING AND EXPLORING Q-CHEM	13

CHAPTER 3 Q-CHEM INPUTS 15

3.1	GENERAL FORM.....	15
3.2	MOLECULAR COORDINATE INPUT (\$MOLECULE)	17
3.2.1	<i>Reading Molecular Coordinates From a Previous Calculation</i>	18
3.2.2	<i>Example</i>	19
3.2.3	<i>Reading molecular Coordinates from another file</i>	19
3.3	CARTESIAN COORDINATES.....	20
3.3.1	<i>Examples:</i>	20

3.4	Z-MATRIX COORDINATES	21
3.4.1	<i>Dummy atoms</i>	23
3.5	JOB SPECIFICATION: THE \$REM ARRAY CONCEPT	24
3.6	\$REM ARRAY FORMAT IN Q-CHEM INPUT	25
3.7	MINIMUM \$REM ARRAY REQUIREMENTS.....	26
3.8	COMMENTS (\$COMMENT).....	26
3.9	USER-DEFINED BASIS SET (\$BASIS).....	26
3.10	USER-DEFINED PSEUDOPOTENTIALS (\$ECP).....	26
3.11	GEOMETRY OPTIMIZATION WITH GENERAL CONSTRAINTS (\$OPT).....	27
3.12	USER-DEFINED OCCUPIED GUESS ORBITALS (\$OCCUPIED)	27
3.13	NATURAL BOND ORBITAL PACKAGE (\$NBO)	27
3.14	ADDITION OF EXTERNAL CHARGES (\$EXTERNAL_CHARGES).....	27
3.15	APPLYING A MULTIPOLE FIELD (\$MULTIPOLE_FIELD)	27
3.16	ORBITALS, DENSITIES AND ESPs ON A MESH (\$PLOTS).....	28
3.17	USER-DEFINED VAN DER WAALS RADII (\$VAN_DER_WAALS).....	28
3.18	USER-DEFINED EXCHANGE-CORRELATION DENSITY FUNCTIONALS (\$XC_FUNCTIONAL).....	28
3.16	MULTIPLE JOBS IN A SINGLE FILE: Q-CHEM BATCH JOB FILES	30
3.17	Q-CHEM TEXT INPUT SUMMARY	33
3.17.1	<i>Keyword Format Summary</i>	34
3.18	Q-CHEM OUTPUT FILE.....	39
3.19	Q-CHEM SCRATCH FILES	39
CHAPTER 4 SELF-CONSISTENT FIELD GROUND STATE METHODS		41
4.1	INTRODUCTION.....	41
4.1.1	<i>Overview of Chapter</i>	41
4.1.2	<i>Theoretical Background</i>	42
4.2	HARTREE-FOCK CALCULATIONS.....	46
4.2.1	<i>Hartree-Fock Equations</i>	46
4.2.2	<i>Basic Hartree-Fock Job Control</i>	48
4.2.3	<i>Additional Hartree-Fock Job Control Options</i>	51
4.2.4	<i>Examples</i>	53
4.2.5	<i>Symmetry</i>	54
4.3	DENSITY FUNCTIONAL THEORY	55
4.3.1	<i>Introduction</i>	55
4.3.2	<i>Kohn-Sham Density Functional Theory</i>	56
4.3.3	<i>Exchange-Correlation Functionals</i>	58
4.3.4	<i>DFT Numerical Quadrature</i>	59
4.3.5	<i>Consistency Check and Cutoffs for Numerical Integration</i>	61
4.3.6	<i>Basic DFT Job Control</i>	62
4.3.7	<i>User-Defined Density Functionals</i>	63
4.3.8	<i>Examples</i>	64
4.4	LARGE MOLECULES AND LINEAR SCALING METHODS.....	65
4.4.1	<i>Introduction</i>	65
4.4.2	<i>Continuous Fast Multipole Method (CFMM)</i>	66
4.4.3	<i>Linear Scaling Exchange (LinK) Matrix Evaluation</i>	68
4.4.4	<i>Incremental and Variable Thresh Fock Matrix Building</i>	69
4.4.5	<i>Examples</i>	71
4.5	SCF INITIAL GUESS	72
4.5.1	<i>Introduction</i>	72
4.5.2	<i>Simple Initial Guesses</i>	73
4.5.3	<i>Reading MOs from Disk</i>	74
4.5.4	<i>Modifying the Occupied Molecular Orbitals</i>	75

4.5.5	<i>Basis Set Projection</i>	78
4.5.6	<i>Examples</i>	79
4.6	CONVERGING SCF CALCULATIONS	81
4.6.1	<i>Introduction</i>	81
4.6.2	<i>Basic Convergence Control Options</i>	82
4.6.3	<i>Direct Inversion in the Iterative Subspace (DIIS)</i>	83
4.6.5	<i>Geometric Direct Minimization (GDM)</i>	85
4.6.5	<i>Direct Minimization (DM)</i>	86
4.6.6	<i>Maximum Overlap Method (MOM)</i>	87
4.6.7	<i>Examples</i>	89
4.7	UNCONVENTIONAL SCF CALCULATIONS	90
4.7.1	<i>Case Approximation</i>	90
4.7.2	<i>Polarized Atomic Orbital (PAO) Calculations</i>	91
4.8	GROUND STATE METHOD SUMMARY	93
4.9	REFERENCES AND FURTHER READING	94

CHAPTER 5 WAVEFUNCTION-BASED CORRELATION METHODS..... 97

5.1	INTRODUCTION.....	97
5.2	MØLLER-PLESSET PERTURBATION THEORY	99
5.2.1	<i>Introduction</i>	99
5.2.2	<i>Theoretical Background</i>	99
5.3	EXACT MP2 METHODS.....	101
5.3.1	<i>Algorithm</i>	101
5.3.2	<i>Algorithm Control and Customization</i>	103
5.3.3	<i>Examples</i>	105
5.4	LOCAL MP2 METHODS.....	106
5.4.1	<i>Local Triatomics in Molecules (TRIM) Model</i>	106
5.4.2	<i>EPAO Evaluation Options</i>	109
5.4.3	<i>Algorithm Control and Customization</i>	111
5.4.4	<i>Examples</i>	112
5.5	SELF-CONSISTENT PAIR CORRELATION METHODS.....	113
5.5.1	<i>Coupled Cluster Singles and Doubles (CCSD)</i>	113
5.5.2	<i>Quadratic Configuration Interaction (QCISD)</i>	114
5.5.3	<i>Optimized Orbital Coupled Cluster Doubles (OD)</i>	115
5.5.4	<i>Quadratic Coupled Cluster Doubles (QCCD)</i>	116
5.5.5	<i>Job Control Options</i>	118
5.5.6	<i>Examples</i>	120
5.6	NON-ITERATIVE CORRECTIONS TO COUPLED CLUSTER ENERGIES	121
5.6.1	<i>(T) Triples Corrections</i>	121
5.6.2	<i>(2) Triples and Quadruples Corrections</i>	121
5.6.3	<i>Job Control Options</i>	123
5.6.4	<i>Examples</i>	124
5.7	COUPLED CLUSTER ACTIVE SPACE METHODS	125
5.7.1	<i>Introduction</i>	125
5.7.2	<i>VOD and VOD(2) Methods</i>	127
5.7.3	<i>VQCCD</i>	127
5.7.4	<i>Convergence Strategies and More Advanced Options</i>	128
5.7.5	<i>Examples</i>	131
5.8	REFERENCES AND FURTHER READING	133

CHAPTER 6	EXCITED STATE METHODS	135
6.1	GENERAL EXCITED STATE FEATURES	135
6.2	NON-CORRELATED WAVEFUNCTION METHODS	136
6.2.1	<i>Single Excitation Configuration Interaction (CIS)</i>	136
6.2.2	<i>Random Phase Approximation (RPA)</i>	138
6.2.3	<i>Extended CIS (XCIS)</i>	138
6.2.4	<i>Basic Job Control Options</i>	140
6.2.5	<i>Customization</i>	142
6.2.6	<i>CIS Analytical Derivatives</i>	144
6.2.7	<i>Examples</i>	146
6.3	TIME-DEPENDENT DENSITY FUNCTIONAL THEORY (TDDFT).....	149
6.3.1	<i>A Brief Introduction to TDDFT</i>	149
6.3.2	<i>Job Control for TDDFT</i>	149
6.3.3	<i>Examples</i>	150
6.4	EXCITED STATE CORRELATED METHODS.....	151
6.4.1	<i>CIS(D)</i>	151
6.4.2	<i>CIS(D) Job Control</i>	152
6.4.3	<i>Coupled Cluster Excited State Methods</i>	154
6.4.4	<i>Coupled Cluster Excited State Job Control</i>	155
6.4.5	<i>Examples</i>	159
6.5	ATTACHMENT/DETACHMENT DENSITY ANALYSIS	160
6.5	FURTHER READING	162
CHAPTER 7	BASIS SETS	163
7.1	INTRODUCTION.....	163
7.2	BUILT-IN BASIS SETS.....	164
7.2.1	<i>Features</i>	164
7.2.2	<i>Job Control</i>	164
7.3	BASIS SET SYMBOLIC REPRESENTATION	165
7.3.1	<i>Customization</i>	168
7.4	USER-DEFINED BASIS SETS (\$BASIS)	169
7.4.1	<i>Introduction</i>	169
7.4.2	<i>Features</i>	170
7.4.3	<i>Job Control</i>	170
7.4.4	<i>Example</i>	171
7.5	FORMAT FOR USER-DEFINED BASIS SETS	172
7.5.1	<i>Customization</i>	172
7.5.2	<i>Examples</i>	173
7.6	FURTHER READING	178
CHAPTER 8	EFFECTIVE CORE POTENTIALS	179
8.1	INTRODUCTION.....	179
8.2	BUILT-IN PSEUDOPOTENTIALS.....	180
8.2.1	<i>Overview</i>	180
8.2.2	<i>Job Control for Built-in ECP's</i>	180
8.2.3	<i>Combining Pseudopotentials</i>	181
8.2.4	<i>Examples</i>	181
8.3	USER-DEFINED PSEUDOPOTENTIALS	182
8.3.1	<i>Job Control for User-Defined ECP's</i>	182

8.3.2	<i>Example</i>	184
8.4	PSEUDOPOTENTIALS AND DENSITY FUNCTIONAL THEORY	185
8.4.1	<i>Example</i>	185
8.5	PSEUDOPOTENTIALS AND ELECTRON CORRELATION.....	186
8.5.1	<i>Example</i>	186
8.6	PSEUDOPOTENTIALS AND VIBRATIONAL FREQUENCIES	187
8.6.1	<i>Example</i>	187
8.7	A BRIEF GUIDE TO Q-CHEM'S BUILT-IN ECP'S.....	188
8.7.1	<i>The HWMB Pseudopotential at a Glance</i>	189
8.7.2	<i>The LANL2DZ Pseudopotential at a Glance</i>	190
8.7.3	<i>The SBKJC Pseudopotential at a Glance</i>	191
8.7.4	<i>The CRENBS Pseudopotential at a Glance</i>	192
8.7.5	<i>The CRENBL Pseudopotential at a Glance</i>	193
8.7.6	<i>The SRLC Pseudopotential at a Glance</i>	194
8.7.7	<i>The SRSC Pseudopotential at a Glance</i>	196
8.8	FURTHER READING	197

CHAPTER 9 MOLECULAR GEOMETRY CRITICAL POINTS..... 199

9.1	EQUILIBRIUM GEOMETRIES AND TRANSITION STRUCTURES	199
9.2	USER-CONTROLLABLE PARAMETERS.....	200
9.2.1	<i>Features</i>	200
9.2.2	<i>Job Control</i>	200
9.2.3	<i>Customization</i>	203
9.2.4	<i>Example</i>	206
9.3	CONSTRAINED OPTIMIZATION	206
9.3.1	<i>Introduction</i>	206
9.3.2	<i>Geometry Optimization with General Constraints</i>	207
9.3.3	<i>Frozen Atoms</i>	208
9.3.4	<i>Dummy Atoms</i>	208
9.3.5	<i>Dummy Atom Placement in Dihedral Constraints</i>	209
9.3.6	<i>Additional Atom Connectivity</i>	210
9.3.7	<i>Example</i>	211
9.3.8	<i>Summary</i>	212
9.4	FURTHER READING	212

CHAPTER 10 MOLECULAR PROPERTIES AND ANALYSIS..... 213

10.1	INTRODUCTION.....	213
10.2	CHEMICAL SOLVENT MODELS	213
10.2.1	<i>Onsager Dipole Continuum Solvent</i>	213
10.2.2	<i>Langevin Dipoles Solvation Model</i>	214
10.2.3	<i>Customizing Langevin Dipoles Solvation Calculations</i>	217
10.2.4	<i>Example</i>	218
10.3	WAVEFUNCTION ANALYSIS.....	219
10.3.1	<i>Population Analysis</i>	219
10.3.2	<i>Multipole Moments</i>	220
10.3.3	<i>Symmetry Decomposition</i>	221
10.4	VIBRATIONAL ANALYSIS.....	221
10.4.1	<i>Job Control</i>	221
10.4.2	<i>Example</i>	224
10.5	INTERFACE TO THE NBO PACKAGE	225

10.5.1	<i>Job Control</i>	225
10.6	PLOTTING DENSITIES AND ORBITALS	226
10.7	ELECTROSTATIC POTENTIALS	229
10.8	FURTHER READING	230
CHAPTER 11 EXTENDED CUSTOMIZATION		231
11.1	USER-DEPENDENT AND MACHINE-DEPENDENT CUSTOMIZATION	231
11.1.1	<i>.qchemrc and Preferences File Format</i>	232
11.1.2	<i>Recommendations</i>	233
11.2	Q-CHEM AUXILIARY FILES (\$QCAUX)	234
11.3	ADDITIONAL Q-CHEM OUTPUT	234
11.3.1	<i>Third Party FCHK File</i>	234
APPENDIX A GEOMETRY OPTIMIZATION WITH Q-CHEM		A-1
A.1	INTRODUCTION	A-1
A.2	THEORETICAL BACKGROUND	A-3
A.2.1	<i>The Newton-Raphson Step</i>	A-3
A.2.2	<i>The Eigenvector Following (EF) Algorithm</i>	A-5
A.3	DELOCALIZED INTERNAL COORDINATES	A-8
A.4	CONSTRAINED OPTIMIZATION	A-12
A.4.1	<i>Cartesian Coordinates</i>	A-12
A.4.2	<i>Delocalized internal coordinates</i>	A-15
A.5	GDIIS	A-17
A.6	REFERENCES	A-20
APPENDIX B AOINTS		B-1
B.1	INTRODUCTION	B-1
B.2	HISTORICAL PERSPECTIVE	B-1
B.3	AOINTS: CALCULATING ERIS WITH Q-CHEM	B-3
B.3.1	<i>Shell-Pair Data</i>	B-3
B.3.2	<i>Shell-Quartets and Integral Classes</i>	B-4
B.3.3	<i>Fundamental ERI</i>	B-5
B.3.4	<i>Angular Momentum Problem</i>	B-5
B.3.5	<i>Contraction Problem</i>	B-6
B.3.6	<i>Quadratic Scaling</i>	B-6
B.3.7	<i>Algorithm Selection</i>	B-7
B.3.8	<i>User Controllable Variables</i>	B-7
B.4	REFERENCES	B-8
APPENDIX C \$REM VARIABLE REFERENCE		C-1
C.1	INPUT DECK FORMAT	C-1
C.2	THEORETICAL CHEMICAL MODEL	C-2
C.3	GENERAL	C-6
C.4	SCF GROUND STATE CALCULATIONS	C-8
C.5	LARGE MOLECULES	C-15
C.6	CORRELATED METHODS	C-17

C.7	EXCITED STATES	C-29
C.8	MOLECULAR GEOMETRY CRITICAL POINTS.....	C-37
C.9	MOLECULAR PROPERTIES AND ANALYSIS	C-41
C.10	PRINTING	C-45

CHAPTER 1 INTRODUCTION

1.1 ABOUT THIS MANUAL

This manual is intended as a general purpose user's guide for Q-Chem, a modern electronic structure program. The manual contains background information that describes Q-Chem methods and user-selected parameters. It is assumed that the user has some familiarity with the UNIX environment, an ASCII file editor and a basic understanding of quantum chemistry.

The manual is divided into 11 chapters and 3 appendices which are briefly summarized below. After installing Q-Chem and making necessary adjustments to your user account, it is recommended that particular attention be given to Chapters Three and Four. The latter chapter has been formatted so that advanced users can quickly find the information they require, while supplying new users with a moderate level of important background information. This format has been maintained throughout the manual, and every attempt has been made to guide the user forward and backward to other relevant information so that a logical progression through this manual, while recommended, is not necessary.

1.2 CHAPTER SUMMARIES

- Chapter 1: General overview of the Q-Chem program, its features and capabilities, the people behind it and contact information
- Chapter 2: Procedures to install, test and run Q-Chem on your machine
- Chapter 3: Basic attributes of the Q-Chem command line input.
- Chapter 4: Running self-consistent field ground state calculations.
- Chapter 5: Running wavefunction-based correlation methods for ground states.
- Chapter 6: Running excited state calculations
- Chapter 7: Using Q-Chem's built-in basis sets and running user-defined basis sets
- Chapter 8: Using Q-Chem's effective core potential capabilities
- Chapter 9: Options available for determining potential surface critical points such as transition states and local minima
- Chapter 10: Techniques available for computing molecular properties and performing wavefunction analysis
- Chapter 11: Important customization options available to enhance user flexibility
- Appendix A: OPTIMIZE package used in Q-Chem for determining Molecular Geometry Critical Points
- Appendix B: Q-Chem's AOINTS library, which contains some of the fastest two-electron integral codes currently available
- Appendix C: *\$rem* variable reference

1.3 CONTACT INFORMATION

1.3.1 GENERAL INQUIRIES AND SALES

For general information regarding broad aspects and features of the Q-Chem program, see Q-Chem's WWW home page (<http://www.q-chem.com>). Alternatively, contact Q-Chem, Inc. headquarters:

Q-Chem, Inc.
Four Triangle Lane
Suite 160
Export, PA 15632-9255
Telephone: (724) 325-9969
Fax: (724) 325-9560
email: sales@q-chem.com
support@q-chem.com
info@q-chem.com

1.3.2 CUSTOMER SUPPORT

Full customer support is promptly provided though telephone or email for those customers who have purchased Q-Chem's maintenance contract. The maintenance contract offers free customer support and discounts on future releases and updates. For details of the maintenance contract see Q-Chem's home page (<http://www.q-chem.com>).

1.4 Q-CHEM, INC.

Q-Chem, Inc. is based in Export, Pennsylvania and was founded in 1993. Q-Chem's scientific contributors and board members includes leading quantum chemistry software developers - Martin Head-Gordon (Berkeley), Peter Gill (Cambridge), Fritz Schaefer (Georgia) and John Pople (Northwestern). The close coupling between leading university research groups, and Q-Chem Inc. ensures that the methods and algorithms available in Q-Chem are state-of-the-art.

In order to create this technology, the founders of Q-Chem, Inc. built entirely new methodologies from the ground up, using the latest algorithms and modern programming techniques. Since 1993, well over 50 man-years have been devoted to the development of the Q-Chem program. The author list of the program shows the full list of contributors to the current version, consisting of some 35 people.

1.5 COMPANY MISSION

The mission of Q-Chem, Inc. is to develop, distribute and support innovative quantum chemistry software for industrial, government and academic researchers in the chemical, petrochemical, biochemical, pharmaceutical and material sciences.

1.6 Q-CHEM FEATURES

Quantum chemistry methods have proven invaluable for studying chemical and physical properties of molecules. The Q-Chem system brings together a variety of advanced computational methods and tools in an integrated *ab initio* software package, greatly improving the speed and accuracy of calculations being performed. In addition, Q-Chem will accommodate far larger molecular structures than previously possible and with no loss in accuracy, thereby bringing the power of quantum chemistry to critical research projects for which this tool was previously unavailable.

1.6.1 WHAT'S NEW IN Q-CHEM 2.0?

In a word--- lots!! We are very proud of the many new features that this new release of Q-Chem contains. The main ones are listed below, but they are by no means all, and you as a user should also notice significant improvements in performance and robustness over earlier versions. So, without further ado, here is a list of the main new goodies, and who is primarily to thank for them.

- New algorithms for large-molecule density functional calculations
 - ◊ New J engine and J force engine by Yihan Shao (Berkeley)
 - ◊ LinK for exchange energies and forces by Christian Ochsenfeld (Mainz)
- Effective core potentials for energies and gradients (Chapter 8)
 - ◊ Highly efficient PRISM-based algorithms to evaluate ECP matrix elements
 - ◊ Wide range of ECP's and ECP basis sets built-in.
 - ◊ By Ross Adamson and Peter Gill (Cambridge and Nottingham)
- Analytical second derivatives for density functional theory calculations
 - ◊ Substantially more efficient than previous finite-differences of gradients
 - ◊ By Jing Kong (Q-Chem)
- Local second order Møller-Plesset (MP2) methods
 - ◊ Accurate and efficient TRIM method for energy evaluation
 - ◊ By Mike Lee and Martin Head-Gordon (Berkeley)
- Standard high level electron correlation methods (Chapter 5)
 - ◊ QCISD, QCISD(T), CCSD, CCSD(T), MP3, MP4
 - ◊ By Steve Gwaltney, Anna Krylov, David Sherrill and Ed Byrd (Berkeley)
- New high level electron correlation methods (Chapter 8, Chapter 9)
 - ◊ Optimized orbital coupled cluster (Anna Krylov, David Sherrill, Ed Byrd)
 - ◊ CCSD(2) and OD(2) methods (Steve Gwaltney)
 - ◊ Excited state coupled cluster methods for OD and VOD (Anna Krylov)

- Time-dependent density functional theory for excited states
 - ◊ Computationally inexpensive and often much more accurate than CIS
 - ◊ Developed by So Hirata and Cherri Hsu (Berkeley)
- Langevin dipoles solvation model
 - ◊ A leading continuum solvation treatment
 - ◊ Developed by Jan Florian and Arieh Warshel (USC)

1.6.2 SUMMARY OF METHODS AND FEATURES

- Linear scaling methods for Hartree-Fock and DFT calculations
 - ◊ CFMM for Coulomb interactions (energies and gradients)
 - ◊ LinK for exchange interactions (energies and gradients)
 - ◊ Linear scaling exchange-correlation quadrature
- Local, Gradient-Corrected and Hybrid DFT functionals
 - ◊ Slater, Becke, GGA91 and Gill '96 exchange functionals
 - ◊ VWN, PZ81, Wigner, Perdew86, LYP and GGA91 correlation functionals
 - ◊ EDF1 exchange-correlation functional
 - ◊ B3LYP, B3P and user-definable hybrids
 - ◊ Analytical gradients and analytical frequencies
- Post-Hartree-Fock wavefunction-based electron correlation methods
 - ◊ Efficient semidirect MP2 energies and gradients
 - ◊ Local MP2 for energies using the TRIM and DIM models
 - ◊ MP3, MP4, QCISD, CCSD energies
 - ◊ OD and QCCD energies and analytical gradients
 - ◊ QCISD(T), CCSD(T) and OD(T) energies
 - ◊ CCSD(2) and OD(2) energies
 - ◊ active space coupled cluster methods: VOD, VQCCD, VOD(2)
- Extensive excited state capabilities
 - ◊ CIS energies, analytical gradients and analytical frequencies
 - ◊ CIS(D) energies
 - ◊ Time-dependent density functional theory energies (TDDFT)
 - ◊ Coupled cluster excited state energies (OD, and VOD)
- Evaluation and visualization of molecular properties
 - ◊ Langevin dipoles solvation model
 - ◊ Evaluate densities, electrostatic potentials, orbitals over cubes for plotting
 - ◊ Natural Bond Orbital (NBO) analysis
 - ◊ Attachment-detachment densities for excited states via CIS, TDDFT
 - ◊ Vibrational analysis after evaluation of the nuclear coordinate Hessian
- High performance geometry and transition structure optimization
 - ◊ Optimizes in Cartesian, Z-matrix or delocalized internal coordinates
 - ◊ Impose bond angle, dihedral angle (torsion) or out-of-plane bend constraints
 - ◊ Freezes atoms in Cartesian coordinates
 - ◊ Constraints do not need to be satisfied in the starting structure
 - ◊ Geometry optimization in the presence of fixed point charges

1.7 HIGHLIGHTED FEATURES

Developed by Q-Chem, Inc. and its collaborators, fundamental features include COLD PRISM, CFMM, CIS(D), OPTIMIZE packages. The features, which are highlighted below, are elaborated in later relevant sections.

1.7.1 THEORETICAL ADVANCEMENTS

COLD PRISM

The COLD PRISM is the latest in a number of high performance two-electron integral algorithms developed by Peter Gill and his collaborators at Massey University and the University of Cambridge. The development of COLD PRISM began with the realization that all methods for computing two-electron integral matrix elements involve four steps (represented by the COLD acronym), namely - contraction (C), operator (O), momentum (L) and density (D). This has culminated in the unification and augmentation of the previous PRISM and J engine methodologies into a generalised scheme, for the construction of two-electron matrix elements from shell-pair data. New in version 2.0 are extensions to the PRISM scheme to permit highly efficient evaluation of the matrix elements associated with effective core potentials.

Continuous Fast Multipole Method (CFMM)

One of the main driving forces in the evolution of Q-Chem is the implementation of the Continuous Fast Multipole Method (CFMM) developed by Chris White at the University of California at Berkeley. This enables Q-Chem to calculate the electronic Coulomb interactions (the rate-limiting step in large DFT calculations) in less time than other programs, and the time saved actually increases as the molecule becomes larger. New in version 2.0 is an improved treatment of the short-range interactions, developed by Yihan Shao at Berkeley, that significantly speeds up energy evaluation and dramatically speeds up force evaluation, with no loss of accuracy.

Local MP2

Q-Chem's local MP2 methods are unique, and were developed by Michael Lee, Paul Maslen and Martin Head-Gordon at Berkeley. Unlike other local correlation methods these satisfy all the properties of a theoretical model chemistry, and yield strictly continuous potential energy surfaces. Local MP2 reduces disk requirements compared to conventional MP2 by a factor proportional to the number of atoms in the molecule, and permits calculations in the 1000 to 1500 basis function range on workstations.

High Level Coupled Cluster Methods

Q-Chem's coupled cluster capabilities have been developed new from the ground up by Anna Krylov (USC) and David Sherrill (Georgia Tech) while they were postdocs in the research group of Martin Head-Gordon at Berkeley. In addition to conventional methods such as QCISD, CCSD, CCSD(T), Q-Chem also contains novel optimized orbital coupled cluster methods developed by Krylov, Sherrill and Ed Byrd at Berkeley, that can be performed in active spaces. Additionally new high level methods developed by Steve

Gwaltney in Head-Gordon's group are available exclusively in Q-Chem. These methods, denoted as CCSD(2) and OD(2), are superior to CCSD(T) and QCISD(T) for problems involving bond-breaking and radicals.

CHEMSOL

The Q-Chem program incorporates the CHEMSOL package developed by Jan Florian and Arieh Warshel at the University of Southern California to treat solvation by the method of Langevin dipoles. This method has already been proven in many applications studies, and contains important physical features such as dielectric saturation that are not accurately captured by other commonly used continuum solvation models.

OPTIMIZE

The Q-Chem program incorporates the latest version of Jon Baker's OPTIMIZE package, containing a suite of state-of-the-art algorithms for geometry optimization including the extremely efficient delocalized internal coordinates. Dr. Baker wrote the optimization algorithms in the Spartan package and the optimization code in the Biosym-distributed versions of DMol, Turbomole and Zindo.

1.7.2 GRAPHICAL USER INTERFACE (GUI)

Linux Spartan

Under development jointly between Wavefunction (www.wavefun.com) and Q-Chem for release early in 2001, Linux Spartan will bring the ease of use of Wavefunction's graphical user interface together with the full version of Q-Chem as a back-end for electronic structure calculations in a single integrated package.

1.8 CURRENT DEVELOPMENT AND FUTURE RELEASES

All details of functionality currently under development, information relating to future releases, and patch information are regularly updated on the Q-Chem web page (<http://www.q-chem.com>). Users are referred to this page for updates on developments, release information and further information on ordering and licenses. For any additional information, please contact Q-Chem, Inc. headquarters.

1.9 CITING Q-CHEM

The official Q-Chem citation for this release is a journal article that has been written describing the main technical features of the program. The full citation for this article is:

“Q-Chem 2.0: A high-performance ab initio electronic structure program”,

J. Kong, C. A. White, A. I. Krylov, C. D. Sherrill, R. D. Adamson,
T. R. Furlani, M. S. Lee, A. M. Lee, S. R. Gwaltney, T. R. Adams,
C. Ochsenfeld, A. T. B. Gilbert, G. S. Kedziora, V. A. Rassolov,
D. R. Maurice, N. Nair, Y. Shao, N. A. Besley, P. E. Maslen,
J. P. Dombroski, H. Daschel, W. Zhang, P. P. Korambath,
J. Baker, E. F. C. Byrd, T. Van Voorhis, M. Oumi, S. Hirata,
C.-P. Hsu, N. Ishikawa, J. Florian, A. Warshel, B. G. Johnson,
P. M. W. Gill, M. Head-Gordon, and J. A. Pople,

J. Comput. Chem. (2000) **21**, 1532-1548.

CHAPTER 2 INSTALLATION

2.1 Q-CHEM INSTALLATION REQUIREMENTS

2.1.1 EXECUTION ENVIRONMENT

Q-Chem is shipped as a collection of executables and scripts for the computer systems you will run Q-Chem on. No compilation is needed. Once the package is installed, it is ready to run. The installation instructions are provided on the CD cover.

The software required to run Q-Chem on your platform is minimal and includes:

- a suitable operating system
- runtime libraries (usually provided with your operating system)
- perl, version 5.

2.1.2 HARDWARE PLATFORMS AND OPERATING SYSTEMS

Q-Chem will run on a range of UNIX-based computer systems, ranging from Pentium and Athlon based PCs running Linux, to high performance workstations and servers running other versions of UNIX. For the availability of a specific platform/operating system, please check Q-Chem web page at <http://www.q-chem.com/products/platforms.html>.

2.1.3 MEMORY AND HARD DISK

Memory

Q-Chem, Inc. has endeavored to minimize memory requirements and maximize the efficiency of its use. Still the larger the structure or the higher the level of theory, the more random access memory (RAM) is needed. Although Q-Chem can be run with 16 MB RAM, we recommend 64 MB as a minimum. Q-Chem also offers the ability for user control of important memory intensive aspects of the program, an important consideration for non-batch constrained multi-user systems. In general, the more memory your system has, the larger the calculations that become feasible.

Disk

The Q-Chem executables, shell scripts, auxiliary files, samples and documentation require between 360 to 400 MB of disk space, depending on the platform. The default Q-Chem output, which is printed to the designated output file, is usually only a few KB's. This will be exceeded, of course, in difficult geometry optimizations, and in cases where users invoke non-default print options. In order to maximize the capabilities of

your copy of Q-Chem, additional disk space is required for scratch files created during execution; these are normally automatically deleted on termination of a job. The amount of disk space required for scratch files depends critically on the type of job, the size of the molecule and the basis set chosen. Q-Chem uses direct methods for Hartree-Fock and density functional theory calculations, which do not require large amount of scratch disk space. Wave function-based correlation methods, such as MP2 and coupled cluster theory require substantial amounts of temporary (scratch) disk storage, and the faster the access speeds, the better these jobs will perform. With the low cost of disk drives, it is feasible to have between 10 and 100GB of scratch space available relatively inexpensively, as a dedicated file system for these large temporary job files. The more you have available, the larger the jobs that will be feasible, and, in the case of some jobs like MP2, the jobs will also run faster as two-electron integrals are computed less often.

2.2 INSTALLING Q-CHEM

Users are referred to the guide on the CD cover for installation instructions pertinent to the release and platform. Should any difficulties arise during installation, please refer to the Q-Chem website (<http://www.q-chem.com>, FAQ's, telephone and facsimile numbers) or directly contact Q-Chem customer support (email: support@q-chem.com) for assistance.

2.3 LICENSE REQUIREMENTS

In order to run Q-Chem you must obtain the necessary encrypted license password file. The license consists of two files. Place these files, filenames *qchem.license.dat* and *qchem.aux*, in the *\$QCAUX/license* directory.

Do not alter these files unless directed by Q-Chem, Inc.

2.4 ENVIRONMENT VARIABLES

Q-Chem requires three shell environment variables in order to run calculations.

QC	defines the location of the Q-Chem directory structure. The <i>qchem.install</i> shell script determines this automatically.
QCAUX	defines the location of the auxiliary information required by Q-Chem, which includes the license required to run Q-Chem. This defaults to <i>\$QC/aux</i> . The user may redefine this location.
QCSCRATCH	defines the directory in which all scratch files will be placed during a run. The directory and the files in it will be removed by the script <i>qchem</i> at the end of the job, unless the job is terminated abnormally or is invoked with three arguments. Note that many of the files can be

quite large, and it should be ensured that sufficient disk space is available. The *QCSCRATCH* directory should be periodically checked for scratch files remaining from abnormally terminated jobs. *QCSCRATCH* defaults to the working directory.

2.5 USER ACCOUNT ADJUSTMENTS

In order for individual users to run Q-Chem, their user environment must be modified as follows:

- User file access permissions must be set so that the user can read, write and execute the necessary Q-Chem files. It may be advantageous to create a Q-Chem User's UNIX group on your machine and recursively change the group ownership of the Q-Chem files to that of the new group.
- A few lines need to be added to user login files or to the system default login files. The Q-Chem environment variables need to be defined and the Q-Chem set up file needs to be initiated prior to use of Q-Chem (once, on login).

2.5.1 EXAMPLE LOGIN FILE MODIFICATIONS

For users using the csh shell (or equivalent), add the following lines to their home directory *.cshrc* file:

```
##### Q-Chem Configuration Begin #####
setenv QC directory_name
setenv QCAUX directory_name
setenv QCSCRATCH directory_name
if (-e ${QC}/bin/qchem.setup) source ${QC}/bin/qchem.setup
unset noclobber
##### Q-Chem Configuration End #####
```

For users using the Bourne shell (or equivalent), add the following lines to their home directory *.profile* file:

```
##### Q-Chem Configuration Begin #####
QC=directory_name; export QC
QCAUX=directory_name; export QCAUX
QCSCRATCH=directory_name; export QCSCRATCH
noclobber=""
if [ -e ${QC}/bin/qchem.setup.sh ] ; then
  . ${QC}/bin/qchem.setup.sh
fi
##### Q-Chem Configuration End #####
```

Alternatively, these lines can be added to system wide *profile* or *cshrc* files or their equivalents.

2.6 THE QCHEM.SETUP FILE

When sourced on login from the *.cshrc* (or *.profile*, or equivalent), the *qchem.setup(.sh)* file makes a number of changes to the operating environment to enable the user to fully exploit Q-Chem capabilities, without adversely affecting any other aspect of the login session. The file:

- defines a number of environment variables used by various parts of the Q-Chem program
- sets the default directory for *QCAUX*, if not already defined
- adjusts the *PATH* environment variable so that the user can access Q-Chem's executables from the users working directory

2.7 RUNNING Q-CHEM

Once installation is complete and any necessary adjustments are made to the user account, the user is now able to run Q-Chem. There are two ways to invoke Q-Chem:

- *qchem* command line shell script (if you have purchased Q-Chem as a stand-alone package. The simple format for command line execution is given below. The remainder of this manual covers the creation of input files in detail.
- Via a supported Graphical User Interface. If you find the creation of text-based input, and examination of the text output tedious and difficult (which, frankly, it can be), then Q-Chem can be invoked transparently through Wavefunction's Spartan user interface on some platforms. Contact Wavefunction (www.wavefun.com) or Q-Chem for full details of current availability.

Using the Q-Chem command line shell script, *qchem*, is straightforward provided Q-Chem has been correctly installed on your machine and the necessary environment variables have been set in *.cshrc* or *.profile* (or equivalent) login files. If done correctly, necessary changes will have been made to the *PATH* variable automatically on login so that Q-Chem can be invoked from your working directory. The *qchem* shell script can be used in either of the following ways:

```
qchem infile outfile
```

```
qchem infile outfile savename
```

```
qchem -save infile outfile savename
```

where *infile* is the name of a suitably formatted Q-Chem input file (detailed in Chapter 3, and the remainder of this manual), and the *outfile* is the name of the file to which Q-Chem will place the job output information.

Note: If the *outfile* already exists in the working directory, it will be overwritten.

The use of the *savename* command line variable allows the saving of a few key scratch files between runs and is necessary when instructing Q-Chem to read information from previous jobs. Otherwise Q-Chem deletes all the scratch files at the end of a run. The saved files are in *\$QCSCRATCH/savename/*, and include files with the current molecular geometry, the current molecular orbitals and density matrix, the current force constants (if available), etc.

The `-save` option in conjunction with *savename* means that all temporary files are saved, rather than just the few essential files described above. Normally this is not required.

The name of the input parameters *infile*, *outfile* and *save* can be chosen at the discretion of the user (usual UNIX file and directory name restrictions apply). It may be helpful to use the same jobname for *infile* and *outfile*, but with varying suffixes. For example:

```
localhost-1> qchem water.in water.out &
```

invokes Q-Chem where the input is taken from *water.in* and the output is placed into *water.out*. The `&` places the job into the background so that you may continue to work in the current shell.

```
localhost-2> qchem water.com water.log water &
```

invokes Q-Chem where the input is assumed to reside in *water.com*, the output is placed into *water.log* and the key scratch files are saved in a directory *\$QCSCRATCH/water/*.

2.8 TESTING AND EXPLORING Q-CHEM

Q-Chem is shipped with a small number of test jobs, which are situated in the *\$QC/samples* directory. If you wish to test your version of Q-Chem, run the test jobs in the *samples* directory and compare the output files with the reference files (suffixed *.ref*) of the same name.

These test jobs are not an exhaustive quality control test (a small subset of the test suite used at Q-Chem, Inc.), but they should all run correctly on your platform. However, if any fault is identified in these or any output files created by your version, do not hesitate to contact customer service immediately.

These jobs are also an excellent way to begin learning about Q-Chem's text-based input and output formats in detail. In many cases you can use these inputs as starting points for building your own input files, if you wish to avoid reading the rest of this manual!

CHAPTER 3 Q-CHEM INPUTS

3.1 GENERAL FORM

A graphical interface is the simplest way to control Q-Chem. However, the low level command line interface is available to enable maximum customization and user exploitation of all Q-Chem features. The command line interface requires a Q-Chem input file which is simply an ASCII text file. This input file can be created using your favorite editor (*e.g.* vi, emacs, jot, *etc.*) following the basic steps outlined in the next few chapters.

Q-Chem's input mechanism uses a series of **keywords** to signal user input sections of the input file. As required, the Q-Chem program searches the input file for supported keywords. When Q-Chem finds a keyword, it then reads the section of the input file beginning at the keyword until that keyword section is terminated *\$end*. A short description of all Q-Chem keywords is provided in Figure 3.1. The user **must** understand the function and format of the *\$molecule* (section 3.2) and *\$rem* (section 3.5) keywords, as these keyword sections are where the user places the molecular geometric information and job specification details.

The keywords *\$rem* and *\$molecule* are requisites of Q-Chem input files.

As each keyword has a different function, the format required for specific keywords varies somewhat, to account for the different specialised information (format requirements are summarised at the end of this chapter). But, because each keyword in the input file is sought out independently by the program as the information is required, the overall format requirements of Q-Chem input files are much less stringent. *e.g.*, it is not necessary to enter a user-defined basis set in a particular part of the input, if it is contained within the appropriate keyword (*\$basis*) section and in the correct format.

<code>\$molecule</code>	Contains the molecular coordinate input (input file requisite)
<code>\$end</code>	Terminates each keyword section
<code>\$rem</code>	Job specification and customization parameters (input file requisite)
<code>\$basis</code>	User-defined basis set information (see Chapter 7)
<code>\$comment</code>	User comments for inclusion into output file
<code>\$external_charges</code>	External charges and their positions
<code>\$ecp</code>	User-defined effective core potentials (see Chapter 8)
<code>\$multipole_field</code>	Details of a multipole field to apply
<code>\$nbo</code>	Natural Bond Orbital package
<code>\$occupied</code>	Guess orbitals to be occupied
<code>\$opt</code>	Constraint definitions for geometry optimizations
<code>\$plots</code>	Generate plotting information over a grid of points (see Chapter 10).
<code>\$van_der_waals</code>	User-defined atomic radii for Langevin dipoles solvation (see Chapter 10)
<code>\$xc_functional</code>	Details of user-defined DFT exchange-correlation functionals

Figure 3.1 Q-Chem user input section keywords

- Notes:**
- (1) Users are able to enter keyword sections in any order.
 - (2) Each keyword section must be terminated with the *\$end* keyword.
 - (3) It is not necessary to have all keywords in an input file.
 - (4) Each keyword section will be described below.
 - (5) The entire Q-Chem input is case-insensitive.

In general, users will need to enter variables for the *\$molecule* and *\$rem* keyword section and are encouraged to add a *\$comment* for future reference. The necessity of other keyword input will become apparent throughout the manual, and is summarized at the end of this Chapter.

See the Appendix and/or the *\$QC/samples* directory with your release for specific examples of Q-Chem input using the keywords in Figure 3.1.

The second general aspect of Q-Chem input, is that there are effectively four input sources:

1. User input file (required)
2. *.qchemrc* file in *\$HOME* (optional)
3. *preferences* files in *\$QC/config* (optional)
4. Internal program defaults and calculation results (built-in)

These are summarised in order of preference in Figure 3.2. Thus, the input mechanism offers a program default over-ride for *all* users, default override for *individual* users and, of course, the input file provided by the user overrides all defaults. Refer to Chapter 11 for details of *.qchemrc* and *preferences*.

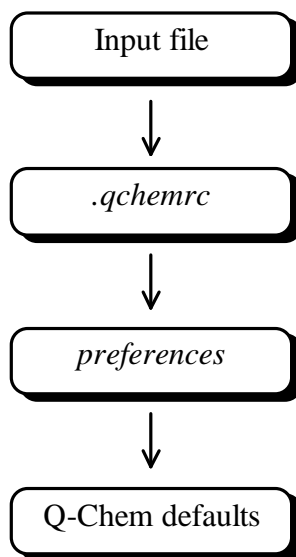


Figure 3.2 Diagram of input initialization override settings. The higher mechanism indicates override preference of lower mechanisms.

Currently, Q-Chem only supports the *\$rem* keyword in *.qchemrc* and *preferences* files.

3.2 MOLECULAR COORDINATE INPUT (\$MOLECULE)

The *\$molecule* section communicates to the program the charge, spin multiplicity and geometry of the molecule under investigation. The molecular coordinate input begins with two integers: the net charge and the spin multiplicity of the molecule. The net charge must be between -50 and 50, inclusively (0 for neutral molecules, 1 for cations, -1 for anions, *etc.*). The multiplicity must be between 1 and 10, inclusively (1 for a singlet, 2 for a doublet, 3 for a triplet, *etc.*). Each subsequent line of the molecular coordinate input corresponds to a single atom in the molecule (or dummy atom), irrespective of whether using *Z*-matrix internal coordinates or Cartesian coordinates.

Note: The coordinate system used for declaring an initial molecular geometry by default does not affect that used in a geometry optimization procedure. See the appendix which discusses the OPTIMIZE package in further detail.

Q-Chem begins all calculations by rotating and translating the user-defined molecular geometry into a Standard Nuclear Orientation whereby the centre of nuclear charge is placed at the origin. This is a standard feature of most quantum chemistry programs.

Note: Q-Chem ignores commas and equal signs, and requires all distances, positions and angles to be entered as Angstroms and degrees.

```
$molecule
0 1
O
H1  O    distance
H2  O    distance      H1  theta

distance = 1.0
theta = 104.5
$end
```

Figure 3.3 Example of molecular coordinate input for a water molecule in Z-matrix coordinates. Note that the *\$molecule* input begins with the charge and multiplicity.

3.2.1 READING MOLECULAR COORDINATES FROM A PREVIOUS CALCULATION

Often users wish to perform several calculations in quick succession, whereby the later calculations rely on results obtained from previous calculations. For example, geometry optimization at a low level of theory, followed by vibrational analysis and then, perhaps, single point energy at a higher level. Rather than having the user manually transfer the coordinates from the output of the optimization to the input file of a vibrational analysis or single point energy calculation, Q-Chem can transfer them directly from job to job.

To achieve this requires that:

1. The *READ* variable is entered into the molecular coordinate input
2. Scratch files from a previous calculation have been saved. These may be obtained *explicitly* by using the save option across multiple job runs as described below and in Chapter 2, or *implicitly* when running multiple calculations in one input file, as described later in this chapter.

```
$molecule  
READ  
$end
```

Figure 3.4 Reading a geometry from a prior calculation

3.2.2 EXAMPLE

```
localhost-1> qchem job1.in job1.out job1  
localhost-2> qchem job2.in job2.out job1
```

Example 3.1 In this example, the *job1* scratch files are saved in a directory *\$QCSCRATCH/job1* and are then made available to the *job2* calculation.

Note: The program must be instructed to read specific scratch files by the input of *job2*.

Users are also able to use the *READ* function for molecular coordinate input using Q-Chem's batch job file (see later in this chapter).

3.2.3 READING MOLECULAR COORDINATES FROM ANOTHER FILE

Users are able to use the *READ* function to read molecular coordinates from a second input file. The format for the coordinates in the second file follows that for standard Q-Chem input, and must be delimited with the *\$molecule* and *\$end* keywords.

```
$molecule  
READ      filename  
$end
```

Figure 3.5 Reading molecular coordinates from another file. *filename* maybe given either as the full file path or, path relative to the working directory.

3.3 CARTESIAN COORDINATES

Q-Chem can accept a list of N atoms and their $3N$ Cartesian coordinates. The atoms can be entered either as atomic numbers or atomic symbols where each line corresponds to a single atom. The Q-Chem format for declaring a molecular geometry using Cartesian coordinates (in Angstroms) is:

```
atom      x-coordinate  y-coordinate  z-coordinate
```

3.3.1 EXAMPLES:

```
$molecule
0 1
8          0.000000      0.000000      -0.212195
1          1.370265      0.000000      0.848778
1          -1.370265     0.000000      0.848778
$end
```

Example 3.2 Atomic number Cartesian coordinate input for H₂O.

```
$molecule
0 1
O          0.000000      0.000000      -0.212195
H          1.370265      0.000000      0.848778
H          -1.370265     0.000000      0.848778
$end
```

Example 3.3 Atomic symbol Cartesian coordinate input for H₂O.

Notes:

- Atoms can be declared by either atomic number or symbol
- Coordinates can be entered either as variables/parameters or real numbers
 - ◊ Variables/parameters can be declared in any order
 - ◊ A single blank line separates parameters from the atom declaration

Once all the molecular Cartesian coordinates have been entered, terminate the Molecular Coordinate Input with the *\$end* keyword.

3.4 Z-MATRIX COORDINATES

Z-matrix notation is one of the most common molecular coordinate input forms. The Z-matrix defines the positions of atoms relative to previously defined atoms using a length, an angle and a dihedral angle. Again, note that all bond lengths and angles must be in Angstroms and degrees.

Note: As with the Cartesian coordinate input method, Q-Chem begins a calculation by taking the user-defined coordinates and translating and rotating them into a Standard Nuclear Orientation.

The first three atom entries of a Z-matrix are different from the subsequent entries. The first Z-matrix line declares a single atom. The second line of the Z-matrix input declares a second atom, refers to the first atom and gives the distance between them. The third line declares the third atom, refers to either the first or second atom, gives the distance between them, refers to the remaining atom and gives the angle between them. All subsequent entries begin with an atom declaration, a reference atom and a distance, a second reference atom and an angle, a third reference atom and a dihedral angle. This can be summarised as:

1. First atom
2. Second atom, reference atom, distance
3. Third atom, reference atom A, distance between A and the third atom, reference atom B, angle defined by atoms A, B and the third atom
4. Fourth atom, reference atom A, distance, reference atom B, angle, reference atom C, dihedral angle (A, B, C and the fourth atom)
5. All subsequent atoms follow the same basic form as (4)

```
O1
O2 O1 OO
H1 O1 HO O2 HOO
H2 O2 HO O1 HOO H1 HOOH
```

Example 3.4 Z-matrix for hydrogen peroxide

Line 1 declares an oxygen atom (O1). Line 2 declares the second oxygen atom (O2), followed by a reference to the first atom (O1) and a distance between them denoted OO. Line 3 declares the first hydrogen atom (H1), indicates it is separated from the first oxygen atom (O1) by a distance HO and makes an angle with the second oxygen atom (O2) of HOO. Line 4 declares the fourth atom and the second hydrogen atom (H2), indicates it is separated from the second oxygen atom (O2) by a distance HO and makes an angle with the first oxygen atom (O1) of HOO and makes a dihedral angle with the first hydrogen atom (H1) of HOOH.

Some further points to note are:

- atoms can be declared by either atomic number or symbol
 - ◊ if declared by atomic number, connectivity needs to be indicated by Z-matrix line number
 - ◊ if declared by atomic symbol either
 - number similar atoms (*e.g.*, H1, H2, O1, O2 *etc.*) and refer connectivity using this symbol, or
 - indicate connectivity by the line number of the referred atom
- bond lengths and angles can be entered either as variables/parameters or real numbers
 - ◊ variables/parameters can be declared in any order
 - ◊ a single blank line separates parameters from the Z-matrix

All the following examples are equivalent in the information forwarded to the Q-Chem program.

```
$molecule
0 1
O1
O2 O1 OO
H1 O1 HO O2 HOO
H2 O2 HO O1 HOO H1 HOOH

OO=1.5
HOO =120.0
OH = 1.0
HOOH= 180.0
$end
```

Example 3.5 Using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```
$molecule
0 1
O1
O2 O1 1.5
H1 O1 1.0 O2 120.0
H2 O2 1.0 O1 120.0 H1 180.0
$end
```

Example 3.6 Not using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```

$molecule
0 1
8
8 1 OO
1 1 HO 2 HOO
1 2 HO 1 HOO 3 HOOH

OO=1.5
HOO=120.0
OH=1.0
HOOH=180.0
$end

```

Example 3.7 Using parameters to define bond lengths and angles, and referring to atom connectivities by line number.

```

$molecule
0 1
8
8 1 1.5
1 1 1.0 2 120.0
1 2 1.0 1 120.0 3 180.0
$end

```

Example 3.8 Referring to atom connectivities by line number, and entering bond length and angles directly.

Obviously, a number of the formats outlined above are less appealing to the eye and more difficult for us to interpret than the others, but each communicates *exactly* the same Z-matrix to the Q-Chem program.

3.4.1 DUMMY ATOMS

Dummy atoms are indicated by the identifier *X* and followed, if necessary, by an integer. (*e.g.*, *X1*, *X2*). Dummy atoms are often useful for molecules where symmetry axes and planes are not centred on a real atom, and have also been useful in the past for choosing variables for structure optimization and introducing symmetry constraints.

Note: Dummy atoms play no role in the quantum mechanical calculation, and are used merely for convenience in specifying other atomic positions or geometric variables.

3.5 JOB SPECIFICATION: THE \$REM ARRAY CONCEPT

The *\$rem* array is the means by which users convey to Q-Chem the type of calculation they wish to perform (level of theory, basis set, convergence criteria, *etc.*) The keyword *\$rem* signals the beginning of the overall job specification. Within the *\$rem* section the user inserts ***\$rem* variables** (one per line) which define the essential details of the calculation. The format for entering *\$rem* variables within the *\$rem* keyword section of the input is:

```
REM_VARIABLE   OPTION   [ Comment ]
```

Figure 3.6 Format for declaring *\$rem* variables in the *\$rem* keyword section of the Q-Chem input file. Note, Q-Chem only reads the first two arguments on each line of *\$rem*. All other text is ignored and can be used for placing short user comments.

The *\$rem* array stores all details required to perform the calculation, and details of output requirements. It provides the flexibility to customize a calculation to specific user requirements. If a default *\$rem* variable setting is indicated in this manual, the user does not have to declare the variable in order for the default to be initiated (*e.g.*, the default *JOBTYPE* is a single point energy (*SP*)). Thus, to perform a single point energy calculation, the user does **not** need to set the *\$rem* variable *JOBTYPE* to *SP*. However, to perform an optimisation, for example, it is necessary to override the program default by setting *JOBTYPE* to *OPT*.

A number of the *\$rem* variables have been set aside for internal program use, as they represent variables automatically determined by Q-Chem (*e.g.*, the number of atoms, the number of basis functions). These need not concern the user.

User communication to the internal program *\$rem* array comes in two general forms: (1) long term, machine-specific customization *via* the *.qchemrc* and *preferences* files (Chapter 9) and, (2) the Q-Chem input deck. There are many defaults already set within the Q-Chem program many of which can be overridden by the user. Checks are made to ensure that the user specifications are permissible (*e.g.*, integral accuracy is confined to 10^{-12}) and adjusted, if necessary. If adjustment is not possible, an error message is returned. Details of these checks and defaults will be given as they arise.

The user need not know all elements, options and details of the *\$rem* array in order to fully exploit the Q-Chem program. Many of the necessary elements and options are determined automatically by the program, or the optimized default parameters, supplied according to the user's basic requirements, available disk and memory, and the operating system and platform.

3.6 \$REM ARRAY FORMAT IN Q-CHEM INPUT

All data between the *\$rem* keyword and the next appearance of *\$end*, is assumed to be user *\$rem* array input. On a single line for each *\$rem* variable, the user declares the *\$rem* variable, followed by a blank space (tab stop inclusive) and then the *\$rem* variable option. It is recommended that a comment be placed following a space after the *\$rem* variable option. *\$rem* variables are case insensitive and a full listing is supplied in the appendix. Depending on the particular *\$rem* variable, *\$rem* options are entered either as a case-insensitive keyword, an integer value or logical identifier (true/false). The format for describing each *\$rem* variable in this manual is as follows:

REM_VARIABLE

Gives a short description of what the variable controls

VARIABLE:

Defines the variable as either INTEGER, LOGICAL or STRING

DEFAULT:

Describes Q-Chem's internal default, if any exist

OPTIONS:

Lists options available for the user

RECOMMENDATION:

Gives a quick recommendation

The end of the *\$rem* location declaration is signalled by the string *\$end*.

```
$rem
rem_variable      option      [user_comment ]
rem_variable      option      [user_comment ]
...
...
$end
```

Figure 3.7 General format of the *\$rem* section of the text input file.

Notes: (1) Erroneous lines will terminate the calculation
 (2) Tab stops can be used to format input
 (3) Entire lines can be commented by prefixing the line with an exclamation mark “!”

3.7 MINIMUM \$REM ARRAY REQUIREMENTS

Although Q-Chem provides defaults for most *\$rem* variables, the user will always have to stipulate a few others. For example, in a single point energy calculation, the minimum requirements will be *BASIS* (defining the basis set), *EXCHANGE* (defining the level of theory to treat exchange) and *CORRELATION* (defining the level of theory to treat correlation, if required). Furthermore, if a wavefunction-based correlation treatment (such as MP2) is used, HF is taken as the default for exchange.

```
$rem
BASIS           6-31G*      Just a small basis set
CORRELATION     MP2         MP2 energy
$end
```

Example 3.9 Example of minimum *\$rem* requirements to run an MP2/6-31G* energy calculation.

3.8 COMMENTS (\$COMMENT)

Users are able to add comments to the input file outside keyword input sections, which will be **ignored** by the program. This can be useful as reminders to the user, or perhaps, when teaching another user to set up inputs. Q-Chem has also provided a means of adding comments *via* the *\$comment* which will be placed **into** the output file. For example, an initial geometry obtained at another level of theory, or from a publication, may be used to calculate other properties using Q-Chem. The source of the initial geometry can then automatically be placed into the output file as a comment.

Note: Currently the entire input deck is copied to the top of the output file when a calculation commences.

3.9 USER-DEFINED BASIS SET (\$BASIS)

The *\$rem* variable *BASIS* (Chapter 7) allows the user to indicate that the basis set is being user-defined. The user-defined basis set is entered in the *\$basis* section of the input. For further details of entering a user-defined basis set, see chapter 7.

3.10 USER-DEFINED PSEUDOPOTENTIALS (\$ECP)

The *\$rem* variable *ECP* (Chapter 8) allows the user to indicate that pseudopotentials (effective core potentials) are being user-defined. The user-defined effective core potential is entered in the *\$ecp* section of the input. For further details, see chapter 8

3.11 GEOMETRY OPTIMIZATION WITH GENERAL CONSTRAINTS (\$OPT)

When a user defines the *JOBTYPE* to be a molecular geometry optimization, Q-Chem scans the input deck for the *\$opt* keyword. Distance, angle, dihedral and out-of-plane bend constraints imposed on any atom declared by the user in this section, are then imposed on the optimization procedure. See chapter 7 for details.

3.12 USER-DEFINED OCCUPIED GUESS ORBITALS (\$OCCUPIED)

It is sometimes useful for the occupied guess orbitals to be other than the lowest Nalpha (or Nbeta) orbitals. Q-Chem allows the occupied guess orbitals to be defined using the *\$occupied* keyword. The user defines occupied guess orbitals by listing the alpha orbitals to be occupied on the first line, and beta on the second (see chapter 4).

3.13 NATURAL BOND ORBITAL PACKAGE (\$NBO)

The default action in Q-Chem is not to run the NBO package. To turn the NBO package on, set the *\$rem* variable *NBO* to *ON*. To access further features of NBO, place standard NBO package parameters into a keyword section in the input file headed with the *\$nbo* keyword. Terminate the section with the termination string *\$end*.

3.14 ADDITION OF EXTERNAL CHARGES (\$EXTERNAL_CHARGES)

If the *\$external_charges* keyword is present, Q-Chem scans for a set of external charges to be incorporated into a calculation. The format for a set of external charges is the Cartesian coordinates, followed by the charge size, one charge per line. Charges and coordinates are in atomic units. Coordinates are in the Standard Nuclear Orientation.

```
$external_charges
x-coord1      y-coord1      z-coord1      charge1
x-coord2      y-coord2      z-coord2      charge2
x-coord3      y-coord3      z-coord3      charge3
...
$end
```

Figure 3.8 General format for incorporating a set of external charges.

3.15 APPLYING A MULTIPOLE FIELD (\$MULTIPOLE_FIELD)

Q-Chem has the capability to apply a multipole field to the molecule under investigation. Q-Chem scans the input deck for the *\$multipole_field* keyword, and reads each line (up

to the terminator keyword, *\$end*) as a single component of the applied field. The format is:


```
$multipole_field
field_component_1  value_1
field_component_2  value_2
...
$end
```

Figure 3.9 General format for imposing a multipole field.

The *field_component* is simply stipulated using the Cartesian representation *e.g.*, X, Y, Z, XX, XY, YY ... XXX, *etc.*, and the value or size of the imposed field is in atomic units.

3.16 ORBITALS, DENSITIES AND ESPs ON A MESH (\$PLOTS)

The *\$plots* part of the input permits the evaluation of molecular orbitals, densities, electrostatic potentials, transition densities, electron attachment and detachment densities on a user-defined mesh of points. For more details, see Chapter 10.

3.17 USER-DEFINED VAN DER WAALS RADII (\$VAN_DER_WAALS)

The *\$van_der_waals* section of the input enables the user to customize the Van der Waals radii that are important parameters in the Langevin dipoles solvation model. For more details, see chapter 10.

3.18 USER-DEFINED EXCHANGE-CORRELATION DENSITY FUNCTIONALS (\$XC_FUNCTIONAL)

The *EXCHANGE* and *CORRELATION* *\$rem* variables (Chapter 4) allow the user to indicate that the exchange-correlation density functional will be user-defined. The user defined exchange-correlation is to be entered in the *\$xc_functional* part of the input. The format is:

```
$XC_functional
X          exchange_symbol      coefficient
X          exchange_symbol      coefficient
.
.
.
C          correlation_symbol    coefficient
C          correlation_symbol    coefficient
.
.
.
K                               coefficient
```

Figure 3.10 General form for entering user-defined XC functionals.

Note: Coefficients are real numbers.

3.16 MULTIPLE JOBS IN A SINGLE FILE: Q-CHEM BATCH JOB FILES

It is sometimes useful to place a series of jobs into a single ASCII file. This feature is supported by Q-Chem and is invoked by separating jobs with the string “@@@” on a single line. All output is subsequently appended to the same output file for each job within the file.

Note: The first job will overwrite any existing output file of the same name in the working directory. Restarting the job will also overwrite any existing file.

In general, multiple jobs are placed in a single file for two reasons:

1. To use information from a prior job in a later job
2. To keep projects together in a single file

The “@@@” feature allows these objectives to be met, but the following points should be noted:

- Q-Chem reads all the jobs from the input file on initiation and stores them. The user cannot make changes to the details of jobs which have not been run post command line initiation.
- If any single job fails, Q-Chem proceeds to the next job in the batch file.
- No check is made to ensure that dependencies are satisfied, or that information is consistent (*e.g.*, an optimisation job followed by a frequency job; reading in the new geometry from the optimization for the frequency). No check is made to ensure that the optimization was successful. Similarly, it is assumed that both jobs use the same basis set when reading in MO coefficients from a previous job.
- Scratch files are saved between multi-job/single files runs (*i.e.*, using a batch file with “@@@” separators), but are deleted on completion **unless** a third *qchem* command line argument is supplied (see chapter 2).

Using batch files with the “@@@” separator is clearly most useful for cases relating to point 1 above. The alternative would be to cut and paste output, and/or use a third command line argument to save scratch files between separate runs.

For example, the following input file will optimize the geometry of H₂ at HF/6-31G*, calculate vibrational frequencies at HF/6-31G* using the optimized geometry and the self-consistent MO coefficients from the optimization and, finally, perform a single point energy using the optimized geometry at the MP2/6-311G(d,p) level of theory. Each job will use the same scratch area, reading files from previous runs as instructed.

```
$comment
Optimize H-H at HF/6-31G*
$end
```

```
$molecule
0 1
H
H 1 r
```

```
r = 1.1
$end
```

```
$rem
JOBTYPE          OPT          Optimise the bond length
EXCHANGE         HF
CORRELATION      NONE
BASIS            6-31G*
$end
```

```
@@@
$comment
Frequencies of H-H at HF/6-31G*
$end
```

```
$molecule
READ
$end
```

```
$rem
JOBTYPE          FREQ          Calculate vibrational frequencies
EXCHANGE         HF
CORRELATION      NONE
BASIS            6-31G*
SCF_GUESS        READ          Read the MOs from disk
$end
```

```
@@@
$comment
H-H at MP2/6-311G(d,p)//HF/6-31G*
$end
```

```
$molecule
READ
$end
```

```
$rem  
EXCHANGE          HF  
CORRELATION       MP2  
BASIS             6-311G(d,p)  
$end
```

Example 3.10 Example of using information from previous jobs in a single input file.

Notes: (1) Output is concatenated into the same output file.
(2) Only two arguments are necessarily supplied to the command line interface.

3.17 Q-CHEM TEXT INPUT SUMMARY

- Q-Chem text input file uses a series of **keywords**
- Q-Chem scans the input file for keywords, so they do not have to be placed in any particular order
- Each keyword represents a section of the input file
- Q-Chem reads in data, variables and options from keyword sections
- Each keyword section is terminated with *\$end*
- Not all keywords have to be entered, but *\$rem* and *\$molecule* are compulsory
- Input file is case-insensitive
- Multiple jobs in a single input file are separated by the string “@@@” on a single line

Keyword	Description
<i>\$molecule</i>	Signifies the beginning of the molecular coordinate input. Input file requisite
<i>\$end</i>	Terminates each keyword section
<i>\$rem</i>	Job specification and customization parameters. Input file requisite
<i>\$basis</i>	Indicates the beginning of the basis set information for user defined basis sets. (See chapter 7)
<i>\$ecp</i>	Indicates the beginning of pseudopotential information for user defined effective core potentials. (See chapter 8)
<i>\$comment</i>	All information placed in the section is incorporated into the Q-Chem output file. All other comments remain as input file user comments and are not read
<i>\$external_charges</i>	Section containing external charges and positions
<i>\$multipole_field</i>	Section contains details of a multipole field to apply
<i>\$nbo</i>	Placing Natural Bond Orbital package options
<i>\$occupied</i>	Guess orbitals to be occupied
<i>\$opt</i>	Constraint definitions for geometry optimizations
<i>\$plots</i>	Input for density, orbital, etc evaluation over meshes
<i>\$van_der_waals</i>	User-defined atomic radii for Langevin dipoles solvation
<i>\$xc_functional</i>	Details of user-defined DFT exchange-correlation functionals

Table 3.1 Description summary of all Q-Chem input keywords.

3.17.1 KEYWORD FORMAT SUMMARY

Keyword: \$molecule

Four methods:

1. Z-matrix (Angstroms and degrees)
\$molecule
 {Z-Matrix}
 {blank line, if parameters are being used}
 {Z-matrix parameters, if used}
\$end
2. Cartesian Coordinates (Angstroms)
\$molecule
 {Cartesian coordinates}
 {blank line, if parameter are being used}
 {Coordinate parameters, if used}
\$end
3. Read from a previous calculation
\$molecule
 READ
\$end
4. Read from a file
\$molecule
 READ filename
\$end

Keyword: \$rem

```
$rem  
rem_variable   rem_option       [user comment]  
...  
$end
```

Keyword: \$basis

```

$basis
atomic_symbol      0
ang_mom_sym        contraction_K      scaling
exp_1              coeff_1_Lmin      coeff_1_(Lmin+1)  ...  coeff_1_Lmax
exp_2              coeff_2_Lmin      coeff_2_(Lmin+1)  ...  coeff_2_Lmax
exp_3              coeff_3_Lmin      coeff_2_(Lmin+1)  ...  coeff_3_Lmax
.                  .                  .                  .      .
.                  .                  .                  .      .
exp_K              coeff_K_Lmin      coeff_K_(Lmin+1)  ...  coeff_K_Lmax
****

atomic_symbol      0
ang_mom_sym        contraction_K      scaling
exp_1              coeff_1_Lmin      coeff_1_(Lmin+1)  ...  coeff_1_Lmax
exp_2              coeff_2_Lmin      coeff_2_(Lmin+1)  ...  coeff_2_Lmax
exp_3              coeff_3_Lmin      coeff_2_(Lmin+1)  ...  coeff_3_Lmax
.                  .                  .                  .      .
.                  .                  .                  .      .
exp_K              coeff_K_Lmin      coeff_K_(Lmin+1)  ...  coeff_K_Lmax
****

...
****
$end

```

Keyword: \$ecp

```

$ecp
For each atom that will bear an ECP
    Chemical symbol for the atom
    ECP name ; the L value for the ECP ; number of core electrons removed
    For each ECP component (in the order unprojected,  $\hat{P}_0, \hat{P}_1, \dots, \hat{P}_{L-1}$ )
        The component name
        The number of Gaussians in the component
        For each Gaussian in the component
            The power of r ; the exponent ; the contraction coefficient
    A sequence of four asterisks (i.e. ****)
$end

```

Keyword: \$comment

```

$comment
{User comments - copied to output file}
$end

```


Keyword: \$external_charges (atomic units)

```
$external_charges
x-coord1      y-coord1      z-coord1      charge1
x-coord2      y-coord2      z-coord2      charge2
...
$end
```

Keyword: \$multipole_field (atomic units)

```
$multipole_field
field_component1  value1
field_component2  value2
...
$end
```

Keyword: \$nbo

```
$nbo
{Refer to Chapter 10 and NBO Program manual}
{must set $rem NBO to ON to initiate NBO package}
$end
```

Keyword: \$occupied

```
$occupied
 $\alpha_i$   $\alpha_j$   $\alpha_k$   $\alpha_l$  ... {alpha guess orbitals to be occupied}
 $\beta_1$   $\beta_m$   $\beta_n$   $\beta_o$  ... {beta guess orbitals to be occupied}
$end
```

Keyword: \$opt (Angstroms and degrees)

```
$opt
CONSTRAINT
stre      atom1      atom2      value
...
bend      atom1      atom2      atom3      value
...
outp      atom1      atom2      atom3      atom4      value
...
tors      atom1      atom2      atom3      atom4      value
...
linc      atom1      atom2      atom3      atom4      value
...
linp      atom1      atom2      atom3      atom4      value
...
ENDCONSTRAINT
FIXED
atom      coordinate_reference
...
ENDFIXED
DUMMY
idum      type      list_length      defining_list
...
ENDDUMMY
CONNECT
atom      list_length      list
...
ENDCONNECT
$end
```

Keyword: \$plots

```
$plots
A comment line here...
Nx  xmin  xmax      (# x points, x range)
Ny  ymin  ymax
Nz  zmin  zmax
NMO NRho  NTrans  NDA      (number of quantities to plot)
MO(1)  MO(2)  ...  MO(NMO)      (only if NMO > 0)
Rho(1)  Rho(2)  ...  Rho(NRho)      (only if NRho > 0)
Trans(1)  Trans(2)  ...  Trans(NTrans)      (only if Ntrans > 0)
DA(1)  DA(2)  ...  DA(NDA)      (only if NDA > 0)
$end
```

Keyword: \$van_der_waals

\$van_der_waals

1

atomic number

VdW-radius (Å)

...

\$end

(alternative format)

\$van_der_waals

2

sequential atom number

VdW-radius (Å)

...

\$end

Keyword: \$xc_functional

\$xc_functional

X

exchange_symbol

coefficient

X

exchange_symbol

coefficient

.

.

.

C

correlation_symbol

coefficient

C

correlation_symbol

coefficient

.

.

.

K

coefficient

\$end

3.18 Q-CHEM OUTPUT FILE

The Q-Chem output file is the file to which details of the job invoked by the user are printed. The type of information printed to this files depends on the type of job (single point energy, geometry optimisation *etc.*) and the $\$rem$ variable print levels. The general and default form is as follows:

- User input
- Q-Chem citation
- Molecular geometry in Cartesian coordinates
- Molecular point group, nuclear repulsion energy, number of alpha and beta electrons
- Basis set information (number of functions, shells and function pairs)
- SCF details (method, guess, optimization procedure)
- SCF iterations (for each iteration, energy and DIIS error is reported)
- {depends on job type}
- Molecular orbital symmetries
- Mulliken population analysis
- Cartesian multipole moments
- Job completion

Note: Q-Chem overwrites any existing output files in the working directory when it is invoked with an existing file as the output file parameter.

3.19 Q-CHEM SCRATCH FILES

The directory represented by the environment variable *QCSCRATCH* is the location Q-Chem places scratch files it creates on execution. Users may wish to use the information created for subsequent calculations. See chapter 2 for information on saving files.

CHAPTER 4 SELF-CONSISTENT FIELD GROUND STATE METHODS

4.1 INTRODUCTION

4.1.1 OVERVIEW OF CHAPTER

Theoretical chemical models [1] involve two principal approximations. One must specify the type of atomic orbital basis set used (See Chapters 7 and 8), and one must specify the way in which the instantaneous interactions (or correlations) between electrons are treated. Self-consistent field (SCF) methods are the simplest and most widely used electron correlation treatments, and contain as special cases all Kohn-Sham density functional methods and the Hartree-Fock method. This chapter summarizes Q-Chem's SCF capabilities, while the next chapter discusses more complex (and computationally expensive!) wavefunction-based methods for describing electron correlation. If you are new to quantum chemistry, we recommend that you also purchase an introductory textbook on the physical content and practical performance of standard methods [1,2,3].

This chapter is organized so that the earlier sections provide a mixture of basic theoretical background, and a description of the minimum number of program input options that must be specified to run SCF jobs. Specifically, this includes the sections on:

- Hartree-Fock theory
- Density functional theory. Note that all basic input options described in the Hartree-Fock section (4.2) also apply to density functional calculations.

Later sections introduce more specialized options that can be consulted as needed:

- Large molecules and linear scaling methods. A short overview of the ideas behind methods for very large systems and the options that control them.
- Initial guesses for SCF calculations. Changing the default initial guess is sometimes important for SCF calculations that do not converge.
- Converging the SCF calculation. This section describes the iterative methods available to control SCF calculations in Q-Chem. Altering the standard options is essential for SCF jobs that have failed to converge with the default options.
- Unconventional SCF calculations. Some nonstandard SCF methods with novel physical and mathematical features. Explore further if you are interested!

4.1.2 THEORETICAL BACKGROUND

In 1926 Schrödinger [4] combined the wave nature of the electron with the statistical knowledge of the electron *viz.* Heisenberg's Uncertainty Principle [5] to formulate an eigenvalue equation for the total energy of a molecular system. If we focus on stationary states and ignore the effects of relativity, we have the time independent, non-relativistic equation

$$H(\mathbf{R}, \mathbf{r})\Psi(\mathbf{R}, \mathbf{r}) = E(\mathbf{R})\Psi(\mathbf{R}, \mathbf{r}) \quad (0.1)$$

where the coordinates \mathbf{R} and \mathbf{r} refer to nuclei and electron position vectors respectively and H is the Hamiltonian operator (in atomic units)

$$H = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \frac{1}{2} \sum_{A=1}^M \frac{1}{M_A} \nabla_A^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N \sum_{j>i}^N \frac{1}{r_{ij}} + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{R_{AB}} \quad (0.2)$$

∇^2 is the Laplacian operator

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (0.3)$$

Z is the nuclear charge, M_A is the ratio of the mass of nucleus A to the mass of an electron, $R_{AB} = |\mathbf{R}_A - \mathbf{R}_B|$ is the distance between the A^{th} and B^{th} nucleus, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between the i^{th} and j^{th} electrons, $r_{iA} = |\mathbf{r}_i - \mathbf{R}_A|$ is the distance between the i^{th} electron and A^{th} nucleus, M is the number of nuclei and N is the number of electrons. E is an eigenvalue of H , equal to the total energy, and the wave function Ψ , is an eigenfunction of H .

Separating the motions of the electrons from that of the nuclei, an idea originally due to Born and Oppenheimer [6], yields the electronic Hamiltonian operator.

$$H_{elec} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N \sum_{j>i}^N \frac{1}{r_{ij}} \quad (0.4)$$

The solution of the corresponding electronic Schrödinger equation

$$H_{elec} \Psi_{elec} = E_{elec} \Psi_{elec} \quad (0.5)$$

gives the total electronic energy (E_{elec}) and electronic wave function Ψ_{elec} , which describes the motion of the electrons for a fixed nuclear position. The total energy is obtained by simply adding the nuclear-nuclear repulsion energy (fifth term of (0.2)) to the total electronic energy

$$E_{Tot} = E_{elec} + E_{nuc} \quad (0.6)$$

Solving the eigenproblem (0.5) yields a set of eigenfunctions ($\Psi_0, \Psi_1, \Psi_2 \dots$) with corresponding eigenvalues ($E_0, E_1, E_2 \dots$) where $E_0 \leq E_1 \leq E_2 \leq E_3 \dots$

Our interest lies in determining the lowest eigenvalue and associated eigenfunction which correspond to the ground state energy and wavefunction of the molecule. However, solving (0.5) for other than the most trivial systems is extremely difficult and the best we can do in practice is to find approximate solutions.

The first approximation used to solve (0.5) is that electrons move independently within molecular orbitals (MO), each of which describes the probability distribution of a single electron. Each MO is determined by considering the electron as moving within an average field of all the other electrons. Ensuring that the wave function is antisymmetric upon electron interchange, yields the well known Slater [7,8] determinant wavefunction

$$\Psi = \frac{1}{\sqrt{n!}} \begin{vmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_n(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_n(2) \\ \vdots & \vdots & & \vdots \\ \chi_1(n) & \chi_2(n) & \cdots & \chi_n(n) \end{vmatrix} \quad (0.7)$$

where χ_i , a spin orbital, is the product of a molecular orbital ψ_i and a spin function (α or β).

One obtains the optimum set of MOs by variationally minimizing the energy in what is called a “self-consistent field” or SCF approximation to the many-electron problem. The archetypal SCF method is the Hartree-Fock approximation, but these SCF methods also include Kohn-Sham Density Functional Theories (see section 4.5). All SCF methods lead to equations of the form

$$f(i)\chi(\mathbf{x}_i) = \epsilon\chi(\mathbf{x}_i) \quad (0.8)$$

where the Fock operator $f(i)$ can be written

$$f(i) = -\frac{1}{2}\nabla_i^2 + v^{eff}(i) \quad (0.9)$$

Here \mathbf{x}_i are spin and spatial coordinates of the i^{th} electron, χ are the spin orbitals and v^{eff} is the effective potential “seen” by the i^{th} electron which depends on the spin orbitals of the other electrons. The nature of the effective potential v^{eff} depends on the SCF methodology and will be elaborated on in further sections.

The second approximation usually introduced when solving (0.5), is the introduction of an Atomic Orbital (AO) basis. AOs (ϕ_μ) are usually combined linearly to approximate the true MOs. There are many standardized, atom-centered basis sets and details of these are discussed in Chapter 7.

After eliminating the spin components in (0.8) and introducing a finite basis,

$$\psi_i = \sum_{\mu} c_{\mu i} \phi_{\mu} \quad (0.10)$$

(0.8) reduces to the Roothaan-Hall matrix equation

$$\mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC} \quad (0.11)$$

where \mathbf{F} is the Fock matrix, \mathbf{C} is a square matrix of molecular orbital coefficients, \mathbf{S} is the overlap matrix with elements

$$S_{\mu\nu} = \int \phi_{\mu}(\mathbf{r})\phi_{\nu}(\mathbf{r}) d\mathbf{r} \quad (0.12)$$

and $\boldsymbol{\varepsilon}$ is a diagonal matrix of the orbital energies. Generalizing to an unrestricted formalism by introducing separate spatial orbitals for α and β spin in (0.7) yields the Pople-Nesbet [9] equations

$$\begin{aligned} \mathbf{F}^{\alpha}\mathbf{C}^{\alpha} &= \boldsymbol{\varepsilon}^{\alpha}\mathbf{S}\mathbf{C}^{\alpha} \\ \mathbf{F}^{\beta}\mathbf{C}^{\beta} &= \boldsymbol{\varepsilon}^{\beta}\mathbf{S}\mathbf{C}^{\beta} \end{aligned} \quad (0.13)$$

Solving (0.11) or (0.13) yields the restricted or unrestricted finite basis Hartree-Fock approximation. This approximation inherently neglects the instantaneous electron-electron correlations which are averaged out by the SCF procedure, and while the chemistry resulting from HF calculations often offers valuable qualitative insight, quantitative energetics are often poor. In principle, the DFT SCF methodologies are able to capture all the correlation energy (the difference in energy between the HF energy and the true energy). In practice, the best currently available density functionals perform well, but not perfectly and conventional HF-based approaches to calculating the correlation energy are still often required. They are discussed separately in the following chapter.

In self-consistent field methods, an initial guess is calculated for the MOs and, from this, an average field seen by each electron can be calculated. A new set of MOs can be obtained by solving the Roothaan-Hall or Pople-Nesbet eigenvalue equations ((0.11) or (0.13)). This procedure is repeated until the new MOs differ negligibly from those of the previous iteration.

Because they often yield acceptably accurate chemical predictions at a reasonable computational cost, self-consistent field methods are the corner stone of most quantum chemical programs and calculations. The formal costs of many SCF algorithms is $O(N^4)$, that is, they grow with the fourth power of the size (N) of the system. This is slower than the growth of the cheapest conventional correlated methods but recent work by Q-Chem, Inc. and its collaborators has dramatically reduced it to $O(N)$, an improvement that now allows SCF methods to be applied to molecules previously considered beyond the scope of *ab initio* [1] treatment.

In order to carry out an SCF calculation using Q-Chem, three *\$rem* variables need to be set:

- *BASIS* to specify the basis set (see chapter 7)
- *EXCHANGE* method for treating Exchange
- *CORRELATION* method for treating Correlation (defaults to *NONE*)

Types of ground state energy calculations currently available in Q-Chem are summarized in Table 4.1.

Calculation	\$rem Variable JOBTYP
Single point energy (default)	SINGLE_POINT, SP
Force	FORCE
Equilibrium Structure Search	OPTIMIZATION, OPT
Transition Structure Search	TS
Frequency	FREQUENCY, FREQ

Table 4.1 The type of calculation to be run by Q-Chem is controlled by the *\$rem* variable JOBTYP.

4.2 HARTREE-FOCK CALCULATIONS

4.2.1 HARTREE-FOCK EQUATIONS

As with much of the theory underlying modern quantum chemistry, the Hartree-Fock approximation was developed shortly after publication of the Schrödinger equation, but remained a qualitative theory until the advent of the computer. Although the HF approximation tends to yield qualitative chemical accuracy, rather than quantitative information, and is generally inferior to many of the DFT approaches available, it remains as a useful tool in the quantum chemist's toolkit. In particular, for organic chemistry, HF predictions of molecular structure are very useful.

Consider once more the Roothaan-Hall equations, (0.11) or (0.13), which can be traced back to the integro-differential equation (0.8) where the effective potential v^{eff} depends on the SCF methodology. In a restricted HF (RHF) formalism, the effective potential can be written as

$$v^{eff} = \sum_a^{N/2} [2J_a(1) - K_a(1)] - \sum_{A=1}^M \frac{Z_A}{r_{1A}} \quad (0.14)$$

where the Coulomb and exchange operators are defined as

$$J_a(1) = \int \psi_a^*(2) \frac{1}{r_{12}} \psi_a(2) d\mathbf{r}_2 \quad (0.15)$$

$$K_a(1)\psi_i(1) = \left[\int \psi_a^*(2) \frac{1}{r_{12}} \psi_i(2) d\mathbf{r}_2 \right] \psi_a(1) \quad (0.16)$$

respectively. By introducing an atomic orbital basis, we obtain Fock matrix elements

$$F_{\mu\nu} = H_{\mu\nu}^{core} + J_{\mu\nu} - K_{\mu\nu} \quad (0.17)$$

where the core Hamiltonian matrix elements

$$H_{\mu\nu}^{core} = T_{\mu\nu} + V_{\mu\nu} \quad (0.18)$$

consist of kinetic energy elements

$$T_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left[-\frac{1}{2} \nabla^2 \right] \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (0.19)$$

and nuclear attraction elements

$$V_{\mu\nu} = \int \phi_{\mu}(\mathbf{r}) \left[-\sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} \right] \phi_{\nu}(\mathbf{r}) d\mathbf{r} \quad (0.20)$$

The Coulomb and Exchange elements are given by

$$J_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} (\mu\nu | \lambda\sigma) \quad (0.21)$$

$$K_{\mu\nu} = \frac{1}{2} \sum_{\lambda\sigma} P_{\lambda\sigma} (\mu\lambda | \nu\sigma) \quad (0.22)$$

where the density matrix elements are

$$P_{\mu\nu} = 2 \sum_{a=1}^{N/2} C_{\mu a} C_{\nu a} \quad (0.23)$$

and the two electron integrals are

$$(\mu\nu | \lambda\sigma) = \iint \phi_{\mu}(\mathbf{r}_1) \phi_{\nu}(\mathbf{r}_1) \left[\frac{1}{r_{12}} \right] \phi_{\lambda}(\mathbf{r}_2) \phi_{\sigma}(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad (0.24)$$

Note: The formation and utilization of two-electron integrals is a topic central to the overall performance of SCF methodologies. The performance of the SCF methods in new quantum chemistry software programs can be quickly estimated simply by considering the quality of their atomic orbital integrals packages. See the appendix for details of Q-Chem's AOINTS package.

Substituting the matrix element (0.17) back into the Roothaan-Hall equations (0.11) and solving until self-consistency is achieved will yield the Restricted Hartree-Fock (RHF) energy and wavefunction. Alternatively, one could have adopted the unrestricted form of the wavefunction by defining an alpha and beta density matrix

$$P_{\mu\nu}^{\alpha} = \sum_{a=1}^{n_{\alpha}} C_{\mu a}^{\alpha} C_{\nu a}^{\alpha} \quad (0.25)$$

$$P_{\mu\nu}^{\beta} = \sum_{a=1}^{n_{\beta}} C_{\mu a}^{\beta} C_{\nu a}^{\beta}$$

and the total electron density matrix P^T is simply the sum of the alpha and beta density matrices. The unrestricted alpha Fock matrix

$$F_{\mu\nu}^{\alpha} = H_{\mu\nu}^{core} + J_{\mu\nu} - K_{\mu\nu}^{\alpha} \quad (0.26)$$

differs from the restricted one only in the exchange contributions where the alpha exchange matrix elements are given by

$$K_{\mu\nu}^{\alpha} = \sum_{\lambda}^N \sum_{\sigma}^N P_{\lambda\sigma}^{\alpha} (\mu\lambda | \nu\sigma) \quad (0.27)$$

4.2.2 BASIC HARTREE-FOCK JOB CONTROL

In brief, Q-Chem supports the 3 main variants of the Hartree-Fock method. They are:

- Restricted Hartree-Fock (RHF) for closed shell molecules. It is typically appropriate for closed shell molecules at their equilibrium geometry, where electrons occupy orbitals in pairs.
- Unrestricted Hartree-Fock (UHF) for open shell molecules. Appropriate for radicals with an odd number of electrons, and also for molecules with even numbers of electrons where not all electrons are paired (for example stretched bonds and diradicaloids).
- Restricted open shell Hartree-Fock (ROHF) for open shell molecules, where the alpha and beta orbitals are constrained to be identical.

Only 2 *\$rem* variables are required in order to run Hartree-Fock (HF) calculations:

- 1) *EXCHANGE* must be specified as HF.
- 2) A valid keyword for *BASIS* must be specified (see Chapter 7).

In slightly more detail, here is a list of basic *\$rem* variables associated with running Hartree-Fock calculations. See chapter 7 for further detail on basis sets available and chapter 8 for specifying effective core potentials.

JOBTYPE

Specifies the calculation

VARIABLE:

STRING

DEFAULT:

SP Single point energy

OPTIONS:

SP Single point energy

OPT Geometry Minimization

TS Transition Structure Search

FREQ Frequency Calculation

FORCE Analytical Force calculation

EXCHANGE

Specifies the exchange level of theory

VARIABLE:

STRING

DEFAULT:

No default

OPTIONS:

HF

Exact (Hartree-Fock)

RECOMMENDATION:

HF for Hartree-Fock calculations

BASIS

Sets the basis sets to be used

VARIABLE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen

User defined (*\$basis* keyword required)

Symbol

Use standard basis sets as per Chapter 7

RECOMMENDATIONS:

Consult literature and reviews to aid your selection

PRINT_ORBITALS

Prints orbital coefficients with atom labels in analysis part of output.

VARIABLE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

Prints occupied orbitals plus 5 virtuals.

NVIRT

Number of virtuals to print.

RECOMMENDATION:

Use TRUE unless more virtuals are desired.

THRESH

Cutoff for neglect of two electron integrals. $10^{-\text{THRESH}}$ ($\text{THRESH} \leq 14$)

VARIABLE:

INTEGER

DEFAULT:

8 for single point energies

10 for optimizations and frequency calculations

OPTIONS:

User-defined

RECOMMENDATION:

Should be at least three greater than *SCF_CONVERGENCE*. Increase for more significant figures, at greater computational cost.

SCF_CONVERGENCE

SCF is considered converged when the wavefunction error is less than $10^{-\text{SCF_CONVERGENCE}}$. Adjust the value of *THRESH* at the same time.

VARIABLE:

INTEGER

DEFAULT:

5 For single point energy calculations

8 for geometry optimizations and vibrational analysis

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost.

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals

VARIABLE:

LOGICAL

DEFAULT:

FALSE (Restricted) Closed-shell systems

TRUE (Unrestricted) Open-shell systems

OPTIONS:

FALSE Restricted open-shell HF (ROHF)

RECOMMENDATION:

Use default unless ROHF is desired. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break alpha-beta symmetry in the initial guess, by using *SCF_GUESS_MIX* or providing *\$occupied* information (see initial guess section).

4.2.3 ADDITIONAL HARTREE-FOCK JOB CONTROL OPTIONS

Listed below are a number of useful options to customize a Hartree-Fock calculation. This is only a short summary of the function of these *\$rem* variables. A full list of all SCF-related variables is provided in Appendix C. A number of other specialized topics (large molecules, customizing initial guesses, and converging the calculation) are discussed separately in Sections 4.4, 4.5, and 4.6 respectively.

INCORE_INTS_BUFFER

Controls the size of in-core integral storage buffer

VARIABLE:

INTEGER

DEFAULT:

2,000,000 words (1 word = 8 bytes)

OPTIONS:

User defined size; hardware dependent.

DIRECT_SCF

Controls direct SCF

VARIABLE:

LOGICAL

DEFAULT:

Determined by program

OPTIONS:

TRUE Forces direct SCF

FALSE Do not use direct SCF

RECOMMENDATION:

Use default; direct SCF switches off in-core integrals

METECO

Sets the threshold criteria for discarding shell-pairs

VARIABLE:

INTEGER

DEFAULT:

2 Discard shell-pairs below $10^{-\text{THRESH}}$

OPTIONS:

1 Discard shell-pairs four orders of magnitude below machine precision

2 Discard shell-pairs below $10^{-\text{THRESH}}$

RECOMMENDATION:

Use default

SCF_PRINT

Controls level of output from SCF procedure to Q-Chem output file

VARIABLE:

INTEGER

DEFAULT:

0 Minimal, concise, useful and necessary output

OPTIONS:

0 Minimal, concise, useful and necessary output

1 Level 0 plus component breakdown of SCF electronic energy

2 Level 1 plus density, Fock and MO matrices on each cycle

3 Level 2 plus two-electron Fock matrix components (Coulomb, HF exchange and DFT exchange-correlation matrices) on each cycle

RECOMMENDATIONS:

Proceed with care; can result in *extremely* large output files at level 2 or higher. These levels are primarily for program debugging.

SCF_FINAL_PRINT

Controls level of output from SCF procedure to Q-Chem output file at the end of the SCF

VARIABLE:

INTEGER

DEFAULT:

0 No extra print out

OPTIONS:

0 No extra print out

1 Orbital Energies only

2 Level 1 plus MOs

3 Level 2 plus Fock and density matrices

4.2.4 EXAMPLES

Provided below are examples of Q-Chem input files to run ground state, Hartree-Fock single point energy calculations. See the appendix for more examples of Q-Chem input files.

```
$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH
```

```
OH = 1.2
HOH = 120.0
$end
```

```
$rem
JOBTYPE          SP          Single Point energy
EXCHANGE         HF          Exact HF exchange
CORRELATION      None        No correlation
BASIS            STO-3G      Basis set
$end
```

```
$comment
HF/STO-3G water single point calculation
$end
```

Example 4.1 Example Q-Chem input for a single point energy calculation on water. Note that the declaration of the single point *\$rem* variable and level of theory to treat correlation are redundant because they are the same as the Q-Chem defaults.

```

$molecule
0,2
3
$end

$rem
EXCHANGE      HF          Hartree-Fock
BASIS         6-311G      Basis set
$end

```

Example 4.2 UHF/6-311G calculation on the Lithium atom. Note that correlation and the job type were not indicated because Q-Chem defaults automatically to no correlation and single point energies. Note also that, since the number of alpha and beta electron differ, MOs default to an unrestricted formalism.

```

$molecule
0,2
3
$end

$rem
EXCHANGE      HF          Hartree-Fock
UNRESTRICTED  FALSE         Restricted MOs
BASIS         6-311G      Basis set
$end

```

Example 4.3 ROHF/6-311G calculation on the Lithium atom. Note again that correlation and the job type need not be indicated.

4.2.5 SYMMETRY

Symmetry is a powerful branch of mathematics and is often exploited in quantum chemistry, both to reduce the computational workload and to classify the final results obtained [10,11,12]. Q-Chem is able to determine the point group symmetry of the molecular nuclei and, on completion of the SCF procedure, classify the symmetry of molecular orbitals, and provide symmetry decomposition of kinetic and nuclear attraction energy (see Chapter 10).

Molecular systems possessing point group symmetry offer the possibility of large savings of computational time, by avoiding calculations of integrals which are equivalent. *i.e.*, those integrals which can be mapped on to one another under one of the symmetry operations of the molecular point group.

The Q-Chem default is to use symmetry to reduce computational time, when possible. Some algorithms, such as the CFMM, do not yet have symmetry efficiencies implemented and these cases the symmetry flag (*\$rem* variable *SYMMETRY*) is ignored.

SYMMETRY

Controls the use of efficiency through the use of point group symmetry

VARIABLE:

LOGICAL

DEFAULT:

TRUE Use symmetry when available

OPTIONS:

TRUE Use symmetry when available

FALSE Do not use symmetry

RECOMMENDATION:

Use default unless benchmarking

4.3 DENSITY FUNCTIONAL THEORY

4.3.1 INTRODUCTION

In recent years, Density Functional Theory [13,14,15] has emerged as an accurate alternative first-principles approach to quantum mechanical molecular investigations. DFT currently accounts for approximately 90% of all quantum chemical calculations being performed, not only because of its proven chemical accuracy, but also because of its relatively cheap computational expense. These two features suggest that DFT is likely to remain a leading method in the quantum chemist's toolkit well into the future. Q-Chem contains fast, efficient and accurate algorithms for all popular density functional theories, which make calculations on quite large molecules possible and practical.

DFT is primarily a theory of electronic ground state structures based on the electron density, $\rho(\mathbf{r})$, as opposed to the many-electron wavefunction $\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N)$. There are a number of distinct similarities and differences to traditional wavefunction approaches and modern DFT methodologies. Firstly, the essential building blocks of the many electron wavefunction are single-electron orbitals are directly analogous to the Kohn-Sham (see below) orbitals in the current DFT framework. Secondly, both the electron density and the many-electron wavefunction tend to be constructed *via* a SCF approach that requires the construction of matrix elements which are remarkably and conveniently very similar.

However, traditional approaches using the many electron wavefunction as a foundation must resort to a post-SCF calculation (Chapter 5) to incorporate correlation effects, whereas DFT approaches do not. Post-SCF methods, such as perturbation theory or coupled cluster theory are extremely expensive relative to the SCF procedure. On the other hand, the DFT approach is, in principle, exact, but in practice relies on modeling

the unknown exact exchange correlation energy functional. While more accurate forms of such functionals are constantly being developed, there is no systematic way to improve the functional to achieve an arbitrary level of accuracy. Thus, the traditional approaches offer the possibility of achieving an arbitrary level of accuracy, but can be computationally demanding, whereas DFT approaches offer a practical route but the theory is currently incomplete.

4.3.2 KOHN-SHAM DENSITY FUNCTIONAL THEORY

The Density Functional Theory by Hohenberg, Kohn and Sham [16,17] stems from the original work of Dirac [18], who found that the exchange energy of a uniform electron gas may be calculated exactly, knowing only the charge density. However, while the more traditional DFT constitutes a direct approach and the necessary equations contain only the electron density, difficulties associated with the kinetic energy functional obstructed the extension of DFT to anything more than a crude level of approximation. Kohn and Sham developed an indirect approach to the kinetic energy functional which transformed DFT into a practical tool for quantum chemical calculations.

Within the Kohn-Sham formalism [17], the ground state electronic energy, E , can be written as

$$E = E_T + E_V + E_J + E_{XC} \quad (0.28)$$

where E_T is the kinetic energy, E_V is the electron-nuclear interaction energy, E_J is the Coulomb self-interaction of the electron density $\rho(\mathbf{r})$ and E_{XC} is the exchange-correlation energy. Adopting an unrestricted format, the alpha and beta total electron densities can be written as

$$\begin{aligned} \rho_\alpha(\mathbf{r}) &= \sum_{i=1}^{n_\alpha} |\psi_i^\alpha|^2 \\ \rho_\beta(\mathbf{r}) &= \sum_{i=1}^{n_\beta} |\psi_i^\beta|^2 \end{aligned} \quad (0.29)$$

where n_α and n_β are the number of alpha and beta electron respectively and, ψ_i are the Kohn-Sham orbitals. Thus, the total electron density is

$$\rho(\mathbf{r}) = \rho_\alpha(\mathbf{r}) + \rho_\beta(\mathbf{r}) \quad (0.30)$$

which within a finite basis [19] is represented by

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}^T \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \quad (0.31)$$

The components of (0.28) can now be written as

$$\begin{aligned}
E_T &= \sum_{i=1}^{n_\alpha} \langle \psi_i^\alpha | -\frac{1}{2} \nabla^2 | \psi_i^\alpha \rangle + \sum_{i=1}^{n_\beta} \langle \psi_i^\beta | -\frac{1}{2} \nabla^2 | \psi_i^\beta \rangle \\
&= \sum_{\mu\nu} P_{\mu\nu}^T \langle \phi_\mu(\mathbf{r}) | -\frac{1}{2} \nabla^2 | \phi_\nu(\mathbf{r}) \rangle
\end{aligned} \tag{0.32}$$

$$\begin{aligned}
E_V &= -\sum_{A=1}^M Z_A \frac{\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{R}_A|} d\mathbf{r} \\
&= -\sum_{\mu\nu} P_{\mu\nu}^T \sum_A \langle \phi_\mu(\mathbf{r}) | \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} | \phi_\nu(\mathbf{r}) \rangle
\end{aligned} \tag{0.33}$$

$$\begin{aligned}
E_J &= \frac{1}{2} \langle \rho(\mathbf{r}_1) | \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} | \rho(\mathbf{r}_2) \rangle \\
&= \frac{1}{2} \sum_{\mu\nu} \sum_{\lambda\sigma} P_{\mu\nu}^T P_{\lambda\sigma}^T (\mu\nu | \lambda\sigma)
\end{aligned} \tag{0.34}$$

$$E_{XC} = \int f(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}), \dots) d\mathbf{r} \tag{0.35}$$

Minimizing E with respect to the unknown Kohn-Sham orbital coefficients yields a set of matrix equations exactly analogous to the UHF case

$$\begin{aligned}
\mathbf{F}^\alpha \mathbf{C}^\alpha &= \boldsymbol{\varepsilon}^\alpha \mathbf{S} \mathbf{C}^\alpha \\
\mathbf{F}^\beta \mathbf{C}^\beta &= \boldsymbol{\varepsilon}^\beta \mathbf{S} \mathbf{C}^\beta
\end{aligned} \tag{0.36}$$

where the Fock matrix elements are generalised to

$$\begin{aligned}
F_{\mu\nu}^\alpha &= H_{\mu\nu}^{core} + J_{\mu\nu} - F_{\mu\nu}^{XC\alpha} \\
F_{\mu\nu}^\beta &= H_{\mu\nu}^{core} + J_{\mu\nu} - F_{\mu\nu}^{XC\beta}
\end{aligned} \tag{0.37}$$

where $F_{\mu\nu}^{XC\alpha}$ and $F_{\mu\nu}^{XC\beta}$ are the exchange-correlation parts of the Fock matrices dependent on the exchange-correlation functional used. The Pople-Nesbet equations are obtained simply by allowing

$$F_{\mu\nu}^{XC\alpha} = K_{\mu\nu}^\alpha \tag{0.38}$$

and similarly for the beta equation.

Thus, the density and energy are obtained in a manner analogous to that for the Hartree-Fock method. Initial guesses are made for the MO coefficients and an iterative process applied until self consistency is obtained.

4.3.3 EXCHANGE-CORRELATION FUNCTIONALS

There are an increasing number of exchange and correlation functionals and hybrid DFT methods available to the quantum chemist, many of which are very effective. In short, there are two basic types of functionals: those based on the local spin density approximation (LSDA) and those based on generalized gradient approximations (GGA). Explicit definitions of each of these approximations vary amongst theoreticians and the reader is referred to the literature for further details.

Q-Chem includes the following LSDA functionals:

- Slater-Dirac (Exchange) [18]
- Vosko-Wilk-Nusair (Correlation) [20]
- Perdew-Zunger (Correlation) [21]
- Wigner (Correlation) [22]

the following GGA functionals

- Becke88 (Exchange) [23]
- Gill96 (Exchange) [24]
- Gilbert-Gill99 (Exchange) [25]
- Lee-Yang-Parr (Correlation) [26]
- Perdew86 (Correlation) [27]
- GGA91 (Exchange and correlation) [28]

In addition to the established density functionals, Q-Chem contains the recent Empirical Density Functional 1 (EDF1), developed by Adamson, Gill and Pople [29]. EDF1 is a combined exchange + correlation functional that is specifically adapted to yield good results with the relatively modest-sized 6-31+G* basis set, by direct fitting to thermochemical data. It has the interesting feature that exact exchange mixing was not found to be helpful with a basis set of this size. Furthermore, for a basis set of this size, the performance substantially exceeded the popular B3LYP functional, while the cost of the calculations is considerably lower because there is no need to evaluate exact (nonlocal) exchange. We recommend consideration of EDF1 instead of either B3LYP or BLYP for density functional calculations on large molecules, for which basis sets larger than 6-31+G* may be too computationally demanding.

Hybrid exchange-correlation functionals [30], whereby several different exchange and correlation functionals are combined linearly to form a new functional, have proven successful in a number of reported applications. However, since Hybrid functionals contain HF exchange they are more expensive than pure DFT functionals. Q-Chem has incorporated two of the most popular hybrid functionals, B3LYP [31] and B3PW91 [30], with the additional option for users to define their own hybrid functionals via the `$xc_functional` keyword (see user-defined functionals in Section 4.3.7, below).

Note: The hybrid functionals are not simply a pairing of an exchange and correlation functional, but are a combined exchange-correlation functional (*i.e.*, B-LYP and B3LYP vary in the correlation contribution in addition to the exchange part).

4.3.4 DFT NUMERICAL QUADRATURE

In practical DFT calculations, the forms of the approximate exchange-correlation functionals (0.35) used are quite complicated, such that the required integrals involving the functionals generally cannot be evaluated analytically. Q-Chem evaluates these integrals through numerical quadrature directly applied to the exchange-correlation integrand (*i.e.*, no fitting of the XC potential in an auxiliary basis is done). Q-Chem provides a standard quadrature grid by default which is sufficient for most purposes.

The quadrature approach in Q-Chem is generally similar to that found in many DFT programs. The multicenter XC integrals are first partitioned into "atomic" contributions using a nuclear weight function. Q-Chem uses the nuclear partitioning of Becke [32], though without the "atomic size adjustments". The atomic integrals are then evaluated through standard one-center numerical techniques.

Thus, the exchange-correlation energy E_{xc} (0.35) is obtained as

$$E_{xc} = \sum_A \sum_i w_{Ai} f(\mathbf{r}_{Ai}) \quad (0.39)$$

where the first summation is over the atoms and the second is over the numerical quadrature grid points for the current atom. The f function is the exchange-correlation functional. The w_{Ai} are the quadrature weights, and the grid points \mathbf{r}_{Ai} are given by

$$\mathbf{r}_{Ai} = \mathbf{R}_A + \mathbf{r}_i \quad (0.40)$$

where \mathbf{R}_A is the position of nucleus A , with the \mathbf{r}_i defining a suitable one-centre integration grid, which is independent of the nuclear configuration.

The single-centre integrations are further separated into radial and angular integrations. The radial part is treated by the Euler-Maclaurin scheme (this method, proposed by Handy [33], is based on the Euler-Maclaurin formula for summation of a series).

Angular quadrature rules may be characterized by their degree, which is the highest degree of spherical harmonics for which the formula is exact, and their efficiency, which is the number of spherical harmonics exactly integrated per degree of freedom in the formula. Q-Chem supports the following types of angular grids:

- **Lebedev**

These are specially constructed grids for quadrature on the surface of a sphere [34,35,36] based on the octahedral group. Lebedev grids of the following degrees are available:

- 3rd degree, 6 points
- 5th degree, 18 points
- 7th degree, 26 points
- 9th degree, 38 points
- 11th degree, 50 points
- 15th degree, 86 points
- 17th degree, 110 points
- 19th degree, 146 points
- 23rd degree, 194 points
- 29th degree, 302 points

Lebedev grids typically have efficiencies near one, with efficiencies greater than one in some cases.

- **Gauss-Legendre**

These are spherical product rules separating the two angular dimensions θ and ϕ . Integration in the θ dimension is carried out with a Gaussian quadrature rule derived from the Legendre polynomials (orthogonal on $[-1,1]$ with weight function unity), while the ϕ integration is done with equally spaced points.

A Gauss-Legendre grid is selected by specifying the total number of points, $2N_\theta^2$, to be used for the integration. This gives a grid with $2N_\theta$ ϕ -points, N_θ θ -points, and a degree of $2N_\theta-1$.

In contrast with Lebedev grids, Gauss-Legendre grids have efficiency of only $2/3$ (hence more Gauss-Legendre points are required to attain the same accuracy as Lebedev). However, since Gauss-Legendre grids of general degree are available, this is a convenient mechanism for achieving arbitrary accuracy in the angular integration if desired.

The default grid used in Q-Chem is the SG-1 standard quadrature grid [37]. This grid was designed to yield the performance of a large, accurate quadrature grid, but with as few points as possible for the sake of computational efficiency. This is accomplished by reducing the number of angular points in regions where sophisticated angular quadrature is not necessary, such as near the nuclei where the charge density is nearly spherically symmetric, while retaining large numbers of angular points in the valence region where angular accuracy is critical.

The SG-1 grid is derived in this fashion from a Euler-Maclaurin-Lebedev-(50,194) grid (*i.e.*, 50 radial points, and 194 angular points per radial point). This grid has been found to give numerical integration errors of the order of 0.2 kcal/mol for medium-sized molecules, including particularly demanding test cases such as isomerization energies of alkanes. This error is deemed acceptable since it is significantly smaller than the accuracy typically achieved by quantum chemical methods. In SG-1 the total number of points is reduced to approximately 1/4 of that of the original EML-(50,194) grid, with SG-1

generally giving the same total energies as EML-(50,194) to within a few microhartrees (0.01 kcal/mol). Therefore, the SG-1 grid is relatively efficient while still maintaining the numerical accuracy necessary for chemical reliability in the majority of applications.

4.3.5 CONSISTENCY CHECK AND CUTOFFS FOR NUMERICAL INTEGRATION

Whenever Q-Chem calculates numerical density functional integrals, the electron density itself is also integrated numerically as a test on the quality of the quadrature formula used. The deviation of the numerical result from the number of electrons in the system is an indication of the accuracy of the other numerical integrals. If the relative error in the numerical electron count reaches 0.01%, a warning is printed; this is an indication that the numerical XC results may not be reliable. If the warning appears at the first SCF cycle, it is probably not serious, because the initial-guess density matrix is sometimes not idempotent, as is the case with the SAD guess and the density matrix taken from a different geometry in a geometry optimization. If that is the case, the problem will be corrected as the idempotency is restored in later cycles. On the other hand, if the warning is persistent to the end of SCF iterations, then either a finer grid is needed, or choose an alternative method for generating the initial guess.

Users should be aware, however, of the potential flaws that have been discovered in some of the grids currently in use. Jarecki and Davidson [38], for example, have recently shown that correctly integrating the density is a necessary, but not sufficient, test of grid quality.

By default, Q-Chem will estimate the magnitude of various XC contributions on the grid and eliminate those determined to be numerically insignificant. Q-Chem uses specially developed cutoff procedures which permits evaluation of the XC energy and potential in only $O(N)$ work for large molecules, where N is the size of the system. This is a significant improvement over the formal $O(N^3)$ scaling of the XC cost, and is critical in enabling DFT calculations to be carried out on very large systems. In very rare cases, however, the default cutoff scheme can be too aggressive, eliminating contributions that should be retained; this is almost always signalled by an inaccurate numerical density integral. An example of when this could occur is in calculating anions with multiple sets of diffuse functions in the basis. As mentioned above, when an inaccurate electron count is obtained, it may be possible to remedy the problem by increasing the size of the quadrature grid.

Finally we note that early implementations of quadrature-based Kohn-Sham DFT employing standard basis sets were plagued by lack of rotational invariance. That is, rotation of the system yielded a significantly energy change. Clearly, such behavior is highly undesirable. Johnson *et al.* rectified the problem of rotational invariance by completing the specification of the grid procedure [39] to ensure that the computed XC energy is the same for any orientation of the molecule in any Cartesian coordinate system.

4.3.6 BASIC DFT JOB CONTROL

3 \$rem variables are required to run a DFT job: *EXCHANGE*, *CORRELATION* and *BASIS*. In addition, all of the basic input options discussed for Hartree-Fock calculations in Section 4.2.2, and the extended options discussed in Section 4.2.3 are all valid for DFT calculations. Below we list only the basic DFT-specific options.

EXCHANGE

Specifies the exchange functional or exchange-correlation functional for hybrids

VARIABLE:

STRING

DEFAULT:

No default exchange functional

OPTIONS:

HF	exact Hartree-Fock
Slater, S	Slater
Becke, B	Becke
Gill96, Gill	Gill 1996
GG99	Gilbert and Gill, 1999
Becke(EDF1), B(EDF1)	Becke (EDF1)
PW91, PW	Perdew
B3PW91, Becke3PW91, B3P	B3PW91 hybrid
B3LYP, Becke3LYP	B3LYP
B3LYP5	original B3LYP (using VWN5)
EDF1	EDF1
General, Gen	User defined combination of K, X and C (refer next section)

CORRELATION

Specifies the correlation functional

VARIABLE:

STRING

DEFAULT:

None No correlation

OPTIONS:

None	No correlation
VWN	Vosko-Wilk-Nusair parameterization #5
LYP	Lee-Yang-Parr (LYP)
PW91, PW	GGA91 (Perdew)
LYP(EDF1)	LYP(EDF1) parameterization
Perdew86, P86	Perdew 1986
PZ81, PZ	Perdew-Zunger 1981
Wigner	Wigner

XC_GRID

Specifies the type of grid to use for DFT calculations.

DEFAULT:

1 SG-1

OPTIONS:

1 SG-1

2 Low Quality

mn The first six integers correspond to *m* radial points and the second six integers correspond to *n* angular points where possible numbers of Lebedev angular points are listed in section 4.7.2

-mn The first six integers correspond to *m* radial points and the second six integers correspond to *n* angular points where the number of Gauss-Legendre angular points $n = 2N_{\theta}^2$

RECOMMENDATION:

SG-1 or larger.

4.3.7 USER-DEFINED DENSITY FUNCTIONALS

The format for entering user-defined exchange-correlation density functionals is one line for each component of the functional. Each line requires three variables: the first defines whether the component is an exchange or correlation functional by declaring an *X* or *C*, respectively. The second variable is the symbolic representation of the functional as used for the *EXCHANGE* and *CORRELATION* *\$rem* variables. The final variable is a real number corresponding to the contribution of the component to the functional. Hartree-Fock exchange contributions (required for hybrid density functionals) can be entered using only two variables (*K*, for HF exchange) followed by a real number.

```
$XC_functional
X          exchange_symbol      coefficient
X          exchange_symbol      coefficient
.
.
.
C          correlation_symbol    coefficient
C          correlation_symbol    coefficient
.
.
.
K          coefficient
$end
```

- Notes:** (1) Coefficients are real.
 (2) A user-defined functional does not require all *X*, *C* and *K* components.

4.3.8 EXAMPLES

```

$comment
B-LYP/STO-3G water single point calculation
$end

$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH

OH = 1.2
HOH = 120.0
$end

$rem
EXCHANGE          Becke          Becke88 exchange
CORRELATION       LYP            LYP correlation
BASIS             STO-3G         Basis set
$end

```

Example 4.4 Example Q-Chem input for a DFT single point energy calculation on water

```

$comment
EDF1/6-31+G* water single point calculation
$end

$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH

OH = 1.2
HOH = 120.0
$end

$rem
EXCHANGE          EDF1          EDF1 exchange-correlation
BASIS             6-31+G*       Basis set
$end

```

Example 4.5 Example Q-Chem input for a DFT single point energy calculation on water

4.4 LARGE MOLECULES AND LINEAR SCALING METHODS

4.4.1 INTRODUCTION

Construction of the effective Hamiltonian, or Fock matrix, has traditionally been the rate-determining step in self-consistent field calculations, due primarily to the cost of two-electron integral evaluation, even with the efficient methods available in Q-Chem (see AOINTS appendix). However, for large enough molecules, significant speedups are possible by employing recently developed linear-scaling methods for each of the nonlinear terms that can arise. Linear scaling means that if the molecule size is doubled, then the computational effort likewise only doubles. There are three computationally significant terms:

- 1) Electron-electron Coulomb interactions, for which Q-Chem incorporates the Continuous Fast Multipole Method (CFMM) discussed in Sec. 4.4.2
- 2) Exact exchange interactions, which arise in hybrid DFT calculations and Hartree-Fock calculations, for which Q-Chem incorporates the LinK method discussed in Section 4.4.3 below.
- 3) Numerical integration of the exchange and correlation functionals in DFT calculations, which we have already discussed in Section 4.3.4

Q-Chem supports energies and efficient analytical gradients for all three of these high performance methods to permit structure optimization of large molecules, as well as relative energy evaluation. Note that analytical second derivatives of SCF energies do not exploit these methods at present.

For the most part, these methods are switched on automatically as the program chooses, on the basis of whether they offer a significant speedup for the job at hand. Nevertheless it is useful to have a general idea of the key concepts behind each of these algorithms, and what input options are necessary to control them. That is the primary purpose of this section, in addition to briefly describing 2 more conventional methods for reducing computer time in large calculations in Section 4.4.4.

There is one other computationally significant step in SCF calculations, and that is diagonalization of the Fock matrix, once it has been constructed. This step scales with the cube of molecular size (or basis set size), with a small prefactor. So, for large enough SCF calculations (very roughly in the vicinity of 2000 basis functions and larger), diagonalization becomes the rate determining step. The cost of cubic scaling with a small prefactor at this point exceeds the cost of the linear scaling Fock build, which has a very large prefactor, and the gap rapidly widens thereafter. This sets an effective upper limit on the size of SCF calculation for which Q-Chem is useful at several thousand basis functions.

4.4.2 CONTINUOUS FAST MULTIPOLE METHOD (CFMM)

The quantum chemical Coulomb problem, perhaps better known as the DFT bottleneck, has been at the forefront of many research efforts throughout the 1990s. The quadratic computational scaling behavior conventionally seen in the construction of the Coulomb

matrix in DFT or HF calculations has prevented the application of *ab initio* methods to molecules containing many hundreds of atoms. Q-Chem, Inc., in collaboration with White and Head-Gordon at the University of California at Berkeley, and Gill at Massey University in New Zealand, were the first to develop the generalization of Greengard's Fast Multipole Method (FMM) [40] to Continuous charged matter distributions in the form of the CFMM, which is the first linear scaling algorithm for DFT calculations. This initial breakthrough has since lead to an increasing number of linear scaling alternatives and analogues, but for Coulomb interactions, the CFMM remains state of the art. There are two computationally intensive contributions to the Coulomb interactions which we discuss in turn:

- Long-range interactions, which are treated by the CFMM
- Short-range interactions, corresponding to overlapping charge distributions, which are treated by a specialized "J-matrix engine" together with Q-Chem's state-of-the art two-electron integral methods.

The Continuous Fast Multipole Method was the first implemented linear scaling algorithm for the construction of the \mathbf{J} matrix. In collaboration with Q-Chem, Inc., Dr. Chris White began the development of the CFMM by more efficiently deriving [41] the original Fast Multipole Method before generalizing to CFMM [42]. The generalization applied by White *et al.* allowed the principles underlying the success of the FMM to be applied to *arbitrary* (subject to constraints in evaluating the related integrals) continuous, but localized, matter distributions. White and co-workers further improved the underlying CFMM algorithm [43,44] then implemented it efficiently [45], achieving performance that is an order of magnitude faster than some competing implementations.

The success of the CFMM follows similarly with that of the FMM, in that the charge system is subdivided into a hierarchy of boxes. Local charge distributions are then systematically organized into multipole representations so that each distribution interacts with local expansions of the potential due to all distant charge distributions. Local and distant distributions are distinguished by a well-separated (WS) index, which is the number of boxes that must separate two collections of charges before they may be considered distant and can interact through multipole expansions; near-field interactions must be calculated directly. In the CFMM each distribution is given its own WS index and is sorted on the basis of the WS index, and the position of their space centers. The implementation in Q-Chem has allowed the efficiency gains of contracted basis functions to be maintained.

The CFMM algorithm can be summarized in five steps:

1. Form and translate multipoles.
2. Convert multipoles to local Taylor expansions.
3. Translate Taylor information to the lowest level.
4. Evaluate Taylor expansions to obtain the far-field potential.
5. Perform direct interactions between overlapping distributions.

Accuracy can be carefully controlled by due consideration of tree depth, truncation of the multipole expansion and the definition of the extent of charge distributions in accordance with a rigorous mathematical error bound. As a rough guide, 10 poles are adequate for single point energy calculations, while 25 poles yield sufficient accuracy for gradient calculations. Subdivision of boxes to yield a one-dimensional length of about 8 boxes works quite well for systems of up to about one hundred atoms. Larger molecular systems, or ones which are extended along one dimension, will benefit from an increase in this number. The program automatically selects an appropriate number of boxes by default.

For the evaluation of the remaining short-range interactions, Q-Chem incorporates efficient J-matrix engines, originated by White and Head-Gordon [46]. These are analytically exact methods that are based on standard two-electron integral methods, but with an interesting twist. If one knows that the two-electron integrals are going to be summed into a Coulomb matrix, one can ask whether they are in fact the most efficient intermediates for this specific task. Or, can one instead find a more compact and computationally efficient set of intermediates by folding the density matrix into the recurrence relations for the two-electron integrals. For integrals that are not highly contracted (i.e. are not linear combinations of more than a few Gaussians), the answer is a dramatic yes. This is the basis of the J-matrix approach, and Q-Chem includes the latest algorithm developed by Yihan Shao working with Martin Head-Gordon at Berkeley for this purpose. Shao's J engine is employed for both energies [47] and forces [48] and gives substantial speedups relative to the use of two-electron integrals without any approximation (roughly a factor of 10 (energies) and 30 (forces) at the level of an uncontracted dddd shell quartet, and increasing with angular momentum). Its use is automatically selected for integrals with low degrees of contraction, while regular integrals are employed when the degree of contraction is high, following the state of the art PRISM approach of Gill and coworkers [49].

The CFMM is controlled by the following input parameters:

CFMM_ORDER

Controls the order of the multipole expansions in CFMM calculation

VARIABLE:

INTEGER

DEFAULT:

15 For single point SCF accuracy

25 For tighter convergence

OPTIONS:

n Use multipole expansions of order n

RECOMMENDATION:

Use default

GRAIN

Controls the number of lowest-level boxes in one dimension for CFMM

VARIABLE:

INTEGER

DEFAULT:

-1 Program decides best value, turning on CFMM when useful

OPTIONS:

-1 Program decides best value, turning on CFMM when useful

1 Do not use CFMM

$n \geq 8$ Use CFMM with n lowest-level boxes in one dimension

RECOMMENDATIONS:

This is an expert option; either use the default, or use a value of 1 if CFMM is not desired

4.4.3 LINEAR SCALING EXCHANGE (LINK) MATRIX EVALUATION

Hartree-Fock calculations and the popular hybrid density functionals such as B3LYP also require two-electron integrals to evaluate the exchange energy associated with a single determinant. There is no useful multipole expansion for the exchange energy, because the bra and ket of the two-electron integral are coupled by the density matrix, which carries the effect of exchange. Fortunately, density matrix elements decay exponentially with distance for systems that have a HOMO-LUMO gap [50]. The better the insulator, the more localized the electronic structure, and the faster the rate of exponential decay. Therefore, for insulators, there are only a linear number of numerically significant contributions to the exchange energy. With intelligent numerical thresholding, it is possible to rigorously evaluate the exchange matrix in linear scaling effort. For this purpose, Q-Chem contains the linear scaling K (LinK) method [51] to evaluate both exchange energies and their gradients [52] in linear scaling effort (provided the density matrix is highly sparse). The LinK method essentially reduces to the conventional direct SCF method for exchange in the small molecule limit (by adding no significant overhead), while yielding large speedups for (very) large systems where the density matrix is indeed highly sparse. For full details, we refer the reader to the original papers [51,52]. LinK can be explicitly requested by the following option (although Q-Chem automatically switches it on when the program believes it is the preferable algorithm).

LIN_K

Controls whether linear scaling evaluation of exact exchange (LinK) is used.

VARIABLE:

LOGICAL

DEFAULT:

Program chooses, switching on LinK whenever CFMM is used.

OPTIONS:

TRUE Use LinK

FALSE Do not use LinK

RECOMMENDATION:

Use for HF and hybrid DFT calculations with large numbers of atoms

4.4.4 INCREMENTAL AND VARIABLE THRESH FOCK MATRIX BUILDING

The use of a variable integral threshold, operating for the first few cycles of an SCF, is justifiable on the basis that the MO coefficients are usually of poor quality in these cycles. In Q-Chem, the integrals in the first iteration are calculated at a threshold of 10^{-6} (for an anticipated final integral threshold greater than, or equal to 10^{-6}) to ensure the error in the first iteration is solely sourced from the poor MO guess. Following this, the integral thresh-hold used is computed as

$$tmp_thresh = varthresh \times DIIS_error \quad (0.41)$$

where the *DIIS_error* is that calculated from the previous cycle, *varthresh* is the variable threshold set by the program (by default) and *tmp_thresh* is the temporary threshold used for integral evaluation. Each cycle requires recalculation of all integrals. The variable integral threshold procedure has the greatest impact in early SCF cycles.

In an incremental Fock matrix build [53], **F** is computed recursively as

$$\mathbf{F}^m = \mathbf{F}^{m-1} + \Delta\mathbf{J}^m - \frac{1}{2}\Delta\mathbf{K}^m \quad (0.42)$$

where *m* is the SCF cycle, and $\Delta\mathbf{J}^m$ and $\Delta\mathbf{K}^m$ are computed using the difference density

$$\Delta\mathbf{P}^m = \mathbf{P}^m - \mathbf{P}^{m-1} \quad (0.43)$$

Using Schwartz integrals and elements of the difference density, Q-Chem is able to determine at each iteration which ERIs are required, and if necessary, recalculated. As the SCF nears convergence, $\Delta\mathbf{P}^m$ becomes sparse and the number of ERIs that need to be recalculated declines dramatically, saving the user large amounts of computational time.

Incremental Fock matrix builds and variable thresholds are only used when the SCF is carried out using the direct SCF algorithm and are clearly complementary algorithms. These options are controlled by the following input parameters, which are only used with direct SCF calculations.

INCFOCK

Iteration number after which the incremental Fock matrix algorithm is initiated

VARIABLE:

INTEGER

DEFAULT:

1 Start INCFOCK after iteration number 1

OPTIONS:

User-defined (0 switches INCFOCK off)

RECOMMENDATIONS:

May be necessary to allow several iterations before switching on INCFOCK

VARTHRESH

Controls the temporary integral cut-off threshold.

$\text{tmp_thresh} = 10^{-\text{VARTHRESH}} \times \text{DIIS_error}$

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

User-defined threshold

RECOMMENDATIONS:

3 has been found to be a practical level, and can slightly speed up SCF evaluation.

4.4.5 EXAMPLES

```
$comment
HF/3-21G single point calculation on a large molecule
read in the molecular coordinates from file
$end

$molecule
READ      dna.inp
$end

$rem
EXCHANGE      HF           HF exchange
BASIS         3-21G       Basis set
LIN_K         TRUE        Calculate K using LinK
$end
```

Example 4.6 Example Q-Chem input for a large single point energy calculation. The CFMM is switched on automatically when LinK is requested.

```
$comment
HF/3-21G single point calculation on a large molecule
read in the molecular coordinates from file
$end

$molecule
READ      dna.inp
$end

$rem
EXCHANGE      HF      HF exchange
BASIS         3-21G   Basis set
INCFOCK       5      Incremental Fock after 5 cycles
VARTHRESH     3      1.0d-03 variable threshold
$end
```

Example 4.7 Example Q-Chem input for a large single point energy calculation. This would be appropriate for a medium-sized molecule, but for truly large calculations, the CFMM and LinK algorithms are far more efficient.

4.5 SCF INITIAL GUESS

4.5.1 INTRODUCTION

The Roothaan-Hall and Pople-Nesbet equations of SCF theory are non-linear in the molecular orbital coefficients. Like many mathematical problems involving non-linear equations, prior to the application of a technique to search for a numerical solution, an initial guess for the solution must be generated. If the guess is poor, the iterative procedure applied to determine the numerical solutions may converge very slowly, requiring a large number of iterations, or at worst, the procedure may diverge.

Thus, in an *ab initio* SCF procedure, the quality of the initial guess is of utmost importance for (at least) two main reasons:

- (1) To ensure that the SCF converges to an appropriate ground state. Often SCF calculations can converge to different local minima in wavefunction space, depending upon which part of that space the initial guess places the system in.
- (2) When considering jobs with many basis functions requiring the recalculation of ERIs at each iteration, using a good initial guess that is close to the final solution can reduce the total job time significantly by decreasing the number of SCF iterations.

For these reasons, sooner or later most users will find it helpful to have some understanding of the different options available for customizing the initial guess. Q-Chem currently offers five options for the initial guess:

- Superposition of Atomic Density (SAD)
- Core Hamiltonian
- Generalized Wolfsberg-Helmholtz (GWH)
- Reading previously obtained MOs from disk.
- Basis set projection

The first 3 of these guesses are built-in, and are briefly described in Section 4.5.2. The option of reading MO's from disk is described in Section 4.5.3. The initial guess MO's can be modified, either by mixing, or altering the order of occupation. These options are discussed in Section 4.5.4. Finally, Q-Chem's novel basis set projection method is discussed in Section 4.5.5.

4.5.2 SIMPLE INITIAL GUESSES

There are three simple initial guesses available in Q-Chem. While they are all simple, they are by no means equal in quality, as we discuss below.

- (1) **Superposition of Atomic Densities (SAD):** The SAD guess is almost trivially constructed by summing together atomic densities that have been spherically

averaged to yield a trial density matrix. The SAD guess is far superior to the other two options below, particularly when large basis sets and/or large molecules are employed. There are three issues associated with the SAD guess to be aware of:

- No molecular orbitals are obtained, which means that SCF algorithms requiring orbitals (the direct minimization methods discussed in Section 4.6) cannot directly use the SAD guess, and,
- The SAD guess is not available for general (read-in) basis sets. All internal basis sets support the SAD guess.
- The SAD guess is not idempotent and thus requires *at least* two SCF iterations to ensure proper SCF convergence (idempotency of the density).

(2) **Generalized Wolfsberg-Helmholtz (GWH):** The GWH guess procedure [54] uses a combination of the overlap matrix elements (0.12), and the diagonal elements of the Core Hamiltonian matrix (0.18). This initial guess is most satisfactory in small basis sets for small molecules. It is constructed according to the relation given below, where c_x is a constant.

$$H_{\mu\nu} = c_x S_{\mu\nu} (H_{\mu\mu} + H_{\nu\nu})/2 \quad (0.44)$$

(3) **Core Hamiltonian:** The core Hamiltonian guess simply obtains the guess MO coefficients by diagonalizing the core Hamiltonian matrix (0.18). This approach works best with small basis sets, and degrades as both the molecule size and the basis set size are increased.

The selection of these choices (or whether to read in the orbitals) is controlled by the following *\$rem* variables:

SCF_GUESS

Specifies the initial guess procedure to use for the SCF

VARIABLE:

STRING

DEFAULT:

SAD Superposition of atomic density (available only with standard basis sets)

GWH For ROHF where a set of orbitals are required.

OPTIONS:

CORE Diagonalize core Hamiltonian

SAD Superposition of atomic density

GWH Apply generalized Wolfsberg-Helmholtz approximation

READ Read previous MOs from disk

RECOMMENDATION:

SAD guess for standard basis sets. For general basis sets, it is best to use the *BASIS2* REM. Alternatively, try the GWH or core Hamiltonian guess. For ROHF it can be useful to READ guesses from an SCF calculation on the corresponding cation or anion.

SCF_GUESS_ALWAYS

Switch to force the regeneration of a new initial guess for each series of SCF iterations (for use in geometry optimization)

VARIABLE:

LOGICAL

DEFAULT:

False Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available

OPTIONS:

False Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available

True Generate a new guess for each series of SCF iterations in a geometry optimization

4.5.3 READING MOS FROM DISK

There are two methods by which MO coefficients can be used from a previous job by reading them from disk:

1. Running two independent jobs sequentially invoking *qchem* with three command line variables:

```
localhost-1> qchem job1.in job1.out save
localhost-2> qchem job2.in job2.out save
```

Notes: (1) The *\$rem* variable *SCF_GUESS* must be set to *READ* in *job2.in*.
(2) Scratch files remain in *\$QCSCRATCH/save* on exit.

2. Running a batch job where two jobs are placed into a single input file separated by the string “@@@” on a single line.

Notes: (1) The *\$rem* variable *SCF_GUESS* must be set to *READ* in the second job of the batch file.
(2) A third qchem command line variable is not necessary.
(3) As for the SAD guess, Q-Chem requires at least two SCF cycles to ensure proper SCF convergence (idempotency of the density).

Most important note: It is up to the user to make sure that the basis sets match between the 2 jobs. There is no internal checking for this, although the occupied orbitals are reorthogonalized in the current basis after being read in. If you want to project from a smaller basis into a larger basis, consult section 4.5.5

4.5.4 MODIFYING THE OCCUPIED MOLECULAR ORBITALS

It is sometimes useful for the occupied guess orbitals to be other than the lowest N_{α} (or N_{β}) orbitals. Reasons why one may need to do this include:

- To converge to a state of different symmetry or orbital occupation
- To break spatial symmetry
- To break spin symmetry, as in unrestricted calculations on molecules with an even number of electrons.

There are 2 mechanisms for modifying a set of guess orbitals: either by *SCF_GUESS_MIX*, or by specifying the orbitals to occupy. Q-Chem users may define the occupied guess orbitals using the *\$occupied* keyword. Occupied guess orbitals are defined by listing the alpha orbitals to be occupied on the first line and beta on the second. The need for orbitals renders this option incompatible with the SAD guess.

```

$occupied
 $\alpha_i$   $\alpha_j$   $\alpha_k$   $\alpha_l$  ... {alpha guess orbitals to be occupied}
 $\beta_1$   $\beta_m$   $\beta_n$   $\beta_o$  ... {beta guess orbitals to be occupied}
$end

```

Figure 4.1 Format for modifying occupied guess orbitals.

The other \$rem variables related to altering the orbital occupancies are:

SCF_GUESS_PRINT

Controls printing of guess MOs, Fock and density matrices

VARIABLE:

INTEGER

DEFAULT:

0 Do not print guesses

OPTIONS:

0 Do not print guesses

SAD

1 Atomic density matrices and molecular matrix

2 Level 1 plus density matrices

CORE and GWH

1 No extra output

2 Level 1 plus Fock and density matrices and, MO coefficients and eigenvalues

READ

1 No extra output

2 Level 1 plus density matrices, MO coefficients and eigenvalues

SCF_GUESS_MIX

Controls mixing of LUMO and HOMO to break symmetry in the initial guess. For unrestricted jobs, the mixing is performed only for the alpha orbitals.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess

OPTIONS:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess

1 (TRUE) Add 10% of LUMO to HOMO to break symmetry

n Add $n \times 10\%$ of LUMO to HOMO ($0 < n < 10$)

RECOMMENDATION:

When performing unrestricted calculations on molecules with an even number of electrons, it is often necessary to break alpha-beta symmetry in the initial guess with this option, or by specifying input for *\$occupied*.

4.5.5 BASIS SET PROJECTION

Q-Chem also includes a novel basis set projection method developed by Dr. Jing Kong of Q-Chem. It permits a calculation in a large basis set to bootstrap itself up via a calculation in a small basis set that is automatically spawned when the user requests this option. When basis set projection is requested (by providing a valid small basis for *BASIS2*), the program executes the following steps:

- (1) A simple DFT calculation is performed in the small basis, *BASIS2*, yielding a converged density matrix in this basis.
- (2) The large basis set SCF calculation (with different values of *EXCHANGE* and *CORRELATION* set by the input) begins by constructing the DFT Fock operator in the large basis but with the density matrix obtained from the small basis set.
- (3) By diagonalizing this matrix, an accurate initial guess for the density matrix in the large basis is obtained, and the target SCF calculation commences.

Basis set projection is a very effective option for general basis sets, where the SAD guess is not available. In detail, this initial guess is controlled by the following *\$rem* variables:

BASIS2

Sets the small basis set to use in basis set projection

VARIABLE:

STRING

DEFAULT:

No second basis set default

OPTIONS:

Symbol Use standard basis sets as per Chapter 7

RECOMMENDATIONS:

BASIS2 should be smaller than BASIS

4.5.6 EXAMPLES

```

$molecule
0 1
O
H 1 r
H 1 r 2 a

r 0.9
a 104.0
$end

$rem
  exchange          hf
  correlation       mp2
  basis             general
  basis2           sto-3g
$end

$basis
O    0
S    3    1.000000
      3.22037000E+02    5.92394000E-02
      4.84308000E+01    3.51500000E-01
      1.04206000E+01    7.07658000E-01
SP   2    1.000000
      7.40294000E+00   -4.04453000E-01    2.44586000E-01
      1.57620000E+00    1.22156000E+00    8.53955000E-01
SP   1    1.000000
      3.73684000E-01    1.00000000E+00    1.00000000E+00
SP   1    1.000000
      8.45000000E-02    1.00000000E+00    1.00000000E+00
****
H    0
S    2    1.000000
      5.44717800E+00    1.56285000E-01
      8.24547000E-01    9.04691000E-01
S    1    1.000000
      1.83192000E-01    1.00000000E+00
****
$end

```

Example 4.8 Input where basis set projection is used to generate a good initial guess for a calculation employing a general basis set, for which the default initial guess is not available.

```
$comment
  OH radical, part 1.  Do 1 iteration of cation orbitals.
$end

$molecule
  1 1
  o   0.0   0.0   0.0
  h   0.0   0.0   1.0
$end

$rem
  basis           = 6-311++G(2df)
  exchange        = hf
  max_scf_cycles  = 1
  thresh          = 10
$end

@@@
$comment
  OH radical, part 2.  Read cation orbitals, do the radical
$end

$molecule
  0 2
  o   0.0   0.0   0.0
  h   0.0   0.0   1.0
$end

$rem
  basis           = 6-311++G(2df)
  exchange        = hf
  unrestricted    = false
  scf_algorithm   = dm
  scf_convergence = 7
  scf_guess       = read
  thresh          = 10
end
```

Example 4.9 Input for an ROHF calculation on the OH radical. One SCF cycle is initially performed on the cation, to get reasonably good initial guess orbitals, which are then read in as the guess for the radical. This avoids the use of Q-Chem's default GWH guess for ROHF, which is often poor.

```
$molecule
  0    1
  H    0.0    0.0    0.0
  H    0.0    0.0   -10.0
$end

$rem
  unrestricted    = true
  exchange        = hf
  basis           = 6-31g**
  scf_algorithm   = diis_gdm
  max_diis_cycles = 1
  scf_guess       = gwh
  scf_guess_mix   = 2
$end
```

Example 4.10 Input for an unrestricted HF calculation on H_2 in the dissociation limit, showing the use of $SCF_GUESS_MIX = 2$ (corresponding to 20% of the alpha LUMO mixed with the alpha HOMO). Geometric direct minimization with DIIS (see 4.6.4) is used to converge the SCF, together with $MAX_DIIS_CYCLES = 1$ (using the default value for MAX_DIIS_CYCLES , the DIIS procedure just oscillates).

4.6 CONVERGING SCF CALCULATIONS

4.6.1 INTRODUCTION

As for any numerical optimization procedure, the rate of convergence of the SCF procedure is dependent on the initial guess, and on the algorithm used to step towards the stationary point. Q-Chem features a number of alternative SCF optimization algorithms, which are discussed in the following sections, along with the \$rem variables that are used to control the calculations. The main options are discussed in sections which follow, and are, in brief:

- The highly successful DIIS procedures, which are the default.
- The new geometric direct minimization (GDM) method, which is highly robust, and the recommended fall-back when DIIS fails. It can also be invoked after a few initial iterations with DIIS to improve the initial guess.
- The older and less robust direct minimization method (DM), which is retained because it is the only method implemented for restricted open shell SCF. As for GDM, it can also be invoked after a few DIIS iterations (except for RO jobs).
- The maximum overlap method (MOM) which ensures that DIIS always occupies a continuous set of orbitals and does not oscillate between different occupancies.

4.6.2 BASIC CONVERGENCE CONTROL OPTIONS

See also more detailed options in the following sections, and note that *SCF_CONVERGENCE* and *THRESH* must be set in a compatible manner.

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted

VARIABLE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

SCF_ALGORITHM

Algorithm used for converging the SCF

VARIABLE:

STRING

DEFAULT:

DIIS Pulay DIIS

OPTIONS:

DIIS Pulay DIIS

DM Direct minimizer

DIIS_DM Uses DIIS initially, switching to direct minimizer for later iterations (See *THRESH_DIIS_SWITCH*, *MAX_DIIS_CYCLES*)

DIIS_GDM Use DIIS and then later switch to geometric direct minimization (See *THRESH_DIIS_SWITCH*, *MAX_DIIS_CYCLES*)

GDM Geometric Direct Minimization

ROOTHAAN Roothaan repeated diagonalization

RECOMMENDATION:

Use DIIS unless wanting ROHF, in which case direct minimization must be used. If DIIS fails, DIIS_GDM is the recommended fall-back option.

SCF_CONVERGENCE

SCF is considered converged when the wavefunction error is less than $10^{-\text{SCF_CONVERGENCE}}$. Adjust the value of *THRESH* at the same time.

VARIABLE:

INTEGER

DEFAULT:

5

For single point energy calculations

8

for geometry optimizations and vibrational analysis

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost.

4.6.3 DIRECT INVERSION IN THE ITERATIVE SUBSPACE (DIIS)

The SCF implementation of the Direct Inversion in the Iterative Subspace (DIIS) method [55,56] uses the property of an SCF solution which requires the density matrix to commute with the Fock matrix

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} \quad (0.45)$$

During the SCF cycles, prior to achieving self-consistency, it is possible to define an error vector \mathbf{e}_i , which is non-zero

$$\mathbf{SP}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i\mathbf{S} = \mathbf{e}_i \quad (0.46)$$

where \mathbf{P}_i is obtained from diagonalization of $\hat{\mathbf{F}}_i$, and

$$\hat{\mathbf{F}}_k = \sum_{j=1}^{k-1} c_j \mathbf{F}_j \quad (0.47)$$

The DIIS coefficients c_k , are obtained by a least squares constrained minimisation of the error vectors, viz

$$Z = \left(\sum_k c_k \mathbf{e}_k \right) \cdot \left(\sum_k c_k \mathbf{e}_k \right) \quad (0.48)$$

where the constraint

$$\sum_k c_k = 1 \quad (0.49)$$

is imposed to yield a set of linear equations, of dimension $(N+1)$

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_1 \cdot \mathbf{e}_N & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_N \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_N \cdot \mathbf{e}_N & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (0.50)$$

Convergence criteria requires the largest element of the N^{th} error vector to be below a cutoff threshold, usually 10^{-5} for single point energies, often increased to 10^{-8} for optimizations and frequency calculations.

The rate of convergence may be improved by restricting the number of previous Fock matrices (size of the DIIS subspace, *\$rem* variable DIIS_SUBSPACE_SIZE) used for determining the DIIS coefficients

$$\hat{\mathbf{F}}_k = \sum_{j=k-(L+1)}^{k-1} c_j \mathbf{F}_j \quad (0.51)$$

where L is the size of the DIIS subspace. As the Fock matrix nears self-consistency the linear matrix equations (0.50) tend to become severely ill-conditioned and it is often necessary to reset the DIIS subspace (this is automatically carried out by the program).

Finally, on a practical note, we observe that DIIS has a tendency to converge to global minima rather than local minima when employed for SCF calculations. This seems to be because only at convergence is the density matrix in the DIIS iterations idempotent. On the way to convergence, one is not on the “true” energy surface, and this seems to permit DIIS to “tunnel” through barriers in wavefunction space. This is usually a desirable property, and is the motivation for the options that permit initial DIIS iterations before switching to direct minimization to converge to the minimum in difficult cases.

The following *\$rem* variables permit some customization of the DIIS iterations:

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS subspace during the SCF

VARIABLE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

DIIS_PRINT

Controls the output from DIIS SCF optimization

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

0

1 Chosen method and DIIS coefficients & solutions

2 Level 1 plus changes in multipole moments

3 Level 2 plus Multipole moments

4 Level 3 plus extrapolated Fock matrices

4.6.5 GEOMETRIC DIRECT MINIMIZATION (GDM)

Troy Van Voorhis, working at Berkeley with Martin Head-Gordon, has developed a novel direct minimization method that is extremely robust, and at the same time is only slightly less efficient than DIIS. This method is called geometric direct minimization (GDM) because it takes steps in an orbital rotation space that correspond properly to the hyperspherical geometry of that space. In other words, rotations are variables that describe a space which is curved like a many-dimensional sphere. Just like the optimum flight paths for airplanes are not straight lines but great circles, so too are the optimum steps in orbital rotation space. GDM takes this correctly into account, which is the origin of its efficiency and its robustness. For full details, we refer the reader to a paper submitted for publication [57]. GDM is a good alternative to DIIS for SCF jobs that exhibit convergence difficulties with DIIS.

In section 4.6.3, we discussed the fact that DIIS can efficiently head towards the global SCF minimum in the early iterations. This can be true even if DIIS fails to converge in later iterations. For this reason, a hybrid scheme has been implemented which uses the DIIS minimization procedure to achieve convergence to an intermediate cutoff threshold. Thereafter, the geometric direct minimization algorithm is used. This scheme combines the strengths of the two methods quite nicely: the ability of DIIS to recover from initial guesses that may not be close to the global minimum, and the ability of GDM to robustly converge to a local minimum, even when the local surface topology is challenging for DIIS. This is the recommended procedure with which to invoke GDM (i.e. setting *SCF_ALGORITHM = DIIS_GDM*). This hybrid procedure is also compatible with the SAD guess, while GDM itself is not, because it requires an initial guess set of orbitals. If one wishes to disturb the initial guess as little as possible before switching on GDM, one should additionally specify *MAX_DIIS_CYCLES = 1* to obtain only a single Roothaan step (which also serves up a properly orthogonalized set of orbitals).

\$rem options relevant to GDM are the following.

SCF_ALGORITHM

Algorithm used for converging the SCF

VARIABLE:

STRING

OPTIONS:

DIIS_GDM	Use DIIS and then later switch to geometric direct minimization (See <i>THRESH_DIIS_SWITCH</i> , <i>MAX_DIIS_CYCLES</i>)
GDM	Geometric Direct Minimization

MAX_DIIS_CYCLES

The maximum number of DIIS iterations before switching to (geometric) direct minimization when *SCF_ALGORITHM* is *DIIS_GDM* or *DIIS_DM*. See also *THRESH_DIIS_SWITCH*.

VARIABLE:

INTEGER

DEFAULT:

50

OPTIONS:

1	Only a single Roothaan step before switching to (G)DM
<i>n</i>	<i>n</i> DIIS iterations before switching to (G)DM.

THRESH_DIIS_SWITCH

The threshold for switching between DIIS extrapolation and direct minimization of the SCF energy is $10^{-\text{THRESH_DIIS_SWITCH}}$ when *SCF_ALGORITHM* is *DIIS_GDM* or *DIIS_DM*. See also *MAX_DIIS_CYCLES*.

VARIABLE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined

4.6.5 DIRECT MINIMIZATION (DM)

Direct minimization (DM) is a less sophisticated forerunner of the geometric direct minimization (GDM) method discussed in the previous section. DM does not properly step along great circles in the hyperspherical space of orbital rotations, and therefore converges less rapidly and less robustly than GDM, in general. It is retained for legacy purposes, and because it is at present the only method available for restricted open shell (RO) SCF calculations in Q-Chem. In general, the input options are the same as for GDM, with the exception of the specification of *SCF_ALGORITHM*, which can be either *DIIS_DM* (recommended) or *DM*.

4.6.6 MAXIMUM OVERLAP METHOD (MOM)

In general, the DIIS procedure is remarkably successful. One difficulty that is occasionally encountered is the problem of an SCF that occupies two different sets of orbitals on alternating iterations, and therefore oscillates and fails to converge. This can be overcome by choosing orbital occupancies that maximize the overlap of the new occupied orbitals with the set previously occupied. However this combinatorial matching problem has computational complexity that scales factorially with the number of occupied orbitals if implemented straightforwardly. Q-Chem contains the maximum overlap method (MOM) [58], developed by Andrew Gilbert and Peter Gill at Nottingham, which, remarkably, reduces the combinatorial problem to cubic in the number of orbitals.

MOM is therefore a useful adjunct to DIIS in convergence problems involving flipping of orbital occupancies. It is controlled by the *\$rem* variable *MOM_START*, which specifies the DIIS iteration on which the MOM procedure is first enabled. There are two strategies that are useful in setting a value for *MOM_START*. To help maintain an initial configuration it should be set to start on the first cycle. On the other hand, to assist convergence it should come on later to avoid holding on to an initial configuration that may be far from the converged one.

The MOM-related *\$rem* variables in full are the following:

MOM_ECONOMIZE:

Determines if any computational savings are to be used with MOM

VARIABLE:

INTEGER

DEFAULT:

4

OPTIONS:

1	Include everything
2	Freeze core electrons
3	Only use orbitals within 1Eh of the HOMO
4	Combined frozen core and window

RECOMMENDATION:

Use default

MOM_PRINT

Switches printing on within the MOM procedure

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Printing is turned off
TRUE	Printing is turned on

MOM_START

Determines when MOM is switched on to stabilize DIIS iterations

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE)

MOM is not used

n

MOM begins on cycle n

4.6.7 EXAMPLES

```

$molecule
  0 2
  c1
  x1 c1 1.0
  c2 c1 rc2 x1 90.0
  x2 c2 1.0 c1 90.0 x1 0.0
  c3 c1 rc3 x1 90.0 c2 tc3
  c4 c1 rc3 x1 90.0 c2 -tc3
  c5 c3 rc5 c1 ac5 x1 -90.0
  c6 c4 rc5 c1 ac5 x1 90.0
  h1 c2 rh1 x2 90.0 c1 180.0
  h2 c3 rh2 c1 ah2 x1 90.0
  h3 c4 rh2 c1 ah2 x1 -90.0
  h4 c5 rh4 c3 ah4 c1 180.0
  h5 c6 rh4 c4 ah4 c1 180.0

  rc2=2.67298593
  rc3=1.35449831
  tc3=62.85150452
  rc5=1.37290399
  ac5=116.45436983
  rh1=1.08573521
  rh2=1.08534214
  ah2=122.157328
  rh4=1.08721616
  ah4=119.52349629
$end

$rem
  basis           = 6-31G*
  exchange        = hf
  memory          = 5000000
  intsbuffersize = 15000000
  scf_algorithm   = diis_gdm
  scf_convergence = 7
  thresh         = 10
$end

```

Example 4.11 Q-Chem input for a UHF calculation using geometric direct minimization (GDM) on the phenyl radical, after initial iterations with DIIS. This example fails to converge if DIIS is employed directly.

```

$molecule
0 2
b
$end

$rem
BASIS          6-31G*
EXCHANGE       GG99
MOM_PRINT      TRUE
MOM_START      3
$end

```

Example 4.12 Q-Chem input for a MOM-stabilized DIIS calculation. This job fails to converge without the use of MOM.

4.7 UNCONVENTIONAL SCF CALCULATIONS.

4.7.1 CASE APPROXIMATION

The Coulomb Attenuated Schrödinger Equation (CASE) [59] approximation follows from the KWIK [60] algorithm in which the Coulomb operator is separated into two pieces

$$\frac{1}{r_{12}} \equiv \frac{\operatorname{erfc}(\omega r_{12})}{r_{12}} + \frac{\operatorname{erf}(\omega r_{12})}{r_{12}} \quad (0.52)$$

The first of these two terms is singular but short-range and the second is non-singular but long-range. The CASE approximation is applied by smoothly attenuating all occurrences of the Coulomb operator in (0.2) by neglecting the long-range portion of the identity in (0.52). The parameter ω can be used to tune the level of attenuation. Although the total energies from Coulomb attenuated calculations are significantly different from non-attenuated energies, it is found that relative energies, correlation energies and, in particular, wavefunctions, are not, provided a reasonable value of ω is chosen.

By virtue of the exponential decay of the attenuated operator, ERIs can be neglected on a proximity basis yielding a rigorous $O(N)$ algorithm for single point energies. CASE may also be applied in geometry optimizations and frequency calculations.

OMEGA

Controls the degree of attenuation of the Coulomb operator

VARIABLE:

INTEGER

DEFAULT:

No default

OPTIONS:

n $\omega = n/1000$

INTEGRAL_2E_OPR

Determines the two-electron operator

VARIABLE:

INTEGER

DEFAULT:

-2 Coulomb Operator

OPTIONS:

-1 Apply the CASE approximation

-2 Coulomb Operator

4.7.2 POLARIZED ATOMIC ORBITAL (PAO) CALCULATIONS

Polarized atomic orbital (PAO) calculations are an interesting unconventional SCF method, in which the molecular orbitals and the density matrix are not expanded directly in terms of the basis of atomic orbitals. Instead, an intermediate molecule-optimized minimal basis of polarized atomic orbitals (PAO's) is used [61]. The polarized atomic orbitals are defined by an atom-blocked linear transformation from the fixed atomic orbital basis, where the coefficients of the transformation are optimized to minimize the energy, at the same time as the density matrix is obtained in the PAO representation. Thus a PAO-SCF calculation is a constrained variational method, whose energy is above that of a full SCF calculation in the same basis. However, a molecule optimized minimal basis is a very compact and useful representation for purposes of chemical analysis, and it also has potential computational advantages in the context of MP2 or local MP2 calculations, as can be done after a PAO-HF calculation is complete to obtain the PAO-MP2 energy.

PAO-SCF calculations tend to systematically underestimate binding energies (since by definition the exact result is obtained for atoms, but not for molecules). In tests on the G2 database, PAO-B3LYP/6-311+G(2df,p) atomization energies deviated from full B3LYP/6-311+G(2df,p) atomization energies by roughly 20 kcal/mol, with the error being essentially extensive with the number of bonds [62]. This deviation can be reduced to only 0.5 kcal/mol [62] with the use of a simple non-iterative second order correction for "beyond-minimal basis" effects [63]. The second order correction is evaluated at the end of each PAO-SCF calculation, as it involves negligible computational cost. Analytical gradients are available using PAO's, to permit structure

optimization. For additional discussion of the PAO-SCF method and its uses, see the references cited above.

Calculations with PAO's are determined controlled by the following *\$rem* variables. *PAO_METHOD = PAO* invokes PAO-SCF calculations, while the algorithm used to iterate the PAO's can be controlled with *PAO_ALGORITHM*.

PAO_ALGORITHM

Algorithm used to optimise polarized atomic orbitals (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

0

use efficient (and riskier) strategy to converge PAO's

OPTIONS:

1

use conservative (and slower) strategy to converge PAO's

PAO_METHOD

Controls evaluation of polarized atomic orbitals (PAO's)

VARIABLE:

STRING

DEFAULT:

EPAO

For local MP2 calculations

Otherwise no default

OPTIONS:

PAO

Perform PAO-SCF instead of conventional SCF

EPAO

Obtain EPAO's after a conventional SCF.

4.8 GROUND STATE METHOD SUMMARY

To summarize the main features of Q-Chem's ground state self-consistent field capabilities, the user needs to consider:

1. Input a molecular geometry (*\$molecule* keyword)
 - Cartesian
 - Z-matrix
 - Read from prior calculations

2. Declare the job specification (*\$rem* keyword)
 - JOBTYPED
 - ◊ Single point
 - ◊ Optimization
 - ◊ Frequency
 - BASIS
 - ◊ Refer to Chapter 7 (note: *\$basis* keyword for user defined basis sets).
 - ◊ Effective core potentials, as described in Chapter 8.
 - EXCHANGE
 - ◊ Linear scaling algorithms for all methods
 - ◊ Arsenal of exchange density functionals
 - ◊ User definable functionals and hybrids
 - CORRELATION
 - ◊ DFT or conventional methods
 - ◊ Linear scaling (CPU and memory) incorporation of correlation with DFT
 - ◊ Arsenal of correlation density functionals
 - ◊ User definable functionals and hybrids
 - ◊ See Chapter 5 for wavefunction-based correlation methods.

3. Exploit Q-Chem's special features
 - CFMM, LinK large molecule options
 - SCF rate of convergence increased through improved guessers and alternative minimization algorithms
 - Explore novel methods if desired: CASE approximation, PAO's.

4.9 REFERENCES AND FURTHER READING

Basis sets (Chapter 7)

Effective Core Potentials (Chapter 8)

Molecular Geometry Critical Points (Chapter 9)

Molecular Properties analysis (Chapter 10)

AOINTS (Appendix)

- [1] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York, 1986).
- [2] For a general introduction to Hartree-Fock theory, see, A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Dover, New York, 1996).
- [3] F. Jensen, *Introduction to Computational Chemistry* (Wiley, New York, 1999).
- [4] E. Schrödinger, *Ann. Physik.* (1926), **79**, 361.
- [5] W. Heisenberg, *Z. Physik.* (1926), **39**, 499.
- [6] M. Born and J. R. Oppenheimer, *Ann. Physik.* (1927), **84**, 457.
- [7] J. C. Slater, *Phys. Rev.* (1929), **34**, 1293.
- [8] J. C. Slater, *Phys. Rev.* (1930), **35**, 509.
- [9] J. A. Pople and R. K. Nesbet, *J. Chem. Phys.* (1954), **22**, 571.
- [10] T. Takada, M. Dupuis and H. F. King, *J. Chem. Phys.* (1981), **75**, 332.
- [11] M. Dupuis and H. F. King, *Int. J. Quantum Chem.* (1977), **11**, 613.
- [12] M. Dupuis and H. F. King, *J. Chem. Phys.* (1978), **68**, 3998.
- [13] R. G. Parr and W. Yang, *Density-Functional theory of atoms and molecules* (Oxford University Press: New York, 1989)
- [14] W. Kohn, A. D. Becke and R. G. Parr, *J. Phys. Chem.* (1996) **100**, 12974.
- [15] See for example, T. Ziegler, *Chem. Rev.* (1991) **91**, 651, *Chemical Applications of Density Functional Methods*, edited by B.B.Laird, R.B.Ross, and T.Ziegler (American Chemical Society, Washington, 1996).
- [16] P. Hohenberg and W. Kohn, *Phys. Rev. B* (1964), **136**, 864.
- [17] W. Kohn and L. J. Sham, *Phys. Rev. A* (1965), **140**, 1133.
- [18] P. A. M. Dirac, *Proc. Cam. Phil. Soc.* (1930), **26**, 376.
- [19] J. A. Pople, P. M. W. Gill and B. G. Johnson, *Chem. Phys. Lett.* (1992), **199**, 557.
- [20] S. H. Vosko, L. Wilk and M. Nusair, *Can. J. Phys.* (1980), **58**, 1200.
- [21] J. P. Perdew and A. Zunger, *Phys. Rev. B* (1981), **23**, 5048.
- [22] E. P. Wigner, *Trans. Faraday Soc.* (1938), **34**, 678.
- [23] A. D. Becke, *Phys. Rev. A* (1988), **38**, 3098.
- [24] P. M. W. Gill, *Mol. Phys.* (1996), **89**, 443.
- [25] A. T. B. Gilbert and P. M. W. Gill, *Chem. Phys. Lett.* (1999) **312**, 511.
- [26] C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B* (1988), **37**, 785.
- [27] J. P. Perdew, *Phys. Rev. B* (1986), **33**, 8822.
- [28] J. R. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, *Phys. Rev. B* (1992), **46**, 6671.
- [29] R. D. Adamson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.* (1998) **284**, 6.
- [30] A. D. Becke, *J. Chem. Phys.* (1993), **98**, 1372.

- [31] P. J. Stephens, F. J. Devlin, C. F. Chabalowski and M. J. Frisch, *J. Phys. Chem.* (1994), **98**, 11623.
- [32] A. D. Becke, *J. Chem. Phys.* (1988), **88**, 2547.
- [33] C. W. Murray, N. C. Handy and G. J. Laming, *Mol. Phys.* (1993), **78**, 997.
- [34] Lebedev, *Sibirsk. Mat. Zh.* (1977), **18**, 132.
- [35] Lebedev, *Zh. Vychisl. Mat. Mat. Fiz.* (1975), **15**, 48.
- [36] Lebedev, *Zh. Vychisl. Mat. Mat. Fiz.* (1976), **16**, 293.
- [37] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Chem. Phys. Lett.* (1993), **209**, 506.
- [38] A. A. Jarecki and E. R. Davidson, *Chem. Phys. Lett.* (1999), **300**, 44.
- [39] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.* (1994), **220**, 377.
- [40] L. Greengard, *The rapid evaluation of potential fields in particle systems*; The MIT Press: London, 1987.
- [41] C. A. White and M. Head-Gordon, *J. Chem. Phys.* (1994), **101**, 6593.
- [42] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.* (1994), **230**, 8.
- [43] C. A. White and M. Head-Gordon, *J. Chem. Phys.* (1996), **105**, 5061.
- [44] C. A. White and M. Head-Gordon, *Chem. Phys. Lett.* (1996), **257**, 647
- [45] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.* (1996), **253**, 268.
- [46] C. A. White and M. Head-Gordon, *J. Chem. Phys.* (1996) **104**, 2620.
- [47] Y. Shao and M. Head-Gordon, *Chem. Phys. Lett.* (2000) **323**, 425.
- [48] Y. Shao and M. Head-Gordon, *J. Chem. Phys.* (in press).
- [49] T. R. Adams, R. D. Adamson, and P. M. W. Gill, *J. Chem. Phys.* (1997) **107**, 124.
- [50] E. Schwegler, M. Challacombe and M. Head-Gordon, *J. Chem. Phys.* (1997) **106**, 9708.
- [51] C. Ochsenfeld, C. A. White and M. Head-Gordon, *J. Chem. Phys.* (1998) **109**, 1663.
- [52] C. Ochsenfeld, *Chem. Phys. Lett.* (2000) **327**, 216.
- [53] E. Schwegler and M. Challacombe, *J. Chem. Phys.* (1996), **106**, 9708.
- [54] M. Wolfsberg and L. Helmholz, *J. Chem. Phys.* (1952), **20**, 837.
- [55] P. Pulay, *J. Comput. Chem.* (1982), **3**, 556.
- [56] P. Pulay, *Chem. Phys. Lett.* (1980), **73**, 393.
- [57] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.* (submitted).
- [58] A.T.B. Gilbert, G.B.W. Gill, and P.M.W. Gill, (in preparation).
- [59] R. D. Adamson, J. P. Dombroski and P. M. W. Gill, *Chem. Phys. Lett.* (1996), **254**, 329.
- [60] J. P. Dombroski, S. W. Taylor and P. M. W. Gill, *J. Phys. Chem.* (1996), **100**, 6272.
- [61] M.S. Lee and M. Head-Gordon, *J. Chem. Phys.* (1997) **107**, 9085.
- [62] M.S. Lee and M. Head-Gordon, (to be published).
- [63] M.S. Lee and M. Head-Gordon, *Comp. Chem.* (2000) **24**, 295.

CHAPTER 5 WAVEFUNCTION-BASED CORRELATION METHODS

5.1 INTRODUCTION

The Hartree-Fock procedure, while often qualitatively correct, is frequently quantitatively deficient. The deficiency is due to the underlying assumption of the Hartree-Fock approximation: that electrons move *independently* within molecular orbitals subject to an averaged field imposed by the remaining electrons. The error that this introduces is called the correlation energy and a wide variety of procedures exist for estimating its magnitude. The purpose of this chapter is to introduce the main wavefunction-based methods available in Q-Chem to describe electron correlation.

Wavefunction-based electron correlation methods concentrate on the design of corrections to the wavefunction beyond the mean-field Hartree-Fock description. This is to be contrasted with the density functional theory methods discussed in the previous chapter. While density functional methods yield a description of electronic structure that accounts for electron correlation subject only to the limitations of present-day functionals (which for example omit dispersion interactions), DFT cannot be systematically improved if the results are deficient. Wave function-based approaches for describing electron correlation [1,2] offer this main advantage. Their main disadvantage is relatively high computational cost, particularly for the higher level theories.

There are four broad classes of models for describing electron correlation that are supported within Q-Chem. The first three directly approximate the full time-independent Schrodinger equation. In order of increasing accuracy, and also increasing cost, they are:

- (a) Perturbative treatment of pair correlations between electrons, capable of recovering typically 80% or so of the correlation energy in stable molecules.
- (b) Self-consistent treatment of pair correlations between electrons, capable of recovering on the order of 95% or so of the correlation energy.
- (c) Non-iterative corrections for higher than double substitutions, which can typically account for more than 99% of the correlation energy. They are the basis of many modern methods that are capable of yielding chemical accuracy for ground state reaction energies, as exemplified by the G2 [3] and G3 methods [4].

These methods are discussed in the following 3 subsections.

There is also a 4th class of methods supported in Q-Chem, which have a different objective. These active space methods aim to obtain a balanced description of electron

correlation in highly correlated systems, such as biradicals, or along bond-breaking coordinates. Active space methods are discussed in the 4th part of this chapter.

In order to carry out a wavefunction-based electron correlation calculation using Q-Chem, 3 *\$rem* variables need to be set:

- *BASIS* to specify the basis set (see chapter 6)
- *CORRELATION* method for treating Correlation (defaults to *NONE*)
- *N_FROZEN_CORE* frozen core electrons (0 default, optionally FC, or *n*)

Note that for wavefunction-based correlation methods, the default option for *EXCHANGE* is HF (Hartree-Fock). It can therefore be omitted from the input if desired.

The full range of ground state wavefunction-based correlation methods available (i.e. the recognized options to the *CORRELATION* keyword) are as follows:

CORRELATION

Specifies the correlation level of theory, either DFT or wavefunction-based.

VARIABLE:

STRING

DEFAULT:

None

No Correlation

OPTIONS:

MP2

Sections 5.2 and 5.3

Local_MP2

Section 5.4

MP3

Section 5.2

MP4SDQ

Section 5.2

MP4

Section 5.2

CCD

Section 5.5

CCD(2)

Section 5.6

CCSD

Section 5.5

CCSD(T)

Section 5.6

CCSD(2)

Section 5.6

QCISD

Section 5.5

QCISD(T)

Section 5.6

OD

Section 5.5

OD(T)

Section 5.6

OD(2)

Section 5.6

VOD

Section 5.7

VOD(2)

Section 5.7

QCCD

Section 5.5

VQCCD

Section 5.7

5.2 MØLLER-PLESSET PERTURBATION THEORY

5.2.1 INTRODUCTION

Møller-Plesset Perturbation Theory [5] is a widely used method for approximating the correlation energy of molecules. In particular, second order Møller-Plesset perturbation theory (MP2) is one of the simplest and most *useful* levels of theory beyond the Hartree-Fock approximation. Conventional and local MP2 methods available in Q-Chem are discussed in detail in Sections 5.3 and 5.4 respectively. The MP3 method is still occasionally used, while MP4 calculations are quite commonly employed as part of the G2 and G3 thermochemical methods [3,4]. In the remainder of this section, the theoretical basis of Møller-Plesset theory is reviewed.

5.2.2 THEORETICAL BACKGROUND

The Hartree-Fock wave function (Ψ_0) and energy (E_0) are *approximate* solutions (eigenfunction and eigenvalue) to the exact Hamiltonian eigenvalue problem or Schrödinger's electronic wave equation (4.5). The HF wave function and energy are, however, exact solutions for the Hartree-Fock Hamiltonian (H_0) eigenvalue problem. If we assume that the Hartree-Fock wave function (Ψ_0) and energy (E_0) lie near the exact wave function (Ψ) and energy (E), we can now write the exact Hamiltonian operator as

$$H = H_0 + \lambda V \quad (5.1)$$

where V is the small perturbation and λ is a dimensionless parameter. Expanding the exact wave function and energy in terms of the HF wave function and energy yields

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \lambda^3 E^{(3)} + \dots \quad (5.2)$$

$$\Psi = \Psi_0 + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \lambda^3 \Psi^{(3)} + \dots \quad (5.3)$$

substituting the expansions into the Schrödinger equation and gathering terms in λ^n yields

$$H_0 \Psi_0 = E^{(0)} \Psi_0 \quad (5.4)$$

$$H_0 \Psi^{(1)} + V \Psi_0 = E^{(0)} \Psi^{(1)} + E^{(1)} \Psi_0 \quad (5.5)$$

$$H_0 \Psi^{(2)} + V \Psi^{(1)} = E^{(0)} \Psi^{(2)} + E^{(1)} \Psi^{(1)} + E^{(2)} \Psi_0 \quad (5.6)$$

and so forth. Multiplying each of the above equations by Ψ_0 and integrating over all space yields the following expression for the n^{th} order (MP n) energy

$$E^{(0)} = \langle \Psi_0 | H_0 | \Psi_0 \rangle \quad (5.7)$$

$$E^{(1)} = \langle \Psi_0 | V | \Psi_0 \rangle \quad (5.8)$$

$$E^{(2)} = \langle \Psi_0 | V | \Psi^{(1)} \rangle \quad (5.9)$$

Thus, the Hartree-Fock energy

$$E_0 = \langle \Psi_0 | H_0 + V | \Psi_0 \rangle \quad (5.10)$$

is simply the sum of the zeroth- and first- order energies

$$E_0 = E^{(0)} + E^{(1)} \quad (5.11)$$

The correlation energy can then be written

$$E_{corr} = E_0^{(2)} + E_0^{(3)} + E_0^{(4)} + \dots \quad (5.12)$$

of which the first term is the MP2 energy.

It can be shown that the MP2 energy can be written (in terms of spinorbitals) as

$$E_0^{(2)} = -\frac{1}{4} \sum_{ab}^{virt} \sum_{ij}^{occ} \frac{|\langle ab || ij \rangle|^2}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} \quad (5.13)$$

where

$$\langle ab || ij \rangle = \langle ab | ij \rangle - \langle ab | ji \rangle \quad (5.14)$$

and

$$\langle ab | cd \rangle = \int \psi_a(\mathbf{r}_1) \psi_c(\mathbf{r}_1) \left[\frac{1}{r_{12}} \right] \psi_b(\mathbf{r}_2) \psi_d(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad (5.15)$$

which can be written in terms of the two electron repulsion integrals

$$\langle ab | cd \rangle = \sum_{\mu} \sum_{\nu} \sum_{\lambda} \sum_{\sigma} C_{\mu a} C_{\nu c} C_{\lambda b} C_{\sigma d} (\mu\nu | \lambda\sigma) \quad (5.16)$$

Expressions for higher order terms follow similarly, although with much greater algebraic and computational complexity. MP3 and particularly MP4 (the third and fourth order contributions to the correlation energy) are both occasionally used, although they are increasingly supplanted by the coupled cluster methods described in the following sections. The disk and memory requirements for MP3 are similar to the self-consistent pair correlation methods discussed in Section 5.5 while the computational cost of MP4 is similar to the (T) corrections discussed in Section 5.6.

5.3 EXACT MP2 METHODS

5.3.1 ALGORITHM

Second order Møller-Plesset theory (MP2) [5] probably the simplest useful wave function-based electron correlation method. Revived in the mid-1970's, it remains highly popular today, because it offers systematic improvement in optimized geometries and other molecular properties relative to Hartree-Fock (HF) theory [6]. Indeed, in a recent comparative study of small closed shell molecules [7], MP2 outperformed much more expensive singles and doubles coupled-cluster theory for such properties! Relative to state-of-the-art Kohn-Sham density functional theory (DFT) methods, which are the most economical methods to account for electron correlation effects, MP2 has the advantage of properly incorporating long-range dispersion forces. The principal weaknesses of MP2 theory are for open shell systems, and other cases where the HF determinant is a poor starting point.

Q-Chem contains an efficient conventional semi-direct method to evaluate the MP2 energy and gradient [8]. These methods require OVN memory (O, V, N are the numbers of occupied, virtual and total orbitals, respectively), and disk space which is bounded from above by $OVN^2/2$. The latter can be reduced to $IVN^2/2$ by treating the occupied orbitals in batches of size I , and re-evaluating the two-electron integrals O/I times. This approach is tractable on modern workstations for energy and gradient calculations of at least 500 basis functions or so, or molecules of between 15 and 30 first row atoms, depending on the basis set size. The computational cost increases between the 3rd and 5th power of the size of the molecule, depending on which part of the calculation is time-dominant.

The algorithm and implementation in Q-Chem is improved over earlier methods [9,10], particularly in the following areas:

- Uses pure functions, as opposed to Cartesians, for all fifth order steps. This leads to large computational savings for basis sets containing pure functions
- Customized loop unrolling for improved efficiency
- The sortless semi-direct method avoids a read and write operation resulting in a large I/O savings
- Reduction in disk and memory usage
- No extra integral evaluation for gradient calculations
- Full exploitation of frozen core approximation

The implementation offers the user the following alternatives:

1. direct algorithm (energies only)
2. disk-based sortless semi-direct algorithm (energies and gradients)
3. local occupied orbital method (energies only).

The semidirect algorithm is the only choice for gradient calculations. It is also normally the most efficient choice for energy calculations. There are two classes of exceptions:

- If the amount of disk space available is not significantly larger than the amount of memory available, then the direct algorithm is preferred.
- If the calculation involves a very large basis set, then the local orbital method may be faster, because it performs the transformation in a different order. It does not have the large memory requirement (no *OVN* array needed), and always evaluates the integrals 4 times. The *CD_DISK* option is also ignored in this algorithm, which requires up to *OOVN* words of disk space.

There are 3 important options that are not defaults that should be wisely chosen by the user in order to exploit the full efficiency of Q-Chem's direct and semidirect MP2 methods (as discussed above, the *LOCAL_OCCUPIED* method has different requirements).

- (1) *MEMORY*: The value specified for this REM variable must be sufficient to permit efficient integral evaluation (2-10MW) and to hold a large temporary array whose size is *OVN*, the product of the number of occupied, virtual and total numbers of orbitals.
- (2) *CD_DISK*: The value specified for this REM variable should be as large as possible (i.e. perhaps 80% of the free space on your *\$QCSCRATCH* partition where temporary job files are held). The value of this variable will determine how many times the two-electron integrals in the atomic orbital basis must be re-evaluated, which is a major computational step in MP2 calculations.
- (3) *N_FROZEN_CORE*: The computational requirements for MP2 are proportional to the number of occupied orbitals for some steps, and the square of that number for other steps. Therefore the CPU time can be significantly reduced if your job employs the frozen core approximation. Additionally the memory and disk requirements are reduced when the frozen core approximation is employed.

5.3.2 ALGORITHM CONTROL AND CUSTOMIZATION

The direct and semidirect integral transformation algorithms used by Q-Chem (*e.g.*, MP2, CIS(D)) are limited by available disk space (D) and memory (C), the number of basis functions (N), the number of virtual orbitals (V) and the number of occupied orbitals (O), as discussed above. The generic description of the key REM's are as follows:

MEMORY

Sets the memory for individual program modules

VARIABLE:

INTEGER

DEFAULT:

2,000,000 (2 MW)

OPTIONS:

User-defined number of words. For direct and semidirect MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (2-10 MW), as discussed above.

MEMORY_TOTAL

Sets the total memory available to Q-Chem

VARIABLE:

INTEGER

DEFAULT:

Unlimited (1,000 MW)

OPTIONS:

User-defined number of words

RECOMMENDATION:

Use default, or set to the physical memory of your machine.

CD_MAX_DISK

Sets the amount of disk space (in words) available for MP2 calculations

VARIABLE:

INTEGER

DEFAULT:

200,000,000 (200 MW)

OPTIONS:

User-defined: should be set as large as possible, discussed in Sec. 5.3.1

CD_ALGORITHM

Determines the algorithm for MP2 integral transformations

VARIABLE:

STRING

DEFAULT:

Program determined

OPTIONS:

DIRECT Uses fully direct algorithm (energies only)

SEMI_DIRECT Uses disk-based semi-direct algorithm

LOCAL_OCCUPIED Alternative energy algorithm (see 5.3.1)

RECOMMENDATION:

Semidirect is usually most efficient, and will normally be chosen by default.

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

FC Frozen Core approximation (all core orbitals frozen)

n Freeze n core orbitals

RECOMMENDATION:

While the default is not to freeze orbitals, MP2 calculations are more efficient with frozen core orbitals. Use FC if possible.

N_FROZEN_VIRTUAL

Sets the number of frozen virtual orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

n Freeze n virtual orbitals

5.3.3 EXAMPLES

```
$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH

OH = 1.01
HOH = 105
$end
```

```
$rem
JOBTYPE          SP          Single Point energy
CORRELATION      MP2
EXCHANGE         HF          Exact
BASIS            6-31G*
$end
```

Example 5.1 Example of an MP2/6-31G* calculation on the water molecule

```
$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH

OH = 1.01
HOH = 105
$end
```

```
$rem
JOBTYPE          SP          Single Point energy
CORRELATION      MP2
EXCHANGE         HF          Exact
BASIS            6-31G*
N_FROZEN_CORE    FC          Frozen core approximation
$end
```

Example 5.2 Example of an MP2/6-31G* calculation employing the frozen core approximation

5.4 LOCAL MP2 METHODS

5.4.1 LOCAL TRIATOMICS IN MOLECULES (TRIM) MODEL

The development of what may be called “fast methods” for evaluating electron correlation is a problem of both fundamental and practical importance, because of the unphysical increases in computational complexity with molecular size which afflict “exact” implementations of electron correlation methods. Ideally, the development of fast methods for treating electron correlation should not impact either model errors or numerical errors associated with the original electron correlation models. Unfortunately this is not possible at present, as may be appreciated from the following rough argument. *Spatial locality* is what permits reformulations of electronic structure methods that yield the same answer as traditional methods, but faster. The one-particle density matrix decays exponentially with a rate that relates to the HOMO-LUMO gap in periodic systems. When length scales longer than this characteristic decay length are examined, sparsity will emerge in both the one-particle density matrix and also pair correlation amplitudes expressed in terms of localized functions. Very roughly, such a length scale is about 5 to 10 atoms in a line, for good insulators such as alkanes. Hence sparsity emerges beyond this number of atoms in 1-d, beyond this number of atoms squared in 2-d, and this number of atoms cubed in 3-d. Thus for three-dimensional systems, locality only begins to emerge for systems of between hundreds and thousands of atoms.

If we wish to accelerate calculations on systems below this size regime, we must therefore introduce additional errors into the calculation, either as numerical noise through looser tolerances, or by modifying the theoretical model, or perhaps both. Q-Chem’s approach to local electron correlation is based on modifying the theoretical models describing correlation with an additional well-defined local approximation. We do not attempt to accelerate the calculations by introducing more numerical error because of the difficulties of controlling the error as a function of molecule size, and the difficulty of achieving reproducible significant results. From this perspective, local correlation becomes an integral part of specifying the electron correlation treatment. This means that the considerations necessary for a correlation treatment to qualify as a well-defined theoretical model chemistry apply equally to local correlation modeling. The local approximations should be

- (a) *Size-consistent*: meaning that the energy of a supersystem of two noninteracting molecules should be the sum of the energy obtained from individual calculations on each molecule.
- (b) *Uniquely defined*: Require no input beyond nuclei, electrons, and an atomic orbital basis set. In other words, the model should be uniquely specified without customization for each molecule.
- (c) *Yield continuous potential energy surfaces*: The model approximations should be smooth, and not yield energies that exhibit jumps as nuclear geometries are varied.

To ensure that these model chemistry criteria are met, Q-Chem's local MP2 methods [11,12] express the double substitutions (i.e. the pair correlations) in a redundant basis of atom-labeled functions. The advantage of doing this is that local models satisfying model chemistry criteria can be defined by performing an *atomic truncation* of the double substitutions. A general substitution in this representation will then involve the replacement of occupied functions associated with two given atoms by empty (or virtual) functions on two other atoms, coupling together 4 different atoms. We can force one occupied to virtual substitution (of the two that comprise a double substitution) to occur only between functions on the same atom, so that only 3 different atoms are involved in the double substitution. This defines the *triatomics in molecules* (TRIM) local model for double substitutions. The TRIM model offers the potential for reducing the computational requirements of exact MP2 theory by a factor proportional to the number of atoms. We could also force each occupied to virtual substitution to be on a given atom, thereby defining a more drastic *diatomics in molecules* (DIM) local correlation model.

The simplest atom-centered basis that is capable of spanning the occupied space is a *minimal basis* of core and valence atomic orbitals on each atom. Such a basis is necessarily redundant because it also contains sufficient flexibility to describe the empty valence antibonding orbitals necessary to correctly account for nondynamical electron correlation effects such as bond-breaking. This redundancy is actually important for the success of the atomic truncations because occupied functions on adjacent atoms to some extent describe the same part of the occupied space. The minimal functions we use to span the occupied space are obtained at the end of a large basis set calculation, and are called *extracted polarized atomic orbitals* (EPAO's) [13]. We discuss them briefly below. It is even possible to explicitly perform an SCF calculation in terms of a molecule-optimized minimal basis of *polarized atomic orbitals* (PAO's) (see Chapter 4). To span the virtual space, we use the full set of atomic orbitals, appropriately projected into the virtual space.

We summarize the situation. The number of functions spanning the occupied subspace will be the minimal basis set dimension, M , which is greater than the number of occupied orbitals, O , by a factor of up to about 2. The virtual space is spanned by the set of projected atomic orbitals whose number is the atomic orbital basis set size N , which is fractionally greater than the number of virtuals $V=N-O$. The number of double substitutions in such a redundant representation will be typically 3 to 5 times larger than the usual total. This will be more than compensated by reducing the number of retained substitutions by a factor of the number of atoms, A , in the local triatomics in molecules model, or a factor of A^2 in the diatomics in molecules model.

The local MP2 energy in the TRIM and DIM models are given by the following expressions, which can be compared against the full MP2 expression given earlier in Eq. (5.13). First, for the DIM model:

$$E_{DIMP2} = -\frac{1}{2} \sum_{\bar{P}\bar{Q}} \frac{(\bar{P}|\bar{Q})(\bar{P}||\bar{Q})}{\Delta_{\bar{P}} + \Delta_{\bar{Q}}} \quad (5.17)$$

The sums are over the linear number of atomic single excitations after they have been canonicalized. Each term in the denominator is thus an energy difference between occupied and virtual levels in this local basis. Similarly, the TRIM model corresponds to the following local MP2 energy:

$$E_{TRIMP2} = - \sum_{\bar{P}bj} \frac{(\bar{P} | jb)(\bar{P} || jb)}{\Delta_{\bar{P}} + \epsilon_b - \epsilon_j} - E_{DIMP2} \quad (5.18)$$

where the sum is now mixed between atomic substitutions (\bar{P}), and nonlocal occupied (j) to virtual (b) substitutions. See references [11,12] for a full derivation and discussion.

The accuracy of the local TRIM and DIM models has been tested in a series of calculations [11,12]. In particular, the TRIM model has been shown to be quite faithful to full MP2 theory via the following tests:

- (a) The TRIM model recovers around 99.7% of the MP2 correlation energy for covalent bonding. This is significantly higher than the roughly 98-99% correlation energy recovery typically exhibited by the Saebo-Pulay local correlation method [14]. The DIM model recovers around 95% of the correlation energy.
- (b) The performance of the TRIM model for relative energies is very robust, as shown in ref. [11] for the challenging case of torsional barriers in conjugated molecules. The RMS error in these relative energies is only 0.031 kcal/mol, as compared to around 1 kcal/mol when electron correlation effects are completely neglected.
- (c) For the water dimer with the aug-cc-pVTZ basis, 96% of the MP2 contribution to the binding energy is recovered with the TRIM model, as compared to 62% with the Saebo-Pulay local correlation method.
- (d) For calculations of the MP2 contribution to the G3 and G3(MP2) energies with the larger molecules in the G3-99 database [15], introduction of the TRIM approximation results in an RMS error relative to full MP2 theory of only 0.3 kcal/mol, even though the absolute magnitude of these quantities is on the order of tens of kcal/mol.

5.4.2 EPAO EVALUATION OPTIONS

When a local MP2 job (requested by the *LOCAL_MP2* option for *CORRELATION*) is performed, the first new step after the SCF calculation is converged is to extract a minimal basis of polarized atomic orbitals (EPAO's) that spans the occupied space. There are three valid choices for this basis, controlled by the *PAO_METHOD* and *EPAO_ITERATE* keywords described below.

- (1) *Uniterated EPAO's*: The initial guess EPAO's are the default for local MP2 calculations, and are defined as follows. For each atom, the covariant density matrix (SPS) is diagonalized, giving eigenvalues which are approximate natural orbital occupancies, and eigenvectors which are corresponding atomic orbitals. The m eigenvectors with largest populations are retained (where m is the minimal basis dimension for the current atom). This nonorthogonal minimal basis is symmetrically orthogonalized, and then modified as discussed in ref. [13] to ensure that these functions rigorously span the occupied space of the full SCF calculation that has just been performed. These orbitals may be denoted as EPAO(0) to indicate that no iterations have been performed after the guess. In general, the quality of the local MP2 results obtained with this option is very similar to the EPAO option below, but it is much faster and fully robust. For the example of the torsional barrier calculations [11] discussed above, the TRIM RMS deviations of 0.03 kcal/mol from full MP2 calculations are increased to only 0.04 kcal/mol when EPAO(0) orbitals are employed rather than EPAO's.
- (2) *EPAO's*: EPAO's are defined by minimizing a localization functional as described in ref. [13]. These functions were designed to be suitable for local MP2 calculations, and have yielded excellent results in all tests performed so far. Unfortunately the functional is difficult to converge for large molecules, at least with the algorithms that have been developed to this stage. Therefore it is not the default, but is switched on by specifying a (large) value for *EPAO_ITERATE*, as discussed below.
- (3) *PAO*: If the SCF calculation is performed in terms of a molecule-optimized minimal basis, as described in chapter 4, then the resulting PAO-SCF calculation can be corrected with either conventional or local MP2 for electron correlation. PAO-SCF calculations alter the SCF energy, and are therefore not the default. This can be enabled by specifying *PAO_METHOD* as PAO, in a job which also requests *CORRELATION* as *LOCAL_MP2*

PAO_METHOD

Controls the type of PAO calculations requested

VARIABLE:

STRING

DEFAULT:

EPAO For local MP2, EPAO's are chosen by default.

OPTIONS:

EPAO Find EPAO's by minimizing delocalisation function

PAO Do SCF in a molecule-optimized minimal basis

EPAO_ITERATE

Controls iterations for EPAO calculations (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

0 Use uniterated EPAO's based on atomic blocks of SPS.

OPTIONS:

n Optimize the EPAO's for up to n iterations.

RECOMMENDATION:

Use default. For molecules that are not too large, one can test the sensitivity of the results to the type of minimal functions by the use of optimised EPAO's in which case a value of $n=500$ is reasonable.

EPAO_WEIGHTS

Controls algorithm and weights for EPAO calculations (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

115 Standard weights, use 1st and 2nd order optimisation

OPTIONS:

15 Standard weights, with 1st order optimisation only.

RECOMMENDATION:

Use default, unless convergence failure is encountered.

5.4.3 ALGORITHM CONTROL AND CUSTOMIZATION

A local MP2 calculation (requested by the *LOCAL_MP2* option for *CORRELATION*) consists of the following steps:

- After the SCF is converged, a minimal basis of EPAO's are obtained.
- The TRIM (and DIM) local MP2 energies are then evaluated (gradients are not yet available).

Details of the efficient implementation of the local MP2 method described above are reported in the recent thesis of Dr. Michael Lee [16], and will shortly be published in the scientific literature [17]. Here we simply summarize the capabilities of the program. The computational advantage associated with these local MP2 methods varies depending upon the size of molecule and the basis set. As a rough general estimate, TRIM-MP2 calculations are feasible on molecule sizes about twice as large as those for which conventional MP2 calculations are feasible on a given computer, and this is their primary advantage. Our implementation is well suited for large basis set calculations. The AO basis two-electron integrals are evaluated four times. DIM-MP2 calculations are performed as a by-product of TRIM-MP2 but no separately optimized DIM algorithm has been implemented.

The resource requirements for local MP2 calculations are as follows:

- *Memory_total*: The memory requirement for the integral transformation does not exceed *OON*, and is thresholded so that it asymptotically grows linearly with molecule size. Additional memory of approximately $32N^2$ is required to complete the local MP2 energy evaluation.
- *Disk*: The disk space requirement is only about $8OVN$, but is not thresholded. This is a very large reduction from the case of a full MP2 calculation, where, in the case of 4 integral evaluations, $OVN^2/4$ disk space is required. As the local MP2 disk space requirement is not adjustable, the *CD_DISK* keyword is ignored for *LOCAL_MP2* calculations.

The evaluation of the local MP2 energy does not require any further customization. An adequate amount of *MEMORY* (5 to 10 million words) should be specified to permit efficient AO basis two-electron integral evaluation, but all large scratch arrays are allocated from *MEMORY_TOTAL*.

5.4.4 EXAMPLES

```

$molecule
0 1
C
C,1,1.32095
C,2,1.478448,1,121.185244
O,3,1.18974,2,123.834335,1,180.
H,1,1.076863,2,121.497711,3,0.
H,1,1.074497,2122.085528,3,180.
H,2,1.075494,1,122.339969,3,180.
H,3,1.094859,2,115.27087,4,180.
$end

```

```

$rem
correlation      local_mp2
basis            6-311g**
$end

```

```

@@@
$molecule
0 1
C
C,1 316563
C,2,1.498384,1,123.439464
O,3,1.18747,2,123.811782,1,92.283611
H,1,1.07631,2,122.029762,3,-0.310147
H,1,1.074844,2,121.429505,3,180.276629
H,2,1.078134,1,120.962285,3,180.340519
H,3,1.093867,2,115.870616,4,179.067691
$end

```

```

$rem
correlation      local_mp2
basis            6-311g**
$end

```

Example 5.3 A relative energy evaluation using the local TRIM model for MP2 with the 6-311G** basis set. The energy difference is the internal rotation barrier in propenal, with the first geometry being planar trans, and the second the transition structure.

5.5 SELF-CONSISTENT PAIR CORRELATION METHODS

The following sections give short summaries of the various pair correlation methods available in Q-Chem, all of which are variants of coupled cluster theory. The basic object-oriented tools necessary to permit the implementation of these methods in Q-Chem was accomplished by Dr. Anna Krylov and Dr. David Sherrill, working at Berkeley with Martin Head-Gordon, and then continuing independently at the University of Southern California and Georgia Tech respectively. While at Berkeley, Krylov and Sherrill also developed the optimized orbital coupled cluster method, with additional assistance from Ed Byrd. The extension of this code to MP3, MP4, CCSD and QCISD is the work of Dr. Steve Gwaltney at Berkeley, while the extensions to QCCD were implemented by Ed Byrd at Berkeley.

5.5.1 COUPLED CLUSTER SINGLES AND DOUBLES (CCSD)

The standard approach for treating pair correlations self-consistently are coupled cluster methods where the cluster operator contains all single and double substitutions [18], abbreviated as CCSD. CCSD yields results that are only slightly superior to MP2 for structures and frequencies of stable closed shell molecules. However, it is far superior for reactive species, such as transition structures and radicals, for which the performance of MP2 is quite erratic. Q-Chem supports only energy evaluation for CCSD at present.

A full textbook presentation of CCSD is beyond the scope of this manual, and several comprehensive references are available. However, it may be useful to briefly summarize the main equations. The CCSD wavefunction is:

$$|\Psi_{CCSD}\rangle = \exp(\hat{T}_1 + \hat{T}_2)|\Phi_0\rangle \quad (5.19)$$

where the single and double excitation operators may be defined by their actions on the reference single determinant (which is normally taken as the Hartree-Fock determinant in CCSD):

$$\hat{T}_1|\Phi_0\rangle = \sum_i^{\text{occ}} \sum_a^{\text{virt}} t_i^a |\Phi_i^a\rangle \quad (5.20)$$

$$\hat{T}_2|\Phi_0\rangle = \frac{1}{4} \sum_{ij}^{\text{occ}} \sum_{ab}^{\text{virt}} t_{ij}^{ab} |\Phi_{ij}^{ab}\rangle \quad (5.21)$$

It is unfeasible to determine the CCSD energy by variational minimization of $\langle E \rangle_{CCSD}$ with respect to the singles and doubles amplitudes because the expressions terminate at the same level of complexity as full configuration interaction (!). So, instead, the Schrödinger equation is satisfied in the subspace spanned by the reference determinant, all single substitutions, and all double substitutions. Projection with these functions and integration over all space provides sufficient equations to determine the energy, the

singles and doubles amplitudes as the solutions of sets of nonlinear equations. These equations may be symbolically written as follows:

$$\begin{aligned} E_{CCSD} &= \langle \Phi_0 | \hat{H} | \Psi_{CCSD} \rangle \\ &= \langle \Phi_0 | \hat{H} | (1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2) \Phi_0 \rangle_C \end{aligned} \quad (5.22)$$

$$\begin{aligned} 0 &= \langle \Phi_i^a | \hat{H} - E_{CCSD} | \Psi_{CCSD} \rangle \\ &= \langle \Phi_i^a | \hat{H} | (1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 + \frac{1}{3!} \hat{T}_1^3) \Phi_0 \rangle_C \end{aligned} \quad (5.23)$$

$$\begin{aligned} 0 &= \langle \Phi_{ij}^{ab} | \hat{H} - E_{CCSD} | \Psi_{CCSD} \rangle \\ &= \langle \Phi_{ij}^{ab} | \hat{H} | (1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 + \frac{1}{3!} \hat{T}_1^3 + \frac{1}{2} \hat{T}_2^2 + \frac{1}{2} \hat{T}_1^2 \hat{T}_2 + \frac{1}{4!} \hat{T}_1^4) \Phi_0 \rangle_C \end{aligned} \quad (5.24)$$

The result is a set of equations which yield an energy that is not necessarily variational (i.e. may not be above the true energy), although it is strictly size-consistent. The equations are also exact for a pair of electrons, and, to the extent that molecules are a collection of interacting electron pairs, this is the basis for expecting that CCSD results will be of useful accuracy.

The computational effort necessary to solve the CCSD equations can be shown to scale with the 6th power of the molecular size, for fixed choice of basis set. Disk storage scales with the 4th power of molecular size, and involves a number of sets of doubles amplitudes, as well as two-electron integrals in the molecular orbital basis. Therefore the improved accuracy relative to MP2 theory comes at a steep computational cost. Given these scalings it is relatively straightforward to estimate the feasibility (or unfeasibility) of a CCSD calculation on a larger molecule (or with a larger basis set) given that a smaller trial calculation is first performed.

5.5.2 QUADRATIC CONFIGURATION INTERACTION (QCISD)

Quadratic configuration interaction with singles and doubles (QCISD) [19] is a widely used alternative to CCSD, that shares its main desirable properties of being size-consistent, exact for pairs of electrons, as well as being also nonvariational. Its computational cost also scales in the same way with molecule size and basis set as CCSD, although with slightly smaller constants. While originally proposed independently of CCSD based on correcting configuration interaction equations to be size-consistent, QCISD is probably best viewed as approximation to CCSD. The defining equations are given below (under the assumption of Hartree-Fock orbitals, which should always be used in QCISD). The QCISD equations can clearly be viewed as the CCSD equations with a large number of terms omitted, which are evidently not very numerically significant:

$$E_{QCISD} = \left\langle \Phi_0 \left| \hat{H} \left(1 + \hat{T}_2 \right) \Phi_0 \right\rangle_C \quad (5.25)$$

$$0 = \left\langle \Phi_i^a \left| \hat{H} \left(\hat{T}_1 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 \right) \Phi_0 \right\rangle_C \quad (5.26)$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \left(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2 \right) \Phi_0 \right\rangle_C \quad (5.27)$$

QCISD energies are available in Q-Chem, and are requested with the QCISD keyword. As discussed in Section 4.6, the noniterative QCISD(T) correction to the QCISD solution is also available to approximately incorporate the effect of higher substitutions.

5.5.3 OPTIMIZED ORBITAL COUPLED CLUSTER DOUBLES (OD)

It is possible to greatly simplify the CCSD equations by omitting the single substitutions (i.e. setting the T_1 operator to zero). If the same single determinant reference is used (specifically the Hartree-Fock determinant), then this defines the coupled cluster doubles (CCD) method, by the following equations:

$$E_{CCD} = \left\langle \Phi_0 \left| \hat{H} \left(1 + \hat{T}_2 \right) \Phi_0 \right\rangle_C \quad (5.28)$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \left(1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2 \right) \Phi_0 \right\rangle_C \quad (5.29)$$

The CCD method cannot itself usually be recommended because while pair correlations are all correctly included, the neglect of single substitutions causes calculated energies and properties to be significantly less reliable than for CCSD. Single substitutions play a role very similar to orbital optimization, in that they effectively alter the reference determinant to be more appropriate for the description of electron correlation (the Hartree-Fock determinant is optimized in the absence of electron correlation).

This suggests an alternative to CCSD and QCISD that has some additional advantages. This is the optimized orbital CCD method (OO-CCD), which we normally refer to as simply optimized doubles (OD) [20]. The OD method is defined by the CCD equations above, plus the additional set of conditions that the cluster energy is minimized with respect to orbital variations. This may be mathematically expressed by:

$$\frac{\partial E_{CCD}}{\partial \theta_i^a} = 0 \quad (5.30)$$

where the rotation angle θ_i^a mixes the i^{th} occupied orbital with the a^{th} virtual (empty) orbital. Thus the orbitals that define the single determinant reference are optimized to minimize the coupled cluster energy, and are variationally best for this purpose. The resulting orbitals are approximate Brueckner orbitals.

The OD method has the advantage of formal simplicity (orbital variations and single substitutions are essentially redundant variables). In cases where Hartree-Fock theory performs poorly (for example artifactual symmetry breaking, or nonconvergence), it is also practically advantageous to use the OD method, where the HF orbitals are not required, rather than CCSD or QCISD. Q-Chem supports both energies and analytical gradients using the OD method. The computational cost for the OD energy is more than twice that of the CCSD or QCISD method, but the total cost of energy plus gradient is roughly similar, although OD remains more expensive. An additional advantage of the OD method is that it can be performed in an active space, as discussed later, in section 5.7.

5.5.4 QUADRATIC COUPLED CLUSTER DOUBLES (QCCD)

The nonvariational determination of the energy in the CCSD, QCISD, and OD methods discussed in the above subsections is not normally a practical problem. However, there are some cases where these methods perform poorly. One such example are potential curves for homolytic bond dissociation, using closed shell orbitals, where the calculated energies near dissociation go significantly below the true energies, giving potential curves with unphysical barriers to formation of the molecule from the separated fragments [21]. The Quadratic Coupled Cluster Doubles (QCCD) method [22] recently proposed by Troy Van Voorhis at Berkeley uses a different energy functional to yield improved behavior in problem cases of this type. Specifically, the QCCD energy functional is defined as:

$$E_{QCCD} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2 + \frac{1}{2} \hat{\Lambda}_2^2 \right) \middle| \hat{H} \middle| \exp(\hat{T}_2) \Phi_0 \right\rangle_C \quad (5.31)$$

where the amplitudes of both the \hat{T}_2 and $\hat{\Lambda}_2$ operators are determined by minimizing the QCCD energy functional. Additionally, the optimal orbitals are determined by minimizing the QCCD energy functional with respect to orbital rotations mixing occupied and virtual orbitals.

To see why the QCCD energy should be an improvement on the OD energy, we first write the latter in a different way than before. Namely, we can write a CCD energy functional which when minimized with respect to the \hat{T}_2 and $\hat{\Lambda}_2$ operators, gives back the same CCD equations defined earlier. This energy functional is:

$$E_{CCD} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2 \right) \middle| \hat{H} \middle| \exp(\hat{T}_2) \Phi_0 \right\rangle_C \quad (5.32)$$

Minimization with respect to the $\hat{\Lambda}_2$ operator gives the equations for the \hat{T}_2 operator presented previously, and, if those equations are satisfied then it is clear that we do not require knowledge of the $\hat{\Lambda}_2$ operator itself to evaluate the energy.

Comparing the two energy functionals, (5.31) and (5.32), we see that the QCCD functional includes up through quadratic terms of the Maclaurin expansion of $\exp(\hat{\Lambda}_2)$ while the conventional CCD functional includes only linear terms. Thus the bra wavefunction and the ket wavefunction in the energy expression are treated more equivalently in QCCD than in CCD. This makes QCCD closer to a true variational treatment [21] where the bra and ket wavefunctions are treated precisely equivalently, but without the exponential cost of the variational method.

In practice QCCD is a dramatic improvement relative to any of the conventional pair correlation methods for processes involving more than two active electrons (i.e. the breaking of at least a double bond, or, two spatially close single bonds). For example calculations, we refer to the original paper [22], and the follow-up paper describing the full implementation [23]. We note that these improvements carry a computational price. While QCCD scales formally with the 6th power of molecule size like CCSD, QCISD, and OD, the coefficient is substantially larger. For this reason, QCCD calculations are by default performed as OD calculations until they are partly converged.

5.5.5 JOB CONTROL OPTIONS

There are a large number of options for the coupled cluster singles and doubles methods. They are documented in Appendix C, and, as the reader will find upon following this link, it is an extensive list indeed. Fortunately, many of them are not necessary for routine jobs. Most of the options for non-routine jobs concern altering the default iterative procedure, which is most often necessary for optimized orbital calculations (OD, QCCD), as well as the active space methods discussed later in Section 5.7. The more common options relating to convergence control are discussed there, in Section 5.7.5. Below we list the options that one should be aware of for routine calculations.

CC_CONVERGENCE

Overall convergence criterion for the coupled cluster codes. This is designed to ensure at least n significant digits in the calculated energy, and automatically sets the other convergence-related variables (*CC_E_CONV*, *CC_T_CONV*, *CC_THETA_CONV*, *CC_THETA_GRAD_CONV*, *CC_Z_CONV*) [$10^{*(-n)}$]

VARIABLE:

INTEGER

DEFAULT:

8 energies
8 gradients

OPTIONS

n $10^{*(-n)}$ convergence criterion

CC_DOV_THRESH

Specifies minimum allowed values for the coupled cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the guess is poor.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{*(-de)}$

DEFAULT:

0.25

RECOMMENDATION:

Increase to 0.5 or 0.75 for nonconvergent coupled cluster calculations.

CC_MAXITER

Maximum number of iterations to optimize the coupled cluster energy.

VARIABLE:

INTEGER

DEFAULT:

200

OPTIONS:

n up to n iterations to achieve convergence

CC_PRINT

Controls the output from post-MP2 coupled cluster module of Q-Chem

VARIABLE:

INTEGER

DEFAULT:

1

OPTIONS:

0 → 7 Higher values can lead to deforestation...

5.5.6 EXAMPLES

```
$molecule
0 2
n
h1 n 1.02805
h2 n 1.02805 h1 103.337613
$end
```

```
$rem
correlation      ccsd
basis            6-31g*
n_frozen_core    fc
$end
```

```
@@@
$molecule
0 2
n
h1 n 1.02805
h2 n 1.02805 h1 103.337613
$end
```

```
$rem
correlation      od
basis            6-31g*
n_frozen_core    fc
$end
```

```
$molecule
0 2
n
h1 n 1.02805
h2 n 1.02805 h1 103.337613
$end
```

```
$rem
correlation      qccd
basis            6-31g*
n_frozen_core    fc
$end
```

Example 5.4 A series of jobs evaluating the correlation energy (with core orbitals frozen) of the ground state of the NH_2 radical with three methods of coupled cluster singles and doubles type: CCSD itself, OD, and QCCD.

5.6 NON-ITERATIVE CORRECTIONS TO COUPLED CLUSTER ENERGIES

5.6.1 (T) TRIPLES CORRECTIONS

To approach chemical accuracy in reaction energies and related properties, it is necessary to account for electron correlation effects that involve 3 electrons simultaneously, as represented by triple substitutions relative to the mean field single determinant reference, which arise in MP4. The best standard methods for including triple substitutions are the CCSD(T) [24] and QCISD(T) methods [19]. The accuracy of these methods is well-documented for many cases [25], and in general is a very significant improvement relative to the starting point (either CCSD or QCISD). The cost of these corrections scales with the 7th power of molecule size (or the 4th power of the number of basis functions for fixed molecule size), although no additional disk resources are required relative to the starting coupled cluster calculation. Q-Chem supports the evaluation of CCSD(T) and QCISD(T) energies, as well as the corresponding OD(T) correction to the optimized doubles method discussed in the previous subsection. Gradients are not currently available for any of these (T) corrections.

5.6.2 (2) TRIPLES AND QUADRUPLES CORRECTIONS

While the (T) corrections discussed above have been extraordinarily successful, there is nonetheless still room for further improvements in accuracy, for at least some important classes of problems. They contain judiciously chosen terms from 4th and 5th order Moller-Plesset perturbation theory, as well as higher order terms that result from the fact that the converged cluster amplitudes are employed to evaluate the 4th and 5th order terms. The (T) correction therefore depends upon the bare reference orbitals and orbital energies, and in this way its effectiveness still depends on the quality of the reference determinant. Since we are correcting a coupled cluster solution rather than a single determinant, this is an aspect of the (T) corrections that can be improved. Deficiencies of the (T) corrections show up computationally in cases where there are near-degeneracies between orbitals, such as stretched bonds, some transition states, open shell radicals, and biradicals.

Recently, Steve Gwaltney working at Berkeley with Martin Head-Gordon has suggested a new class of noniterative correction that offers the prospect of improved accuracy in problem cases of the types identified above [26]. Q-Chem contains Gwaltney's implementation of this new method, for energies only. The new correction is a true second order correction to a coupled cluster starting point, and is therefore denoted as (2). It is available for 2 of the cluster methods discussed above, as OD(2) and CCSD(2) [26,27]. Only energies are available at present.

The basis of the (2) method is to partition not the regular Hamiltonian into perturbed and unperturbed parts, but rather to partition a similarity-transformed Hamiltonian, defined as $\hat{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}}$. In the truncated space (call it the p -space) within which the cluster problem is solved (e.g. singles and doubles for CCSD), the coupled cluster wavefunction is a true eigenvalue of \hat{H} . Therefore we take the zero order Hamiltonian, $\hat{H}^{(0)}$, to be the

full \hat{H} in the p-space, while in the space of excluded substitutions (the q-space) we take only the one-body part of \hat{H} (which can be made diagonal). The fluctuation potential describing electron correlations in the q -space is $\hat{H} - \hat{H}^{(0)}$, and the (2) correction then follows from second order perturbation theory.

The new partitioning of terms between the perturbed and unperturbed Hamiltonians inherent in the (2) correction leads to a correction that shows both similarities and differences relative to the existing (T) corrections. There are two types of higher correlations that enter at second order: not only triple substitutions, but also quadruple substitutions. The quadruples are treated with a factorization ansatz, that is exact in 5th order Moller-Plesset theory [28], to reduce their computational cost from N^9 to N^6 . For large basis sets this can still be larger than the cost of the triples terms, which scale as the 7th power of molecule size, with a factor twice as large as the usual (T) corrections.

These corrections are feasible for molecules containing between four and ten first row atoms, depending on computer resources, and the size of the basis set chosen. There is early evidence that the (2) corrections are superior to the (T) corrections for highly correlated systems [26]. This shows up in improved potential curves, particularly at long range and may also extend to improved energetic and structural properties at equilibrium in problematical cases. It will be some time before sufficient testing on the new (2) corrections has been done to permit a general assessment of the performance of these methods. However, they are clearly very promising, and for this reason they are available in Q-Chem.

5.6.3 JOB CONTROL OPTIONS

The evaluation of a noniterative (T) or (2) correction after a coupled cluster singles and doubles level calculation (either CCSD, QCISD or OD) is controlled by the correlation keyword, and the specification of any frozen orbitals via *N_FROZEN_CORE* (and possibly *N_FROZEN_VIRTUAL*).

There is only one additional job control option. For the (2) correction, it is possible to apply the frozen core approximation in the reference coupled cluster calculation, and then correlate all orbitals in the (2) correction. This is controlled by *CC_INCL_CORE_CORR*, described below.

The default is to include core and core-valence correlation automatically in the CCSD(2) or OD(2) correction, if the reference CCSD or OD calculation was performed with frozen core orbitals. The reason for this choice is that core correlation is economical to include via this method (the main cost increase is only linear in the number of core orbitals), and such effects are important to account for in accurate calculations. This option should be made false if a job with explicitly frozen core orbitals is desired. One good reason for freezing core orbitals in the correction is if the basis set is physically inappropriate for describing core correlation (e.g. standard Pople basis sets, and Dunning cc-pVxZ basis sets are designed to describe valence-only correlation effects). Another good reason is if a direct comparison is desired against another method such as CCSD(T) which is always used in the same orbital window as the CCSD reference.

CC_INCL_CORE_CORR

Whether to include the correlation contribution from frozen core orbitals in noniterative (2) corrections, such as OD(2) and CCSD(2).

VARIABLE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

RECOMMENDATION:

Use default unless no core-valence or core correlation is desired (e.g. for comparison with other methods or because the basis used cannot describe core correlation).

5.6.4 EXAMPLES

```
$molecule
0 2
o
h o 0.97907
$end
```

```
$rem
correlation          CCSD(T)
basis                cc-pVTZ
n_frozen_core       fc
$end
```

```
@@@
$molecule
0 2
o
h o 0.97907
$end
```

```
$rem
correlation          CCSD(2)
basis                cc-pVTZ
n_frozen_core       fc
cc_incl_core_corr   false
$end
```

Example 5.5 Two jobs that compare the correlation energy calculated via the standard CCSD(T) method with the new CCSD(2) approximation, both using the frozen core approximation. This requires that `CC_INCL_CORE_CORR` must be specified as `false` in the CCSD(2) input.

5.7 COUPLED CLUSTER ACTIVE SPACE METHODS

5.7.1 INTRODUCTION

Electron correlation effects can be qualitatively divided into two classes. The first class is static or nondynamical correlation: long wavelength low-energy correlations associated with other electron configurations that are nearly as low in energy as the lowest energy configuration. These correlation effects are important for problems such as homolytic bond breaking, and are the hardest to describe because by definition the single configuration Hartree-Fock description is not a good starting point. The second class is dynamical correlation: short wavelength high-energy correlations associated with atomic-like effects. Dynamical correlation is essential for *quantitative* accuracy, but a reasonable description of static correlation is a prerequisite for a calculation being *qualitatively* correct.

In the methods discussed in the previous several subsections, the objective was to approximate the total correlation energy. However, in some cases, it is useful to instead directly model the nondynamical and dynamical correlation energies separately. The reasons for this are pragmatic: with approximate methods, such a separation can give a better balanced treatment of electron correlation along bond-breaking coordinates, or reaction coordinates that involve biradicaloid intermediates. The nondynamical correlation energy is conveniently defined as the solution of the Schrodinger equation within a small basis set composed of valence bonding, antibonding and lone pair orbitals: the so-called full valence active space. Solved exactly, this is the so-called full valence complete active space SCF (CASSCF) [29], or equivalently, the fully optimized reaction space (FORS) method [30].

Full valence CASSCF and FORS involve computational complexity which increases exponentially with the number of atoms, and is thus unfeasible beyond systems of only a few atoms, unless the active space is further restricted on a case-by-case basis. Q-Chem includes two relatively economical methods that directly approximate these theories using a truncated coupled cluster doubles wave function with optimized orbitals [31]. They are active space generalizations of the OD and QCCD methods discussed previously in Sections 5.5.3 and 5.5.4, and are discussed in the following two subsections. By contrast with the exponential growth of computational cost with problem size associated with exact solution of the full valence CASSCF problem, these cluster approximations have only 6th order growth of computational cost with problem size, while often providing useful accuracy.

The full valence space is a well-defined theoretical chemical model. For these active space coupled cluster doubles methods, it consists of the union of *valence* levels that are occupied in the single determinant reference, and those that are empty. The occupied levels that are to be replaced can only be the occupied valence and lone pair orbitals, whose number is defined by the sum of the valence electron counts for each atom (i.e. 1 for H, 2 for He, 1 for Li, etc.). At the same time, the empty virtual orbitals to which the double substitutions occur are restricted to be empty (usually antibonding) valence

orbitals. Their number is the difference between the number of valence atomic orbitals, and the number of occupied valence orbitals given above. This definition (the full valence space) is the default when either of the “valence” active space methods are invoked (VOD or VQCCD)

There is also a second useful definition of a valence active space, which we shall call the 1:1 or perfect pairing active space. In this definition, the number of occupied valence orbitals remains the same as above. The number of empty correlating orbitals in the active space is defined as being exactly the same number, so that each occupied orbital may be regarded as being associated 1:1 with a correlating virtual orbital. In the water molecule, for example, this means that the lone pair electrons as well as the bond-orbitals are correlated. Generally the 1:1 active space recovers more correlation for molecules dominated by elements on the right of the periodic table, while the full valence active space recovers more correlation for molecules dominated by atoms to the left of the periodic table.

If you wish to specify either the 1:1 active space as described above, or some other choice of active space based on your particular chemical problem, then you must specify the numbers of active occupied and virtual orbitals. This is done via the standard “window options”, documented earlier in the chapter.

Finally we note that the entire discussion of active spaces here leads only to specific numbers of active occupied and virtual orbitals. The orbitals that are contained within these spaces are optimized by minimizing the trial energy with respect to all the degrees of freedom previously discussed: the substitution amplitudes, and the orbital rotation angles mixing occupied and virtual levels. In addition, there are new orbital degrees of freedom to be optimized to obtain the best active space of the chosen size, in the sense of yielding the lowest coupled cluster energy. Thus rotation angles mixing active and inactive occupied orbitals must be varied until the energy is stationary. Denoting inactive orbitals by primes and active orbitals without primes this corresponds to satisfying:

$$\frac{\partial E_{CCD}}{\partial \theta_i^j} = 0 \quad (5.33)$$

Likewise the rotation angles mixing active and inactive virtual orbitals must also be varied until the coupled cluster energy is minimized with respect to these degrees of freedom.

$$\frac{\partial E_{CCD}}{\partial \theta_a^b} = 0 \quad (5.34)$$

5.7.2 VOD AND VOD(2) METHODS

The VOD method is the active space version of the OD method described earlier in Sec. 5.5.3. Both energies and gradients are available for VOD, so structure optimization is possible. There are a few important comments to make about the usefulness of VOD. First, it is a method that is capable of accurately treating problems that fundamentally involve 2 active electrons in a given local region of the molecule. It is therefore a good alternative for describing single bond-breaking, or torsion around a double bond, or some classes of diradicals. However it often performs poorly for problems where there is more than one bond being broken in a local region, with the nonvariational solutions being quite possible. For such problems the newer VQCCD method is substantially more reliable.

Assuming that VOD is a valid zero order description for the electronic structure, then a second order correction, VOD(2), is available for energies only. VOD(2) is a version of OD(2) generalized to valence active spaces. It permits more accurate calculations of relative energies by accounting for dynamical correlation.

5.7.3 VQCCD

The VQCCD method is the active space version of the QCCD method described earlier in Sec. 5.5.3. Both energies and gradients are available for VQCCD, so that structure optimization is possible. VQCCD is applicable to a substantially wider range of problems than the VOD method, because the modified energy functional is not vulnerable to nonvariational collapse. Testing to date suggests that it is capable of describing double bond breaking to similar accuracy as full valence CASSCF, and that potential curves for triple bond-breaking are qualitatively correct, although quantitatively in error by a few 10's of kcal/mol. The computational cost scales in the same manner with system size as the VOD method, albeit with a significantly larger prefactor.

5.7.4 CONVERGENCE STRATEGIES AND MORE ADVANCED OPTIONS

These optimized orbital coupled cluster active space methods enable the use of the full valence space for larger systems than is possible with conventional complete active space codes. However, we should note at the outset that often there are substantial challenges in converging valence active space calculations (and even sometimes optimized orbital coupled cluster calculations without an active space). Active space calculations cannot be regarded as “routine” calculations in the same way as SCF calculations, and often require a considerable amount of computational trial and error to persuade them to converge. These difficulties are largely because of strong coupling between the orbital degrees of freedom and the amplitude degrees of freedom, as well as the fact that the energy surface is often quite flat with respect to the orbital variations defining the active space.

Being aware of this at the outset, and realizing that the program has nothing against you personally is useful information for the uninitiated user of these methods. What the program does have, to assist in the struggle to achieve a converged solution, are accordingly many convergence options, fully documented in Appendix C. In this section, we describe the basic options and the ideas behind using them as a starting point. Experience plays a critical role, however, and so we encourage you to experiment with toy jobs that give rapid feedback in order to become proficient at diagnosing problems.

If the default procedure fails to converge, the first useful option to employ is `CC_PRECONV_T2Z`, with a value of between 10 and 50. This is useful for jobs in which the MP2 amplitudes are very poor guesses for the converged cluster amplitudes, and therefore initial iterations varying only the amplitudes will be beneficial:

CC_PRECONV_T2Z

Whether to pre-converge the cluster amplitudes before beginning orbital optimization in optimized orbital cluster methods.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

10 If `CC_RESTART`, `CC_RESTART_NO_SCF` or `CC_MP2NO_GUESS` are TRUE

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to *n* iterations in this pre-convergence procedure.

RECOMMENDATION

Experiment with this option in cases of convergence failure.

Other options that are useful include those that permit some damping of step sizes, and modify or disable the standard DIIS procedure. The main choices are as follows:

CC_DIIS

Specify the version of Pulay's Direct Inversion of the Iterative Subspace (DIIS) convergence accelerator to be used in the coupled cluster code.

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

- | | |
|---|--|
| 0 | Activates procedure 2 initially, and procedure 1 when gradients are smaller than <i>DIIS12_SWITCH</i> . |
| 1 | Uses error vectors defined as differences between parameter vectors from successive iterations. Most efficient near convergence. |
| 2 | Error vectors are defined as gradients scaled by square root of the approximate diagonal Hessian. Most efficient far from convergence. |

RECOMMENDATION

DIIS1 can be more stable. If DIIS problems are encountered in the early stages of a calculation (when gradients are large) try DIIS1.

CC_DIIS_START

Iteration number when DIIS is turned on. Set to a large number to disable DIIS.

VARIABLE:

INTEGER

DEFAULT:

3

RECOMMENDATION

Occasionally DIIS can cause optimized orbital coupled cluster calculations to diverge through large orbital changes. If this is seen, DIIS should be disabled.

CC_DOV_THRESH

Specifies minimum allowed values for the coupled cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the guess is poor.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{**(-de)}$

DEFAULT:

0.25

RECOMMENDATION:

Increase to 0.5 or 0.75 for nonconvergent coupled cluster calculations.

CC_THETA_STEPSIZE

Scale factor for the orbital rotation step size. The optimal rotation steps should be approximately equal to the gradient vector.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{**(-de)}$

DEFAULT:

1.0 If the initial step is smaller than 0.5, the program will increase step when gradients are smaller than the value of *THETA_GRAD_THRESH*, up to a limit of 0.5.

RECOMMENDATION:

Try a smaller value in cases of poor convergence and very large orbital gradients. For example, a value of 01000 translates to 0.1

An even stronger more or less last resort option permits iteration of the cluster amplitudes without changing the orbitals:

CC_PRECONV_T2Z_EACH

Whether to pre-converge the cluster amplitudes before each change of the orbitals in optimized orbital coupled cluster methods. The maximum number of iterations in this pre-convergence procedure is given by the value of this parameter.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to *n* iterations in this pre-convergence procedure.

RECOMMENDATION:

A very slow last resort option for jobs that do not converge.

5.7.5 EXAMPLES

```
$molecule
0 1
o
h 1 r
h 1 r a

r=1.5
a=104.5
$end

$rem
correlation          VOD
exchange             HF
basis                6-31G
$end

@@@
$molecule
READ
$end

$rem
correlation          VQCCD
exchange             HF
basis                6-31G
$end
```

Example 5.5 Two jobs that compare the correlation energy of the water molecule with partially stretched bonds, calculated via the two coupled cluster active space methods, VOD, and VQCCD. These are relatively “easy” jobs to converge, and may be contrasted with the next example, which is not easy to converge. The orbitals are restricted.

```

$molecule
0 1
o
h 1 r
h 1 r a

r=3.0
a=104.5
$end

$rem
correlation      VOD
exchange         HF
basis            6-31G
scf_convergence  9
thresh           12
cc_preconv_t2z   50
cc_preconv_t2z_each 50
cc_dov_thresh    7500
cc_theta_stepsize 3200
cc_diis_start    75
$end

@@@
$molecule
READ
$end

$rem
correlation      VQCCD
exchange         HF
basis            6-31G
scf_convergence  9
thresh           12
cc_preconv_t2z   50
cc_preconv_t2z_each 50
cc_dov_thresh    7500
cc_theta_stepsize 3200
cc_diis_start    75
$end

```

Example 5.6 The water molecule with highly stretched bonds, calculated via the two coupled cluster active space methods, VOD, and VQCCD. These are “difficult” jobs to converge. The convergence options shown permitted the job to converge after some experimentation (thanks due to Ed Byrd for this!). The difficulty of converging this job should be contrasted with the previous example where the bonds were less stretched. In this case, the VQCCD method yields far better results than VOD!

5.8 REFERENCES AND FURTHER READING

Self-consistent field methods (Chapter 4)

Excited state calculations (Chapter 6)

Basis sets (Chapter 7)

Effective core potentials (Chapter 8)

- [1] For a tutorial introduction to electron correlation methods based on wavefunctions, see, R. J. Bartlett and J. F. Stanton, in *Reviews in Computational Chemistry*, Volume V, edited by K. B. Lipkowitz and D. B. Boyd (VCH, New York, 1994), p 65 (Chapter 2).
- [2] For a general textbook introduction to electron correlation methods, and their strengths and weaknesses, see for example: F. Jensen, *Introduction to Computational Chemistry* (Wiley, New York, 1999).
- [3] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, *J. Chem. Phys.* (1991) **94**, 722.
- [4] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople, *J. Chem. Phys.* (1998) **109**, 7764.
- [5] C. Møller and M. S. Plesset, *Phys. Rev.* (1934), **46**, 618
- [6] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York 1986).
- [7] T. Helgaker, J. Gauss, P. Jorgensen, and J. Olsen, *J. Chem. Phys.* (1997) **106**, 6430.
- [8] M. Head-Gordon, *Mol. Phys.* (1999) **96**, 673.
- [9] M. Head-Gordon, J. A. Pople and M. J. Frisch, *Chem. Phys. Lett.* (1988), **153**, 503.
- [10] M. J. Frisch, M. Head-Gordon, and J. A. Pople, *Chem. Phys. Lett.* (1990), **166**, 275; **166**, 281.
- [11] M. S. Lee, P. E. Maslen, and M. Head-Gordon, *J. Chem. Phys.* (2000) **112**, 3592
- [12] M. Head-Gordon, M. S. Lee and P. E. Maslen, in *Simulation and Theory of Electrostatic Interactions in Solution*, edited by L.R.Pratt and G.Hummer, (AIP Conference Proceedings 492; AIP, New York 1999), p.301.
- [13] M. S. Lee and M. Head-Gordon, *Int. J. Quantum Chem.* (2000) **76**, 169.
- [14] S. Saebo and P.Pulay, *Ann. Rev. Phys. Chem.* (1993) **44**, 213.
- [15] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, *J. Chem. Phys.* (2000) **112**, 7374
- [16]. M. S. Lee, Ph. D. Thesis, University of California, Berkeley, 2000.
- [17] M. S. Lee and M. Head-Gordon (in preparation).
- [18] G. D. Purvis and R. J. Bartlett, *J. Chem. Phys.* (1982) **76**, 1910.
- [19] J. A. Pople, M. Head-Gordon and K. Raghavachari, *J. Chem. Phys.* (1987) **87**, 5968.
- [20] C. D. Sherrill, A. I. Krylov, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.* (1998) **109**, 4171.
- [21] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.* (2000) **113**, 8873.

- [22] T. Van Voorhis and M. Head-Gordon, *Chem. Phys. Lett.* (2000) **330**, 585.
- [23] E. F. C. Byrd, T. Van Voorhis and M. Head-Gordon (in preparation).
- [24] K. Raghavachari, G. W. Trucks, J. A. Pople and M. Head-Gordon, *Chem. Phys. Lett.* (1989) **157**, 479.
- [25] See for example, T.J.Lee and G.E.Scuseria, in *Quantum Mechanical Calculations with Chemical Accuracy*, edited by S.R.Langhoff (Kluwer, Dordrecht, 1995) p 47.
- [26] S. R. Gwaltney and M. Head-Gordon, *Chem. Phys. Lett.* (2000) **323**, 21.
- [27] S. R. Gwaltney and M. Head-Gordon (in preparation).
- [28] S. A. Kucharski and R. J. Bartlett, *J. Chem. Phys.* (1998) **108**, 5243; (1998) **108**, 9221.
- [29] For an introduction to the CASSCF method, see, B. O. Roos, *Adv. Chem. Phys.* (1987) **69**, 399.
- [30] See for example, K. Ruedenberg, M. W. Schmidt, M. M. Gilbert, and S. T. Elbert, *Chem. Phys.* (1982) **71**, 49; (1982) **71**, 51.
- [31] A. I. Krylov, C. D. Sherrill, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.* (1998) **109**, 10669.

CHAPTER 6 EXCITED STATE METHODS

6.1 GENERAL EXCITED STATE FEATURES

As for ground state calculations, performing an adequate excited state calculation involves making an appropriate choice of method and basis set. The development of effective approaches to modeling electronic excited states has historically lagged behind advances in treating the ground state. In part this is because of the much greater diversity in the character of the wave functions for excited states, making it more difficult to develop broadly applicable methods without molecule-specific or even state-specific specification of the form of the wave function. Recently, however, a hierarchy of single-reference *ab initio* methods has begun to emerge for the treatment of excited states. Broadly speaking, Q-Chem contains methods that are capable of giving qualitative agreement, and in many cases quantitative agreement with experiment for lower optically allowed states. The situation is less satisfactory for states that involve bi-electronic excitations, although even here reasonable results can sometimes be obtained.

In excited state calculations, as for ground state calculations, the user must strike a compromise between cost and accuracy. The first three main sections of this chapter summarize Q-Chem's capabilities in 3 general classes of excited state methods:

- Single-electron wavefunction-based methods (Section 6.2). These are excited state treatments of roughly the same level of sophistication as the Hartree-Fock ground state method, in the sense that electron correlation is essentially ignored. Single excitation configuration interaction (CIS) is the workhorse method of this type.
- Time-dependent density functional theory (TDDFT) (Section 6.3). TDDFT is the most useful extension of density functional theory to excited states that has been developed so far. For a cost that is little greater than the simple wavefunction methods such as CIS, a significantly more accurate method results.
- Wavefunction-based electron correlation treatments (Section 6.4) Roughly speaking, these are excited state analogs of the ground state wavefunction-based electron correlation methods discussed in Chapter 5. They are more accurate than the methods of Section 6.2, but also dramatically more computationally expensive.

In general, a basis set appropriate for a ground state density functional theory or Hartree-Fock calculations will be appropriate for describing valence excited states. However, many excited states involve significant contributions from very diffuse Rydberg orbitals, and, therefore, it is often advisable to use basis sets that include additional diffuse functions. The 6-31+G* basis set is a reasonable compromise for the low-lying valence

excited states of many organic molecules, but to describe true Rydberg excited states, Q-Chem allows the user to add second and higher sets of diffuse functions (see Chapter 7), as in basis sets of the 6-311(2+)G* type, that are generally adequate for description of both valence and Rydberg excited states.

Q-Chem supports three main types of excited state calculation:

- **Vertical absorption spectrum**

This is the calculation of the excited states of the molecule at the ground state geometry, as appropriate for absorption spectroscopy. The methods supported for performing a vertical absorption calculation are: CIS, RPA, XCIS and CIS(D), each of which will be discussed in turn. In addition, it is possible to visualise the excited states either by attachment-detachment density analysis or by plotting the transition density (see *\$plots* descriptions in chapters 3 and 10). The visual analysis options are available only for the CIS method, and the theoretical basis of this method is discussed in Section 6.4 of this chapter.

- **Excited state optimization**

Optimization of the geometry of stationary points on excited state potential energy surfaces is valuable for understanding the geometric relaxation that occurs between the ground and excited state. Excited state optimization is currently available for UCIS and RCIS only.

- **Excited state vibrational analysis**

Given an optimised excited state geometry, Q-Chem can calculate the force constants at the stationary point to predict excited state vibrational frequencies. Stationary points can also be characterized as minima, transition structures or n^{th} -order saddle points. Excited state vibrational analyses can only be performed using the UCIS and RCIS methods, for which efficient analytical second derivatives are available.

6.2 NON-CORRELATED WAVEFUNCTION METHODS

Q-Chem includes several excited state methods which do not incorporate correlation: CIS, XCIS and RPA. These methods are sufficiently inexpensive that calculations on large molecules are possible, and are roughly comparable to the HF treatment of the ground state in terms of performance. They tend to yield qualitative rather than quantitative insight. Excitation energies tend to exhibit errors on the order of an electron volt, consistent with the neglect of electron correlation effects, which are generally different in the ground state and the excited state.

6.2.1 SINGLE EXCITATION CONFIGURATION INTERACTION (CIS)

The derivation of the CI-singles [1,2] energy and wave function begins by selecting the HF single determinantal wave function as reference for the ground state of the system

$$\Psi_{HF} = \frac{1}{\sqrt{n!}} \det \{ \chi_1 \chi_2 \cdots \chi_i \chi_j \cdots \chi_n \} \quad (6.1)$$

where n is the number of electrons, and the spin orbitals

$$\chi_i = \sum_{\mu}^N c_{\mu i} \phi_{\mu} \quad (6.2)$$

are expanded in a finite basis of N atomic orbital basis functions. Molecular orbital coefficients $\{c_{\mu i}\}$ are usually found by SCF procedures which solve the Hartree-Fock equations

$$\mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC} \quad (6.3)$$

where \mathbf{S} is the overlap matrix, \mathbf{C} is the matrix of molecular orbital coefficients, $\boldsymbol{\varepsilon}$ is a diagonal matrix of orbital eigenvalues and \mathbf{F} is the Fock matrix with elements

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda\sigma} \sum_i c_{\mu i} c_{\nu i} (\mu\lambda || \nu\sigma) \quad (6.4)$$

involving the core Hamiltonian and the antisymmetrized two-electron integrals

$$(\mu\nu || \lambda\sigma) = \iint \phi_{\mu}(\mathbf{r}_1) \phi_{\nu}(\mathbf{r}_2) (1/r_{12}) [\phi_{\lambda}(\mathbf{r}_1) \phi_{\sigma}(\mathbf{r}_2) - \phi_{\lambda}(\mathbf{r}_2) \phi_{\sigma}(\mathbf{r}_1)] d\mathbf{r}_1 d\mathbf{r}_2 \quad (6.5)$$

On solving (6.3), the total energy of the ground state single determinant can be expressed as

$$E_{HF} = \sum_{\mu\nu} P_{\mu\nu}^{HF} H_{\mu\nu} + \frac{1}{2} \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}^{HF} P_{\lambda\sigma}^{HF} (\mu\lambda || \nu\sigma) + V_{nuc} \quad (6.6)$$

where P^{HF} is the HF density matrix and V_{nuc} is the nuclear repulsion energy.

Equation (6.1) represents only one of many possible determinants made from orbitals of the system; there are in fact $n(N - n)$ possible singly substituted determinants constructed by replacing an orbital occupied in the ground state (i, j, k, \dots) with an orbital unoccupied in the ground state (a, b, c, \dots). Such wave functions and energies can be written

$$\Psi_i^a = \frac{1}{\sqrt{n!}} \det \{ \chi_1 \chi_2 \cdots \chi_a \chi_j \cdots \chi_n \} \quad (6.7)$$

$$E_{ia} = E_{HF} + \varepsilon_a - \varepsilon_i - (ia || ia) \quad (6.8)$$

where we have introduced the antisymmetrized two-electron integrals in the molecular orbital basis

$$(pq || rs) = \sum_{\mu\nu\lambda\sigma} c_{\mu p} c_{\nu q} c_{\lambda r} c_{\sigma s} (\mu\lambda || \nu\sigma) \quad (6.9)$$

These singly excited wave functions and energies could be considered crude approximations to the excited states of the system. However, determinants of the form (6.7) are deficient in that they:

- (1) do not yield pure spin states
- (2) resemble more closely ionization rather than excitation
- (3) are not appropriate for excitation into degenerate states

These deficiencies can be partially overcome by representing the excited state wavefunction as a linear combination of *all* possible singly excited determinants

$$\Psi_{CIS} = \sum_{ia} a_i^a \Psi_i^a \quad (6.10)$$

where the coefficients $\{a_{ia}\}$ can be obtained by diagonalizing the many-electron Hamiltonian, \mathbf{A} , in the space of all single substitutions, where the matrix elements are

$$\begin{aligned} A_{ia,jb} &= \langle \Psi_i^a | H | \Psi_j^b \rangle \\ &= [E_{HF} + \varepsilon_a - \varepsilon_j] \delta_{ij} \delta_{ab} - (ja || ib) \end{aligned} \quad (6.11)$$

By Brillouin's theorem single substitutions do not interact directly with a reference HF determinant, so the resulting eigenvectors from the CIS excited state represent a treatment roughly comparable to that of the HF ground state. The excitation energy is simply the difference between HF ground state energy and CIS excited state energies, and the eigenvectors of \mathbf{A} correspond to the amplitudes of the single-electron promotions.

CIS calculations can be performed in Q-Chem using restricted (RCIS) [1,2], unrestricted (UCIS), or restricted open shell (ROCIS) [3] spin orbitals.

6.2.2 RANDOM PHASE APPROXIMATION (RPA)

The Random Phase Approximation (RPA) [4,5] is an alternative to CIS for uncorrelated calculations of excited states. It offers some advantages for computing oscillator strengths, and is roughly comparable in accuracy to CIS for excitation energies to singlet states, but is inferior for triplet states. RPA energies are non-variational.

6.2.3 EXTENDED CIS (XCIS)

The motivation for the extended CIS procedure (XCIS) [6] stems from the fact that ROCIS and UCIS are less effective for radicals than CIS is for closed shell molecules. Using the attachment-detachment density analysis procedure [7], the failing of ROCIS and UCIS methodologies for the nitromethyl radical was traced to the neglect of a

particular class of double substitution which involves the simultaneous promotion of an α spin electron from the singly occupied orbital and the promotion of a β spin electron into the singly occupied orbital. In particular, the spin adapted configurations

$$|\tilde{\Psi}_i^a(1)\rangle = \frac{1}{\sqrt{6}}(\Psi_{\bar{i}}^{\bar{a}} - \Psi_i^a) + \frac{2}{\sqrt{6}}\Psi_{p\bar{i}}^{a\bar{p}} \quad (6.12)$$

(where $a, b, c \dots$ are virtual orbitals, $i, j, k \dots$ are occupied orbitals and, $p, q, r \dots$ are singly occupied orbitals) are of crucial importance and, it is quite likely that similar excitations are also very significant in other radicals of interest.

The XCIS proposal, a more satisfactory generalization of CIS to open shell molecules, is to simultaneously include a restricted class of double substitutions similar to those in (6.12). To illustrate this, consider the resulting orbital spaces of an ROHF calculation: doubly occupied (d), singly occupied (s) and virtual (v). From this starting point we can distinguish three types of single excitations of the same multiplicity as the ground state: $d \rightarrow s$, $s \rightarrow v$ and $d \rightarrow v$. Thus, the spin adapted ROCIS wave function is

$$|\Psi_{\text{ROCIS}}\rangle = \frac{1}{\sqrt{2}} \sum_{ia}^{dv} a_i^a (\Psi_i^a + \Psi_{\bar{i}}^{\bar{a}}) + \sum_{pa}^{sv} a_p^a \Psi_p^a + \sum_{ip}^{ds} a_i^{\bar{p}} \Psi_{\bar{i}}^{\bar{p}} \quad (6.13)$$

The extension of CIS theory to incorporate higher excitations maintains the ROHF as the ground state reference and adds terms to the ROCIS wave function similar to that of equation (6.13), as well as those where the double excitation occurs through different orbitals in the α and β space

$$|\Psi_{\text{XCIS}}\rangle = \frac{1}{\sqrt{2}} \sum_{ia}^{dv} a_i^a (\Psi_i^a + \Psi_{\bar{i}}^{\bar{a}}) + \sum_{pa}^{sv} a_p^a \Psi_p^a + \sum_{ip}^{ds} a_i^{\bar{p}} \Psi_{\bar{i}}^{\bar{p}} + \sum_{iap}^{dvs} \tilde{a}_i^a(p) \tilde{\Psi}_i^a(p) + \sum_{ia, p \neq q}^{dv, ss} a_{p\bar{i}}^{a\bar{q}} \Psi_{p\bar{i}}^{a\bar{q}} \quad (6.14)$$

XCIS is defined only from a restricted open shell Hartree-Fock ground state reference, as it would be difficult to uniquely define singly occupied orbitals in a UHF wave function. In addition, β unoccupied orbitals, through which the spin-flip double excitation proceeds, may not match the half-occupied α orbitals in either character or even symmetry.

For molecules with closed shell ground states, both the HF ground and CIS excited states emerge from diagonalization of the Hamiltonian in the space of the HF reference and singly excited substituted configuration state functions. The XCIS case is different because the restricted class of double excitations included could mix with the ground state and lower its energy. This mixing is avoided to maintain the size consistency of the ground state energy.

With the inclusion of the restricted set of doubles excitations in the excited states, but not in the ground state, it could be expected that some fraction of the correlation energy be recovered, resulting in anomalously low excited state energies. However, the fraction of the total number of doubles excitations included in the XCIS wave function is very small and those introduced cannot account for the pair correlation of any pair of electrons. Thus, the XCIS procedure can be considered one that neglects electron correlation.

The computational cost of XCIS is approximately four times greater than CIS and ROCIS, and its accuracy for open shell molecules is generally comparable to that of the CIS method for closed shell molecules. In general, it achieves qualitative agreement with experiment. XCIS is available for doublet and quartet excited states beginning from a doublet ROHF treatment of the ground state, for excitation energies only.

6.2.4 BASIC JOB CONTROL OPTIONS

See also *JOBTYPE*, *BASIS*, *EXCHANGE* and *CORRELATION*. *EXCHANGE* must be *HF* and *CORRELATION* must be *NONE*. The minimum input required above a ground state HF calculation is to specify a nonzero value for *CIS_N_ROOTS*.

CIS_N_ROOTS

Sets the number of CI-Singles (CIS) excited state roots to find

VARIABLE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

$n > 0$ Looks for n CIS excited states

CIS_SINGLET

Solve for singlet excited states in RCIS calculations (ignored for UCIS)

VARIABLE:

LOGICAL

DEFAULT:

True Solve for singlet states

OPTIONS:

True Solve for singlet states

False Do not solve for singlet states

CIS_TRIPLETS

Solve for triplet excited states in RCIS calculations (ignored for UCIS)

VARIABLE:

LOGICAL

DEFAULT:

True Solve for triplet states

OPTIONS:

True Solve for triplet states

False Do not solve for triplet states

RPA

Do an RPA calculation in addition to a CIS calculation

VARIABLE:

LOGICAL

DEFAULT:

False Do not do an RPA calculation

OPTIONS:

False Do not do an RPA calculation

True Do an RPA calculation

XCIS

Do an XCIS calculation in addition to a CIS calculation

VARIABLE:

LOGICAL

DEFAULT:

False Do not do an XCIS calculation

OPTIONS:

False Do not do an XCIS calculation

True Do an XCIS calculation (requires ROHF ground state)

6.2.5 CUSTOMIZATION

N_FROZEN_CORE

Controls the number of frozen core orbitals

VARIABLE:

INTEGER

DEFAULT:

0 No frozen core orbitals

OPTIONS:

FC Frozen core approximation

n Freeze n core orbitals

RECOMMENDATIONS:

There is no computational advantage to using frozen core for CIS, and analytical derivatives are only available when no orbitals are frozen. It is helpful when calculating CIS(D) corrections (see Sec. 6.4).

N_FROZEN_VIRTUAL

Controls the number of frozen virtual orbitals.

VARIABLE:

INTEGER

DEFAULT:

0 No frozen virtual orbitals

OPTIONS:

n Freeze n virtual orbitals

RECOMMENDATIONS:

There is no computational advantage to using frozen virtuals for CIS, and analytical derivatives are only available when no orbitals are frozen.

MAX_CIS_CYCLES

Maximum number of CIS iterative cycles allowed

VARIABLE:

INTEGER

DEFAULT:

30

OPTIONS:

User-defined

RECOMMENDATIONS:

Default is usually sufficient

CIS_CONVERGENCE

CIS is considered converged when error is less than $10^{-\text{CIS_CONVERGENCE}}$

VARIABLE:

INTEGER

DEFAULT:

6 CIS convergence threshold 10^{-6}

OPTIONS:

User-defined

CIS_RELAXED_DENSITY

Use the relaxed CIS density for attachment/detachment density analysis

VARIABLE:

LOGICAL

DEFAULT:

False Do not use the relaxed CIS density in analysis

OPTIONS:

False Do not use the relaxed CIS density in analysis

True Use the relaxed CIS density in analysis

CIS_GUESS_DISK

Read the CIS guess from disk (previous calculation)

VARIABLE:

LOGICAL

DEFAULT:

False Create a new guess

OPTIONS:

False Create a new guess

True Read the guess from disk

RECOMMENDATIONS:

Requires a guess from previous calculation

CIS_GUESS_DISK_TYPE

Determines the type of guesses to be read from disk

VARIABLE:

INTEGER

DEFAULT:

Nil

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATIONS:

Must be specified if *CIS_GUESS_DISK* is *TRUE*

6.2.6 CIS ANALYTICAL DERIVATIVES

While CIS excitation energies are relatively inaccurate, with errors of the order of 1eV, CIS excited state properties, such as structures and frequencies, are much more useful. This is very similar to the manner in which ground state Hartree-Fock (HF) structures and frequencies are much more accurate than HF relative energies. Generally speaking, for low-lying excited states, it is expected that CIS vibrational frequencies will be systematically 10% higher or so relative to experiment [8,9,10]. If the excited states are of pure valence character, then basis set requirements are generally similar to the ground state. Excited states with partial Rydberg character require the addition of one or preferably two sets of diffuse functions.

Q-Chem includes efficient analytical first and second derivatives of the CIS energy [11,12], to yield analytical gradients, excited state vibrational frequencies, force constants, polarizabilities, and infrared intensities. Their evaluation is controlled by two REM's, listed below. Analytical gradients can be evaluated for any job where the CIS excitation energy calculation itself is feasible.

JOBTYPE

Specifies the type of calculation

VARIABLE:

STRING

DEFAULT:

SP Single point energy

OPTIONS:

SP Single point energy

FORCE Analytical Force calculation

OPT Geometry Minimization

TS Transition Structure Search

FREQ Frequency Calculation

RECOMMENDATION:

Defaults to single point

CIS_STATE_DERIV

Sets CIS state for excited state optimizations and vibrational analysis

VARIABLE:

INTEGER

DEFAULT:

0 Does not select any of the excited states

OPTIONS:

n Select the n^{th} state

The semi-direct method [11] used to evaluate the frequencies is generally similar to the semi-direct method used to evaluate Hartree-Fock frequencies for the ground state. Memory and disk requirements (see below) are similar, and the computer time scales approximately as the cube of the system size for large molecules.

The main complication associated with running analytical CIS second derivatives is ensuring Q-Chem has sufficient memory to perform the calculations. For most purposes, the defaults will be adequate, but if a large calculation fails due to a memory error, then the following additional information may be useful in fine tuning the input, and understanding why the job failed. Note that the analytical CIS second derivative code does not currently support frozen core or virtual orbitals (unlike Q-Chem's MP2 code). Unlike MP2 calculations, applying frozen core/virtual orbital approximations does not lead to large computational savings in CIS calculations as all computationally expensive steps are performed in the atomic basis.

The memory requirements for CIS (and HF) analytical frequencies are primarily extracted from "C" memory, which is defined as

$$\text{"C" memory} = MEMORY_TOTAL - MEMORY \quad (6.15)$$

"C" memory must be large enough to contain a number of arrays whose size is $3*NAtoms*NBasis*NBasis$ ($NAtoms$ is the number of atoms and $NBasis$ refers to the number of basis functions). The value of the *\$rem* variable $MEMORY$ should be set sufficiently large to permit efficient integral evaluation. If too large, it reduces the amount of "C" memory available. If too small, the job may fail due to insufficient scratch space. For most purposes, a value of about 5MW is sufficient and by default $MEMORY_TOTAL$ is set to a very large number (large than physical memory on most computers) and thus *malloc* (memory allocation) errors may occur on jobs where the memory demands exceeds physical memory.

6.2.7 EXAMPLES

```

$molecule
  0 1
  C
  O,1,CO
  H,1,CH,2,A
  H,1,CH,2,A,3,D

  CO = 1.2
  CH = 1.0
  A  = 120.0
  D  = 180.0
$end

$rem
  jobtype           =  opt
  exchange          =  HF
  basis             =  6-31G*
$end

@@@
$molecule
  READ
$end

$rem
  exchange          =  HF
  basis             =  6-311(2+)G*
  cis_n_roots       =  15           Do 15 states
  cis_singlets      =  true         Do   do singlets
  cis_triplets      =  false        Don't do Triplets
$end

```

Example 6.1 A basic CIS excitation energy calculation on formaldehyde at the HF/6-31G* optimized ground state geometry, which is obtained in the first part of the job. Above the first singlet excited state, the states have Rydberg character, and therefore a basis with two sets of diffuse functions is used.

```

$comment
  C6H5 phenyl radical C2v symmetry mp2(full)/6-31g* = -
230.7777459
$end

$molecule
  0 2
  c1
  x1 c1 1.0
  c2 c1 rc2 x1 90.0
  x2 c2 1.0 c1 90.0 x1 0.0
  c3 c1 rc3 x1 90.0 c2 tc3
  c4 c1 rc3 x1 90.0 c2 -tc3
  c5 c3 rc5 c1 ac5 x1 -90.0
  c6 c4 rc5 c1 ac5 x1 90.0
  h1 c2 rh1 x2 90.0 c1 180.0
  h2 c3 rh2 c1 ah2 x1 90.0
  h3 c4 rh2 c1 ah2 x1 -90.0
  h4 c5 rh4 c3 ah4 c1 180.0
  h5 c6 rh4 c4 ah4 c1 180.0
  rc2=2.67298593
  rc3=1.35449831
  tc3=62.85150452
  rc5=1.37290399
  ac5=116.45436983
  rh1=1.08573521
  rh2=1.08534214
  ah2=122.157328
  rh4=1.08721616
  ah4=119.52349629
$end

$rem
  basis           = 6-31+G*
  exchange        = hf
  memory          = 5000000
  intsbuffersize  = 15000000
  scf_convergence = 8
  cis_n_roots     = 5
  xcis            = true
$end

```

Example 6.2 An XCIS calculation of excited states of an unsaturated radical, the phenyl radical, for which double substitutions make considerable contributions to low-lying excited states.

```

$comment
singlet n --> pi* state optimization and frequencies for
formaldehyde
$end

$molecule
  0 1
  C
  O,1,CO
  H,1,CH,2,A
  H,1,CH,2,A,3,D

  CO = 1.2
  CH = 1.0
  A  = 120.0
  D  = 150.0
$end

$rem
  jobtype           =  opt
  exchange           =  HF
  basis              =  6-31+G*
  cis_state_deriv   =  1           Optimize state 1
  cis_n_roots       =  3           Do 3 states
  cis_singlets      =  true        Do   do singlets
  cis_triplets     =  false       Don't do Triplets
$end

@@@
$molecule
  READ
$end

$rem
  jobtype           =  freq
  exchange           =  HF
  basis              =  6-31+G*
  cis_state_deriv   =  1           Focus on state 1
  cis_n_roots       =  3           Do 3 states
  cis_singlets      =  true        Do   do singlets
  cis_triplets     =  false       Don't do Triplets
$end

```

Example 6.3 This example illustrates a CIS geometry optimization followed by a vibrational frequency analysis on the lowest singlet excited state of formaldehyde. This $n \rightarrow \pi^*$ excited state is non-planar, unlike the ground state. The optimization converges to a nonplanar structure with zero forces, and all frequencies real.

6.3 TIME-DEPENDENT DENSITY FUNCTIONAL THEORY (TDDFT)

6.3.1 A BRIEF INTRODUCTION TO TDDFT

Excited states may be obtained from density functional theory by time-dependent density functional theory [13,14], which calculates poles in the response of the ground state density to a time-varying applied electric field. These poles are Bohr frequencies or excitation energies, and are available in Q-Chem [15], together with the CIS-like Tamm-Dancoff approximation [16]. TDDFT is becoming very popular as a method for studying excited states because the computational cost is roughly similar to the simple CIS method (scaling as roughly the square of molecular size), but a description of differential electron correlation effects is implicit in the method. The excitation energies for low-lying valence excited states of molecules (below the ionization threshold, or more conservatively, below the first Rydberg threshold) are often remarkably improved relative to CIS, with an accuracy of roughly 0.3 eV being observed with either gradient corrected or local density functionals.

However, standard density functionals do not yield a potential with the correct long-range Coulomb tail (due to the so-called self-interaction problem), and therefore excited states which sample this tail (for example diffuse Rydberg states, and some charge transfer excited states) are not given accurately [17,18]. Hence it is advisable to only employ TDDFT for low-lying valence excited states that are below the first ionization potential of the molecule. This makes radical cations a particularly favorable choice of system, as exploited in ref. [19]. TDDFT for low-lying valence excited states of radicals is in general a remarkable improvement relative to CIS, including some states, that, when treated by wave function-based methods can involve a significant fraction of double excitation character [15].

6.3.2 JOB CONTROL FOR TDDFT

Input for time-dependent density functional theory calculations follows very closely the input already described for the uncorrelated excited state methods described in the previous section (in particular, see Section 6.2.4). There are two points to be aware of:

- The exchange and correlation functionals are specified exactly as for a ground state DFT calculation, through EXCHANGE and CORRELATION.
- If RPA is set to true, a full TDDFT calculation will be performed. This is not the default. The default is to RPA = false, which leads to a calculation employing the Tamm-Dancoff approximation (TDA), which is usually a good approximation to full TDDFT.

TDDFT and TDDFT/TDA are both available only for excitation energies at present.

6.3.3 EXAMPLES

```

$comment
methyl peroxy radical
TDDFT/TDA and full TDDFT with 6-31+G*
$end

$molecule
0 2
C      1.004123   -0.180454   0.000000
O     -0.246002    0.596152   0.000000
O     -1.312366   -0.230256   0.000000
H      1.810765    0.567203   0.000000
H      1.036648   -0.805445  -0.904798
H      1.036648   -0.805445   0.904798
$end

$rem
exchange      B
correlation   LYP
cis_n_roots   5
basis         6-31+G*
scf_convergence 7
$end

@@@
$molecule
READ
$end

$rem
exchange      B
correlation   LYP
cis_n_roots   5
rpa           true
basis         6-31+G*
scf_convergence 7
$end

```

Example 6.4 This example shows 2 jobs which request variants of time-dependent density functional theory calculations. The first job, using the default value of $RPA = \text{false}$, performs TDDFT in the Tamm-Dancoff approximation (TDA). The second job, with $RPA = \text{true}$ performs a both TDA and full TDDFT calculations.

6.4 EXCITED STATE CORRELATED METHODS

6.4.1 CIS(D)

CIS(D) [20,21] is a simple size-consistent doubles correction to CIS which has a computational cost which scales as the fifth power of the basis set for each excited state. In this sense, CIS(D) can be considered an excited state analog of the ground state MP2 method. CIS(D) yields useful improvements in the accuracy of excitation energies relative to CIS, and yet can still be applied to relatively large molecules using Q-Chem's efficient integrals transformation package.

The CIS(D) excited state procedure is a second-order perturbative approximation to the computationally expensive CCSD, based on a single excitation configuration interaction (CIS) reference. The coupled cluster wavefunction, truncated at single and double excitations, is the exponential of the single and double substitution operators acting on the Hartree-Fock determinant

$$|\Psi\rangle = \exp(T_1 + T_2)|\Psi_0\rangle \quad (6.16)$$

Determining the singles and doubles amplitudes requires solving the two equations

$$\langle \Psi_i^a | H - E | (1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3) \Psi_0 \rangle = 0 \quad (6.17)$$

$$\langle \Psi_{ij}^{ab} | H - E | (1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3 + \frac{1}{2}T_2^2 + \frac{1}{2}T_1^2T_2 + \frac{1}{4!}T_1^4) \Psi_0 \rangle = 0 \quad (6.18)$$

which lead to the CCSD excited state equations, which can be written

$$\langle \Psi_i^a | H - E | (U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \frac{1}{2}T_1^2U_1) \Psi_0 \rangle = \omega b_i^a \quad (6.19)$$

$$\begin{aligned} \langle \Psi_i^a | H - E | (U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \frac{1}{2}T_1^2U_1 + T_2U_2 \\ + \frac{1}{2}T_1^2U_2 + T_1T_2U_1 + \frac{1}{3!}T_1^3U_1) \Psi_0 \rangle = \omega b_{ij}^{ab} \end{aligned} \quad (6.20)$$

This is an eigenvalue equation $\mathbf{A}\mathbf{b} = \omega\mathbf{b}$ for the transition amplitudes (\mathbf{b} vectors), which are also contained in the U operators.

The second-order approximation to the CCSD eigenvalue equation yields a second-order contribution to the excitation energy which can be written in the form

$$\omega^{(2)} = \mathbf{b}^{(0)\dagger} \mathbf{A}^{(1)} \mathbf{b}^{(1)} + \mathbf{b}^{(0)\dagger} \mathbf{A}^{(2)} \mathbf{b}^{(0)} \quad (6.21)$$

or in the alternative form

$$\begin{aligned}\omega^{(2)} &= \omega^{CIS(D)} \\ &= E^{CIS(D)} - E^{MP2}\end{aligned}\quad (6.22)$$

where

$$E^{CIS(D)} = \langle \Psi^{CIS} | V | U_2 \Psi^{HF} \rangle + \langle \Psi^{CIS} | V | T_2 U_1 \Psi^{HF} \rangle \quad (6.23)$$

and

$$E^{MP2} = \langle \Psi^{HF} | V | T_2 \Psi^{HF} \rangle \quad (6.24)$$

The output of a CIS(D) calculation contains useful information beyond the CIS(D) corrected excitation energies themselves. The stability of the CIS(D) energies is tested by evaluating a diagnostic, termed the “theta diagnostic” [22]. The theta diagnostic calculates a mixing angle that measures the extent to which electron correlation causes each pair of calculated CIS states to couple. Clearly the most extreme case would be a mixing angle of 45°, which would indicate breakdown of the validity of the initial CIS states and any subsequent corrections. On the other hand small mixing angles on the order of only a degree or so are an indication that the calculated results are reliable. The code reports the largest mixing angle for each state to all others that have been calculated.

6.4.2 CIS(D) JOB CONTROL

The algorithms used to evaluate the CIS(D) energy have much in common with Q-Chem’s semidirect MP2 methods, discussed earlier in Section 5.3. As such, there are again 3 nondefaulted options that the user should understand in order to run CIS(D) calculations.

- (1) *MEMORY*: The value specified for this REM variable must be sufficient to permit efficient integral evaluation (2-10MW) and to hold a large temporary array whose size is $2OVN$, the product of the number of occupied, virtual and total numbers of orbitals.
- (2) *CD_DISK*: The value specified for this REM variable should be as large as possible (i.e. perhaps 80% of the free space on your $\$QCSCRATCH$ partition where temporary job files are held). The value of this variable will determine how many times the two-electron integrals in the atomic orbital basis must be re-evaluated, which is a major computational step in CIS(D) calculations.
- (3) *N_FROZEN_CORE*: The computational requirements for CIS(D) are proportional to the number of occupied orbitals for some steps, and the square of that number for other steps. Therefore the CPU time can be significantly reduced if your job employs the frozen core approximation. Additionally the memory and disk requirements are reduced when the frozen core approximation is employed.

CORRELATION

OPTION:

CIS(D) CIS(D) excited states

CIS_N_ROOTS

Sets the number of CI-Singles (CIS) excited state roots to find.

VARIABLE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

 $n > 0$ Look for n CIS excited states**MEMORY**

Sets the memory for individual program modules

VARIABLE:

INTEGER

DEFAULT:

2,000,000 (2 MW)

OPTIONS:

User-defined number of words. For direct and semidirect MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (2-10 MW), as discussed above.

MEMORY_TOTAL

Sets the total memory available to Q-Chem

VARIABLE:

INTEGER

DEFAULT:

Unlimited (1,000 MW)

OPTIONS:

User-defined number of words

RECOMMENDATION:

Use default, or set to the physical memory of your machine.

CD_MAX_DISK

Sets the amount of disk space (in words) available for MP2 calculations

VARIABLE:

INTEGER

DEFAULT:

60,000,000 (60 MW)

OPTIONS:

User-defined: should be set as large as possible, discussed in Sec. 5.3.1

CD_ALGORITHM

Determines the algorithm for MP2 integral transformations

VARIABLE:

STRING

DEFAULT:

Program determined

OPTIONS:

DIRECT Uses fully direct algorithm (energies only)

SEMI_DIRECT Uses disk-based semi-direct algorithm

RECOMMENDATION:

Semidirect is usually most efficient, and will normally be chosen by default.

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

FC Frozen Core approximation (all core orbitals frozen)

n Freeze *n* core orbitals

N_FROZEN_VIRTUAL

Sets the number of frozen virtual orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

n Freeze *n* virtual orbitals

6.4.3 COUPLED CLUSTER EXCITED STATE METHODS

It is possible to obtain a description of electronic excited states at a level of theory similar to that associated with coupled cluster theory for the ground state, by applying either linear response theory [23] or equations of motion methods [24]. A number of groups have demonstrated that excitation energies based on a coupled cluster singles and doubles ground state are generally very accurate for states that are primarily single electron promotions. The error observed in calculated excitation energies to such states is approximately 0.3 eV, including both valence and Rydberg excited states. This of course assumes that a basis set large and flexible enough to describe valence and Rydberg states is employed. The accuracy of excited state coupled cluster methods is much lower for excited states that involve a substantial component of double excitation character, where errors may be 1 eV or even more. Such errors arise because the description of electron

correlation is better in the ground state than for an excited state with substantial double excitation character.

Q-Chem includes coupled cluster methods for excited states based on the optimized orbital coupled cluster doubles (OD) method, described earlier. OD excitation energies are essentially identical in numerical performance to CCSD excited states, as has been recently demonstrated [25]. This method, while far more computationally expensive than TDDFT, is nevertheless useful as a proven high accuracy method for the study of excited states of small molecules. Also, when studying a series of related molecules it can be very useful to compare the performance of TDDFT and coupled cluster theory for at least a small example to understand its performance. Along similar lines, the CIS(D) method described earlier as an economical correlation energy correction to CIS excitation energies is in fact an approximation to coupled cluster excitation energies. It is useful to assess the performance of CIS(D) for a class of problems by benchmarking against the full coupled cluster treatment. Finally, Q-Chem also includes excited states by the equation of motion version of the valence optimized doubles (VOD) method (see Section 5.7 of the previous chapter), whose validity and use is fully discussed in ref. [25].

6.4.4 COUPLED CLUSTER EXCITED STATE JOB CONTROL

There are quite a rich range of input control options for coupled cluster excited state calculations. The minimal requirement is the input for the ground state OD or VOD calculation (see Chapter 5), plus specification of the number of excited states requested, through `CC_NLOWSPIN` and `CC_NHIGHSPIN`. The full range of input options that are directly relevant to coupled cluster excited states follows:

CC_DCONVERGENCE

Convergence criterion for the RMS residuals of excited state vectors [$10^{*(-n)}$].

VARIABLE:

INTEGER

DEFAULT:

5

OPTIONS:

n $10^{*(-n)}$ convergence criterion

RECOMMENDATIONS:

Should be consistent with `CC_DTHRESHOLD`.

CC_DO_DISCONNECTED

Are disconnected terms included in the excited state coupled cluster equations?

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATIONS:

Inclusion of disconnected terms has very small effects and is not necessary.

CC_DTHRESHOLD

Specifies threshold for including a new expansion vector in the iterative Davidson diagonalization. Their norm must be above this threshold.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{**}(-de)$

DEFAULT:

0.00001

RECOMMENDATIONS:

Should be consistent with *CC_DCONVERGENCE*.

CC_DMAXITER

Maximum number of iteration allowed for Davidson diagonalization procedure.

VARIABLE:

INTEGER

DEFAULT:

30

CC_HOW_DO_DOUBLES

Distinguishes different implementations of active space coupled cluster excited state (VOD) calculations.

VARIABLE:

INTEGER

DEFAULT:

2

OPTIONS:

- 0 Double excitations for excited states are defined in the whole orbital space. This model is unbalanced and should be used for testing purposes only.
- 1 Double excitations for the excited states are restricted to an active (valence) orbital space. This approximates linear response CASCCF, with no orbital relaxation.
- 2 Approximate treatment of orbital relaxation by including all internal and semi-internal excited state double excitations.

CC_NGUESS_DOUBLES

Specifies number of excited state guess vectors which are double excitations.

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

n Include n guess vectors that are double excitations

RECOMMENDATIONS:

This should be set to the expected number of doubly excited states (see also *CC_PRECONV_DOUBLES*), otherwise they may not be found.

CC_NGUESS_SINGLES

Specifies number of excited state guess vectors that are single excitations.

VARIABLE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

n Include n guess vectors that are single excitations

RECOMMENDATIONS:

Should be greater or equal than the number of excited states requested.

CC_NHIGHSPIN

Sets the number of coupled cluster high-spin excited state roots to find. Works only for singlet ground state and triplet excited states. The program will increase this number if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations

VARIABLE:

INTEGER

DEFAULT:

0

Do not look for any coupled cluster excited states

OPTIONS:

$n > 0$

Find n CC excited states (after an OD ground state job)

CC_NLOWSPIN

Sets the number of coupled cluster excited state roots to find with the same multiplicity as the ground state. For a spin-unrestricted ground state, (e.g., doublet radicals), this is the total number of states of all multiplicities. The program will increase this number if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations.

VARIABLE:

INTEGER

DEFAULT:

0 Do not look for any coupled cluster excited states

OPTIONS:

$n > 0$ Find n CC excited states (after an OD ground state job)

CC_NVEC_PER_ROOT

Specifies maximum number of vectors per root in Davidson' diagonalization.

VARIABLE:

INTEGER

DEFAULT:

6

OPTIONS:

n Up to n vectors per root before the subspace is reset

RECOMMENDATIONS:

Larger values increase storage but speed convergence.

CC_PRECONV_DOUBLES

When TRUE, doubly-excited vectors are converged prior to a full excited states calculation.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATIONS:

Occasionally necessary to ensure a doubly excited state is found.

CC_PRECONV_SINGLES

When TRUE, singly-excited vectors are converged prior to a full excited states calculation.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

6.4.5 EXAMPLES

```
$comment
  CIS(D) excitation energies for formaldehyde
$end

$molecule
  0 1
  C      .000000      .000000      .519517
  O      .000000      .000000     -.664800
  H      .924162      .000000      1.100647
  H     -.924162      .000000      1.100647
$end

$rem
  correlation      =  CIS(D)
  exchange         =  HF
  basis            =  6-311(2+)G*
  cis_n_roots     =  5           5 states
  memory          =  12500000    100 MB of memory
  cd_disk         =  250000000    2 GB of disk
$end
```

Example 6.5 Evaluating the 5 lowest singlet and triplet excited states of formaldehyde using CIS(D), which treats excited states at a level of theory that is roughly similar to the MP2 method for the ground state.

```

$comment
  OD for ground + lowest 5 excited states of formaldehyde
$end

$molecule
  0 1
  C      .000000      .000000      .519517
  O      .000000      .000000     -.664800
  H      .924162      .000000      1.100647
  H     -.924162      .000000      1.100647
$end

$rem
  correlation      =  OD
  exchange         =  HF
  basis            =  6-311(2+)G*
  cc_nlowspin     =  5          5 states
  memory          =  12500000  100 MB of memory
$end

```

Example 6.6 Evaluating the 5 lowest singlet excited states of formaldehyde using OD. This calculation is much more expensive than the CIS(D) example shown above, but also yields more accurate results.

6.5 ATTACHMENT/DETACHMENT DENSITY ANALYSIS

As methods for *ab initio* calculations of excited states are becoming increasingly more routine, the question is how best to extract chemical meaning from such calculations. Recently, a new method of analyzing molecular excited states has been proposed [7] which has proven successful in applications reported so far [7,26,27]. This section describes the theoretical background to this attachment-detachment density analysis, while details of the input for creating data suitable for plotting these quantities is described separately in the Molecular Properties Chapter.

Consider the one-particle density matrices of the initial and final states of interest, \mathbf{P}_1 and \mathbf{P}_2 respectively. Assuming that each state is represented in a finite basis of spin-orbitals, such as the molecular orbital basis, and each state is at the same geometry. Subtracting these matrices yields the difference density

$$\Delta = \mathbf{P}_1 - \mathbf{P}_2 \quad (6.25)$$

Now, the eigenvectors of the one-particle density matrix \mathbf{P} describing a single state are termed the natural orbitals, and provide the best orbital description that is possible for the state, in that a CI expansion using the natural orbitals as the single-particle basis is the most compact. The basis of the attachment/detachment analysis is to consider what could

be termed natural orbitals of the electronic transition and their occupation numbers (associated eigenvalues). These are defined as the eigenvectors \mathbf{U} defined by

$$\mathbf{U}^t \Delta \mathbf{U} = \delta \quad (6.26)$$

The sum of the occupation numbers δ_p of these orbitals is then

$$tr(\Delta) = \sum_{p=1}^N \delta_p = n \quad (6.27)$$

where n is the net gain or loss of electrons in the transition. The net gain in an electronic transition which does not involve ionisation or electron attachment will obviously be zero.

The detachment density

$$\mathbf{D} = \mathbf{U} \mathbf{d} \mathbf{U}^t \quad (6.28)$$

is defined as the sum of all natural orbitals of the difference density with negative occupation numbers, weighted by the absolute value of their occupations where \mathbf{d} is a diagonal matrix with elements

$$d_p = -\min(\delta_p, 0) \quad (6.29)$$

The detachment density corresponds to the electron density associated with single particle levels vacated in an electronic transition or hole density.

The attachment density

$$\mathbf{A} = \mathbf{U} \mathbf{a} \mathbf{U}^t \quad (6.30)$$

is defined as the sum of all natural orbitals of the difference density with positive occupation numbers where \mathbf{a} is a diagonal matrix with elements

$$a_p = \max(\delta_p, 0) \quad (6.31)$$

The attachment density corresponds to the electron density associated with the single particle levels occupied in the transition or particle density. The difference between the attachment and detachment densities yields the original difference density matrix

$$\Delta = \mathbf{A} - \mathbf{D} \quad (6.32)$$

6.5 FURTHER READING

Basis sets (Chapters 7 and 8)

Ground state methods (Chapters 4 and 5)

- [1] J. E. Del Bene, R. Ditchfield and J. A. Pople, *J. Chem. Phys.* (1971), **55**, 2236.
- [2] J. B. Foresman, M. Head-Gordon, J. A. Pople and M. J. Frisch, *J. Phys. Chem.* (1992), **96**, 135.
- [3] D. Maurice and M. Head-Gordon, *Int. J. Quantum Chem.* (1995), **29**, 361.
- [4] T. D. Bouman and A. E. Hansen, *Int. J. Quantum Chem. Sym* (1989), **23**, 381.
- [5] A. E. Hansen, B. Voigt and S. Rettrup, *Int. J. Quantum Chem.* (1983), **23**, 595.
- [6] D. Maurice and M. Head-Gordon, *J. Phys. Chem.* (1996), **100**, 6131.
- [7] M. Head-Gordon, A. M. Grana, D. Maurice and C. A. White, *J. Phys. Chem.* (1995), **99**, 14261.
- [8] J. F. Stanton, J. Gauss, N. Ishikawa and M. Head-Gordon, *J. Chem. Phys.* (1995), **103**, 4160.
- [9] S. Zilberg and Y. Haas, *J. Chem. Phys.* (1995), **103**, 20.
- [10] C. M. Gittins, E. A. Rohlfing and C. M. Rohlfing, *J. Chem. Phys.* (1996), **105**, 7323.
- [11] D. Maurice and M. Head-Gordon, *Mol. Phys.* (1999) **96**, 1533.
- [12] D. Maurice, Ph.D. Thesis. "Single Electron Theories of Excited States" University of California: Berkeley, 1998.
- [13] E. Runge and E. K. U. Gross, *Phys. Rev. Lett.* (1984) **52**, 997.
- [14] M. E. Casida, in *Recent Advances in Density Functional Methods*, Part I, edited by D. P. Chong (World Scientific, Singapore, 1995), p 155.
- [15] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* (1999) **302**, 375.
- [16] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* (1999) **314**, 291.
- [17] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub, *J. Chem. Phys.* (1998) **108**, 4439.
- [18] D. J. Tozer and N. C. Handy, *J. Chem. Phys.* (1998) **109**, 10180.
- [19] S. Hirata, T. J. Lee, and M. Head-Gordon, *J. Chem. Phys.* (1999) **111**, 8904.
- [20] M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.* (1994), **219**, 21.
- [21] M. Head-Gordon, D. Maurice and M. Oumi, *Chem. Phys. Lett.* (1995), **246**, 114.
- [22] M. Oumi, D. Maurice, T. J. Lee and M. Head-Gordon, *Chem. Phys. Lett.* (1997), **279**, 151.
- [23] See for example, H. Koch and P. Jorgensen, *J. Chem. Phys.* (1990) **93**, 3333.
- [24] J.F.Stanton, and R.J.Bartlett, *J. Chem. Phys.* (1993) **98**, 7029.
- [25] A. I. Krylov, C. D. Sherrill, and M. Head-Gordon, *J. Chem. Phys.* (2000) **113**, 6509.
- [26] M. Oumi, D. Maurice and M. Head-Gordon, *Spectrochim. Acta A* (1999) **55**, 525.
- [27] C.-P. Hsu, S. Hirata and M. Head-Gordon, *J. Phys. Chem.* (in press).

CHAPTER 7 BASIS SETS

7.1 INTRODUCTION

A basis set is a set of functions combined linearly to model molecular orbitals. Basis functions can be considered as representing the atomic orbitals of the atoms and are introduced in quantum chemical calculations because the equations defining the molecular orbitals are otherwise very difficult to solve directly.

Many standard basis sets have been carefully optimized and tested over the years. In principle, a user would employ the largest basis set available in order to model molecular orbitals as accurately as possible. In practice, however, computational cost grows rapidly with the size of the basis set so a compromise must be sought between accuracy and cost. If this is systematically pursued, it leads to a “theoretical model chemistry” [1], that is, a well-defined energy procedure (*e.g.*, Hartree-Fock) in combination with a well defined basis set.

Basis sets have been constructed from Slater, Gaussian, plane wave and delta functions. Slater functions were initially employed because they are considered “natural” and have the correct behaviour at the origin and asymptotically. However, the two electron repulsion integrals (ERIs) encountered when using Slater basis functions are expensive and difficult. Delta functions are used in several quantum chemistry programs. However, while codes incorporating delta functions are simple, thousands of functions are required to achieve accurate results, even for small molecules. Plane waves are widely used and highly efficient for calculations on periodic systems, but are not so convenient or natural for molecular calculations.

The most important basis sets are contracted sets of atom-centered Gaussian functions where the numbers of basis functions used are related to the number of core and valence atomic orbitals, and whether the atom is light (H or He) or heavy (everything else). Contracted basis sets have been shown to be computationally efficient and to have the ability to yield chemical accuracy (see the Appendix on AOINTS). The Q-Chem program has been optimized to exploit basis sets of the contracted Gaussian function type and has a large number of built-in standard basis sets (developed by Dunning and Pople, among others) which the user can access quickly and easily.

The selection of a basis set for quantum chemical calculations is very important. It is sometimes possible to use small basis sets to obtain good chemical accuracy, but calculations can often be significantly improved by the addition of diffuse and polarization functions. Consult the literature and reviews [1,2,3,4,5] to aid your selection. Refer to the “Further Reading” section at the end of this chapter.

7.2 BUILT-IN BASIS SETS

Q-Chem is equipped with many standard basis sets [6]. Q-Chem allows the user to identify the required basis set by its standard symbolic representation. The available built in basis sets are presented below. The four types are:

1. Pople basis sets
2. Old Dunning basis sets
3. New, correlation consistent Dunning basis sets
4. Ahlrichs basis sets

7.2.1 FEATURES

- Extra diffuse functions necessary for high quality excited state calculations
- Pople's standard basis sets
- Arsenal of Dunning derived basis sets
- Dunning's correlation consistent basis sets
- Ahlrichs basis sets
- Standard polarisation functions
- Basis sets are requested by symbolic representation
- *s*, *p*, *d*, *f* and *g* angular momentum types of basis functions
- Maximum number of shells per atom is 100
- Pure and Cartesian basis functions

7.2.2 JOB CONTROL

BASIS

Sets the basis set to be used

VARIABLE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen

User-defined. See section below

Symbol

Use standard basis sets as in the table below

RECOMMENDATIONS:

Consult literature and reviews to aid your selection.

7.3 BASIS SET SYMBOLIC REPRESENTATION

Examples are given in the tables below and follow the standard format generally adopted for specifying basis sets. The single exception applies to additional diffuse functions. These are best inserted in a similar manner to the polarisation functions; in parentheses with the light atom designation following heavy atom designation. (*i.e.*, *heavy*, *light*). Use a period (.) as a place holder (see examples).

	STO- $j(k+,l+)G(m,n)$	$j-21(k+,l+)G(m,n)$	$j-31(k+,l+)G(m,n)$	$j-311(k+,l+)G(m,n)$		
j	2, 3, 6	3	4, 6	6		
k	# sets of heavy atom diffuse functions					
l	# sets of heavy atom diffuse functions					
m	d	2d	3d	df	2df	3df
n	p	2p	3p	pd	2pd	3pd

Table 7.1a Summary of Pople type basis sets available in the Q-Chem program

Symbolic name	Atoms Supported
STO-2G	H, He, Li→Ne, Na→Ar, K, Ca, Sr
STO-3G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Sb
STO-6G	H, He, Li→Ne, Na→Ar, K→Kr
3-21G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Xe, Cs
4-31G	H, He, Li→Ne, P→Cl
6-31G	H, He, Li→Ne, Na→Ar, K→Kr
6-311G	H, He, Li→Ne, Na→Ar, Ga→Kr
G3LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 7.1b Atoms supported for Pople basis sets available in Q-Chem (see the Table below for specific examples).

Basis set	Atoms Supported
3-21G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Xe, Cs
3-21+G	H, He, Na→Cl, Na→Ar
3-21G*	H, He, Na→Cl
6-31G	H, He, Li→Ne, Na→Ar, K→Zn
6-31+G	H, He, Li→Ne, Na→Ar
6-31G*	H, He, Li→Ne, Na→Ar, K→Zn
6-31G(d,p)	H, He, Li→Ne, Na→Ar, K→Zn
6-31G(.,+)G	H, He, Li→Ne, Na→Ar
6-31+G*	H, He, Li→Ne, Na→Ar
6-311G	H, He, Li→Ne, Na→Ar, Ga→Kr
6-311+G	H, He, Li→Ne, Na→Ar
6-311G*	H, He, Li→Ne, Na→Ar, Ga→Kr
6-311G(d,p)	H, He, Li→Ne, Na→Ar, Ga→Kr
G3LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 7.1c Examples of extended Pople basis sets

	$SV(k+,l+)(md,np)$	$DZ(k+,l+)(md,np)$	$TZ(k+,l+)(md,np)$
k	# sets of heavy atom diffuse functions		
l	# sets of heavy atom diffuse functions		
m	# sets of d functions on heavy atoms		
n	# sets of p functions on light atoms		

Table 7.2a Summary of Dunning-type basis sets available in the Q-Chem program

Symbolic name	Atoms Supported
SV	H, Li→Ne
DZ	H, Li→Ne, Al→Cl
TZ	H, Li→Ne

Table 7.2b Atoms supported for old Dunning basis sets available in Q-Chem

Basis set	Atoms Supported
SV	H, Li→Ne
SV*	H, B→Ne
SV(d,p)	H, B→Ne
DZ	H, Li→Ne, Al→Cl
DZ+	H, B→Ne
DZ++	H, B→Ne
DZ*	H, Li→Ne
DZ**	H, Li→Ne
DZ(d,p)	H, Li→Ne
TZ	H, Li→Ne
TZ+	H, Li→Ne
TZ++	H, Li→Ne
TZ*	H, Li→Ne
TZ**	H, Li→Ne
TZ(d,p)	H, Li→Ne

Table 7.2c Examples of extended Dunning basis sets

Basis Set	Basis Set
cc-pVDZ	aug-cc-pVDZ
cc-pVTZ	aug-cc-pVTZ
cc-pVQT	aug-cc-pVQT
cc-pCVDZ	aug-cc-pCVDZ
cc-pCVTZ	aug-cc-pCVTZ
cc-pCVQZ	aug-cc-pCVQZ

Table 7.3a Summary of Dunning's correlation-consistent basis sets available in Q-Chem

Symbolic name	Atoms Supported
cc-pVDZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pVTZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pVQZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pCVDZ	B→Ne
cc-pCVTZ	B→Ne
cc-pCVQZ	B→Ne
aug-cc-pVDZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pVTZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pVQZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pCVDZ	B→F
aug-cc-pCVTZ	B→Ne
aug-cc-pCVQZ	B→Ne

Table 7.3b Atoms supported Dunning correlation-consistent basis sets available in Q-Chem

Symbolic name	Atoms Supported
TZV	Li→Kr
VDZ	H→Kr
VTZ	H→Kr

Table 7.4 Atoms supported for Ahlrichs basis sets available in Q-Chem

7.3.1 CUSTOMIZATION

Q-Chem offers a number of standard and special customization features. One of the most important is that of supplying additional diffuse functions. Diffuse functions are often important for the purpose of studying anions and excited states of molecules. For the latter, it is often important to supply several additional diffuse functions. This can be achieved by splitting the standard basis set diffuse function set into multiple diffuse sets, using a special scaling factor (3.32) applied to the standard diffuse function exponent. This yields a geometric series of diffuse function exponents, starting with the original standard exponent value.

PRINT_GENERAL_BASIS

Controls print out of built in basis sets in input format

VARIABLE:

LOGICAL

DEFAULT:

False Do not print out standard basis set information

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATIONS:

Useful for modification of standard basis sets

7.4 USER-DEFINED BASIS SETS (\$BASIS)

7.4.1 INTRODUCTION

Users may, on occasion, prefer to use non-standard basis sets and, it is possible to declare user-defined basis sets in Q-Chem input (see Chapter 3 on Q-Chem inputs). The format for inserting a non-standard user-defined basis set is both logical and flexible and is described in detail in the job control section below.

Note that the SAD guess is not currently supported with non-standard user-defined basis sets. The simplest alternative is to specify the *GWH* or *CORE* options for *SCF_GUESS*, but these are relatively ineffective other than for small basis sets. The recommended alternative is to employ basis set projection, by specifying a standard basis set for the *BASIS2* keyword. See the section in Chapter 4 on initial guesses for more information.

7.4.2 FEATURES

- Insertion of user-defined contracted basis sets
- Angular momentum types S, P, SP, D, F & G
- Pure or Cartesian functions
- Maximum number of shells per atom is 100

7.4.3 JOB CONTROL

BASIS

OPTION:

General, Gen User-defined basis via *\$basis* keyword

PURECART

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

None.

OPTIONS:

gfd Use 1 for pure and 2 for Cartesian.
Must be defined for user supplied basis sets.

When using a non-standard basis set which incorporates *d* or higher basis functions of angular momentum, the *\$rem* variable *PURECART* needs to be initiated. This *\$rem* variable indicates to the Q-Chem program how to handle the angular form of the basis functions. As indicated above, each integer represents an angular momentum type which can be defined as either pure (1) or Cartesian (2). For example, 111 would indicate to the Q-Chem program to treat all G, F and D basis functions as being in the pure form. 121 would indicate G and D's pure and F's Cartesian, *etc.*

7.4.4 EXAMPLE

```

$molecule
0 1
O
H O OH
H O OH 2 HOH

OH = 1.2
HOH = 110.0
$end

$rem
EXCHANGE          HF          HF Exchange
BASIS             Gen          User-defined general basis
BASIS2            STO-3G       STO-3G orbitals as initial guess
PURECART          112         Cartesian D functions, Pure F and G
$end

$basis
H                0
S                1    1.00
                1.30976    0.430129
                0.233136    0.678914
****
O                0
S                2    1.00
                49.9810    0.430129
                8.89659    0.678914
SP               2    1.00
                1.94524    0.0494720    0.511541
                0.493363    0.963782    0.612820
D                1    1.00
                0.39    1.00
F                1    1.00
                4.1    1.00
G                1    1.00
                3.35    1.00
****
$end

```

Example 7.1 Example of adding a user-defined non-standard basis set. Note that since D, F and G functions are incorporated, the *\$rem* variable *PURECART* must be set. Note the use of *BASIS2* for the initial guess.

7.5 FORMAT FOR USER-DEFINED BASIS SETS

<i>atomic_symbol</i>	0			
<i>ang_mom_sym</i>		<i>contraction_K</i>	<i>scaling</i>	
<i>exp_1</i>	<i>coeff_1_Lmin</i>	<i>coeff_1_(Lmin+1)</i>	...	<i>coeff_1_Lmax</i>
<i>exp_2</i>	<i>coeff_2_Lmin</i>	<i>coeff_2_(Lmin+1)</i>	...	<i>coeff_2_Lmax</i>
<i>exp_3</i>	<i>coeff_3_Lmin</i>	<i>coeff_2_(Lmin+1)</i>	...	<i>coeff_3_Lmax</i>
.
.
<i>exp_K</i>	<i>coeff_K_Lmin</i>	<i>coeff_K_(Lmin+1)</i>	...	<i>coeff_K_Lmax</i>

<i>atomic_symbol</i>	Atomic symbol of the atom (atomic number not accepted)
<i>ang_mom_sym</i>	Angular momentum symbol (S, P, SP, D, F, G)
<i>contraction</i>	Degree of contraction of the shell (integer)
<i>scaling</i>	Scaling to be applied to exponents (default is 1.00)
<i>exp</i>	Gaussian primitive exponent (positive real number)
<i>coeff</i>	Contraction coefficient for each angular momentum (non-zero real numbers)

Atoms are terminated with **** and the complete basis set is terminated with the *\$end* keyword terminator. No blank lines can be incorporated within the general basis set input. As with all Q-Chem input deck information, all input is case-insensitive.

7.5.1 CUSTOMIZATION

In addition to defining one's own basis set, it is possible to define separate standard basis sets for individual atoms or a combination of standard and non-standard basis sets, exclusively, for individual atoms. These can be entered by the more familiar symbolic representation.

- Notes:** (1) It is not possible to augment a standard basis set in this way; the whole basis needs to be inserted manually (angular momentum, exponents, contraction coefficients) and additional functions added. Standard basis set exponents and coefficients can be easily obtained by appropriately setting the *PRINT_GENERAL_BASIS \$rem* variable to *TRUE*.
- (2) The *PURECART* flag must be set for *all* general basis input containing D angular momentum or higher functions, regardless of whether standard basis sets are entered in this non-standard manner.

7.5.2 EXAMPLES

```
$molecule
0 1
O
H O OH
H O OH 2 HOH
```

```
OH = 1.2
HOH = 110.0
$end
```

```
$rem
EXCHANGE          HF          HF Exchange
BASIS             Gen          User-defined general basis
SCF_GUESS         GWH          SAD unavailable for general basis
$end
```

```
$basis
H          0
S          2    1.00
           1.30976    0.430129
           0.233136    0.678914
****
O          0
S          2    1.00
           49.9810    0.430129
           8.89659    0.678914
SP         2    1.00
           1.94524    0.0494720    0.511541
           0.493363    0.963782    0.612820
****
$end
```

Example 7.2 Example of adding a user-defined non-standard basis set.

```

$molecule
0 1
O
H O OH
H O OH 2 HOH

OH = 1.2
HOH = 110.0
$end

$rem
EXCHANGE          HF          HF Exchange
CORRELATION       None        No correlation energy
BASIS             General    User-defined general basis
SCF_GUESS        GWH        SAD unavailable for general basis
PURECART         1          Pure D functions
$end

$basis
H                0
S                2    1.00
                1.30976    0.430129
                0.233136    0.678914
****
O                0
S                2    1.00
                49.9810    0.430129
                8.89659    0.678914
SP               2    1.00
                1.94524    0.0494720    0.511541
                0.493363    0.963782    0.612820
D                1    1.00
                0.39    1.00000
****
$end

```

Example 7.3 Example of adding a user-defined non-standard basis set. Note that since D functions are incorporated the *\$rem* variable *PURECART* has been initiated.

```
$molecule
0 1
O
H O OH
H O OH 2 HOH

OH = 1.2
HOH = 110.0
$end

$rem
EXCHANGE      HF      HF Exchange
BASIS         General  User-defined general basis
SCF_GUESS     GWH     SAD unavailable for general basis
$end

$basis
H              0
STO-2G
****
O              0
STO-6G
****
$end
```

Example 7.4 Example of adding a user-defined non-standard basis set where the user defines different standard basis sets for each atom.

```
$molecule
0 1
O
H O OH
H O OH 2 HOH
```

```
OH = 1.2
HOH = 110.0
$end
```

```
$rem
EXCHANGE          HF          HF Exchange
BASIS             General     User Defined general basis
PURECART          2          Cartesian D functions
BASIS2            STO-3G     Use STO-3G as initial guess
$end
```

```
$basis
H                0
6-31G
****
O                0
6-311G(d)
****
$end
```

Example 7.5 Example of adding a user defined non-standard basis set. The user is able to specify different standard basis sets for different atoms.

```

$molecule
0 1
O
H O OH
H O OH 2 HOH

OH = 1.2
HOH = 110.0
$end

$rem
EXCHANGE          HF          HF Exchange
BASIS              General     User-defined general basis
BASIS2             STO-3G      Use STO-3G as initial guess
$end

$basis
H                  0
S                  2    1.00
                   1.30976      0.430129
                   0.233136      0.678914
****
O                  0
STO-6G
****
$end

```

Example 7.6 Example of adding a user-defined non-standard basis set. The user is able to specify standard basis sets for some atoms and supply user-defined exponents and contraction coefficients for others. This might be particularly useful in cases where the user has constructed exponents and contraction coefficients for atoms not defined in standard basis sets so that only the non-defined atoms need have the exponents and contraction coefficients entered.

7.6 FURTHER READING

Ground State Methods (Chapters 4 and 5)

Effective Core Potentials (Chapter 8)

- [1] W. J. Hehre, L. Radom, P. v. R. Schleyer and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York, 1986)
- [2] S. Huzinaga, *Comp. Phys. Rep.* (1985), **2**, 279.
- [3] E. R. Davidson and D. Feller, *Chem. Rev.* (1986), **86**, 681.
- [4] D. Feller and E. R. Davidson, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd (Wiley-VCH, New York, 1990) Volume 1, pg. 1.
- [5] F. Jensen, *Introduction to Computational Chemistry* (Wiley, New York, 1999).
- [6] Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version 1.0, as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multi-program laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. Contact David Feller, Karen Schuchardt or Don Jones for further information.

CHAPTER 8 EFFECTIVE CORE POTENTIALS

8.1 INTRODUCTION

The application of quantum chemical methods to elements in the lower half of the Periodic Table is more difficult than for the lighter atoms. There are two key reasons for this:

- the number of electrons in heavy atoms is large
- relativistic effects in heavy atoms are often non-negligible

Both of these problems stem from the presence of large numbers of core electrons and, given that such electrons do not play a significant *direct* role in chemical behavior, it is natural to ask whether it is possible to model their effects in some simpler way. Such enquiries led to the invention of Effective Core Potentials (ECPs) or pseudopotentials. For reviews of relativistic effects in chemistry, see for example refs [1,2,3,4].

If we seek to replace the core electrons around a given nucleus by a pseudopotential, while affecting the chemistry as little as possible, the pseudopotential should have the same effect on nearby valence electrons as the core electrons. The most obvious effect is the simple electrostatic repulsion between the core and valence regions but the requirement that valence orbitals must be orthogonal to core orbitals introduces additional subtler effects that cannot be neglected.

The most widely used ECPs today are of the form first proposed by Kahn *et al* [5] in the 1970s. These model the effects of the core by a one-electron operator $U(r)$ whose matrix elements are simply added to the one-electron Hamiltonian matrix. The ECP operator is given by

$$U(r) = U_L(r) + \sum_{l=0}^{L-1} \sum_{m=-l}^{+l} |Y_{lm}\rangle [U_l(r) - U_L(r)] \langle Y_{lm}| \quad (8.1)$$

where the $|Y_{lm}\rangle$ are spherical harmonic projectors and the $U_l(r)$ are linear combinations of Gaussians, multiplied by r^{-2} , r^{-1} or r^0 . In addition, $U_L(r)$ contains a Coulombic term N_c / r , where N_c is the number of core electrons.

One of the key issues in the development of pseudopotentials is the definition of the “core”. So-called “large-core” ECPs include all shells except the outermost one, but “small-core” ECPs include all except the outermost *two* shells. Although the small-core ECPs are more expensive to use (because more electrons are treated explicitly), it is often found that their enhanced accuracy justifies their use.

When an ECP is constructed, it is usually based either on non-relativistic, or quasi-relativistic all-electron calculations. As one might expect, the quasi-relativistic ECPs tend to yield better results than their non-relativistic brethren, especially for atoms beyond the 3*d* block.

8.2 BUILT-IN PSEUDOPOTENTIALS

8.2.1 OVERVIEW

Q-Chem is equipped with several standard ECP sets which are specified using the *ECP* keyword within the `$rem` block. The built-in ECPs, which are described in some detail at the end of this Chapter, fall into four families:

- The Hay-Wadt (or Los Alamos) sets (HWMB and LANL2DZ)
- The Stevens-Basch-Krauss-Jansien-Cundari set (SBKJC)
- The Christiansen-Ross-Ermler-Nash-Bursten sets (CRENBS and CRENBL)
- The Stuttgart-Bonn sets (SRLC and SRSC)

References and information about the definition and characteristics of most of these sets can be found at the EMSL site of the Pacific Northwest National Laboratory [6]

<http://www.emsl.pnl.gov:2080/forms/basisform.html>

Each of the built-in ECPs comes with a matching orbital basis set for the valence electrons. In general, it is advisable to use these together and, if you select a basis set other than the matching one, Q-Chem will print a warning message in the output file. If you omit the BASIS keyword entirely, Q-Chem will automatically provide the matching one.

8.2.2 JOB CONTROL FOR BUILT-IN ECP'S

ECP

Defines the effective core potential and associated basis set to be used

VARIABLE:

STRING

DEFAULT:

No pseudopotential

OPTIONS:

General, Gen User defined. (*\$ecp* keyword required)

Symbol Use standard pseudopotentials discussed above.

RECOMMENDATIONS:

Pseudopotentials are recommended for first row transition metals and heavier elements. Consult Ch. 8 and reviews for more details.

8.2.3 COMBINING PSEUDOPOTENTIALS

If you wish, you can use different ECP sets for different elements in the system. This is especially useful if you would like to use a particular ECP but find that it is not available for all of the elements in your molecule. To combine different ECP sets, you set the ECP and BASIS keywords to “Gen” or “General” and then add a `$ecp` block and a `$basis` block to your input file. In each of these blocks, you must name the ECP and the orbital basis set that you wish to use, separating each element by a sequence of four asterisks.

8.2.4 EXAMPLES

```
$molecule
  0 1
   Ag
  Cl Ag r

  r = 2.4
$end

$rem
  EXCHANGE   hf           Hartree-Fock calculation
  ECP        lanl2dz      Using the Hay-Wadt ECP
  BASIS      lanl2dz      And the matching basis set
$end
```

Example 8.1 Computing the HF/LANL2DZ energy of AgCl at a bond length of 2.4 Å.

```

$molecule
  0 1
  Cd
  Br1 Cd r
  Br2 Cd r Br1 180.

  r = 2.4
$end
$rem
  JOBTYP          opt          Geometry optimization
  EXCHANGE        hf          Hartree-Fock theory
  ECP             gen          Combine ECPs
  BASIS           gen          Combine basis sets
  PURECART        1           Use pure d functions
$end
$ecp
  Cd
    srsc
    ****
  Br
    srlc
    ****
$end
$basis
  Cd
    srsc
    ****
  Br
    srlc
    ****
$end

```

Example 8.2 Computing the HF geometry of CdBr_2 using the Stuttgart relativistic ECPs. The small-core ECP and basis are employed on the Cd atom and the large-core ECP and basis on the Br atoms.

8.3 USER-DEFINED PSEUDOPOTENTIALS

Many users will find that the library of built-in pseudopotentials is adequate for their needs. However, if you need to use an ECP that is not built into Q-Chem, you can enter it in much the same way as you can enter a user-defined orbital basis set (see Chapter 7).

8.3.1 JOB CONTROL FOR USER-DEFINED ECP'S

To apply a user-defined pseudopotential, you must set the ECP and BASIS keywords in `$rem` to “Gen”. You then add a `$ecp` block that defines your ECP, element by element, and a `$basis` block that defines your orbital basis set, separating elements by asterisks.

The syntax within the `$basis` block is described in Chapter 7.

The syntax for each record within the `$ecp` block is as follows:

\$ecp

For each atom that will bear an ECP

Chemical symbol for the atom

ECP name ; the L value for the ECP ; number of core electrons removed

For each ECP component (in the order unprojected, $\hat{P}_0, \hat{P}_1, \dots, \hat{P}_{L-1}$)

The component name

The number of Gaussians in the component

For each Gaussian in the component

The power of r ; the exponent ; the contraction coefficient

A sequence of four asterisks (i.e. ****)

\$end

Figure 8.1 The format for the input of user-defined pseudopotentials.

Notes about the implementation in Q-Chem 2.0

- All of the information in the \$ecp block is case-insensitive
- The L value may not exceed 4. That is, nothing beyond G projectors is allowed
- The power of r (which includes the Jacobian r^2 factor) must be 0, 1 or 2.

8.3.2 EXAMPLE

```

$molecule
  0 1
  al
  h1 al r
  h2 al r h1 120.
  h3 al r h1 120. h2 180.

  r = 1.6
$end
$rem
  JOBTYP  opt      Geometry optimization
  EXCHANGE hf       Hartree-Fock theory
  ECP      gen      User-defined ECP
  BASIS    gen      User-defined basis
$end
$ecp
  Al
  Stevens_ECP  2  10
  d potential
  1
  1      1.95559      -3.03055
  s-d potential
  2
  0      7.78858      6.04650
  2      1.99025      18.87509
  p-d potential
  2
  0      2.83146      3.29465
  2      1.38479      6.87029
  ****
$end
$basis
  Al
  SP  3  1.00
  0.9011      -0.30377      -0.07929
  0.4495      0.13382      0.16540
  0.1405      0.76037      0.53015
  SP  1  1.00
  0.04874      0.32232      0.47724
  ****
  H
  3-21G
  ****
$end

```

Example 8.3 Optimizing the HF geometry of AlH_3 using a user-defined ECP and basis set on Al and the 3-21G basis on H

8.4 PSEUDOPOTENTIALS AND DENSITY FUNCTIONAL THEORY

Q-Chem's pseudopotential package and DFT package are tightly integrated and facilitate the application of advanced density functionals to molecules containing heavy elements. Any of the local, gradient-corrected and hybrid functionals discussed in Chapter 4 may be used and you may also perform ECP calculations with user-defined hybrid functionals.

In a DFT calculation with pseudopotentials, the exchange-correlation energy is obtained entirely from the non-core electrons. This will be satisfactory if there are no chemically important core-valence effects but may introduce significant errors, particularly if you are using a "large-core" ECP.

Q-Chem's default quadrature grid is SG-1 (see Chapter 4) which was originally defined only for the elements up to argon. In Q-Chem 2.0, however, the SG-1 grid has been extended and it is now defined for all atoms up to, and including, the actinides.

8.4.1 EXAMPLE

```
$molecule
  1 1
  xe
  f1 xe r1
  f2 xe r2 f1 a
  f3 xe r2 f1 a f2 90.
  f4 xe r2 f1 a f3 90.
  f5 xe r2 f1 a f4 90.

  r1 = 2.07
  r2 = 2.05
  a = 80.0
$end
$rem
  JOBTYP      opt
  EXCHANGE    b3lyp
  ECP         sbkjc
$end
```

Example 8.4 Optimization of the structure of XeF_5^+ using B3LYP theory and the ECPs of Stevens and collaborators. Note that the BASIS keyword has been omitted and, therefore, the matching SBKJC orbital basis set will be used.

8.5 PSEUDOPOTENTIALS AND ELECTRON CORRELATION

The pseudopotential package is integrated with the electron correlation package and it is therefore possible to apply any of Q-Chem's post-Hartree-Fock methods to systems in which some of the atoms may bear pseudopotentials. Of course, the correlation energy contribution arising from core electrons that have been replaced by an ECP is *not* included. In this sense, correlation energies with ECPs are comparable to correlation energies from frozen core calculations. However, the use of ECPs effectively removes both core electrons *and* the corresponding virtual (unoccupied) orbitals.

8.5.1 EXAMPLE

```
$molecule
  0 1
  x1
  x2  x1  xx
  se1 x1 sx x2 90.
  se2 x1 sx x2 90. se1 90.
  se3 x1 sx x2 90. se2 90.
  se4 x1 sx x2 90. se3 90.
  se5 x2 sx x1 90. se1 45.
  se6 x2 sx x1 90. se5 90.
  se7 x2 sx x1 90. se6 90.
  se8 x2 sx x1 90. se7 90.

  xx = 1.2
  sx = 2.8
$end
$rem
  JOBTYP      opt      Geometry optimization
  EXCHANGE    hf       Hartree-Fock theory
  ECP         lanl2dz   Hay-Wadt ECP and basis
$end

@@@

$molecule
  read
$end
$rem
  JOBTYP      sp       Single-point energy
  CORRELATION mp2      MP2 correlation energy
  ECP         lanl2dz   Hay-Wadt ECP and basis
  SCF_GUESS   read     Read in the MOs
$end
```

Example 8.5 Optimization of the structure of Se_8 using HF/LANL2DZ, followed by a single-point energy calculation at the MP2/LANL2DZ level

8.6 PSEUDOPOTENTIALS AND VIBRATIONAL FREQUENCIES

The pseudopotential package is also integrated with the vibrational analysis package and it is therefore possible to compute the vibrational frequencies (and hence the infra-red and Raman spectra) of systems in which some of the atoms may bear pseudopotentials.

Q-Chem 2.0 cannot compute second derivatives of the required ECP integrals analytically and it therefore estimates the required second-derivatives by taking finite-differences of analytically computed first derivatives. This can be time-consuming in large systems.

8.6.1 EXAMPLE

```
$molecule
  0 1
  Te
  O1 Te r
  O2 Te r O1 a

  r = 1.8
  a = 108.
$end
$rem
  JOBTYP      opt
  EXCHANGE    hf
  ECP         srlc
$end

@@@

$molecule
  read
$end
$rem
  JOBTYP      freq
  EXCHANGE    hf
  ECP         srlc
  SCF_GUESS   read
$end
```

Example 8.6 Structure and vibrational frequencies of TeO_2 using Hartree-Fock theory and the Stuttgart relativistic large-core ECPs. Note that the vibrational frequency job reads both the optimized structure and the molecular orbitals from the geometry optimization job that precedes it.

8.7 A BRIEF GUIDE TO Q-CHEM'S BUILT-IN ECP'S

The remainder of this Chapter consists of a brief reference guide to Q-Chem's built-in ECPs. The ECPs vary in their complexity and their accuracy and the purpose of the guide is to enable the user quickly and easily to decide which ECP to use in a planned calculation.

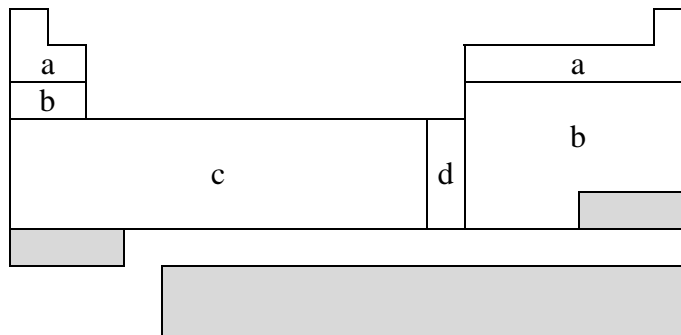
The following information is provided for each ECP:

- The elements for which the ECP is available in Q-Chem. This is shown on a schematic Periodic Table by shading all the elements that are *not* supported.
- The literature reference for each element for which the ECP is available in Q-Chem.
- The matching orbital basis set that Q-Chem will use for light (i.e. non-ECP atoms). For example, if the user requests SRSC pseudopotentials – which are defined only for atoms beyond argon – Q-Chem will use the 6-311G* basis set for all atoms up to Ar.
- The core electrons that are replaced by the ECP. For example, in the LANL2DZ pseudopotential for the Fe atom, the core is [Ne], indicating that the 1s, 2s and 2p electrons are removed.
- The maximum spherical harmonic projection operator (see Section 7.1) that is used for each element. This often, but not always, corresponds to the maximum orbital angular momentum of the core electrons that have been replaced by the ECP. For example, in the LANL2DZ pseudopotential for the Fe atom, the maximum projector is of P-type.
- The number of valence basis functions of each angular momentum type that are present in the matching orbital basis set. For example, in the matching basis for the LANL2DZ pseudopotential for the Fe atom, there are three *s* shells, three *p* shells and two *d* shells. This basis is therefore almost of triple-split valence quality.

Finally, we note the limitations of the current ECP implementation within Q-Chem:

- Energies can be calculated only for *s*, *p*, *d* and *f* basis functions with G projectors. Consequently, Q-Chem cannot perform energy calculations on actinides using SRLC.
- Gradients can be calculated only for *s*, *p* and *d* basis functions with F projectors and only for *s* and *p* basis functions with G projectors.

8.7.1 THE HWMB PSEUDOPOTENTIAL AT A GLANCE

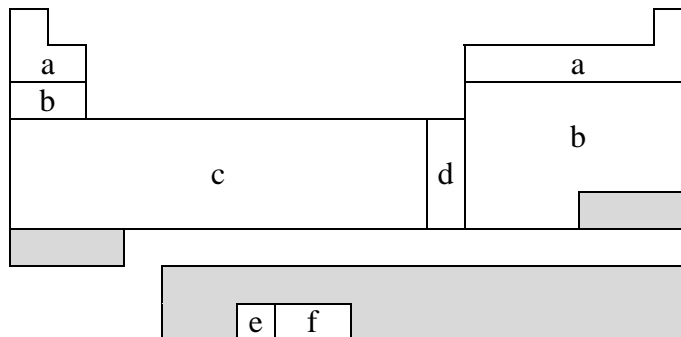


HWMB is not available for shaded elements

- (a) No pseudopotential; Pople STO-3G basis used
- (b) Wadt & Hay, J. Chem. Phys. 82 (1985) 285
- (c) Hay & Wadt, J. Chem. Phys. 82 (1985) 299
- (d) Hay & Wadt, J. Chem. Phys. 82 (1985) 270

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(1s)
Li – Ne	<i>none</i>	<i>none</i>	(2s,1p)
Na – Ar	[Ne]	P	(1s,1p)
K – Ca	[Ne]	P	(2s,1p)
Sc – Cu	[Ne]	P	(2s,1p,1d)
Zn	[Ar]	D	(1s,1p,1d)
Ga – Kr	[Ar]+3d	D	(1s,1p)
Rb – Sr	[Ar]+3d	D	(2s,1p)
Y – Ag	[Ar]+3d	D	(2s,1p,1d)
Cd	[Kr]	D	(1s,1p,1d)
In – Xe	[Kr]+4d	D	(1s,1p)
Cs – Ba	[Kr]+4d	D	(2s,1p)
La	[Kr]+4d	D	(2s,1p,1d)
Hf – Au	[Kr]+4d+4f	F	(2s,1p,1d)
Hg	[Xe]+4f	F	(1s,1p,1d)
Tl – Bi	[Xe]+4f+5d	F	(1s,1p)

8.7.2 THE LANL2DZ PSEUDOPOTENTIAL AT A GLANCE

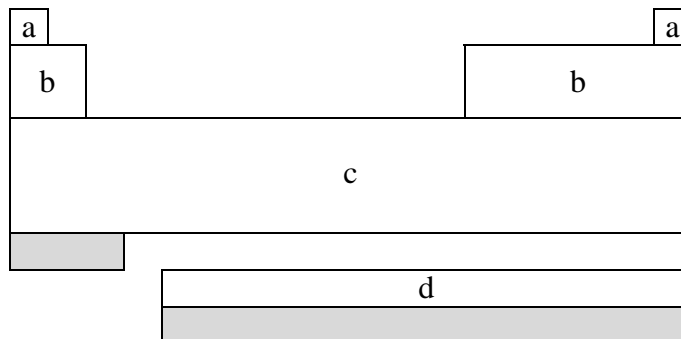


LANL2DZ is not available for shaded elements

- (a) No pseudopotential; Pople 6–31G basis used
- (b) Wadt & Hay, J. Chem. Phys. 82 (1985) 285
- (c) Hay & Wadt, J. Chem. Phys. 82 (1985) 299
- (d) Hay & Wadt, J. Chem. Phys. 82 (1985) 270
- (e) Hay, J. Chem. Phys. 79 (1983) 5469
- (f) Wadt, to be published

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(2s)
Li – Ne	<i>none</i>	<i>none</i>	(3s,2p)
Na – Ar	[Ne]	P	(2s,2p)
K – Ca	[Ne]	P	(3s,3p)
Sc – Cu	[Ne]	P	(3s,3p,2d)
Zn	[Ar]	D	(2s,2p,2d)
Ga – Kr	[Ar]+3d	D	(2s,2p)
Rb – Sr	[Ar]+3d	D	(3s,3p)
Y – Ag	[Ar]+3d	D	(3s,3p,2d)
Cd	[Kr]	D	(2s,2p,2d)
In – Xe	[Kr]+4d	D	(2s,2p)
Cs – Ba	[Kr]+4d	D	(3s,3p)
La	[Kr]+4d	D	(3s,3p,2d)
Hf – Au	[Kr]+4d+4f	F	(3s,3p,2d)
Hg	[Xe]+4f	F	(2s,2p,2d)
Tl	[Xe]+4f+5d	F	(2s,2p,2d)
Pb – Bi	[Xe]+4f+5d	F	(2s,2p)
U – Pu	[Xe]+4f+5d	F	(3s,3p,2d,2f)

8.7.3 THE SBKJC PSEUDOPOTENTIAL AT A GLANCE

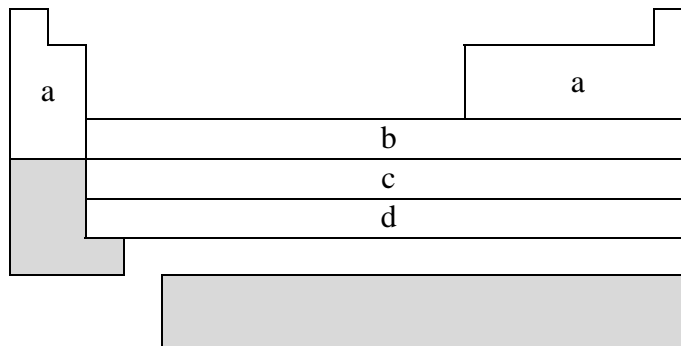


SBKJC is not available for shaded elements

- (a) No pseudopotential; Pople 3–21G basis used
- (b) W.J. Stevens, H. Basch & M. Krauss, *J. Chem. Phys.* 81 (1984) 6026
- (c) W.J. Stevens, M. Krauss, H. Basch & P.G. Jasien, *Can. J. Chem* 70 (1992) 612
- (d) T.R. Cundari & W.J. Stevens, *J. Chem. Phys.* 98 (1993) 5555

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(2s)
Li – Ne	[He]	S	(2s,2p)
Na – Ar	[Ne]	P	(2s,2p)
K – Ca	[Ar]	P	(2s,2p)
Sc – Ga	[Ne]	P	(4s,4p,3d)
Ge – Kr	[Ar]+3d	D	(2s,2p)
Rb – Sr	[Kr]	D	(2s,2p)
Y – In	[Ar]+3d	D	(4s,4p,3d)
Sn – Xe	[Kr]+4d	D	(2s,2p)
Cs – Ba	[Xe]	D	(2s,2p)
La	[Kr]+4d	F	(4s,4p,3d)
Ce – Lu	[Kr]+4d	D	(4s,4p,1d,1f)
Hf – Tl	[Kr]+4d+4f	F	(4s,4p,3d)
Pb – Rn	[Xe]+4f+5d	F	(2s,2p)

8.7.4 THE CRENBNS PSEUDOPOTENTIAL AT A GLANCE

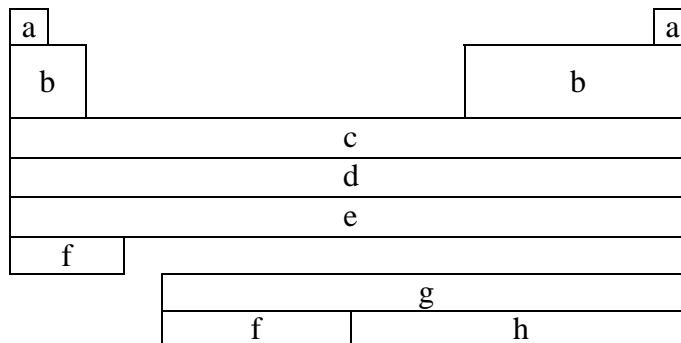


CRENBNS is not available for shaded elements

- (a) No pseudopotential; Pople STO-3G basis used
 (b) Hurley, Pacios, Christiansen, Ross & Ermler, *J. Chem. Phys.* 84 (1986) 6840
 (c) LaJohn, Christiansen, Ross, Atashroo & Ermler, *J. Chem. Phys.* 87 (1987) 2812
 (d) Ross, Powers, Atashroo, Ermler, LaJohn & Christiansen, *J. Chem. Phys.* 93 (1990) 6654

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(1s)
Li – Ne	<i>none</i>	<i>none</i>	(2s,1p)
Na – Ar	<i>none</i>	<i>none</i>	(3s,2p)
K – Ca	<i>none</i>	<i>none</i>	(4s,3p)
Sc – Zn	[Ar]	P	(1s,0p,1d)
Ga – Kr	[Ar]+3d	D	(1s,1p)
Y – Cd	[Kr]	D	(1s,1p,1d)
In – Xe	[Kr]+4d	D	(1s,1p)
La	[Xe]	D	(1s,1p,1d)
Hf – Hg	[Xe]+4f	F	(1s,1p,1d)
Tl – Rn	[Xe]+4f+5d	F	(1s,1p)

8.7.5 THE CRENBL PSEUDOPOTENTIAL AT A GLANCE

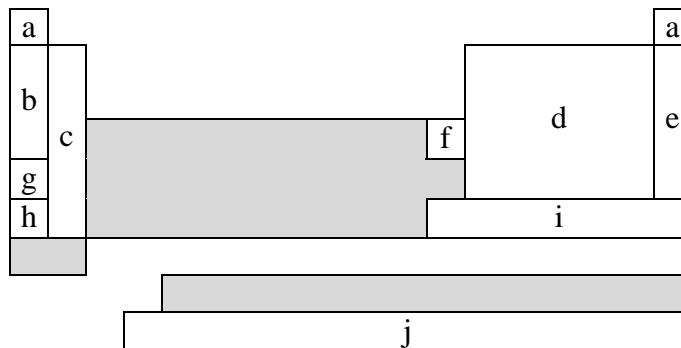


CRENBL is not available for shaded elements

- (a) No pseudopotential; Pople 6–311G* basis used
- (b) Pacios & Christiansen, J. Chem. Phys. 82 (1985) 2664
- (c) Hurley, Pacios, Christiansen, Ross & Ermler, J. Chem. Phys. 84 (1986) 6840
- (d) LaJohn, Christiansen, Ross, Atashroo & Ermler, J. Chem. Phys. 87 (1987) 2812
- (e) Ross, Powers, Atashroo, Ermler, LaJohn & Christiansen, J. Chem. Phys. 93 (1990) 6654
- (f) Ermler, Ross & Christiansen, Int. J. Quantum Chem. 40 (1991) 829
- (g) Ross, Gayen & Ermler, J. Chem. Phys. 100 (1994) 8145
- (h) Nash, Bursten & Ermler, J. Chem. Phys. 106 (1997) 5133

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(3s)
Li – Ne	[He]	S	(4s,4p)
Na – Mg	[He]	S	(6s,4p)
Al – Ar	[Ne]	P	(4s,4p)
K – Ca	[Ne]	P	(5s,4p)
Sc – Zn	[Ne]	P	(7s,6p,6d)
Ga – Kr	[Ar]	P	(3s,3p,4d)
Rb – Sr	[Ar]+3d	D	(5s,5p)
Y – Cd	[Ar]+3d	D	(5s,5p,4d)
In – Xe	[Kr]	D	(3s,3p,4d)
Cs – La	[Kr]+4d	D	(5s,5p,4d)
Ce – Lu	[Xe]	D	(6s,6p,6d,6f)
Hf – Hg	[Kr]+4d+4f	F	(5s,5p,4d)
Tl – Rn	[Xe]+4f	F	(3s,3p,4d)
Fr – Ra	[Xe]+4f+5d	F	(5s,5p,4d)
Ac – Pu	[Xe]+4f+5d	F	(5s,5p,4d,4f)
Am – Lr	[Xe]+4f+5d	F	(0s,2p,6d,5f)

8.7.6 THE SRLC PSEUDOPOTENTIAL AT A GLANCE

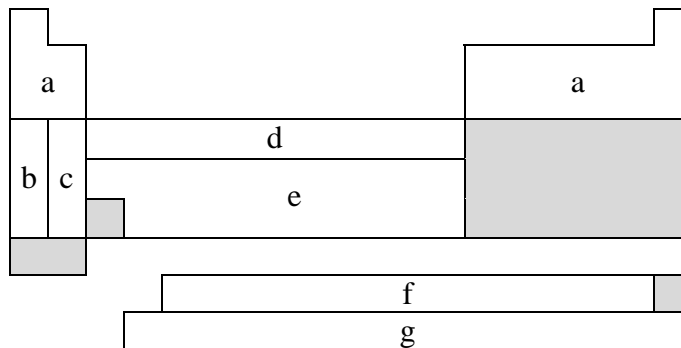


SRLC is not available for shaded elements

- (a) No pseudopotential; Pople 6–31G basis used
- (b) Fuentealba, Preuss, Stoll & Szentpaly, Chem. Phys. Lett. 89 (1982) 418
- (c) Fuentealba, Szentpály, Preuss & Stoll, J. Phys. B 18 (1985) 1287
- (d) Bergner, Dolg, Küchle, Stoll & Preuss, Mol. Phys. 80 (1993) 1431
- (e) Nicklass, Dolg, Stoll & Preuss, J. Chem. Phys. 102 (1995) 8942
- (f) Schautz, Flad & Dolg, Theor. Chem. Acc. 99 (1998) 231
- (g) Fuentealba, Stoll, Szentpaly, Schwerdtfeger & Preuss, J. Phys. B 16 (1983) L323
- (h) Szentpaly, Fuentealba, Preuss & Stoll, Chem. Phys. Lett. 93 (1982) 555
- (i) Küchle, Dolg, Stoll & Preuss, Mol. Phys. 74 (1991) 1245
- (j) Küchle, to be published

Element	Core	Max Projector	Valence
H – He	<i>none</i>	<i>none</i>	(2s)
Li – Be	[He]	P	(2s,2p)
B – N	[He]	D	(2s,2p)
O – F	[He]	D	(2s,3p)
Ne	[He]	D	(4s,4p,3d,1f)
Na – P	[Ne]	D	(2s,2p)
S – Cl	[Ne]	D	(2s,3p)
Ar	[Ne]	F	(4s,4p,3d,1f)
K – Ca	[Ar]	D	(2s,2p)
Zn	[Ar]+3d	D	(3s,2p)
Ga – As	[Ar]+3d	F	(2s,2p)
Se – Br	[Ar]+3d	F	(2s,3p)
Kr	[Ar]+3d	G	(4s,4p,3d,1f)
Rb – Sr	[Kr]	D	(2s,2p)
In – Sb	[Kr]+4d	F	(2s,2p)
Te – I	[Kr]+4d	F	(2s,3p)
Xe	[Kr]+4d	G	(4s,4p,3d,1f)
Cs – Ba	[Xe]	D	(2s,2p)
Hg – Bi	[Xe]+4f+5d	G	(2s,2p,1d)
Po – At	[Xe]+4f+5d	G	(2s,3p,1d)
Rn	[Xe]+4f+5d	G	(2s,2p,1d)
Ac – Lr	[Xe]+4f+5d	G	(5s,5p,4d,3f,2g)

8.7.7 THE SRSC PSEUDOPOTENTIAL AT A GLANCE



SRSC is not available for shaded elements

- (a) No pseudopotential; Pople 6–311G* basis used
 (b) Leininger, Nicklass, Küchle, Stoll, Dolg & Bergner, Chem. Phys. Lett. 255 (1996) 274
 (c) Kaupp, Schleyer, Stoll & Preuss, J. Chem. Phys. 94 (1991) 1360
 (d) Dolg, Wedig, Stoll & Preuss, J. Chem. Phys. 86 (1987) 866
 (e) Andrae, Haeussermann, Dolg, Stoll & Preuss, Theor. Chim. Acta 77 (1990) 123
 (f) Dolg, Stoll & Preuss, J. Chem. Phys. 90 (1989) 1730
 (g) Küchle, Dolg, Stoll & Preuss, J. Chem. Phys. 100 (1994) 7535

Element	Core	Max Projector	Valence
H – Ar	<i>none</i>	<i>none</i>	(3s)
Li – Ne	<i>none</i>	<i>none</i>	(4s,3p,1d)
Na – Ar	<i>none</i>	<i>none</i>	(6s,5p,1d)
K	[Ne]	F	(5s,4p)
Ca	[Ne]	F	(4s,4p,2d)
Sc – Zn	[Ne]	D	(6s,5p,3d)
Rb	[Ar]+3d	F	(5s,4p)
Sr	[Ar]+3d	F	(4s,4p,2d)
Y – Cd	[Ar]+3d	F	(6s,5p,3d)
Cs	[Kr]+4d	F	(5s,4p)
Ba	[Kr]+4d	F	(3s,3p,2d,1f)
Ce – Yb	[Ar]+3d	G	(5s,5p,4d,3f)
Hf – Pt	[Kr]+4d+4f	G	(6s,5p,3d)
Au	[Kr]+4d+4f	F	(7s,3p,4d)
Hg	[Kr]+4d+4f	G	(6s,6p,4d)
Ac – Lr	[Kr]+4d+4f	G	(8s,7p,6d,4f)

8.8 FURTHER READING

Ground State Methods (Chapters 4 and 5)

Basis Sets (Chapter 7)

- [1] P. A. Christiansen, W. C. Ermler and K. S. Pitzer, *Ann. Rev. Phys. Chem.* (1985) **36**, 407
- [2] P. Pyykko, *Chem. Rev.* (1988) **88**, 563.
- [3] M. S. Gordon, and T. R. Cundari, *Coord. Chem. Rev.* (1996) **147**, 87
- [4] See articles by G. Frenking et al, T. R. Cundari et al, and J. Almlof and O. Gropen, in *Reviews in Computational Chemistry*, volume 8, edited by K. B. Lipkowitz and D. B. Boyd (Wiley-VCH, 1996).
- [5] L. R. Kahn and W. A. Goddard III, *J. Chem. Phys.* (1972) **56**, 2685.
- [6] Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version , as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multi-program laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. Contact David Feller or Karen Schuchardt for further information.

CHAPTER 9 MOLECULAR GEOMETRY CRITICAL POINTS

9.1 EQUILIBRIUM GEOMETRIES AND TRANSITION STRUCTURES

Molecular potential energy surfaces rely on the Born-Oppenheimer separation of nuclear and electronic motion. Minima on such energy surfaces correspond to the classical picture of equilibrium geometries and first-order saddle points to transition structures. Both equilibrium and transition structures are stationary points. Gradients of equilibrium and transition structures should vanish; characterisation of the critical point requires consideration of the eigenvalues of the Hessian (second derivative matrix). Equilibrium geometries have Hessians whose eigenvalues are all positive. Transition structures, on the other hand, have Hessians with exactly one negative eigenvalue. That is, a transition structure is a maximum along a reaction path between two local minima, but a minimum in all directions perpendicular to the path.

The quality of a geometry optimization algorithm is of major importance; even the fastest integral code in the world will be useless if combined with an inefficient optimization algorithm that requires excessive numbers of steps to converge. Thus, Q-Chem incorporates the most advanced geometry optimisation features currently available through Jon Baker's OPTIMIZE package (see Appendix), a product of over ten years of research and development.

The key to optimizing a molecular geometry successfully is to proceed from the starting geometry to the final geometry in as few steps as possible. Four factors influence the path and number of steps:

- starting geometry
- optimization algorithm
- quality of the Hessian (and gradient)
- coordinate system

Q-Chem controls the last three of these, but the starting geometry is solely determined by the user, and the closer it is to the converged geometry, the fewer optimization steps will be required. Decisions regarding the optimizing algorithm and the coordinate system are generally made by the OPTIMIZE package to maximise the rate of convergence. Users are able to override these decisions, but in general, this is not recommended.

Another consideration in minimising optimization time concerns gradient and Hessian quality. A higher quality Hessian (*i.e.*, analytical *vs.* approximate) will in many cases lead to faster convergence and hence, fewer optimization steps. However, the construction of an analytical Hessian requires significant computational effort and may

outweigh the advantage of fewer optimization cycles. Currently available analytical gradients and Hessians are summarized in Table 9.1.

Level of Theory/ Algorithm	Analytical Gradients	Maximum Angular Momentum Type	Analytical Hessian	Maximum Angular Momentum Type
DFT	✓	00	✗	
HF	✓	00	✓	f
MP2	✓	00	✗	
(V)OD	✓	00	✗	
(V)QCCD	✓	00	✗	
CIS (except RO)	✓	00	✓	f
CFMM	✓	00	✗	

Table 9.1 Gradients and Hessians currently available for geometry optimizations with maximum angular momentum types for analytical derivative calculations (for higher angular momentum, derivatives are computed numerically)

9.2 USER-CONTROLLABLE PARAMETERS

Note: Users input starting geometry through the *\$molecule* keyword.

9.2.1 FEATURES

- Cartesian, Z-matrix or internal coordinate systems
- Eigenvector Following (EF) or GDIIS algorithms
- Constrained optimizations
- Equilibrium structure searches
- Transition structure searches
- Initial Hessian and Hessian update options

9.2.2 JOB CONTROL

Users must first define what level of theory is required. Refer back to previous sections regarding enhancements and customization of these features. *EXCHANGE*, *CORRELATION* (if required) and *BASIS \$rem* variables must be set.

The remaining *\$rem* variables are those specifically relating to the OPTIMIZE package.

JOBTYPE

Specifies the calculation

VARIABLE:

STRING

OPT Equilibrium structure optimization

TS Transition structure optimization

GEOM_OPT_HESSIAN

Hessian status

VARIABLE:

STRING

DEFAULT:

DIAGONAL

OPTIONS:

DIAGONAL Set up (default) diagonal Hessian.

READ Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

GEOM_OPT_COORDS

Controls the type of optimization coordinates

VARIABLE

INTEGER

DEFAULT:

-1 Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue

OPTIONS:

0 Optimize in Cartesian coordinates

1 Generate and optimize in internal coordinates, if this fails abort

-1 Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue

2 Optimize in Z-matrix coordinates, if this fails abort

-2 Optimize in Z-matrix coordinates, if this fails during any stage of the optimization switch to Cartesians and continue

RECOMMENDATION:

Use the default; delocalized internals are more efficient

GEOM_OPT_TOL_GRADIENT

Convergence on maximum gradient component

VARIABLE:

INTEGER

DEFAULT:

300 $\equiv 300 \times 10^{-6}$ tolerance on maximum gradient component

OPTIONS:

Integer value (tolerance = value $\times 10^{-6}$)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_TOL_DISPLACEMENT

Convergence on maximum atomic displacement

VARIABLE:

INTEGER

DEFAULT:

1200 $\equiv 1200 \times 10^{-6}$ tolerance on maximum atomic displacement

OPTIONS:

Integer value (tolerance = value $\times 10^{-6}$)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_TOL_ENERGY

Convergence on energy change of successive optimisation cycles

VARIABLE:

INTEGER

DEFAULT:

100 $\equiv 100 \times 10^{-8}$ tolerance on maximum gradient component

OPTIONS:

Integer value (tolerance = value $\times 10^{-8}$)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_MAX_CYCLES

Maximum number of optimisation cycles

VARIABLE:

INTEGER

DEFAULT:

20

OPTIONS:

User defined positive integer

RECOMMENDATION:

Use the default, increase for difficult cases

GEOM_OPT_PRINT

Amount of OPTIMIZE print output

VARIABLE:

INTEGER

DEFAULT:

3 Error messages, summary, warning, standard information and gradient print out

OPTIONS:

0 Error messages only

1 Level 0 plus summary and warning print out

2 Level 1 plus standard information

3 Level 2 plus gradient print out

4 Level 3 plus hessian print out

5 Level 4 plus iterative print out

6 Level 5 plus internal generation print out

7 Debug print out

RECOMMENDATION:

Use the default

9.2.3 CUSTOMIZATION**GEOM_OPT_SYMFLAG**

Controls the use of symmetry in OPTIMIZE

VARIABLE:

INTEGER

DEFAULT:

1 Make use of point group symmetry

OPTIONS:

1 Make use of point group symmetry

0 Do not make use of point group symmetry

RECOMMENDATION:

Use default

GEOM_OPT_MODE

Determines Hessian mode followed during TS search

VARIABLE:

INTEGER

DEFAULT:

0 Mode following off

OPTIONS:

0 Mode following off

n Maximise along mode *n*

RECOMMENDATION:

Use default

GEOM_OPT_MAX_DIIS

Controls maximum size of subspace for GDIIS

VARIABLE:

INTEGER

DEFAULT:

0 Do not use GDIIS

OPTIONS:

0 Do not use GDIIS

-1 Default size = min(NDEG, NATOMS, 4) NDEG = number of
molecular degrees of freedom

n Size specified by user

RECOMMENDATION:

Use default or do not set *n* too large

GEOM_OPT_DMAX

Maximum allowed step size. Value supplied is multiplied by 10^{-3}

VARIABLE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

n User-defined cutoff

RECOMMENDATION:

Use default

GEOM_OPT_UPDATE

Controls the Hessian update algorithm

VARIABLE:

INTEGER

DEFAULT:

-1 Use the default update algorithm

OPTIONS:

-1 Use the default update algorithm

0 Do not update the Hessian (not recommended)

1 Murtagh-Sargent update

2 Powell update

3 Powell/Murtagh-Sargent update (TS default)

4 BFGS update (OPT default)

5 BFGS with safeguards to ensure retention of positive definiteness (GDISS default)

RECOMMENDATION:

Use default

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees)

VARIABLE:

INTEGER

DEFAULT:

165 degrees

OPTIONS:

n User-defined level

RECOMMENDATION:

Use default

Comment: Molecular Critical Points Beginning With Analytical Hessian

As outlined, the rate of convergence of the iterative optimization process is dependent on a number of factors, one of which is the use of an initial analytic Hessian. This is easily achieved by instructing Q-Chem to calculate an analytic Hessian and proceed then to determine the required critical point. This is illustrated in the following example.

9.2.4 EXAMPLE

```
$molecule
0 1
O
H 1 OH
H 1 OH 2 HOH

OH = 1.1
HOH = 104
$end
```

```
$rem
JOBTYPE          FREQ          Calculate an analytic Hessian
EXCHANGE         HF
BASIS            6-31G(D)
$end
```

Now proceed with the Optimization making sure to read in the analytic Hessian (use other available information too)

```
@@@
$molecule
READ
$end
```

```
$rem
JOBTYPE          OPT
EXCHANGE         HF
BASIS            6-31G(D)
SCF_GUESS       READ
GEOM_OPT_HESSIAN READ          Have the initial Hessian
$end
```

Example 9.1 Geometry optimization for H₂O starting with an analytic Hessian

9.3 CONSTRAINED OPTIMIZATION**9.3.1 INTRODUCTION**

Constrained optimization refers to the optimization of molecular structures (transition or equilibrium) in which certain parameters (*e.g.*, bond lengths, bond angles or dihedral angles) are fixed. Jon Baker's OPTIMIZE package implemented in the Q-Chem program has been modified to handle constraints directly in delocalized internal coordinates using the method of Lagrange multipliers (see appendix). Constraints are imposed in an *\$opt* keyword section of the input file.

Features of constrained optimizations in Q-Chem are:

- Starting geometries do not have to satisfy imposed constraints
- Delocalized internal coordinates are the most efficient system for large molecules
- Q-Chem's free format *\$opt* section allows the user to apply constraints with ease

Note: The *\$opt* input section is case-insensitive and free-format, except that there should be no space at the start of each line.

9.3.2 GEOMETRY OPTIMIZATION WITH GENERAL CONSTRAINTS

CONSTRAINT and *ENDCONSTRAINT* define the beginning and end, respectively, of the constraint section of *\$opt* within which users may specify up to six different types of constraints:

interatomic distances (angstroms, $value > 0.0$):

```
stre      atom1      atom2      value
```

angles (degrees, $180.0 \geq value \geq 0.0$); *atom2* is the middle atom of the bend:

```
bend      atom1      atom2      atom3      value
```

out-of-plane-bends (degrees, $180.0 \geq value \geq -180.0$); angle between *atom4* and the *atom1-atom2-atom3* plane:

```
outp      atom1      atom2      atom3      atom4      value
```

dihedral angles (degrees, $180.0 \geq value \geq -180.0$); angle the plane *atom1-atom2-atom3* makes with the plane *atom2-atom3-atom4*:

```
tors      atom1      atom2      atom3      atom4      value
```

coplanar bends (degrees, $180.0 \geq value \geq -180.0$); bending of *atom1-atom2-atom3* in the plane *atom2-atom3-atom4*:

```
linc      atom1      atom2      atom3      atom4      value
```

perpendicular bends (degrees, $180.0 \geq value \geq -180.0$); bending of *atom1-atom2-atom3* perpendicular to the plane *atom2-atom3-atom4*:

```
linp      atom1      atom2      atom3      atom4      value
```

9.3.3 FROZEN ATOMS

Absolute atom positions can be frozen with the *FIXED* section. The section starts with the *FIXED* keyword as the first line and ends with the *ENDFIXED* keyword on the last. The format to fix a coordinate or coordinates of an atom is:

```
atom      coordinate_reference
```

coordinate_reference can be any combination of up to three characters *X*, *Y* and *Z* to specify the coordinate(s) to be fixed: *X*, *Y*, *Z*, *XY*, *XZ*, *YZ*, *XYZ*. The fixing characters must be next to each other. *e.g.*,

```
FIXED
2      XY
ENDFIXED
```

means the *x*-coordinate and *y*-coordinate of atom 2 are fixed, whereas

```
FIXED
2      X   Y
ENDFIXED
```

will yield erroneous results.

Note: When the *FIXED* section is specified within *\$opt*, the optimization coordinates will be Cartesian.

9.3.4 DUMMY ATOMS

DUMMY defines the beginning of the dummy atom section and ENDDUMMY its conclusion. Dummy atoms are used to help define constraints during constrained optimizations in Cartesian coordinates. They cannot be used with delocalized internals.

All dummy atoms are defined with reference to a list of real atoms, that is, dummy atom coordinates are generated from the coordinates of the real atoms from the dummy atoms defining list (see below). There are three types of dummy atom:

1. Positioned at the arithmetic mean of the up to 7 real atoms in the defining list
2. Positioned a unit distance along the normal to a plane defined by three atoms, centred on the middle atom of the three
3. Positioned a unit distance along the bisector of a given angle

The format for declaring dummy atoms is:

```
DUMMY
idum      type      list_length  defining_list
ENDDUMMY
```

idum center number of defining atom (must be one greater than the total number of real atoms for the first dummy atom, two greater for second *etc.*)
type type of dummy atom (either 1, 2 or 3; see above)
list_length number of atoms in the defining list
defining_list list of up to 7 atoms defining the position of the dummy atom

Once defined, dummy atoms can be used to define standard internal (distance, angle) constraints as per the constraints section, above.

Warning: The use of dummy atoms of type 1 has never progressed beyond the experimental stage.

9.3.5 DUMMY ATOM PLACEMENT IN DIHEDRAL CONSTRAINTS

Bond and dihedral angles cannot be constrained in Cartesian optimizations to exactly 0° or $\pm 180^\circ$. This is because the corresponding constraint normals are zero vectors. Also, dihedral constraints near these two limiting values (within, say 20°) tend to oscillate and are difficult to converge.

These difficulties can be overcome by defining dummy atoms and redefining the constraints with respect to the dummy atoms. For example, a dihedral constraint of 180° can be redefined to two constraints of 90° with respect to a suitably positioned dummy atom. The same thing can be done with a 180° bond angle (long a familiar use in *Z*-matrix construction).

Typical usage is as follows:

Internal Coordinates	Cartesian Coordinates
\$opt	\$opt
CONSTRAINT	DUMMY
tors I J K L 180.0	M 2 I J K
ENDCONSTRAINT	ENDDUMMY
\$end	CONSTRAINT
	tors I J K M 90
	tors M J K L 90
	ENDCONSTRAINT
	\$end

Table 9.2 Comparison of dihedral angle constraint method for adopted coordinates.

The order of atoms is important to obtain the correct signature on the dihedral angles. For a 0° dihedral constraint, J and K should be switched in the definition of the second torsion constraint in Cartesian coordinates.

Note: In almost all cases the above discussion is somewhat academic, as internal coordinates are now best imposed using delocalized internal coordinates and there is no restriction on the constraint values.

9.3.6 ADDITIONAL ATOM CONNECTIVITY

Normally delocalized internal coordinates are generated automatically from the input Cartesian coordinates. This is accomplished by first determining the atomic connectivity list (*i.e.*, which atoms are formally bonded) and then constructing a set of individual primitive internal coordinates comprising all bond stretches, all planar bends and all proper torsions that can be generated based on the atomic connectivity. The delocalized internal are in turn constructed from this set of primitives.

The atomic connectivity depends simply on distance and there are default bond lengths between all pairs of atoms in the code. In order for delocalized internals to be generated successfully, all atoms in the molecule must be formally bonded so as to form a closed system. In molecular complexes with long, weak bonds or in certain transition states where parts of the molecule are rearranging or dissociating, distances between atoms may be too great for the atoms to be regarded as formally bonded, and the standard atomic connectivity will separate the system into two or more distinct parts. In this event, the generation of delocalized internal coordinates will fail.

Additional atomic connectivity can be included for the system to overcome this difficulty.

CONNECT defines the beginning of the additional connectivity section and *ENDCONNECT* the end. The format of the *CONNECT* section is:

```
CONNECT
atom      list_length  list
ENDCONNECT
```

atom atom for which additional connectivity is being defined
list_length number of atoms in the list of bonded atoms
list list of up to 8 atoms considered as being bonded to the given atom

9.3.7 EXAMPLE

```
$comment
methanol geom opt with constraints in bond length and bond
angles.
$end
```

```
$molecule
0 1
C      0.141915      0.332682      0.000000
O      0.141915     -1.088318      0.000000
H      1.186989      0.656186      0.000000
H     -0.348433      0.742676      0.887862
H     -0.348433      0.742676     -0.887862
H     -0.773953     -1.385902      0.000000
$end
```

```
$rem
GEOM_OPT_PRINT      6
GEOM_OPT_COORDS    2
JOBTYPE             OPT
EXCHANGE            HF
BASIS               3-21G      Basis Set
$end
```

```
$opt
CONSTRAINT
stre 1 6 1.8
bend 2 1 4 110.0
bend 2 1 5 110.0
ENDCONSTRAINT
$end
```

Example 9.2 Methanol geometry optimization with constraints.

9.3.8 SUMMARY

```

$opt
CONSTRAINT
stre      atom1      atom2      value
...
bend      atom1      atom2      atom3      value
...
outp      atom1      atom2      atom3      atom4      value
...
tors      atom1      atom2      atom3      atom4      value
...
linc      atom1      atom2      atom3      atom4      value
...
linp      atom1      atom2      atom3      atom4      value
...
ENDCONSTRAINT
FIXED
atom      coordinate_reference
...
ENDFIXED
DUMMY
idum      type      list_length      defining_list
...
ENDDUMMY
CONNECT
atom      list_length      list
...
ENDCONNECT
$end

```

9.4 FURTHER READING

Appendix A on geometry optimization.

CHAPTER 10 MOLECULAR PROPERTIES AND ANALYSIS

10.1 INTRODUCTION

Q-Chem has incorporated a number of molecular properties and wavefunction analysis tools, summarised as follows:

- Chemical solvent models
- Population analysis
- Vibrational analysis
- Interface to the Natural Bond Orbital package
- Molecular orbital symmetries
- Multipole moments
- Data generation for 2-d or 3-d plots

10.2 CHEMICAL SOLVENT MODELS

Ab initio quantum chemical programs enable the accurate study of large molecules properties in the gas phase. However, some of these properties change significantly in solution. The largest changes are expected when going from vapour to polar solutions. Although in principle it is possible to model solvation effects upon the solute properties by supermolecular (cluster) calculations (*e.g.*, by averaging over several possible configurations of the first solvation shell), these calculations are very demanding. Furthermore, the supermolecular calculations cannot, at present, provide accurate and stable *hydration energies*, for which long-range effects are very important. An accurate prediction of the hydration free energies is necessary for computer modelling of chemical reactions and ligand-receptor interactions in aqueous solution. Q-Chem contains two solvent models, which differ greatly in their level of sophistication and realism. The crude and simple Onsager model is described first, followed by the much more advanced Langevin dipoles model developed by Jan Florian and Arieh Warshel of the University of Southern California.

10.2.1 ONSAGER DIPOLE CONTINUUM SOLVENT

Q-Chem offers a solvent model based on that originally attributed to Onsager [1] in which the solute is placed in a spherical cavity surrounded by a continuous medium. The Onsager model requires two parameters: the cavity radius a_0 and a dielectric constant ϵ . Typically, the cavity radius is calculated using

$$a_0^3 = 3V_m/4\pi N \quad (10.1)$$

where V_m is obtained from experiment (molecular weight/density) [2] and N is Avogadro's number. It is also common to add 0.5 Å to the value of a_0 from (10.1) to account for the first solvation shell [3].

See the review by Tomasi and Perisco [4] for further insights into continuum solvent models.

The $\$rem$ variables associated with running Onsager reaction field calculations are documented below. Q-Chem requires at least single point energy calculation $\$rem$ variables *BASIS*, *EXCHANGE* and *CORRELATION* (if required) in addition to the Onsager specific variables *SOLUTE_RADIUS* and *SOLVENT_DIELECTRIC*.

SOLUTE_RADIUS

Sets the Onsager solvent model cavity radius

VARIABLE:

INTEGER $a_0 = \text{SOLUTE_RADIUS}/10000$

DEFAULT:

No default

OPTIONS:

User-defined

RECOMMENDATION:

Use equation (10.1)

SOLVENT_DIELECTRIC

Sets the dielectric constant of the Onsager solvent continuum

VARIABLE:

INTEGER $\epsilon = \text{SOLVENT_DIELECTRIC}/10000$

DEFAULT:

No default

OPTIONS:

User-defined

RECOMMENDATION:

As per required solvent

10.2.2 LANGEVIN DIPOLES SOLVATION MODEL

Q-Chem provides the option to calculate molecular properties in aqueous solution and the magnitudes of the hydration free energies by the Langevin dipoles (LD) solvation model developed by Jan Florián and Arieh Warshel [5,6], of the University of Southern California. In this model, a solute molecule is surrounded by a sphere of point dipoles, with centers on a cubic lattice. Each of these dipoles (called Langevin dipoles) changes its size and orientation in the electrostatic field of the solute and the other Langevin dipoles. The electrostatic field from the solute is determined rigorously by the integration of its charge density, whereas for dipole-dipole interactions, a 12 Å cutoff is used. The QChem/ChemSol 1.0 implementation of the LD model is fully self-consistent

in that the molecular quantum mechanical calculation takes into account solute-solvent interactions. Further details on the implementation and parametrization of this model can be found in the original literature [5,7].

The results of ChemSol calculations are printed in the standard output file. Below is a part of the output for a calculation on the methoxide anion (corresponding to the sample input given later on, and the sample file in the *\$QC/samples* directory).

```
Iterative Langevin Dipoles (ILD) Results (kcal/mol):  
-----  
LD Electrostatic energy                -86.14  
Hydrophobic energy                     0.28  
van der Waals energy (VdW)             -1.95  
Bulk correction                        -10.07  
Solvation free energy dG(ILD)          -97.87
```

The total hydration free energy, $\Delta G(\text{ILD})$ is calculated as a sum of several contributions. Note that the electrostatic part of ΔG is calculated by using the linear-response approximation [5] and contains contributions from the *polarization* of the solute charge distribution due to its interaction with the solvent. This results from the self-consistent implementation of the Langevin dipoles model within Q-Chem.

In order for an LD calculation to be carried out by the ChemSol program within Q-Chem, the user must specify a single-point HF or DFT calculation (i.e. at least REM variables *BASIS*, *EXCHANGE* and *CORRELATION*) in addition to setting *CHEMSOL* REM variable to 1 in the *\$rem* keyword section.

CHEMSOL

Controls the use of ChemSol in Q-Chem

VARIABLE:

INTEGER

DEFAULT:

0 Do not use ChemSol

OPTIONS:

1 Perform a ChemSol calculation

CHEMSOL_EFIELD

Determines how the solute charge distribution is approximated in evaluating the electrostatic field of the solute.

VARIABLE:

INTEGER

DEFAULT:

1 Exact solute charge distribution is used.

OPTION:

0 Solute charge distribution is approximated by Mulliken atomic charges. This is a faster, but less rigorous procedure.

CHEMSOL_NN

Sets the number of grids used to calculate the average hydration free energy.

VARIABLE:

INTEGER

DEFAULT:

5 ΔG_{hydr} will be averaged over 5 different grids

OPTIONS:

n number of different grids (Max = 20).

10.2.3 CUSTOMIZING LANGEVIN DIPOLES SOLVATION CALCULATIONS

Accurate calculations of hydration free energies require a judicious choice of the solute-solvent boundary in terms of atom-type dependent parameters. The default atomic van der Waals radii available in Q-Chem were chosen to provide reasonable hydration free energies for most solutes and basis sets. These parameters basically coincide with the ChemSol 2.0 radii given in reference [7]. The only difference between the Q-Chem and ChemSol 2.0 atomic radii stems from the fact that Q-Chem parameter set uses hybridization independent radii for carbon and oxygen atoms.

User-defined atomic radii can be specified in the *\$van_der_waals* section of the input file after setting *READ_VDW REM* variable to *true*. Two different (mutually exclusive) formats can be used, as shown below (Table 10.1). The purpose of format 2 is to permit the user to customize the radius of specific atoms, rather than simply by atomic numbers as in format 1. The radii of atoms that are not listed in the *\$van_der_waals* input will be assigned default values. The atomic radii that were used in the calculation are printed in the ChemSol part of the output file in the column denoted *rp*.

```
$van_der_waals
1
atomic number           VdW-radius (Å)
...
$end

$van_der_waals
2
sequential atom number  VdW-radius (Å)
...
$end
```

Figure 10.1 The two different formats available for the input of user-defined atomic radii for ChemSol calculations in Q-Chem. The first format associates input radii with atomic numbers. The second format associates input radii with individual atoms, in the order they appear in the *\$molecule* section.

10.2.4 EXAMPLE

```
$molecule
-1 1
C .0000 .0000 -.5274
O .0000 .0000 .7831
H .0000 1.0140 -1.0335
H .8782 -.5070 -1.0335
H -.8782 -.5070 -1.0335
$end

$rem
EXCHANGE HF
BASIS 6-31G Basis Set
SCF_CONVERGENCE 6
CHEMSOL 1
READ_VDW true
$end

$van_der_waals
2
1 2.5
$end
```

Example 10.1 A Langevin dipoles calculation on the methoxide anion. A customized value is specified for the radius of the C atom.

10.3 WAVEFUNCTION ANALYSIS

Q-Chem performs a number of standard wavefunction analyses by default. Switching the \$rem variable *WAVEFUNCTION_ANALYSIS* to *FALSE* will prevent the calculation of all wavefunction analysis features (described in this section). Alternatively, each wavefunction analysis feature may be controlled by its \$rem variable. (The NBO package which is interfaced with Q-Chem is capable of performing more sophisticated analyses. See later in this chapter and the NBO manual for details).

WAVEFUNCTION_ANALYSIS

Controls the running of the default wavefunction analysis tasks

VARIABLE:

LOGICAL

DEFAULT:

TRUE

Perform default wavefunction analysis

OPTIONS:

TRUE

Perform default wavefunction analysis

FALSE

Do not perform default wavefunction analysis

Note: *WAVEFUNCTION_ANALYSIS* has no effect on NBO, solvent models or vibrational analyses.

10.3.1 POPULATION ANALYSIS

The one-electron charge density, usually written as

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu} \phi_{\mu}(\mathbf{r}) \phi_{\nu}(\mathbf{r}) \quad (10.2)$$

represents the probability of finding an electron at the point \mathbf{r} , but implies little regarding the number of electrons associated with a given nucleus in a molecule. However, since the number of electrons N is related to the occupied orbitals ψ_i by

$$N = 2 \sum_a^{N/2} |\psi_a(\mathbf{r})|^2 d\mathbf{r} \quad (10.3)$$

we can substitute the basis expansion of ψ_a into (8.3) and obtain

$$N = \sum_{\mu} \sum_{\nu} P_{\mu\nu} S_{\mu\nu} = \sum_{\mu} (\mathbf{PS})_{\mu\mu} = tr\mathbf{PS} \quad (10.4)$$

where we interpret $(\mathbf{PS})_{\mu\mu}$ as the number of electrons associated with ϕ_{μ} . If the basis functions are atom-centred, the number of electrons associated with a given atom can be obtained by summing over all the basis functions. This leads to the Mulliken formula for the net charge of the atom A

$$q_A = Z_A - \sum_{\mu \in A} (\mathbf{PS})_{\mu\mu} \quad (10.5)$$

where Z_A is the atom's nuclear charge. This is called a Mulliken population analysis [8]. Q-Chem performs a Mulliken population analysis by default.

POP_MULLIKEN

Controls running of Mulliken population analysis

VARIABLE:

LOGICAL/INTEGER

DEFAULT:

TRUE (1) Calculate Mulliken population

OPTIONS:

FALSE (0) Do not calculate Mulliken Population

TRUE (1) Calculate Mulliken population

2 Also calculate shell populations for each occupied orbital.

RECOMMENDATIONS:

TRUE. Trivial additional calculation

10.3.2 MULTIPOLE MOMENTS

Q-Chem can compute Cartesian multipole moments of the charge density to arbitrary order.

MULTIPOLE_ORDER

Determines highest order of multipole moments to print if wavefunction analysis requested

VARIABLE:

INTEGER

DEFAULT:

4

OPTIONS:

n Calculate moments to n^{th} order

10.3.3 SYMMETRY DECOMPOSITION

Q-Chem's default is to write the SCF wave function molecular orbital symmetries and energies to the output file. If requested, a symmetry decomposition of the kinetic and nuclear attraction energies can also be calculated.

SYMMETRY_DECOMPOSITION

Determines symmetry decompositions to calculate

VARIABLE:

INTEGER

DEFAULT:

1 Calculate MO eigenvalues and symmetry (if available)

OPTIONS:

0 No symmetry decomposition

1 Calculate MO eigenvalues and symmetry (if available)

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices

10.4 VIBRATIONAL ANALYSIS

Vibrational analysis is an extremely important tool for the quantum chemist, supplying a molecular fingerprint which is invaluable for aiding identification of molecular species in many experimental studies. Q-Chem includes a vibrational analysis package that can calculate vibrational frequencies and their Raman [9] and infrared activities. Vibrational frequencies are calculated by either using an analytic Hessian (if available, Table 9.1) or, numerical finite difference of the gradient. The default setting in Q-Chem is to use the highest analytical derivative order available for the requested theoretical method.

Following a vibrational analysis, Q-Chem computes useful statistical thermodynamic properties at standard temperature and pressure, including: zero-point vibration energy (ZPVE) and, translational, rotational and vibrational, entropies and enthalpies.

The performance of various *ab initio* theories in determining vibrational frequencies has been well documented. See references [10,11,12].

10.4.1 JOB CONTROL

In order to carry out a frequency analysis users must *at a minimum* provide a molecule within the *\$molecule* keyword and define an appropriate level of theory within the *\$rem* keyword using the *\$rem* variables *EXCHANGE*, *CORRELATION* (if required) (Chapter 4) and *BASIS* (Chapter 6). Since the default type of job (*JOBTYPE*) is a single point energy (*SP*) calculation, the *JOBTYPE \$rem* variable must be set to *FREQ*.

It is very important to note that a vibrational frequency analysis must be performed at a stationary point on the potential surface that has been optimized at the same level of theory. Therefore a vibrational frequency analysis most naturally follows a geometry

optimization in the same input deck, where the molecular geometry is obtained (see examples).

JOBTYPE

Specifies the calculation.

VARIABLE:

STRING

OPTION:

FREQ Frequency Calculation

The standard output from a frequency analysis includes the following. At the time of writing, isotopic substitution is not yet available.

- Vibrational frequencies
- Raman and IR activities and intensities (requires *\$rem DORAMAN*)
- Atomic masses
- Zero-point vibrational energy
- Translational, rotational, and vibrational, entropies and enthalpies

Several other *\$rem* variables are available that control the vibrational frequency analysis. In detail, they are:

DORAMAN

Controls calculation of Raman intensities. Requires *JOBTYPE* to be set to *FREQ*

VARIABLE:

LOGICAL

DEFAULT:

FALSE Do not calculate Raman intensities

OPTIONS:

FALSE Do not calculate Raman intensities

TRUE Do calculate Raman intensities

VIBMAN_PRINT

Controls level of extra print out for vibrational analysis

VARIABLE:

INTEGER

DEFAULT:

1 Standard full information print out

OPTIONS:

1 Standard full information print out

3 Level 1 plus vibrational frequencies in atomic units

4 Level 3 plus mass-weighted Hessian matrix, projected mass-weighted Hessian matrix

6 Level 4 plus vectors for translations and rotations projection matrix

RECOMMENDATION:

Use default

10.4.2 EXAMPLE

```
$molecule
  0  1
  O
  C, 1, CO
  F, 2, FC, 1, FCO
  H, 2, HC, 1, HCP, 3, 180.0

  CO=1.2
  FC=1.4
  HC=1.0
  FCO=120.0
  HCO=120.0
$end

$rem
  jobtype      = opt
  exchange     = edf1
  basis        = 6-31+G*
$end

@@@

$molecule
  READ
$end

$rem
  jobtype      = freq
  exchange     = edf1
  basis        = 6-31+G*
$end
```

Example 10.2 An EDF1/6-31+G* optimization, followed by a vibrational analysis. Doing the vibrational analysis at a stationary point is necessary for the results to be valid.

10.5 INTERFACE TO THE NBO PACKAGE

Q-Chem has incorporated the Natural Bond Orbital package (v4.0) for molecular properties and wavefunction analysis. The NBO package is invoked either by setting the *\$rem* variable *NBO* to *TRUE* and is initiated after the SCF wavefunction is obtained. Users are referred to the NBO users manual for options and details relating to exploitation of the features offered in this package.

10.5.1 JOB CONTROL

If switched on for a geometry optimization, the NBO package will only be invoked at the end of the last optimization step.

NBO

Controls the use of the NBO package

VARIABLE:

LOGICAL

DEFAULT:

FALSE Do not invoke the NBO package

OPTIONS:

FALSE Do not invoke the NBO package

TRUE Do invoke the NBO package

```
$nbo
{NBO program keywords, parameters and options}
$end
```

Figure 10.2 General format for requesting the NBO program from Q-Chem.

Notes: (1) *\$rem* variable *NBO* must be set to *TRUE* before the *\$nbo* keyword is recognized.
(2) Q-Chem does not currently support facets of the NBO package which require multiple job runs.

10.6 PLOTTING DENSITIES AND ORBITALS

The best way to visualize the charge densities and molecular orbitals that Q-Chem evaluates is with an integrated graphical user interface. Alternatively, Q-Chem can evaluate a range of densities and orbitals on a user-specified grid of points by invoking the *\$plots* option, which is itself enabled by requesting *IANLTY* = 200.

The format of the *\$plots* input is documented below. It permits molecular orbitals to be plotted. Additionally the SCF ground density can be plotted, as well as excited state densities (at either the CIS, RPA or TDDFT/TDA or TDDFT). Also in connection with excited states, either transition densities or attachment-detachment densities (at the same levels of theory given above) can be plotted as well.

The output from the *\$plots* command is one (or several) ASCII files in the working directory, named plots.mo, etc. The results then must be visualized with a third party program capable of making 3-d plots.

An example of the use of the *\$plots* option is the following input deck:

```
$molecule
  0 1
  H 0.0 0.0 0.35
  H 0.0 0.0 -0.35
$end

$rem
  exchange      = hf
  basis         = 6-31g**
  ianlty       = 200
$end

$plots
  plot the HOMO and the LUMO on a line
  1 0.0 0.0
  1 0.0 0.0
  15 -3.0 3.0
  2 0 0 0
  1 2
$end
```

Example 10.3 A job that evaluates the H₂ HOMO and LUMO on a 1 by 1 by 15 grid, along the bond axis. The plotting output is in an ASCII file called plot.mo, which lists for each grid point, x, y, z, and the value of each requested MO.

```

$plots
• One comment line
    My comment here...
• Specification of the 3-d mesh of points on 3 lines:
    Nx  xmin  xmax  (# x points, x range in Angstroms)
    Ny  ymin  ymax
    Nz  zmin  zmax
• A line with 4 integers indicating how many things to plot:
    NMO  NRho  NTrans  NDA
• An optional line with the integer list of MO's to evaluate (only if NMO > 0)
    MO(1)  MO(2)  ...  MO(NMO)
• An optional line with the integer list of densities to evaluate (only if NRho > 0)
    Rho(1)  Rho(2)  ...  Rho(NRho)
• An optional line with the integer list of transition densities (only if Ntrans > 0)
    Trans(1)  Trans(2)  ...  Trans(NTrans)
• An optional line with states for detachment/attachment densities (only if NDA > 0)
    DA(1)  DA(2)  ...  DA(NDA)
$end

```

Figure 10.3 General format for the *\$plots* section of the Q-Chem input deck.

Line 1 of the *\$plots* keyword section is reserved for comments. Lines 2-4 list the number of one dimension points and the range of the grid (note that coordinate ranges are in Angstroms, while all output is in atomic units). Line 5 must contain 4 non-negative integers indicating the number of: molecular orbitals (*Nmo*), electron densities (*Nrho*), transition densities and attach/detach densities (*Nda*), to have mesh values calculated.

The final lines specify which MOs, electron densities, transition densities and CIS attach/detach states are to be plotted (the line can be left blank, or removed, if the number of items to plot is zero). Molecular orbitals are numbered 1 ... $N\alpha$, $N\alpha + 1$... $N\alpha + N\beta$; electron densities numbered where 1 = ground state, 2 = first excited state, 3 = second excited state, *etc.*; and attach/detach specified from state 1 → Nda .

All output data is printed to files in the working directory, overwriting any existing file of the same name. Molecular orbital data is printed to a file called “*plot.mo*”; densities to “*plots.hf*”; restricted unrelaxed attachment/detachment analysis to “*plot.attach.alpha*” and “*plot.detach.alpha*”; unrestricted unrelaxed attachment/detachment analysis to “*plot.attach.alpha*”, “*plot.detach.alpha*”, “*plot.attach.beta*” and “*plot.detach.beta*”; restricted relaxed attachment/detachment analysis to “*plot.attach.rlx.alpha*” and “*plot.detach.rlx.alpha*”; unrestricted relaxed attachment/detachment analysis to “*plot.attach.rlx.alpha*”, “*plot.detach.rlx.alpha*”, “*plot.attach.rlx.beta*” and “*plot.detach.rlx.beta*”. Output is printed in atomic units - coordinates first followed by item value.

```
x1    y1    z1    a1    a2    . . . .  aN
x2    y1    z1    b1    b2    . . . .  bN
. . .
```

Figure 10.4 File output format for all raw plotting data.

10.7 ELECTROSTATIC POTENTIALS

Q-Chem can evaluate electrostatic potentials on a grid of points. Electrostatic potential evaluation is controlled by the *\$rem* variable *IGDESP*, as documented below:

IGDESP

Controls evaluation of the electrostatic potential on a grid of points. If enabled, the output is in an ASCII file, *plot.esp*, in the format *x,y,z,esp* for each point.

VARIABLE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1 read grid input via the *\$plots* section of the input deck
 +*n* read *n* grid points from the ASCII file *ESPGrid*

The following example illustrates the evaluation of electrostatic potentials on a grid:

```
$molecule
  0 1
  H 0.0 0.0 0.35
  H 0.0 0.0 -0.35
$end

$rem
  exchange = hf
  basis = 6-31g**
  ianlty = 200
$end

$plots
  plot the HOMO and the LUMO on a line
  1 0.0 0.0
  1 0.0 0.0
  15 -3.0 3.0
  2 0 0 0
  1 2
$end
```

Example 10.4 A job that evaluates the electrostatic potential for H₂ on a 1 by 1 by 15 grid, along the bond axis. The output is in an ASCII file called *plot.esp*, which lists for each grid point, *x*, *y*, *z*, and the electrostatic potential.

10.8 FURTHER READING

Ground State Methods (Chapters 4 and 5)

Excited State Methods (Chapter 6; particularly attachment-detachment analysis).

Basis Sets (Chapter 7)

NBO manual

- [1] L. Onsager, *J. Am. Chem. Soc* (1936), **58**, 1486.
- [2] R. C. Weast, *Handbook of chemistry and physics*; 70 ed.; R. C. Weast, Ed.; CRC Press: Boca Raton, FL, 1989.
- [3] M. W. Wong, M. J. Frisch and K. B. Wiberg, *J. Am. Chem. Soc* (1991), **113**, 4776.
- [4] J. Tomasi and M. Perisco, *Chem. Rev.* (1994), **94**, 2027.
- [5] J. Florián, and A. Warshel, *J. Phys. Chem. B* (1997), **101**, 5583.
- [6] J. Florián, and A. Warshel: *ChemSol 1.0*, University of Southern California, Los Angeles, 1997.
- [7] J. Florián, and A. Warshel, *J. Phys. Chem.* (1999), **103**, 10282.
- [8] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction To Advanced Electronic Structure Theory*; (Dover, New York, 1998).
- [9] B. G. Johnson and J. Florián, *Chem. Phys. Lett.* (1995), **247**, 120.
- [10] C. W. Murray, G. J. Laming, N. C. Handy and R. D. Amos, *Chem. Phys. Lett.* (1992), **199**, 551.
- [11] A. P. Scott and L. Radom, *J. Phys. Chem.* (1996), **100**, 16502.
- [12] B. G. Johnson, P. M. W. Gill and J. A. Pople, *J. Chem. Phys.* (1993), **98**, 5612.

CHAPTER 11 EXTENDED CUSTOMIZATION

11.1 USER-DEPENDENT AND MACHINE-DEPENDENT CUSTOMIZATION

Q-Chem has developed a simple mechanism for users to set user-defined long-term defaults to override the built-in program defaults. Such defaults may be most suited to machine specific features such as memory allocation, as the total available memory will vary from machine to machine depending on specific hardware and accounting configurations. However, users may identify other important uses for this customization feature.

Q-Chem obtains input initialization variables from four sources:

1. user input file
2. $\$HOME/.qchemrc$ file
3. $\$QC/config/preferences$ file
4. program defaults

The order of preference of initialisation is summarised in the diagram contained in Figure 11.1, where the higher placed input mechanism overrides the lower.

Details of the requirements of the Q-Chem input file have been discussed in detail in this manual and in addition, many of the various program defaults which have been set by Q-Chem. However, in reviewing the variables and defaults, users may identify $\$rem$ variable defaults that they find too limiting or, variables which they find repeatedly need to be set within their input files for maximum exploitation of Q-Chem's features. Rather than continually having to remember to place such variables into the Q-Chem input file, users are able to set long-term defaults which are read each time the user runs a Q-Chem job. This is done by placing these defaults into the file $.qchemrc$ stored in the users home directory. Additionally, system administrators can override Q-Chem defaults with an additional *preferences* file in the $\$QC/config$ directory achieving a hierarchy of input as illustrated in figure 11.1.

Note: The $.qchemrc$ and *preferences* files are not requisites for running Q-Chem and currently only support $\$rem$ keywords.

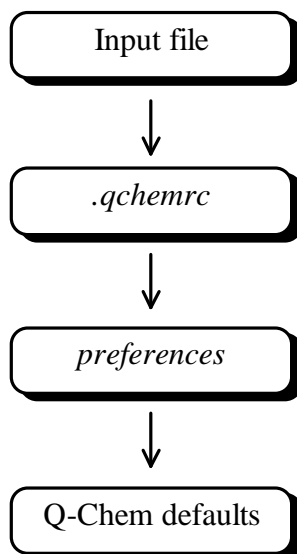


Figure 11.1 Diagram of input initialisation override settings. The higher mechanism indicates override preference of lower mechanisms of input.

11.1.1 .QCHEMRC AND PREFERENCES FILE FORMAT

The format of the *.qchemrc* and *preferences* files is similar to that for the input file, except that only a *\$rem* keyword section may be entered, terminated with the usual *\$end* keyword. Any other keyword sections will be ignored.

It is important that the *.qchemrc* and *preferences* files have appropriate file permissions so that they are readable by the user invoking Q-Chem.

```
$rem
rem_variable          option  comment
rem_variable          option  comment
...
$end
```

Figure 11.2 Format of the *.qchemrc* and *preferences* files

```
$rem
INCORE_INTS_BUFFER   4000000  More integrals in memory
DIIS_SUBSPACE_SIZE   5         Modify max DIIS subspace size
THRESH               10
$end
```

Example 11.1 An example of a *.qchemrc* file to apply program default override *\$rem* settings to all of the user's Q-Chem jobs.

11.1.2 RECOMMENDATIONS

As mentioned, the customization files are specifically suited for placing long-term machine specific defaults, as clearly some of the defaults placed by Q-Chem will not be optimal on large or very small machines. The following *\$rem* variables are examples of those which should be considered, but the user is free to include as few or as many as desired (*CD_DISK*, *INCORE_INTS_BUFFER*, *MEMORY*, *SCF_CONVERGENCE*, *THRESH*, *NBO*).

Q-Chem will print a warning message to advise the user if a *\$rem* keyword section has been detected in either *.qchemrc* or *preferences*.

11.2 Q-CHEM AUXILIARY FILES (\$QCAUX)

The *\$QCAUX* environment variable determines the directory where Q-Chem searches for data files and the machine license. This directory defaults to *\$QC/aux*. Presently, the *\$QCAUX* contains four subdirectories: *atoms*, *basis*, *drivers* and *license*. The *atoms* directory contains data used for the SAD (Chapter 4) SCF density guess; *basis* contains the exponents and contraction coefficients for the standard basis sets available in Q-Chem (Chapter 7); *drivers* contains important information for Q-Chem's AOINTS package and the *license* directory contains the user license. By setting the *\$QCAUX* variable, the *aux* directory may be moved to a separate location from the rest of the program, *e.g.*, to save disk space. Alternatively, one may place a soft link in *\$QC* to the actual *aux* directory.

Users should not alter any files or directories within *\$QCAUX* unless directed by Q-Chem, Inc.

11.3 ADDITIONAL Q-CHEM OUTPUT

The following features are under development and users are advised that those presented, and the format requirements to invoke them, are subject to change in future releases.

11.3.1 THIRD PARTY FCHK FILE

Q-Chem can be instructed to output a third party "fchk" file, "*Test.FChk*", to the working directory by setting the *\$rem* variable *GUI* to 2. Please note that for future releases of Q-Chem this feature, and the method used to invoke it, is subject to change.

APPENDIX A GEOMETRY OPTIMIZATION WITH Q-CHEM

A.1 INTRODUCTION

Geometry optimization refers to the determination of stationary points, principally minima and transition states, on molecular potential energy surfaces. It is an iterative process, requiring the repeated calculation of energies, gradients and (possibly) Hessians at each optimization cycle until convergence is attained. The optimization step involves modifying the current geometry, utilizing current and previous energy, gradient and Hessian information to produce a revised geometry which is closer to the target stationary point than its predecessor was. The art of geometry optimization lies in calculating the step \mathbf{h} , the displacement from the starting geometry on that cycle, so as to converge in as few cycles as possible.

There are four main factors that influence the rate of convergence. These are:

1. initial starting geometry
2. algorithm used to determine the step \mathbf{h}
3. quality of the Hessian (second derivative) matrix
4. coordinate system chosen

The first of these factors is obvious: the closer the initial geometry is to the final converged geometry the fewer optimization cycles it will take to reach it. The second factor is again obvious: if a poor step \mathbf{h} is predicted, this will obviously slow down the rate of convergence. The third factor is related to the second: the best algorithms make use of second derivative (curvature) information in calculating \mathbf{h} , and the better this information is, the better will be the predicted step. The importance of the fourth factor (the coordinate system) has only been generally appreciated relatively recently: a good choice of coordinates can enhance the convergence rate by an order of magnitude (a factor of 10) or more, depending on the molecule being optimized.

Q-Chem includes a powerful suite of algorithms for geometry optimization written by Jon Baker and known collectively as OPTIMIZE. These algorithms have been developed and perfected over the past ten years and the code is robust and has been well tested. OPTIMIZE is a general geometry optimization package for locating both minima and transition states. It can optimize using Cartesian, Z-matrix coordinates or delocalized internal coordinates. The last of these are generated automatically from the Cartesian coordinates and are often found to be particularly effective. It also handles fixed constraints on distances, angles, torsions and out-of-plane bends, between any atoms in the molecule, whether or not the desired constraint is satisfied in the starting geometry. Finally it can freeze atomic positions, or any X, Y, Z Cartesian atomic coordinates.

OPTIMIZE is designed to operate with minimal user input. All that is required is the initial guess geometry, either in Cartesian coordinates (*e.g.* from a suitable model builder such as HyperChem) or as a Z-matrix, the type of stationary point being sought (minimum or transition state) and details of any imposed constraints. All decisions as to the optimization strategy (what algorithm to use, what coordinate system to choose, how to handle the constraints) are made by OPTIMIZE.

Note particularly, that although the starting geometry is input in a particular coordinate system (as a Z-matrix, for example) these coordinates are not necessarily used during the actual optimization. The best coordinates for the majority of geometry optimizations are delocalized internals, and these will be tried first. Only if delocalized internals fail for some reason, or if conditions prevent them being used (*e.g.*, frozen atoms) will other coordinate systems be tried. If all else fails the default is to switch to Cartesian coordinates. Similar defaults hold for the optimization algorithm, maximum step size, convergence criteria, *etc.* You may of course override the default choices and force a particular optimization strategy, but it is not normally necessary to provide OPTIMIZE with anything other than the minimal information outlined above.

The heart of the OPTIMIZE package (for both minima and transition states) is Baker's EF (Eigenvector Following) algorithm [1]. This was developed following the work of Cerjan and Miller [2] and, Simons and coworkers [3, 4]. The Hessian mode-following option incorporated into this algorithm is capable of locating transition states by walking uphill from the associated minima. By following the lowest Hessian mode, the EF algorithm can locate transition states starting from any reasonable input geometry and Hessian.

An additional option available for minimization is Pulay's GDIIS algorithm [5], which is based on the well known DIIS technique for accelerating SCF convergence [6]. GDIIS must be specifically requested, as the EF algorithm is the default.

Although optimizations can be carried out in Cartesian or Z-matrix coordinates, the best choice, as noted above, is usually delocalized internal coordinates. These coordinates were developed very recently by Baker *et al* [7], and can be considered as a further extension of the natural internal coordinates developed by Pulay *et al* [8, 9] and the redundant optimization method of Pulay and Fogarasi [10].

OPTIMIZE incorporates a very accurate and efficient Lagrange multiplier algorithm for constrained optimization. This was originally developed for use with Cartesian coordinates [11, 12] and can handle constraints that are not satisfied in the starting geometry. Very recently the Lagrange multiplier approach has been modified for use with delocalized internals [13]; this is much more efficient and is now the default. The Lagrange multiplier code can locate constrained transition states as well as minima.

A.2 THEORETICAL BACKGROUND

A.2.1 THE NEWTON-RAPHSON STEP

Consider the energy, $E(\mathbf{x}_0)$ at some point \mathbf{x}_0 on a potential energy surface. We can express the energy at a nearby point $\mathbf{x} = \mathbf{x}_0 + \mathbf{h}$ by means of the Taylor series

$$E(\mathbf{x}_0 + \mathbf{h}) = E(\mathbf{x}_0) + \mathbf{h}^t \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \frac{1}{2} \mathbf{h}^t \frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \mathbf{h} + \dots \quad (\text{A.1})$$

If we knew the exact form of the energy functional $E(\mathbf{x})$ and all its derivatives, we could move from the current point \mathbf{x}_0 directly to a stationary point, (*i.e.*, we would know exactly what the step \mathbf{h} ought to be). Since we typically know only the lower derivatives of $E(\mathbf{x})$ at best, then we can estimate the step \mathbf{h} by differentiating the Taylor series with respect to \mathbf{h} , keeping only the first few terms on the right hand side, and setting the left hand side, $dE(\mathbf{x}_0 + \mathbf{h})/d\mathbf{h}$, to zero, which is the value it would have at a genuine stationary point. Thus

$$\frac{dE(\mathbf{x}_0 + \mathbf{h})}{d\mathbf{h}} = \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \mathbf{h} + \text{higher terms (ignored)} \quad (\text{A.2})$$

From which

$$\mathbf{h} = \mathbf{H}^{-1} \mathbf{g} \quad (\text{A.3})$$

where

$$\frac{dE}{d\mathbf{x}} \equiv \mathbf{g} \text{ (gradient vector)}, \quad \frac{d^2E}{d\mathbf{x}_1 d\mathbf{x}_2} \equiv \mathbf{H} \text{ (Hessian matrix)}$$

(A.3) is known as the Newton-Raphson step. It is the major component of almost all geometry optimization algorithms in quantum chemistry.

The above derivation assumed exact first (gradient) and second (Hessian) derivative information. Analytical gradients are available for all methodologies supported in Q-Chem; however analytical second derivatives are not. Furthermore, even if they were, it would not necessarily be advantageous to use them as their evaluation is usually computationally demanding, and, efficient optimizations can in fact be performed without an exact Hessian. An excellent compromise in practice is to begin with an approximate Hessian matrix, and update this using gradient and displacement information generated as the optimization progresses. In this way the starting Hessian can be "improved" at essentially no cost. Using (A.3) with an approximate Hessian is called the quasi Newton-Raphson step.

The nature of the Hessian matrix (in particular its eigenvalue structure) plays a crucial role in a successful optimization. All stationary points on a potential energy surface have a zero gradient vector; however the character of the stationary point (*i.e.*, what type of structure it corresponds to) is determined by the Hessian. Diagonalization of the Hessian matrix can be considered to define a set of mutually orthogonal directions on the energy surface (the eigenvectors) together with the curvature along those directions (the eigenvalues). At a local minimum (corresponding to a well in the potential energy surface) the curvature along all of these directions must be positive, reflecting the fact that a small displacement along any of these directions causes the energy to rise. At a transition state, the curvature is negative (*i.e.*, the energy is a maximum) along one direction, but positive along all the others. Thus, for a stationary point to be a transition state the Hessian matrix at that point must have one and only one negative eigenvalue, while for a minimum the Hessian must have all positive eigenvalues. In the latter case the Hessian is called *positive definite*. If searching for a minimum it is important that the Hessian matrix be positive definite; in fact, unless the Hessian is positive definite there is no guarantee that the step predicted by (A.3) is even a descent step (*i.e.*, a direction that will actually lower the energy). Similarly, for a transition state search, the Hessian must have one negative eigenvalue. Maintaining the Hessian eigenvalue structure is not difficult for minimization, but it can be a difficulty when trying to find a transition state.

In a diagonal Hessian representation the Newton-Raphson step can be written

$$\mathbf{h} = \sum \frac{-F_i}{b_i} \mathbf{u}_i \quad (\text{A.4})$$

where \mathbf{u}_i and b_i are the eigenvectors and eigenvalues of the Hessian matrix \mathbf{H} and $F_i = \mathbf{u}_i^t \mathbf{g}$ is the component of \mathbf{g} along the local direction (eigenmode) \mathbf{u}_i . As discussed by Simons *et al.* [3], the Newton-Raphson step can be considered as minimizing along directions \mathbf{u}_i which have positive eigenvalues and maximizing along directions with negative eigenvalues. Thus, if the user is searching for a minimum and the Hessian matrix is positive definite, then the Newton-Raphson step is appropriate since it is attempting to minimize along all directions simultaneously. However, if the Hessian has one or more negative eigenvalues, then the basic Newton-Raphson step is not appropriate for a minimum search, since it will be maximizing and not minimizing along one or more directions. Exactly the same arguments apply during a transition state search except that the Hessian must have one negative eigenvalue, because the user has to maximize along one direction. However, there must be *only* one negative eigenvalue. A positive definite Hessian is a disaster for a transition state search because the Newton-Raphson step will then lead towards a minimum.

If firmly in a region of the potential energy surface with the right Hessian character, then a careful search (based on the Newton-Raphson step) will almost always lead to a stationary point of the correct type. However, this is only true if the Hessian is exact. If an approximate Hessian is being improved by updating, then there is no guarantee that the Hessian eigenvalue structure will be retained from one cycle to the next unless one is very careful during the update. Updating procedures that "guarantee" conservation of a

positive definite Hessian do exist (or at least warn the user if the update is likely to introduce negative eigenvalues). This can be very useful during a minimum search; but there are no such guarantees for preserving the Hessian character (one and only one negative eigenvalue) required for a transition state.

In addition to the difficulties in retaining the correct Hessian character, there is the matter of obtaining a "correct" Hessian in the first instance. This is particularly acute for a transition state search. For a minimum search it is possible to "guess" a reasonable, positive-definite starting Hessian (for example, by carrying out a molecular mechanics minimization initially and using the mechanics Hessian to begin the *ab initio* optimization) but this option is usually not available for transition states. Even if the user calculates the Hessian exactly at the starting geometry, the guess for the structure may not be sufficiently accurate, and the expensive, exact Hessian may not have the desired eigenvalue structure.

Consequently, particularly for a transition state search, an alternative to the basic Newton-Raphson step is clearly needed, especially when the Hessian matrix is inappropriate for the stationary point being sought.

One of the first algorithms that was capable of taking corrective action during a transition state search if the Hessian had the wrong eigenvalue structure, was developed by Poppinger [14], who suggested that, instead of taking the Newton-Raphson step, if the Hessian had all positive eigenvalues, the lowest Hessian mode be followed uphill; whereas, if there were two or more negative eigenvalues, the mode corresponding to the least negative eigenvalue be followed downhill. While this step should lead the user back into the right region of the energy surface, it has the disadvantage that the user is maximizing or minimizing along one mode only, unlike the Newton-Raphson step which maximizes/minimizes along all modes simultaneously. Another drawback is that successive such steps tend to become linearly dependent, which degrades most of the commonly used Hessian updates.

A.2.2 THE EIGENVECTOR FOLLOWING (EF) ALGORITHM

The work of Cerjan and Miller [2], and later Simons and coworkers [3, 4], showed that there was a better step than simply directly following one of the Hessian eigenvectors. A simple modification to the Newton-Raphson step is capable of guiding the search away from the current region towards a stationary point with the required characteristics. This is

$$\mathbf{h} = \sum \frac{-F_i}{(b_i - \lambda)} \mathbf{u}_i \quad (\text{A.5})$$

in which λ can be regarded as a shift parameter on the Hessian eigenvalue b_i . Scaling the Newton-Raphson step in this manner effectively directs the step to lie primarily, but not exclusively (unlike Poppinger's algorithm [14]), along one of the local eigenmodes,

depending on the value chosen for λ . References [2-4] all utilize the same basic approach (A.5) but differ in the means of determining the value of λ .

The EF algorithm [1] utilizes the rational function approach presented in [4], yielding an eigenvalue equation of the form

$$\begin{pmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} \quad (\text{A.6})$$

from which a suitable λ can be obtained. Expanding (A.6) gives

$$(\mathbf{H} - \lambda)\mathbf{h} + \mathbf{g} = 0 \quad (\text{A.7a})$$

$$\mathbf{g}^t \mathbf{h} = \lambda \quad (\text{A.7b})$$

In terms of a diagonal Hessian representation, (A.7a) rearranges to (A.5), and substitution of (A.5) into the diagonal form of (A.7b) gives

$$\sum \frac{-F_i^2}{(b_i - \lambda)} = \lambda \quad (\text{A.8})$$

which can be used to evaluate λ iteratively.

The eigenvalues, λ , of the RFO equation (A.6) have the following important properties [4]:

1. The $(n+1)$ values of λ bracket the n eigenvalues of the Hessian matrix $\lambda_i < b_i < \lambda_{i+1}$
2. At a stationary point, one of the eigenvalues, λ , of (A.6) is zero and the other n eigenvalues are those of the Hessian at the stationary point.
3. For a saddle point of order m , the zero eigenvalue separates the m negative and the $(n-m)$ positive Hessian eigenvalues.

This last property, the separability of the positive and negative Hessian eigenvalues, enables two shift parameters to be used, one for modes along which the energy is to be maximized and the other for which it is minimized. For a transition state (a first-order saddle point), in terms of the Hessian eigenmodes, we have the two matrix equations

$$\begin{pmatrix} b_1 & F_1 \\ F_1 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ 1 \end{pmatrix} \quad (\text{A.9a})$$

$$\begin{pmatrix} b_2 & & & F_2 \\ & \ddots & \mathbf{0} & \vdots \\ & & \mathbf{0} & b_n & F_n \\ F_2 & \cdots & F_n & & 0 \end{pmatrix} \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} \quad (\text{A.9b})$$

where it is assumed that we are maximizing along the lowest Hessian mode \mathbf{u}_j . Note that λ_p is the highest eigenvalue of (A.9a) (it is always positive and approaches zero at convergence) and λ_n is the lowest eigenvalue of (A.9b) (it is always negative and again approaches zero at convergence).

Choosing these values of λ gives a step that attempts to maximize along the lowest Hessian mode, while at the same time minimizing along all the other modes. It does this regardless of the Hessian eigenvalue structure (unlike the Newton-Raphson step). The two shift parameters are then used in (A.5) to give the final step

$$\mathbf{h} = \frac{-F_1}{(b_1 - \lambda_p)} \mathbf{u}_1 - \sum_{i=2}^n \frac{-F_i}{(b_i - \lambda_n)} \mathbf{u}_i \quad (\text{A.10})$$

If this step is greater than the maximum allowed, it is scaled down. For minimization only one shift parameter, λ_n , would be used which would act on all modes.

In (A.9a) and (A.9b) it was assumed that the step would maximize along the lowest Hessian mode, b_1 , and minimize along all the higher modes. However, it is possible to maximize along modes other than the lowest, and in this way perhaps locate transition states for alternative rearrangements/dissociations from the same initial starting point. For maximization along the k^{th} mode (instead of the lowest), (A.9a) is replaced by

$$\begin{pmatrix} b_k & F_k \\ F_k & 0 \end{pmatrix} \begin{pmatrix} h_k \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_k \\ 1 \end{pmatrix} \quad (\text{A.11})$$

and (A.9b) would now exclude the k^{th} mode but include the lowest. Since what was originally the k^{th} mode is the mode along which the negative eigenvalue is required, then this mode will eventually become the lowest mode at some stage of the optimization. To ensure that the original mode is being followed smoothly from one cycle to the next, the mode that is actually followed is the one with the greatest overlap with the mode followed on the previous cycle. This procedure is known as *mode following*. For more details and some examples, see [1].

A.3 DELOCALIZED INTERNAL COORDINATES

The choice of coordinate system can have a major influence on the rate of convergence during a geometry optimization. For complex potential energy surfaces with many stationary points, a different choice of coordinates can result in convergence to a different final structure.

The key attribute of a good set of coordinates for geometry optimization is the degree of coupling between the individual coordinates. In general, the less coupling the better, as variation of one particular coordinate will then have minimal impact on the other coordinates. Coupling manifests itself primarily as relatively large partial derivative terms between different coordinates. For example, a strong harmonic coupling between two different coordinates, i and j , results in a large off-diagonal element, H_{ij} , in the Hessian (second derivative) matrix. Normally this is the only type of coupling that can be directly “observed” during an optimization, as third and higher derivatives are ignored in almost all optimization algorithms.

In the early days of computational quantum chemistry geometry optimizations were carried out in Cartesian coordinates. Cartesians are an obvious choice as they can be defined for all systems and gradients and second derivatives are calculated directly in Cartesian coordinates. Unfortunately, Cartesians normally make a poor coordinate set for optimization as they are heavily coupled. Recently, Cartesians have been returning to favour because of their very general nature, and because it has been clearly demonstrated that if reliable second derivative information is available (*i.e.*, a good starting Hessian) and the initial geometry is reasonable, then Cartesians can be as efficient as any other coordinate set for small to medium-sized molecules [15, 16]. Without good Hessian data, however, Cartesians are inefficient, especially for long chain acyclic systems.

In the 1970s Cartesians were replaced by Z-matrix coordinates. Initially the Z-matrix was utilized simply as a means of geometry input; it is far easier to describe a molecule in terms of bond lengths, bond angles and dihedral angles (the natural way a chemist thinks of molecular structure) than to develop a suitable set of Cartesian coordinates. It was subsequently found that optimization was generally more efficient in Z-matrix coordinates than in Cartesians, especially for acyclic systems. This is not always the case, and care must be taken in constructing a suitable Z-matrix. A good general rule is ensure that each variable is defined in such a way that changing its value will not change the values of any of the other variables. A brief discussion concerning good Z-matrix construction strategy is given by Schlegel [17].

In 1979 Pulay *et al.* published a key paper, introducing what were termed natural internal coordinates into geometry optimization [8]. These coordinates involve the use of individual bond displacements as stretching coordinates, but linear combinations of bond angles and torsions as deformational coordinates. Suitable linear combinations of bends and torsions (the two are considered separately) are selected using group theoretical arguments based on local pseudosymmetry. For example, bond angles around an sp^3 hybridized carbon atom are all approximately tetrahedral, regardless of the groups

attached, and idealized tetrahedral symmetry can be used to generate deformational coordinates around the central carbon atom.

The major advantage of natural internal coordinates in geometry optimization is their ability to significantly reduce the coupling, both harmonic and anharmonic, between the various coordinates. Compared to natural internals, Z-matrix coordinates arbitrarily omit some angles and torsions (to prevent redundancy), and this can induce strong anharmonic coupling between the coordinates, especially with a poorly constructed Z-matrix. Another advantage of the reduced coupling is that successful minimizations can be carried out in natural internals with only an approximate (*e.g.*, diagonal) Hessian provided at the starting geometry. A good starting Hessian is still needed for a transition state search.

Despite their clear advantages, natural internals have only become used widely more recently. This is because, when used in the early programs, it was necessary for the user to define them. This situation changed in 1992 with the development of computational algorithms capable of automatically generating natural internals from input Cartesians [9]. For minimization, natural internals have become the coordinates of first choice [9, 16].

There are some disadvantages to natural internal coordinates as they are commonly constructed and used:

1. Algorithms for the automatic construction of natural internals are complicated. There are a large number of structural possibilities, and to adequately handle even the most common of them can take several thousand lines of code.
2. For the more complex molecular topologies, most assigning algorithms generate more natural internal coordinates than are required to characterize all possible motions of the system (*i.e.*, the generated coordinate set contains redundancies).
3. In cases with a very complex molecular topology (*e.g.*, multiply fused rings and cage compounds) the assigning algorithm may be unable to generate a suitable set of coordinates.

The redundancy problem has recently been addressed in an excellent paper by Pulay and Fogarasi [10], who have developed a scheme for carrying out geometry optimization directly in the redundant coordinate space.

Very recently, Baker *et al.* have developed a set of delocalized internal coordinates [7] which eliminate all of the above-mentioned difficulties. Building on some of the ideas in the redundant optimization scheme of Pulay and Fogarasi [10], delocalized internals form a complete, non-redundant set of coordinates which are as good as, if not superior to, natural internals, and which can be generated in a simple and straightforward manner for essentially any molecular topology, no matter how complex.

Consider a set of n internal coordinates $\mathbf{q} = (q_1, q_2, \dots, q_n)^t$. Displacements $\Delta\mathbf{q}$ in \mathbf{q} are related to the corresponding Cartesian displacements $\Delta\mathbf{X}$ by means of the usual B-matrix [18]

$$\Delta\mathbf{q} = \mathbf{B}\Delta\mathbf{X} \quad (\text{A.12})$$

If any of the internal coordinates \mathbf{q} are redundant, then the rows of the B-matrix will be linearly dependent.

Delocalized internal coordinates are obtained simply by constructing and diagonalizing the matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^t$. Diagonalization of \mathbf{G} results in two sets of eigenvectors; a set of m (typically $3N-6$, where N is the number of atoms) eigenvectors with eigenvalues $\lambda > 0$, and a set of $n-m$ eigenvectors with eigenvalues $\lambda = 0$ (to numerical precision). In this way, any redundancies present in the original coordinate set \mathbf{q} are isolated (they correspond to those eigenvectors with zero eigenvalues). The eigenvalue equation of \mathbf{G} can thus be written

$$\mathbf{G}(\mathbf{UR}) = (\mathbf{UR}) \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \quad (\text{A.13})$$

where \mathbf{U} is the set of non-redundant eigenvectors of \mathbf{G} (those with $\lambda > 0$) and \mathbf{R} is the corresponding redundant set.

The nature of the original set of coordinates \mathbf{q} is unimportant, as long as it spans all the degrees of freedom of the system under consideration. We include in \mathbf{q} , all bond stretches, all planar bends and all proper torsions that can be generated based on the atomic connectivity. These individual internal coordinates are termed *primitives*. This blanket approach generates far more primitives than are necessary, and the set \mathbf{q} contains much redundancy. This is of little concern, as solution of (A.13) takes care of all redundancies.

Note that eigenvectors in both \mathbf{U} and \mathbf{R} will each be linear combinations of potentially all the original primitives. Despite this apparent complexity, we take the set of non-redundant vectors \mathbf{U} as our working coordinate set. Internal coordinates so defined are much more delocalized than natural internal coordinates (which are combinations of a relatively small number of bends or torsions) hence, the term delocalized internal coordinates.

It may appear that because delocalized internals are such a complicated mixing of the original primitive internals, they are a poor choice for use in an actual optimization. On the contrary, arguments can be made that delocalized internals are, in fact, the "best" possible choice, certainly at the starting geometry. The interested reader is referred to the original literature for more details [7].

The situation for geometry optimization, comparing Cartesian, Z-matrix and delocalized internal coordinates, and assuming a "reasonable" starting geometry, is as follows:

1. For small or very rigid medium-sized systems (up to about 15 atoms), optimizations in Cartesian and internal coordinates ("good" Z-matrix or delocalized internals) should perform similarly.

2. For medium-sized systems (say 15-30 atoms) optimizations in Cartesians should perform as well as optimizations in internal coordinates, provided a reliable starting Hessian is available.
3. For large systems (30+ atoms), unless these are very rigid, neither Cartesian nor *Z*-matrix coordinates can compete with delocalized internals, even with good quality Hessian information. As the system increases, and with less reliable starting geometries, the advantage of delocalized internals can only increase.

There is one particular situation in which Cartesian coordinates may be the best choice. Natural internal coordinates (and by extension delocalized internals) show a tendency to converge to low energy structures [16]. This is because steps taken in internal coordinate space tend to be much larger when translated into Cartesian space, and, as a result, higher energy local minima tend to be “jumped over”, especially if there is no reliable Hessian information available (which is generally not needed for a successful optimization). Consequently, if the user is looking for a local minimum (*i.e.*, a metastable structure) and has both a good starting geometry and a decent Hessian, the user should carry out the optimization in Cartesian coordinates.

A.4 CONSTRAINED OPTIMIZATION

A.4.1 CARTESIAN COORDINATES

Constrained optimization refers to the optimization of molecular structures in which certain parameters (*e.g.*, bond lengths, bond angles or dihedral angles) are fixed. In quantum chemistry calculations, this has traditionally been accomplished using Z-matrix coordinates, with the desired parameter set in the Z-matrix and simply omitted from the optimization space. In 1992, Baker presented an algorithm for constrained optimization directly in Cartesian coordinates [11]. Baker's algorithm used both penalty functions and the classical method of Lagrange multipliers [19], and was developed in order to impose constraints on a molecule obtained from a graphical model builder as a set of Cartesian coordinates. Some improvements widening the range of constraints that could be handled were made in 1993 [12]. Q-Chem includes the latest version of this algorithm, which has been modified to handle constraints directly in delocalized internal coordinates [13].

The essential problem in constrained optimization is to minimize a function of, for example, n variables $F(\mathbf{x})$ subject to a series of m constraints of the form $C_i(\mathbf{x}) = 0$, $i=1 \dots m$. Assuming $m < n$, then perhaps the best way to proceed (if this were possible in practice) would be to use the m constraint equations to eliminate m of the variables, and then solve the resulting unconstrained problem in terms of the $(n-m)$ independent variables. This is exactly what occurs in a Z-matrix optimization. Such an approach cannot be used in Cartesian coordinates as standard distance and angle constraints are non-linear functions of the appropriate coordinates. For example a distance constraint (between atoms i and j in a molecule) is given in Cartesians by $(R_{ij} - R_0) = 0$, with

$$R_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (\text{A.14})$$

and R_0 the constrained distance. This obviously cannot be satisfied by elimination. What can be eliminated in Cartesians are the individual x , y and z coordinates themselves and in this way individual atoms can be totally or partially frozen.

Internal constraints can be handled in Cartesian coordinates by introducing the Lagrangian function

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \sum_{i=1}^m \lambda_i C_i(\mathbf{x}) \quad (\text{A.15})$$

which replaces the function $F(\mathbf{x})$ in the unconstrained case. Here, the λ_i are the so-called Lagrange multipliers, one for each constraint $C_i(\mathbf{x})$. Differentiating (A.15) with respect to \mathbf{x} and λ gives

$$\frac{dL(\mathbf{x}, \lambda)}{dx_j} = \frac{dF(\mathbf{x})}{dx_j} - \sum_{i=1}^m \lambda_i \frac{dC_i(\mathbf{x})}{dx_j} \quad (\text{A.16a})$$

and

$$\frac{dL(\mathbf{x}, \lambda)}{d\lambda_i} = -C_i(\mathbf{x}) \quad (\text{A.16b})$$

At a stationary point of the Lagrangian we have $\nabla \mathbf{L} = 0$, *i.e.*, all $dL/dx_j = 0$ and all $dL/d\lambda_i = 0$. This latter condition means that all $C_i(\mathbf{x}) = 0$ and thus all constraints are satisfied. Hence, finding a set of values (\mathbf{x}, λ) for which $\nabla \mathbf{L} = 0$ will give a possible solution to the constrained optimization problem in exactly the same way as finding an \mathbf{x} for which $\mathbf{g} = \nabla \mathbf{F} = 0$ gives a solution to the corresponding unconstrained problem.

The Lagrangian second derivative matrix, the equivalent of the Hessian matrix in an unconstrained optimization, is given by

$$\nabla^2 \mathbf{L} = \begin{pmatrix} d^2 L(\mathbf{x}, \lambda) / dx_j dx_k & d^2 L(\mathbf{x}, \lambda) / dx_j d\lambda_i \\ d^2 L(\mathbf{x}, \lambda) / dx_j d\lambda_i & d^2 L(\mathbf{x}, \lambda) / d\lambda_j d\lambda_i \end{pmatrix} \quad (\text{A.17})$$

where

$$d^2 L(\mathbf{x}, \lambda) / dx_j dx_k = d^2 F(\mathbf{x}) / dx_j dx_k - \sum \lambda_i d^2 F(\mathbf{x}) / dx_j dx_k \quad (\text{A.17a})$$

$$d^2 L(\mathbf{x}, \lambda) / dx_j d\lambda_i = -dC_i(\mathbf{x}) / dx_j \quad (\text{A.17b})$$

$$d^2 L(\mathbf{x}, \lambda) / d\lambda_j d\lambda_i = 0 \quad (\text{A.17c})$$

Thus in addition to the standard gradient vector and Hessian matrix for the unconstrained function $F(\mathbf{x})$, we need both the first and second derivatives (with respect to coordinate displacement) of the constraint functions. Once these quantities are available, the corresponding Lagrangian gradient, given by (A.16), and Lagrangian second derivative matrix, given by (A.17), can be formed, and the optimization step calculated in a similar manner to that for a standard unconstrained optimization [11].

In the Lagrange multiplier method, the unknown multipliers, λ_i , are an integral part of the parameter set. This means that the optimization space consists of all n variables \mathbf{x} plus all m Lagrange multipliers λ , one for each constraint. The total dimension of the constrained optimization problem, $n+m$, has thus increased by m compared to the corresponding unconstrained case. The Lagrangian Hessian matrix, $\nabla^2 \mathbf{L}$, has m extra modes compared to the standard (unconstrained) Hessian matrix, $\nabla^2 \mathbf{F}$. What normally happens is that these additional modes are dominated by the constraints (*i.e.*, their largest components correspond to the constraint Lagrange multipliers) and they have negative curvature (a negative Hessian eigenvalue). This is perhaps not surprising when one realizes that any motion in the parameter space that breaks the constraints is likely to lower the energy.

Compared to a standard unconstrained minimization, where a stationary point is sought at which the Hessian matrix has all positive eigenvalues, in the constrained problem we are

looking for a stationary point of the Lagrangian function at which the Lagrangian Hessian matrix has as many negative eigenvalues as there are constraints (*i.e.*, we are looking for an m^{th} order saddle point). For further details and practical applications of constrained optimization using Lagrange multipliers in Cartesian coordinates, see [11].

Eigenvector following can be implemented in a constrained optimization in a similar way to the unconstrained case. Considering a constrained minimization with m constraints, then (A.9a) is replaced by

$$\begin{pmatrix} b_1 & & & F_1 \\ & \ddots & \mathbf{0} & \vdots \\ & & \mathbf{0} & b_m & F_m \\ F_1 & \cdots & F_m & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} \quad (\text{A.18a})$$

and (A.9b) by

$$\begin{pmatrix} b_{m+1} & & & F_{m+1} \\ & \ddots & \mathbf{0} & \vdots \\ & & \mathbf{0} & b_{m+n} & F_{m+n} \\ F_{m+1} & \cdots & F_{m+n} & 0 \end{pmatrix} \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} \quad (\text{A.18b})$$

where now the b_i are the eigenvalues of $\nabla^2 \mathbf{L}$, with corresponding eigenvectors \mathbf{u}_i , and $F_i = \mathbf{u}_i^t \nabla \mathbf{L}$. Here (A.18a) includes the m constraint modes along which a negative Lagrangian Hessian eigenvalue is required, and (A.18b) includes all the other modes.

Equations (A.18a) and (A.18b) implement eigenvector following for a constrained minimization. Constrained transition state searches can be carried out by selecting one extra mode to be maximized in addition to the m constraint modes, *i.e.*, by searching for a saddle point of the Lagrangian function of order $m+1$.

It should be realized that, in the Lagrange multiplier method, the desired constraints are only satisfied at convergence, and not necessarily at intermediate geometries. The Lagrange multipliers are part of the optimization space; they vary just as any other geometrical parameter and, consequently the degree to which the constraints are satisfied changes from cycle to cycle, approaching 100% satisfied near convergence. One advantage this brings is that, unlike in a standard Z-matrix approach, constraints do not have to be satisfied in the starting geometry.

Imposed constraints can normally be satisfied to very high accuracy, 10^{-6} or better. However, problems can arise for both bond and dihedral angle constraints near 0° and 180° and, instead of attempting to impose a single constraint, it is better to split angle constraints near these limiting values into two by using a dummy atom [12], exactly analogous to splitting a 180° bond angle into two 90° angles in a Z-matrix.

Note: Exact 0° and 180° single angle constraints cannot be imposed, as the corresponding constraint normals, $\nabla \mathbf{C}_i$, are zero, and would result in rows and columns of zeros in the Lagrangian Hessian matrix.

A.4.2 DELOCALIZED INTERNAL COORDINATES

We do not give further details of the optimization algorithms available in Q-Chem for imposing constraints in Cartesian coordinates, as it is far simpler and easier to do this directly in delocalized internal coordinates.

At first sight it does not seem particularly straightforward to impose any constraints at all in delocalized internals, given that each coordinate is potentially a linear combination of all possible primitives. However, this is deceptive, and in fact all standard constraints can be imposed by a relatively simple Schmidt orthogonalization procedure. In this instance consider a unit vector with unit component corresponding to the primitive internal (stretch, bend or torsion) that one wishes to keep constant. This vector is then projected on to the full set, \mathbf{U} , of active delocalized coordinates, normalized, and then all n , for example, delocalized internals are Schmidt orthogonalized in turn to this normalized, projected constraint vector. The last coordinate taken in the active space should drop out (since it will be linearly dependent on the other vectors and the constraint vector) leaving $(n-1)$ active vectors and one constraint vector.

In more detail, the procedure is as follows (taken directly from [7]). The initial (usually unit) constraint vector \mathbf{C} is projected on to the set \mathbf{U} of delocalized internal coordinates according to

$$\mathbf{C}^{proj} = \sum \langle \mathbf{C} | \mathbf{U}_k \rangle \mathbf{U}_k \quad (\text{A.19})$$

where the summation is over all n active coordinates \mathbf{U}_k . The projected vector \mathbf{C}^{proj} is then normalized and an $(n+1)$ dimensional vector space \mathbf{V} is formed, comprising the normalized, projected constraint vector together with all active delocalized coordinates

$$\mathbf{V} = \{ \mathbf{C}^{proj}, \mathbf{U}_k \ k = 1 \dots n \} \quad (\text{A.20})$$

This set of vectors is Schmidt orthogonalized according to the standard procedure

$$\tilde{\mathbf{V}}_k = \alpha_k \left(\mathbf{V}_k - \sum_{l=1}^{k-1} \langle \mathbf{V}_k | \tilde{\mathbf{V}}_l \rangle \tilde{\mathbf{V}}_l \right) \quad (\text{A.21})$$

where the first vector taken, \mathbf{V}_1 , is \mathbf{C}^{proj} . The α_k in (A.21) is a normalization factor. As noted above, the last vector taken, $\mathbf{V}_{n+1} = \mathbf{U}_k$, will drop out, leaving a fully orthonormal set of $(n-1)$ active vectors and one constraint vector.

After the Schmidt orthogonalization the constraint vector will contain all the weight in the active space of the primitive to be fixed, which will have a zero component in all of

the other ($n-1$) vectors. The fixed primitive has thus been isolated entirely in the constraint vector which can now be removed from the active subspace for the geometry optimization step.

Extension of the above procedure to multiple constraints is straightforward. In addition to constraints on individual primitives, it is also possible to impose combinatorial constraints. For example, if, instead of a unit vector, one started the constraint procedure with a vector in which two components were set to unity, then this would impose a constraint in which the sum of the two relevant primitives were always constant. In theory any desired linear combination of any primitives could be constrained.

Note further that imposed constraints are not confined to those primitive internals generated from the initial atomic connectivity. If we wish to constrain a distance, angle or torsion between atoms that are not formally connected, then all we need to do is add that particular coordinate to our primitive set. It can then be isolated and constrained in exactly the same way as a formal connectivity constraint.

Everything discussed thus far regarding the imposition of constraints in delocalized internal coordinates has involved isolating each constraint in one vector which is then eliminated from the optimization space. This is very similar in effect to a Z-matrix optimization, in which constraints are imposed by elimination. This, of course, can only be done if the desired constraint is satisfied in the starting geometry. We have already seen that the Lagrange multiplier algorithm, used to impose distance, angle and torsion constraints in Cartesian coordinates, can be used even when the constraint is not satisfied initially. The Lagrange multiplier method can also be used with delocalized internals, and its implementation with internal coordinates brings several simplifications and advantages.

In Cartesians, as already noted, standard internal constraints (bond distances, angles and torsions) are somewhat complicated non-linear functions of the x , y and z coordinates of the atoms involved. A torsion, for example, which involves four atoms, is a function of twelve different coordinates. In internals, on the other hand, each constraint is a coordinate in its own right and is therefore a simple linear function of just one coordinate (itself).

If we denote a general internal coordinate by R , then the constraint function $C_i(\mathbf{R})$ is a function of one coordinate, R_i , and it and its derivatives can be written

$$C_i(R_i) = R_i - R_0 \quad (\text{A.22a})$$

$$d C_i(R_i) / d R_i = 1; \quad d C_i(R_i) / d R_j = 0 \quad (\text{A.22b})$$

$$d^2 C_i(R_i) / d R_i d R_j = 0 \quad (\text{A.22c})$$

where in (A.22a), R_0 is the desired value of the constrained coordinate, and R_i is its current value. From (A.22b) we see that the constraint normals, $dC_i(\mathbf{R})/dR_i$, are simply

unit vectors and the Lagrangian Hessian matrix, (A.17), can be obtained from the normal Hessian matrix by adding m columns (and m rows) of, again, unit vectors.

A further advantage, in addition to the considerable simplification, is the handling of 0° and 180° dihedral angle constraints. In Cartesian coordinates it is not possible to formally constrain bond angles and torsions to exactly 0° or 180° because the corresponding constraint normal is a zero vector. Similar difficulties do not arise in internal coordinates, at least for torsions, because the constraint normals are unit vectors regardless of the value of the constraint; thus 0° and 180° dihedral angle constraints can be imposed just as easily as any other value. 180° bond angles still cause difficulties, but near-linear arrangements of atoms require special treatment even in unconstrained optimizations; a typical solution involves replacing a near 180° bond angle by two special linear co-planar and perpendicular bends [20], and modifying the torsions where necessary. A linear arrangement can be enforced by constraining the co-planar and perpendicular bends.

One other advantage over Cartesians is that in internals the constraint coordinate can be eliminated once the constraint is satisfied to the desired accuracy (the default tolerance is 10^{-6} in atomic units: bohrs and radians). This is not possible in Cartesians due to the functional form of the constraint. In Cartesians, therefore, the Lagrange multiplier algorithm must be used throughout the entire optimization, whereas in delocalized internal coordinates it need only be used until all desired constraints are satisfied; as constraints become satisfied they can simply be eliminated from the optimization space and once all constraint coordinates have been eliminated standard algorithms can be used in the space of the remaining unconstrained coordinates. Normally, unless the starting geometry is particularly poor in this regard, constraints are satisfied fairly early on in the optimization (and at more or less the same time for multiple constraints), and Lagrange multipliers only need to be used in the first half-dozen or so cycles of a constrained optimization in internal coordinates.

A.5 GDIIS

Direct inversion in the iterative subspace (DIIS) was originally developed by Pulay for accelerating SCF convergence [6]. Subsequently, Csaszar and Pulay used a similar scheme for geometry optimization, which they termed GDIIS [5]. The method is somewhat different from the usual quasi-Newton type approach and is included in OPTIMIZE as an alternative to the EF algorithm. Tests indicate that its performance is similar to EF, at least for small systems; however there is rarely an advantage in using GDIIS in preference to EF.

In GDIIS, geometries (\mathbf{x}_i) generated in previous optimization cycles are linearly combined to find the "best" geometry on the current cycle

$$\mathbf{x}_n = \sum_{i=1}^m c_i \mathbf{x}_i \quad (\text{A.23})$$

where the problem is to find the best values for the coefficients c_i .

If we express each geometry, \mathbf{x}_i , by its deviation from the sought-after final geometry, \mathbf{x}_f , i.e., $\mathbf{x}_i = \mathbf{x}_f + \mathbf{e}_i$, where \mathbf{e}_i is an error vector, then it is obvious that if the conditions

$$\mathbf{r} = \sum c_i \mathbf{e}_i \quad (\text{A.24})$$

and

$$\sum c_i = 1 \quad (\text{A.25})$$

are satisfied, then the relation

$$\sum c_i \mathbf{x}_i = \mathbf{x}_f \quad (\text{A.26})$$

also holds.

The true error vectors \mathbf{e}_i are, of course, unknown. However, in the case of a nearly quadratic energy function they can be approximated by

$$\mathbf{e}_i = -\mathbf{H}^{-1} \mathbf{g}_i \quad (\text{A.27})$$

where \mathbf{g}_i is the gradient vector corresponding to the geometry \mathbf{x}_i and \mathbf{H} is an approximation to the Hessian matrix. Minimization of the norm of the residuum vector \mathbf{r} , (A.24), together with the constraint equation, (A.25), leads to a system of $(m+1)$ linear equations

$$\begin{pmatrix} B_{11} & \cdots & B_{1m} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ B_{m1} & \cdots & B_{mm} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \\ -\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (\text{A.28})$$

where $B_{ij} = \langle \mathbf{e}_i | \mathbf{e}_j \rangle$ is the scalar product of the error vectors \mathbf{e}_i and \mathbf{e}_j , and λ is a Lagrange multiplier.

The coefficients c_i determined from (A.28) are used to calculate an intermediate interpolated geometry

$$\mathbf{x}_{m+1} = \sum c_i \mathbf{x}_i \quad (\text{A.29})$$

and its corresponding interpolated gradient

$$\mathbf{g}_{m+1} = \sum c_i \mathbf{g}_i \quad (\text{A.30})$$

A new, independent geometry is generated from the interpolated geometry and gradient according to

$$\mathbf{x}_{m+1} = \mathbf{x}'_{m+1} - \mathbf{H}^{-1} \mathbf{g}'_{m+1} \quad (\text{A.31})$$

Note: Convergence is theoretically guaranteed regardless of the quality of the Hessian matrix (as long as it is positive definite), and the original GDIIS algorithm used a static Hessian (*i.e.* the original starting Hessian, often a simple unit matrix, remained unchanged during the entire optimization). However, updating the Hessian at each cycle generally results in more rapid convergence, and this is the default in OPTIMIZE.

Other modifications to the original method include limiting the number of previous geometries used in (A.23) and, subsequently, by neglecting earlier geometries, and eliminating any geometries more than a certain distance (default 0.3 au) from the current geometry.

A.6 REFERENCES

- [1] J. Baker, *J. Comp. Chem.* (1986), **7**, 385.
- [2] C. J. Cerjan and W. H. Miller, *J. Chem. Phys.* (1981), **75**, 2800.
- [3] J. Simons, P. Jorgensen, H. Taylor and J. Ozment, *J. Phys. Chem.* (1983), **87**, 2745.
- [4] A. Banerjee, N. Adams, J. Simons and R. Shepard, *J. Phys. Chem.* (1985), **89**, 52.
- [5] P. Csaszar and P. Pulay, *J. Mol. Struct. Theochem.* (1984), **114**, 31.
- [6] P. Pulay, *J. Comp. Chem.* (1982), **3**, 556.
- [7] J. Baker, A. Kessi and B. Delley, *J. Chem. Phys.* (1996), **105**, 192.
- [8] P. Pulay, G. Fogarasi, F. Pang and J. E. Boggs, *J. Am. Chem. Soc.* (1979), **101**, 2550.
- [9] G. Fogarasi, X. Zhou, P. W. Taylor and P. Pulay, *J. Am. Chem. Soc.* (1992), **114**, 8191.
- [10] P. Pulay and G. Fogarasi, *J. Chem. Phys.* (1992), **96**, 2856.
- [11] J. Baker, *J. Comp. Chem.* (1992), **13**, 240.
- [12] J. Baker and D. Bergeron, *J. Comp. Chem.* (1993), **14**, 1339.
- [13] J. Baker, *J. Comp. Chem.* in press.
- [14] D. Poppinger, *Chem. Phys. Letts.* (1975), **35**, 550.
- [15] J. Baker and W. J. Hehre, *J. Comp. Chem.* (1991), **12**, 606.
- [16] J. Baker, *J. Comp. Chem.* (1993), **14**, 1085.
- [17] H. B. Schlegel, *Theor. Chim. Acta.* (1984), **66**, 333.
- [18] E. B. Wilson, J. C. Decius and P. C. Cross, *Molecular Vibrations*, McGraw-Hill, New York, 1955.
- [19] R. Fletcher, *Practical Methods of Optimization: vol. 2 - Constrained Optimization*, Wiley, New York, 1981.
- [20] S. Califano *Vibrational States*, Wiley, London, 1976.

APPENDIX B AOINTS

B.1 INTRODUCTION

Within the Q-Chem program, an Atomic Orbital INTEGRALS (AOINTS) package has been developed which, while relatively invisible to the user, is one of the keys to the overall speed and efficiency of the Q-Chem program.

“Ever since Boys’ introduction of Gaussian basis sets to quantum chemistry in 1950, the calculation and handling of the notorious two-electron-repulsion integrals (ERIs) over Gaussian functions has been an important avenue of research for practicing computational chemists. Indeed, the emergence of practically useful computer programs ... has been fuelled in no small part by the development of sophisticated algorithms to compute the very large number of ERIs that are involved in calculations on molecular systems of even modest size.” [1].

The ERI engine of any competitive quantum chemistry software package will be one of the most complicated aspects of the package as whole. Coupled with the importance of such an engine’s efficiency, a useful yardstick of a program’s anticipated performance can be quickly measured by considering the components of its ERI engine. In recent times, developers at Q-Chem, Inc. have made significant contributions to the advancement of ERI algorithm technology (for example see [1-10]), and it is not surprising that Q-Chem’s AOINTS package is considered the most advanced of its kind.

B.2 HISTORICAL PERSPECTIVE

Prior to the 1950’s, the most difficult step in the systematic application of Schrödinger wave mechanics to chemistry was the calculation of the notorious two-electron integrals that measure the repulsion between electrons. Boys [11] showed that this step can be made easier (although still time consuming) if Gaussian, rather than Slater, orbitals are used in the basis set. Following the landmark paper of computational chemistry [12] (again due to Boys) programs were constructed that could calculate all the ERIs that arise in the treatment of a general polyatomic molecule with *s* and *p* orbitals. However, the programs were painfully slow and could only be applied to the smallest of molecular systems.

In 1969, Pople constructed a breakthrough ERI algorithm, a hundred time faster than its predecessors. The algorithm remains the fastest available for its associated integral classes and is now referred to as the Pople-Hehre axis-switch method [13].

Over the two decades following Pople’s initial development, an enormous amount of research effort into the construction of ERIs was documented, which built on Pople’s

original success. Essentially, the advances of the newer algorithms could be identified as either better coping with angular momentum (L) or, contraction (K); each new method increasing the speed and application of quantum mechanics to solving real chemical problems.

By 1990, another barrier had been reached. The contemporary programs had become sophisticated and both academia and industry had begun to recognize and use the power of *ab initio* quantum chemistry, but the software was struggling with "dusty deck syndrome" and it had become increasingly difficult for it to keep up with the rapid advances in hardware development. Vector processors, parallel architectures and the advent of the graphical user interface were all demanding radically different approaches to programming and it had become clear that a fresh start, with a clean slate, was both inevitable and desirable. Furthermore, the integral bottleneck had re-emerged in a new guise and the standard programs were now hitting the N^2 wall. Irrespective of the speed at which ERIs could be computed, the unforgiving fact remained that the number of ERIs required scaled quadratically with the size of the system.

The Q-Chem project was established to tackle this problem and to seek new methods that circumvent the N^2 wall. Fundamentally new approaches to integral theory were sought and the ongoing advances that have resulted [14-18] have now placed Q-Chem firmly at the vanguard of the field. It should be emphasized, however, that the $O(N)$ methods that we have developed still require short-range ERIs to treat interactions between nearby electrons, thus the importance of contemporary ERI code remains.

The chronological development and evolution of integral methods can be summarised by considering a timeline showing the years in which important new algorithms were first introduced. These are best discussed in terms of the type of ERI or matrix elements that the algorithm can compute efficiently.

1950	Boys	[11]	ERIs with low L and low K
1969	Pople	[13]	ERIs with low L and high K
1976	Dupuis	[19]	Integrals with any L and low K
1978	McMurchie	[20]	Integrals with any L and low K
1982	Almlöf	[21]	Introduction of the direct SCF approach
1986	Obara	[22]	Integrals with any L and low K
1988	Head-Gordon	[8]	Integrals with any L and low K
1991	Gill	[1, 6]	Integrals with any L and any K
1994	White	[14]	J matrix in linear work
1996	Schwegler	[18, 24]	HF exchange matrix in linear work
1997	Challacombe	[17]	Fock matrix in linear work

B.3 AOINTS: CALCULATING ERIS WITH Q-CHEM

The area of molecular integrals with respect to Gaussian basis functions has recently been reviewed [2] and the user is referred to this review for deeper discussions and further references to the general area. The purpose of this short account is to present the basic approach, and in particular, the implementation of ERI algorithms and aspects of interest to the user in the AOINTS package which underlies the Q-Chem program.

We begin by observing that all of the integrals encountered in an *ab initio* calculation, of which overlap, kinetic energy, multipole moment, internuclear repulsion, nuclear-electron attraction and interelectron repulsion are the best known, can be written in the general form

$$(\mathbf{ab}|\mathbf{cd}) = \iint \phi_{\mathbf{a}}(\mathbf{r}_1)\phi_{\mathbf{b}}(\mathbf{r}_1)\theta(r_{12})\phi_{\mathbf{c}}(\mathbf{r}_2)\phi_{\mathbf{d}}(\mathbf{r}_2)d\mathbf{r}_1d\mathbf{r}_2 \quad (\text{B.1})$$

where the basis functions are contracted Gaussian's (CGTF)

$$\phi_{\mathbf{a}}(\mathbf{r}) = (x - A_x)^{a_x} (y - A_y)^{a_y} (z - A_z)^{a_z} \sum_{i=1}^{K_a} D_{ai} e^{-\alpha_i |\mathbf{r} - \mathbf{A}|^2} \quad (\text{B.2})$$

and the operator θ is a two-electron operator. Of the two-electron operators (Coulomb, CASE, anti-Coulomb and delta-function) used in the Q-Chem program, the most significant is the Coulomb, which leads us to the ERIs.

An ERI is the classical Coulomb interaction ($\theta(x) = 1/x$ in B.1) between two charge distributions referred to as bras $(\mathbf{ab}|$ and kets $|\mathbf{cd})$.

B.3.1 SHELL-PAIR DATA

It is common to characterise a bra, a ket and a bra-ket by their degree of contraction and angular momentum. In general, it is more convenient to compile data for shell-pairs rather than basis-function-pairs. A shell is defined as that sharing common exponents and centres. For example, in the case of a number of Pople derived basis sets, four basis functions, encompassing a range of angular momentum types (*i.e.*, s, p_x, p_y, p_z) on the same atomic centre sharing the same exponents constitute a single shell.

The shell-pair data set is central to the success of any modern integral program for three main reasons. First, in the formation of shell-pairs, all pairs of shells in the basis set are considered and categorized as either significant or negligible. A shell-pair is considered negligible if the shells involved are so far apart, relative to their diffuseness, that their overlap is negligible. Given the rate of decay of Gaussian basis functions, it is not surprising that most of the shell-pairs in a large molecule are negligible, that is, the number of significant shell-pairs increases linearly with the size of the molecule. Second, a number of useful intermediates which are frequently required within ERI algorithms should be computed once in shell-pair formation and stored as part of the shell-pair

information, particularly those which require costly divisions. This prevents re-evaluating simple quantities. Third, it is useful to sort the shell-pair information by type (*i.e.*, angular momentum and degree of contraction). The reasons for this are discussed below.

Q-Chem's shell-pair formation offers the option of two basic integral shell-pair cutoff criteria; one based on the integral threshold (*\$rem* variable *THRESH*) and the other relative to machine precision.

Intelligent construction of shell-pair data scales linearly with the size of the basis set, requires a relative amount of CPU time which is almost entirely negligible for large direct SCF calculations, and for small jobs, constitutes approximately 10% of the job time.

B.3.2 SHELL-QUARTETS AND INTEGRAL CLASSES

Given a sorted list of shell-pair data, it is possible to construct all potentially important shell-quartets by pairing of the shell-pairs with one another. Because the shell-pairs have been sorted, it is possible to deal with batches of integrals of the same type or class (*e.g.*, $(ss|ss)$, $(sp|sp)$, $(dd|dd)$, *etc.*) where an integral class is characterized by both angular momentum (L) and degree of contraction (K). Such an approach is advantageous for vector processors and for semi-direct integral algorithms where the most expensive (high K or L) integral classes can be computed once, stored in memory (or disk) and only less expensive classes rebuilt on each iteration.

While the shell-pairs may have been carefully screened, it is possible for a pair of significant shell-pairs to form a shell-quartet which need not be computed directly. Three cases are:

1. The quartet is equivalent, by point group symmetry, to another quartet already treated.
2. The quartet can be ignored on the basis of cheaply computed ERI bounds [7] on the largest quartet bra-ket.
3. On the basis of an incremental Fock matrix build, the largest density matrix element which will multiply any of the bra-kets associated with the quartet may be negligibly small.

Note: Significance and negligibility is always based on the level of integral threshold set by the *\$rem* variable *THRESH*.

B.3.3 FUNDAMENTAL ERI

The fundamental ERI [2] and the basis of all ERI algorithms is usually represented

$$\begin{aligned} [\mathbf{0}]^{(0)} &= [ss|ss]^{(0)} \\ &= D_A D_B D_C D_D \iint e^{-\alpha|\mathbf{r}_1-\mathbf{A}|^2} e^{-\beta|\mathbf{r}_1-\mathbf{B}|^2} \left[\frac{1}{r_{12}} \right] e^{-\gamma|\mathbf{r}_2-\mathbf{C}|^2} e^{-\delta|\mathbf{r}_2-\mathbf{D}|^2} d\mathbf{r}_1 d\mathbf{r}_2 \end{aligned} \quad (\text{B.3})$$

which can be reduced to a one-dimensional integral of the form

$$[\mathbf{0}]^{(0)} = U(2\vartheta^2)^{1/2} \left(\frac{2}{\pi} \right)^{1/2} \int_0^1 e^{-Tu^2} du \quad (\text{B.4})$$

and can be efficiently computed using a modified Chebyshev interpolation scheme [5]. Equation (B.4) can also be adapted for the general-case $[\mathbf{0}]^{(m)}$ integrals required for most calculations. Following the fundamental ERI, building up to the full bra-ket ERI (or intermediary matrix elements, see later) are the problems of angular momentum and contraction.

Note: Square brackets denote primitive integrals and parentheses fully contracted.

B.3.4 ANGULAR MOMENTUM PROBLEM

The fundamental integral is essentially an integral without angular momentum (*i.e.*, it is an integral of the type $[ss|ss]$). Angular momentum, usually depicted by L , has been problematic for efficient ERI formation, evident in the timeline in section B.2. Initially, angular momentum was calculated by taking derivatives of the fundamental ERI with respect to one of the Cartesian coordinates of the nuclear centre. This is an extremely inefficient route, but it works and was appropriate in the early development of ERI methods. Recursion relations [22, 25] and the newly developed tensor equations [3] are the basis for the modern approaches.

B.3.5 CONTRACTION PROBLEM

The contraction problem may be described by considering a general contracted ERI of s -type functions derived from the STO-3G basis set. Each basis function has degree of contraction $K = 3$. Thus, the ERI may be written

$$\begin{aligned}
 (ss|ss) &= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \sum_{l=1}^3 D_{Ai} D_{Bj} D_{Ck} D_{Dl} \times \\
 &\iint e^{-\alpha_i |\mathbf{r}_1 - \mathbf{A}|^2} e^{-\beta_j |\mathbf{r}_1 - \mathbf{B}|^2} \left[\frac{1}{r_{12}} \right] e^{-\gamma_k |\mathbf{r}_2 - \mathbf{C}|^2} e^{-\delta_l |\mathbf{r}_2 - \mathbf{D}|^2} d\mathbf{r}_1 d\mathbf{r}_2 \quad (\text{B.5}) \\
 &= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \sum_{l=1}^3 [s_i s_j | s_k s_l]
 \end{aligned}$$

and requires 81 primitive integrals for the single ERI. The problem escalates dramatically for more highly contracted sets (STO-6G, 6-311G) and has been the motivation for the development of techniques for shell-pair modelling [26] in which a second shell-pair is constructed with fewer primitives than the first, but introduces no extra error relative to the integral threshold sought.

The Pople-Hehre axis-switch method [13] is excellent for high contraction low angular momentum integral classes.

B.3.6 QUADRATIC SCALING

The success of quantitative modern quantum chemistry, relative to its primitive, qualitative beginnings, can be traced to two sources: better algorithms and better computers. While the two technologies continue to improve rapidly, efforts are heavily thwarted by the fact that the total number of ERIs increases quadratically with the size of the molecular system. Even large increases in ERI algorithm efficiency yield only moderate increases in applicability, hindering the more widespread application of *ab initio* methods to areas of, perhaps, biochemical significance where semi-empirical techniques [27, 28] have already proven so valuable.

Thus, the elimination of quadratic scaling algorithms has been the theme of many research efforts in quantum chemistry throughout the 1990's and has seen the construction of many alternative algorithms to alleviate the problem. Johnson was the first to implement DFT exchange/correlation functionals whose computational cost scaled linearly with system size [23]. This paved the way for the most significant breakthrough in the area with the linear scaling CFMM algorithm [14] leading to linear scaling DFT calculations [29]. Further breakthroughs have been made with traditional theory in the form of the QCTC [17, 30, 31] and ONX [18, 24] algorithms, whilst more radical approaches [15, 16, 32] may lead to entirely new approaches to *ab initio* calculations. Investigations into the quadratic Coulomb problem has not only yielded

linear scaling algorithms, but is also providing large insights into the significance of many molecular energy components.

Linear scaling Coulomb and SCF exchange/correlation algorithms are not the end of the story as the $O(N^3)$ diagonalization step has been rate limiting in semi-empirical techniques and, been predicted [33] to become rate limiting in *ab initio* approaches in the medium term. However, divide-and-conquer techniques [34-37] and the recently developed quadratically convergent SCF algorithm [38] show great promise for reducing this problem.

B.3.7 ALGORITHM SELECTION

No single ERI algorithm is available to efficiently handle all integral classes; rather, each tends to have specific integral classes where the specific algorithm out-performs the alternatives. The PRISM algorithm [6] is an intricate collection of pathways and steps in which the path chosen is that which is the most efficient for a given class. It appears that the most appropriate path for a given integral class depends on the relative position of the contraction step (lowly contracted bra-kets prefer late contraction, highly contracted bra-kets are most efficient with early contraction steps).

Careful studies have provided FLOP counts which are the current basis of integral algorithm selection, although care must be taken to ensure that algorithms are not rate limited by MOPs [4]. Future algorithm selection criteria will take greater account of memory, disk, chip architecture, cache size, vectorization and parallelization characteristics of the hardware, many of which are already exist within Q-Chem.

B.3.8 USER CONTROLLABLE VARIABLES

AOINTS has been optimally constructed so that the fastest integral algorithm for ERI calculation is chosen for the given integral class and batch. Thus, the user has not been provided with the necessary variables for over-riding the program's selection process. The user is, however, able to control the accuracy of the cutoff used during shell-pair formation (*METECO*) and the integral threshold (*THRESH*). In addition, the user can force the use of the direct SCF algorithm (*DIRECT_SCF*) and increase the default size of the integrals storage buffer (*INCORE_INTS_BUFFER*).

Currently, some of Q-Chem's linear scaling algorithms, such as QCTC and ONX algorithms, require the user to specify their use. It is anticipated that further research developments will lead to the identification of situations in which these, or combinations of these and other algorithms, will be selected automatically by Q-Chem in much the same way that PRISM algorithms choose the most efficient pathway for given integral classes.

B.4 REFERENCES

- [1] P. M. W. Gill, M. Head-Gordon and J. A. Pople, *J. Phys. Chem.* (1990), **94**, 5564.
- [2] P. M. W. Gill, *Advan. Quantum Chem.* (1994), **25**, 142.
- [3] T. R. Adams, R. D. Adamson and P. M. W. Gill, *J. Chem. Phys.* (1997), **107**, 124.
- [4] M. J. Frisch, B. G. Johnson, P. M. W. Gill, D. J. Fox and R. H. Nobes, *Chem. Phys. Lett.* (1993), **206**, 225.
- [5] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Int. J. Quantum Chem.* (1991), **40**, 745.
- [6] P. M. W. Gill and J. A. Pople, *Int. J. Quantum Chem.* (1991), **40**, 753.
- [7] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Chem. Phys. Lett.* (1994), **217**, 65.
- [8] M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* (1988), **89**, 5777.
- [9] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.* (1993), **206**, 229.
- [10] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.* (1993), **206**, 239.
- [11] S. F. Boys, *Proc. Roy. Soc., A* (1950), **A200**, 542.
- [12] S. F. Boys, G. B. Cook, C. M. Reeves and I. Shavitt, *Nature* (1956), **178**, 1207.
- [13] J. A. Pople and W. J. Hehre, *J. Comput. Phys.* (1978), **27**, 161.
- [14] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.* (1994), **230**, 8.
- [15] R. D. Adamson, J. P. Dombroski and P. M. W. Gill, *Chem. Phys. Lett.* (1996), **254**, 329.
- [16] J. P. Dombroski, S. W. Taylor and P. M. W. Gill, *J. Phys. Chem.* (1996), **100**, 6272.
- [17] M. Challacombe and E. Schwegler, *J. Chem. Phys.* (1997), **106**, 5526.
- [18] E. Schwegler and M. Challacombe, *J. Chem. Phys.* (1996), **105**, 2726.
- [19] M. Dupuis, J. Rys and H. F. King, *J. Chem. Phys.* (1976), **65**, 111.
- [20] L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.* (1978), **26**, 218.
- [21] J. Almlöf, K. Faegri and K. Korsell, *J. Comput. Chem.* (1982), **3**, 385.
- [22] S. Obara and A. Saika, *J. Chem. Phys.* (1986), **84**, 3963.
- [23] B. G. Johnson, PhD Thesis. "Development, implementation and performance of efficient methodologies for density functional calculations" Carnegie Mellon: Pittsburgh, 1993.
- [24] E. Schwegler and M. Challacombe, *J. Chem. Phys.* (1996), **106**, 9708.
- [25] S. Obara and A. Saika, *J. Chem. Phys.* (1988), **89**, 1540.
- [26] R. D. Adamson, Honours Thesis. "Shell-pair economisation" Massey University: Palmerston North, 1995.
- [27] M. J. S. Dewar, *Org. Mass. Spec.* (1993), **28**, 305.
- [28] M. J. S. Dewar, *The molecular orbital theory of organic chemistry*; McGraw-Hill: New York, 1969.
- [29] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.* (1996), **253**, 268.
- [30] M. Challacombe, E. Schwegler and J. Almlöf, *J. Chem. Phys.* (1996), **104**, 4685.
- [31] M. Challacombe, E. Schwegler and J. Almlöf, "Modern developments in Hartree-Fock theory: Fast methods for computing the Coulomb matrix," Department of

- Chemistry and Minnesota Supercomputer Institute, University of Minnesota, 1995.
- [32] A. M. Lee, R. D. Adamson and P. M. W. Gill, *to be published*.
 - [33] D. L. Strout and G. E. Scuseria, *J. Chem. Phys.* (1995), **102**, 8448.
 - [34] W. Yang, *Phys. Rev. Lett.* (1991), **66**, 1438.
 - [35] W. Yang, *Phys. Rev. A* (1991), **44**, 7823.
 - [36] W. Yang and T.-S. Lee, *J. Chem. Phys.* (1995), **103**, 5674.
 - [37] T.-S. Lee, D. M. York and W. Yang, *J. Chem. Phys.* (1996), **105**, 2744.
 - [38] C. Ochsenfeld and M. Head-Gordon, *Chem. Phys. Lett.* (1997), **270**, 399.

APPENDIX C \$REM VARIABLE REFERENCE

C.1 INPUT DECK FORMAT

The general format of the *\$rem* input for Q-Chem text input files is simply as follows:

```
$rem  
rem_variable    rem_option    [ comment ]  
rem_variable    rem_option    [ comment ]  
$end
```

This input is not case sensitive.

The following sections contain the names and options of available *\$rem* variables for users. The format for describing each *\$rem* variable is as follows

REM_VARIABLE

Gives a short description of what the variable controls

VARIABLE:

Defines the variable as either INTEGER, LOGICAL or STRING

DEFAULT:

Describes Q-Chem's internal default, if any exist

OPTIONS:

Lists options available for the user

RECOMMENDATION:

Gives a quick recommendation

C.2 THEORETICAL CHEMICAL MODEL

BASIS

Defines the basis sets to be used (unless *ECP* is specified)

VARIABLE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen User defined. (*\$basis* keyword required)

Symbol Use standard basis sets as per Chapter 7

RECOMMENDATIONS:

Consult literature and reviews to aid your selection

CORRELATION

Specifies the correlation level of theory, either DFT or wavefunction-based.

VARIABLE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

None No Correlation

VWN Vosko-Wilk-Nusair parameterization #5

LYP Lee-Yang-Parr (LYP)

PW91, PW GGA91 (Perdew)

LYP(EDF1) LYP(EDF1) parameterization

Perdew86, P86 Perdew 1986

PZ81, PZ Perdew-Zunger 1981

Wigner Wigner

MP2

Local_MP2 local MP2 calculations (TRIM and DIM models)

CIS(D) MP2-level correction to CIS for excited states

MP3

MP4SDQ

MP4

CCD

CCD(2)

CCSD

CCSD(T)

CCSD(2)

QCISD

QCISD(T)

OD

OD(T)

OD(2)

VOD

VOD(2)

QCCD

VQCCD

ECP

Defines the effective core potential and associated basis set to be used

VARIABLE:

STRING

DEFAULT:

No pseudopotential

OPTIONS:

General, Gen User defined. (*\$ecp* keyword required)
 Symbol Use standard pseudopotentials as per Chapter 8

RECOMMENDATIONS:

Pseudopotentials are recommended for first row transition metals and heavier elements. Consult Ch. 8 and reviews for more details.

EXCHANGE

Specifies the exchange level of theory

VARIABLE:

STRING

DEFAULT:

HF for wavefunction correlation methods. Otherwise none.

OPTIONS:

HF	exact Hartree-Fock
Slater, S	Slater
Becke, B	Becke
Gill96, Gill	Gill 1996
GG99	Gilbert and Gill, 1999
Becke(EDF1), B(EDF1)	Becke (EDF1)
PW91, PW	Perdew
B3PW91, Becke3PW91, B3P	B3PW91 hybrid
B3LYP, Becke3LYP	B3LYP
B3LYP5	original B3LYP (using VWN5)
EDF1	EDF1
General, Gen	User defined combination of K, X and C (refer DFT section, Chapter 4)

JOBTYPE

Specifies the type of calculation

VARIABLE:

STRING

DEFAULT:

SP Single point energy

OPTIONS:

SP Single point energy

FORCE Analytical Force calculation

OPT Geometry Minimization

TS Transition Structure Search

FREQ Frequency Calculation

RECOMMENDATION:

Defaults to single point

PURECART

Controls the use of either pure or Cartesian basis functions for general basis sets
(ignored for standard basis sets)

DEFAULT:

None

OPTIONS:

gfd Use 1 for pure and a 2 for Cartesian for each angular momentum
type. Must be defined for user supplied basis sets

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals

VARIABLE:

LOGICAL

DEFAULT:

FALSE (Restricted) Closed-shell systems

TRUE (Unrestricted) Open-shell systems

OPTIONS:

True Unrestricted orbitals

False Restricted open-shell HF (ROHF)

RECOMMENDATION:

Use default unless ROHF is desired

C.3 GENERAL

INCORE_INTS_BUFFER

Controls the size of in-core integral storage buffer

VARIABLE:

INTEGER

DEFAULT:

2,000,000 words (1 word = 8 bytes).

OPTIONS:

User defined size; hardware dependent

MEMORY

Sets the "Fortran" memory for individual program modules.

VARIABLE:

INTEGER

DEFAULT:

2,000,000 (2 MW)

OPTIONS:

User-defined number of words

RECOMMENDATIONS:

MP2 energy/gradient calculations have special requirements (see Ch. 5)

MEMORY_TOTAL

Sets the total memory available to Q-Chem

VARIABLE:

INTEGER

DEFAULT:

Unlimited (1,000 MW)

OPTIONS:

User-defined number of words

RECOMMENDATION:

Use default

METECO

Sets the threshold criteria for discarding shell-pairs

VARIABLE:

INTEGER

DEFAULT:

2 Discard shell-pairs below $10^{-\text{THRESH}}$.

OPTIONS:

1 Discard shell-pairs four orders of magnitude below machine precision

2 Discard shell-pairs below $10^{-\text{THRESH}}$

RECOMMENDATION:

Use default

SYMMETRY

Controls the use of efficiency through the use of point group symmetry

VARIABLE:

LOGICAL

DEFAULT:

TRUE Use symmetry when available

OPTIONS:

TRUE Use symmetry when available

FALSE Do not use symmetry

RECOMMENDATION:

Use default

THRESH

Cutoff for neglect of two electron integrals. $10^{-\text{THRESH}}$ ($\text{THRESH} \leq 12$)

VARIABLE:

INTEGER

DEFAULT:

8 for single point energies

10 for optimizations and frequency calculations

OPTIONS:

User-defined

RECOMMENDATION:

Should be at least 8, and 2 or 3 greater than *SCF_CONVERGENCE*

C.4 SCF GROUND STATE CALCULATIONS

See also *BASIS*, *CORRELATION*, *EXCHANGE* and *JOBTYPE*

BASIS2

Sets the small basis set to use in basis set projection

VARIABLE:

STRING

DEFAULT:

No second basis set default

OPTIONS:

Symbol Use standard basis sets as per Chapter 7

RECOMMENDATIONS:

BASIS2 should be smaller than BASIS

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS subspace during the SCF

VARIABLE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

DIRECT_SCF

Controls direct SCF

VARIABLE:

LOGICAL

DEFAULT:

Determined by program

OPTIONS:

TRUE Forces direct SCF

FALSE Do not use direct SCF

RECOMMENDATION:

Use default. *DIRECT_SCF* switches off in-core integrals

EPAO_ITERATE

Controls iterations for EPAO calculations (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE) Use uniterated EPAO's based on atomic blocks of SPS.

OPTIONS:

0 (FALSE) Use uniterated EPAO's based on atomic blocks of SPS.

n Optimize the EPAO's for up to *n* iterations.

RECOMMENDATION:

Use default. For molecules that are not too large, one can test the sensitivity of the results to the type of minimal functions by the use of optimised EPAO's, in which case a value of $n=500$ is reasonable.

EPAO_WEIGHTS

Controls algorithm and weights for EPAO calculations (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

115 Standard weights, use 1st and 2nd order optimisation

OPTIONS:

15 Standard weights, with 1st order optimisation only.

RECOMMENDATION:

Use default, unless convergence failure is encountered.

FAST_XC

Controls direct variable thresholds to accelerate exchange correlation (XC) in DFT

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE *FAST_XC* is on.

FALSE Do not use *FAST_XC*

RECOMMENDATION:

Caution: *FAST_XC* occasionally causes SCF calculations to diverge

INTEGRAL_2E_OPR

Determines the two-electron operator

VARIABLE:

INTEGER

DEFAULT:

-2 Coulomb Operator

OPTIONS:

-1 Apply the CASE approximation

-2 Coulomb Operator

MAX_DIIS_CYCLES

The maximum number of DIIS iterations before switching to (geometric) direct minimization when *SCF_ALGORITHM* is *DIIS_GDM* or *DIIS_DM*. See also *THRESH_DIIS_SWITCH*.

VARIABLE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM
n *n* DIIS iterations before switching to (G)DM.

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted

VARIABLE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

MOM_ECONOMIZE:

Determines if any computational savings are to be used with MOM

VARIABLE:

INTEGER

DEFAULT:

4

OPTIONS:

1 Include everything
 2 Freeze core electrons
 3 Only use orbitals within 1Eh of the HOMO
 4 Combined frozen core and window

RECOMMENDATION:

Use default

MOM_START

Determines when MOM is switched on to stabilize DIIS iterations

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE) MOM is not used
n MOM begins on cycle *n*

OMEGA

Degree of attenuation of Coulomb operator (see *INTEGRAL_2E_OPR*)

VARIABLE:

INTEGER

DEFAULT:

No default

OPTIONS:

n $\omega = n/1000$

PAO_ALGORITHM

Algorithm used to optimise polarized atomic orbitals (see *PAO_METHOD*)

VARIABLE:

INTEGER

DEFAULT:

0 use efficient (and riskier) strategy to converge PAO's

OPTIONS:

1 use conservative (and slower) strategy to converge PAO's

PAO_METHOD

Controls evaluation of polarized atomic orbitals (PAO's)

VARIABLE:

STRING

DEFAULT:

EPAO For local MP2 calculations

Otherwise no default

OPTIONS:

PAO Perform PAO-SCF instead of conventional SCF

EPAO Obtain EPAO's after a conventional SCF.

PSEUDO_CANONICAL

When *SCF_ALGORITHM = DM*, this controls the way the initial step, and steps after subspace resets are taken.

VARIABLE:

LOGICAL

DEFAULT:

FALSE use Roothaan steps when (re)initializing

OPTIONS:

TRUE use a steepest descent step when (re)initializing

RECOMMENDATION

The default (FALSE) is usually more efficient, but choosing TRUE sometimes avoids problems with orbital reordering.

SCF_ALGORITHM

Algorithm used for converging the SCF

VARIABLE:

STRING

DEFAULT:

DIIS Pulay DIIS

OPTIONS:

DIIS Pulay DIIS

DM Direct minimizer

DIIS_DM Uses DIIS initially, switching to direct minimizer for later iterations (See *THRESH_DIIS_SWITCH*, *MAX_DIIS_CYCLES*)

DIIS_GDM Use DIIS and then later switch to geometric direct minimization (See *THRESH_DIIS_SWITCH*, *MAX_DIIS_CYCLES*)

GDM Geometric Direct Minimization

ROOTHAAN Roothaan repeated diagonalization

RECOMMENDATION:

Use DIIS unless wanting ROHF, in which case direct minimization must be used. If DIIS fails, DIIS_GDM is the recommended fall-back option.

SCF_CONVERGENCE

SCF is considered converged when the wavefunction error is less than $10^{-\text{SCF_CONVERGENCE}}$. Adjust the value of *THRESH* at the same time.

VARIABLE:

INTEGER

DEFAULT:

5 For single point energy calculations

8 for geometry optimizations and vibrational analysis

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost.

SCF_GUESS

Specifies the initial guess procedure to use for the SCF

VARIABLE:

STRING

DEFAULT:

SAD Superposition of atomic density (available only with standard basis sets)

GWH For ROHF where a set of orbitals are required.

OPTIONS:

CORE Diagonalize core Hamiltonian

SAD Superposition of atomic density

GWH Apply generalized Wolfsberg-Helmholtz approximation

READ Read previous MOs from disk

RECOMMENDATION:

SAD guess for standard basis sets. For general basis sets, it is best to use the *BASIS2* REM. Alternatively, try the GWH or core Hamiltonian guess. For ROHF it can be useful to READ guesses from an SCF calculation on the corresponding cation or anion.

SCF_GUESS_ALWAYS

Switch to force the regeneration of a new initial guess for each series of SCF iterations (for use in geometry optimization)

VARIABLE:

LOGICAL

DEFAULT:

False Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available

OPTIONS:

False Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available

True Generate a new guess for each series of SCF iterations in a geometry optimization

SCF_GUESS_MIX

Controls mixing of LUMO and HOMO to break symmetry in the initial guess. For unrestricted jobs, the mixing is performed only for the alpha orbitals.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess

OPTIONS:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess

1 (TRUE) Add 10% of LUMO to HOMO to break symmetry

n Add $n \times 10\%$ of LUMO to HOMO ($0 < n < 10$)

RECOMMENDATION:

When performing unrestricted calculations on molecules with an even number of electrons, it is often necessary to break alpha-beta symmetry in the initial guess with this option, or by specifying input for *\$occupied*.

THRESH_DIIS_SWITCH

The threshold for switching between DIIS extrapolation and direct minimization of the SCF energy is $10^{-\text{THRESH_DIIS_SWITCH}}$ when *SCF_ALGORITHM* is *DIIS_GDM* or *DIIS_DM*. See also *MAX_DIIS_CYCLES*.

VARIABLE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined

XC_GRID

Specifies the type of grid to use for DFT calculations.

DEFAULT:

1 SG-1

OPTIONS:

1 SG-1

2 Low Quality

mn The first six integers correspond to m radial points and the second six integers correspond to n angular points where possible numbers of Lebedev angular points are listed in Chapter 4

$-mn$ The first six integers correspond to m radial points and the second six integers correspond to n angular points where the number of Gauss-Legendre angular points $n = 2N_\theta^2$

RECOMMENDATION:

SG-1

C.5 LARGE MOLECULES

See also *JOBTYPE*, *BASIS*, *EXCHANGE* and *CORRELATION*

CFMM_ORDER

Controls the order of multipole expansions in CFMM calculation

VARIABLE:

INTEGER

DEFAULT:

15 For single point SCF accuracy
25 For tighter convergence

OPTIONS:

n Use multipole expansions of order n

RECOMMENDATION:

Use default

GRAIN

Controls the number of lowest-level boxes in one-dimension for CFMM

VARIABLE:

INTEGER

DEFAULT:

-1 Program decides best value, turning CFMM on when useful

OPTIONS:

-1 Program decides best value, turning CFMM on when useful
1 Do not use CFMM
 $n > 7$ Use CFMM with n lowest-level boxes in one-dimension

RECOMMENDATION:

This is an expert option; either use the default, or use a value of 1 if CFMM is not desired

INCFOCK

Iteration number after which the incremental Fock matrix algorithm is initiated

VARIABLE:

INTEGER

DEFAULT:

1 Start INCFOCK after iteration number 1

OPTIONS:

User-defined (0 switches INCFOCK off)

RECOMMENDATIONS:

May be necessary to allow several iterations before switching on INCFOCK

LIN_K

Controls whether linear scaling evaluation of exact exchange (LinK) is used.

VARIABLE:

LOGICAL

DEFAULT:

Program chooses, switching on LinK whenever CFMM is used.

OPTIONS:

TRUE Use LinK

FALSE Do not use LinK

RECOMMENDATION:

Use for HF and hybrid DFT calculations with large numbers of atoms

VARTHRESH

Controls the temporary integral cutoff threshold.

$\text{tmp_thresh} = 10^{-\text{VARTHRESH}} \times \text{DIIS_error}$

VARIABLE:

INTEGER

DEFAULT:

3

OPTIONS:

User-defined threshold

RECOMMENDATIONS:

3 has found to be a practical level

C.6 CORRELATED METHODS

See also *JOBTYPE*, *BASIS*, *EXCHANGE* and *CORRELATION*. Options beginning with “CC_” are for the coupled cluster module.

CC_BLK_TNSR_BUFFSIZE

Specifies maximum size, MB, of buffers for in-core storage of block-tensors.

VARIABLE:

INTEGER

DEFAULT:

80 MB

RECOMMENDATION:

Larger values can give better i/o performance, and are recommended for systems with large memory (add to your *.qchemrc* file)

CC_CANONIZE

Whether to semi-canonicalize orbitals at the start of the calculation (i.e. Fock matrix is diagonalized in each orbital subspace)

VARIABLE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Should not normally have to be altered.

CC_CANONIZE_FINAL

Whether to semi-canonicalize orbitals at the end of the ground state calculation.

VARIABLE:

LOGICAL

DEFAULT:

FALSE unless required

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Should not normally have to be altered.

CC_CANONIZE_FREQ

The orbitals will be semi-canonicalized every n theta resets. The thetas (orbital rotation angles) are reset every *CC_RESET_THETA* iterations. The counting of iterations differs for active space (VOD, VQCCD) calculations, where the orbitals are always canonicalized at the first theta-reset.

VARIABLE:

INTEGER

DEFAULT:

50

RECOMMENDATION:

Smaller values can be tried in cases that do not converge.

CC_CONVERGENCE

Overall convergence criterion for the coupled cluster codes. This is designed to ensure at least n significant digits in the calculated energy, and automatically sets the other convergence-related variables (*CC_E_CONV*, *CC_T_CONV*, *CC_THETA_CONV*, *CC_THETA_GRAD_CONV*, *CC_Z_CONV*) [$10^{*(-n)}$]

VARIABLE:

INTEGER

DEFAULT:

7	energies
8	gradients

OPTIONS

n	$10^{*(-n)}$ convergence criterion
-----	------------------------------------

CC_DIIS

Specify the version of Pulay's Direct Inversion of the Iterative Subspace (DIIS) convergence accelerator to be used in the coupled cluster code.

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Activates procedure 2 initially, and procedure 1 when gradients are smaller than <i>DIIS12_SWITCH</i> .
1	Uses error vectors defined as differences between parameter vectors from successive iterations. Most efficient near convergence.
2	Error vectors are defined as gradients scaled by square root of the approximate diagonal Hessian. Most efficient far from convergence.

RECOMMENDATION

DIIS1 can be more stable. If DIIS problems are encountered in the early stages of a calculation (when gradients are large) try DIIS1.

CC_DIIS_SIZE

Specifies the maximum size of the DIIS space

VARIABLE:

INTEGER

DEFAULT:

7

RECOMMENDATION

Larger values involve larger amounts of disk storage.

CC_DIIS_START

Iteration number when DIIS is turned on. Set to a large number to disable DIIS.

VARIABLE:

INTEGER

DEFAULT:

3

RECOMMENDATION

Occasionally DIIS can cause optimized orbital coupled cluster calculations to diverge through large orbital changes. If this is seen, DIIS should be disabled.

CC_DIIS12_SWITCH

When to switch from DIIS2 to DIIS1 procedure, or when DIIS2 procedure is required to generate DIIS guesses less frequently. Total value of DIIS error vector must be less than $10^{*(-n)}$, where n is the value of this option.

VARIABLE:

INTEGER

DEFAULT:

5

CC_DIIS_MIN_OVERLAP

The DIIS procedure will be halted when the square root of smallest element of the error overlap matrix is less than $10^{*(-n)}$, where n is the value of this option.

Small values of the B matrix mean it will become near-singular, making the DIIS equations difficult to solve.

VARIABLE:

INTEGER

DEFAULT:

11

CC_DIIS_MAX_OVERLAP

DIIS extrapolations will not begin until square root of the maximum element of the error overlap matrix drops below this value.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{*(-de)}$

DEFAULT:

1.0

CC_DIIS_FREQ

DIIS extrapolation will be attempted every *n* iterations. However, DIIS2 will be attempted every iteration while total error vector exceeds *CC_DIIS12_SWITCH*. DIIS1 cannot generate guesses more frequently than every 2 iterations.

VARIABLE:

INTEGER

DEFAULT:

2

CC_DOV_THRESH

Specifies minimum allowed values for the coupled cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the guess is poor.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{*(-de)}$

DEFAULT:

0.25

RECOMMENDATION:

Increase to 0.5 or 0.75 for nonconvergent coupled cluster calculations.

CC_E_CONV

Convergence desired on the change in total energy, $10^{*(-n)}$, between iterations.

VARIABLE:

INTEGER

DEFAULT:

10

OPTIONS

n $10^{*(-n)}$ convergence criterion

CC_HESS_THRESH

Minimum allowed value for the orbital Hessian. Smaller values are replaced by this constant.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{*(-de)}$

DEFAULT:

0.01

CC_INCL_CORE_CORR

Whether to include the correlation contribution from frozen core orbitals in noniterative (2) corrections, such as OD(2) and CCSD(2).

VARIABLE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

CC_ITERATE_OV

In active space calculations, use a "mixed" iteration procedure if the value is greater than 0. Then, if the RMS orbital gradient is larger than the value of *CC_THETA_GRAD_THRESH*, micro-iterations will be performed to converge the occupied-virtual mixing angles for the current active space. The maximum number of such iterations is given by this option.

VARIABLE:

INTEGER

DEFAULT:

0 (no "mixed" iterations)

OPTIONS:

n up to *n* occupied-virtual iterations per overall cycle

RECOMMENDATION:

Can be useful for nonconvergent active space calculations.

CC_MAXITER

Maximum number of iterations to optimize the coupled cluster energy.

VARIABLE:

INTEGER

DEFAULT:

200

OPTIONS:

n up to *n* iterations to achieve convergence

CC_MP2NO_GUESS

Will guess orbitals be natural orbitals of the MP1 wavefunction? Alternatively, it is possible to use an effective one-particle density matrix to define the natural orbitals.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use natural orbitals from an MP2 one-particle density matrix (see *CC_MP2NO_GRAD*).

FALSE Use current molecular orbitals from SCF

CC_MP2NO_GRAD

If *CC_MP2NO_GUESS* is TRUE, what kind of one-particle density matrix is used to make the guess orbitals?

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE 1 PDM from MP2 gradient theory.

FALSE 1 PDM expanded to 2nd order in perturbation theory.

RECOMMENDATION:

The two definitions give generally similar performance.

CC_ORBS_PER_BLOCK

Specifies target (and maximum) size of blocks in orbital space.

VARIABLE:

INTEGER

DEFAULT:

16

OPTIONS

n Orbital block size of *n* orbitals

CC_PRECONV_FZ

In active space methods, whether to pre-converge other wavefunction variables for fixed initial guess of active space.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS

0 No pre-iterations before active space optimization begins.

n Maximum number of pre-iterations via this procedure.

CC_PRECONV_T2Z

Whether to pre-converge the cluster amplitudes before beginning orbital optimization in optimized orbital cluster methods.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

10 If *CC_RESTART*, *CC_RESTART_NO_SCF* or *CC_MP2NO_GUESS* are TRUE

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to *n* iterations in this pre-convergence procedure.

RECOMMENDATION

Experiment with this option in cases of convergence failure.

CC_PRECONV_T2Z_EACH

Whether to pre-converge the cluster amplitudes before each change of the orbitals in optimized orbital coupled cluster methods. The maximum number of iterations in this pre-convergence procedure is given by the value of this parameter.

VARIABLE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to *n* iterations in this pre-convergence procedure.

RECOMMENDATION:

A very slow last resort option for jobs that do not converge.

CC_QCCD_THETA_SWITCH

QCCD calculations switch from OD to QCCD when the rotation gradient is below this threshold [$10^{*(-n)}$]

VARIABLE:

INTEGER

DEFAULT:

2 $10^{*(-2)}$ switchover

OPTIONS

n $10^{*(-n)}$ switchover

CC_RESET_THETA

The reference MO coefficient matrix is reset every *n* iterations to help overcome problems associated with the theta metric as theta becomes large.

VARIABLE:

INTEGER

DEFAULT:

15

OPTIONS:

n *n* iterations between resetting orbital rotations to zero.

CC_RESTART

Allows an optimized orbital coupled cluster calculation to begin with an initial guess for the orbital transformation matrix U other than the unit vector. The scratch file from a previous run must be available for the U matrix to be read successfully.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE use unit initial guess.

TRUE activates *CC_PRECONV_T2Z*, *CC_CANONIZE*, and turns off *CC_MP2NO_GUESS*

RECOMMENDATION:

Useful for restarting a job that did not converge, if files were saved.

CC_RESTART_NO_SCF

Should an optimized orbital coupled cluster calculation begin with optimized orbitals from a previous calculation? When TRUE, molecular orbitals are initially orthogonalized, and *CC_PRECONV_T2Z* and *CC_CANONIZE* are set to TRUE while other guess options are set to FALSE.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

CC_T_CONV

Convergence criterion on the RMS difference between successive sets of coupled cluster doubles amplitudes [$10^{*(-n)}$]

VARIABLE:

INTEGER

DEFAULT:

8 energies

10 gradients

OPTIONS

n $10^{*(-n)}$ convergence criterion

CC_THETA_CONV

Convergence criterion on the RMS difference between successive sets of orbital rotation angles [$10^{*(-n)}$].

VARIABLE:

INTEGER

DEFAULT:

5 energies

6 gradients

OPTIONS:

n $10^{*(-n)}$ convergence criterion

CC_THETA_GRAD_CONV

Convergence desired on the RMS gradient of the energy with respect to orbital rotation angles [$10^{*(-n)}$].

VARIABLE:

INTEGER

DEFAULT:

7 energies

8 gradients

OPTIONS

n $10^{*(-n)}$ convergence criterion

CC_THETA_GRAD_THRESH

RMS orbital gradient threshold [$10^{*(-n)}$] above which “mixed iterations” are performed in active space calculations if *CC_ITERATE_OV* is TRUE.

VARIABLE:

INTEGER

DEFAULT:

2

OPTIONS:

n $10^{*(-n)}$ threshold

RECOMMENDATION

Can be made smaller if convergence difficulties are encountered.

CC_THETA_STEPSIZE

Scale factor for the orbital rotation step size. The optimal rotation steps should be approximately equal to the gradient vector.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{*(-de)}$

DEFAULT:

1.0 If the initial step is smaller than 0.5, the program will increase step when gradients are smaller than the value of *THETA_GRAD_THRESH*, up to a limit of 0.5.

RECOMMENDATION:

Try a smaller value in cases of poor convergence and very large orbital gradients. For example, a value of 01000 translates to 0.1.

CC_TMPBUFFSIZE

Maximum size, in MB, of additional buffers for temporary arrays used to work with individual blocks or matrices. Should not be smaller than size of the largest possible block. The default is 16 MB.

VARIABLE:

INTEGER

DEFAULT:

16 (MB)

OPTIONS

n (MB)

CC_Z_CONV

Convergence criterion on the RMS difference between successive doubles Z-vector amplitudes [$10^{*(-n)}$]

VARIABLE:

INTEGER

DEFAULT:

8 energies
10 gradients

OPTIONS

n $10^{*(-n)}$ convergence criterion

CD_ALGORITHM

Determines the algorithm for integral transformations

VARIABLE:

STRING

DEFAULT:

Program determined

OPTIONS:

DIRECT Uses fully direct algorithm
SEMI_DIRECT Uses disk-based sortless semi-direct algorithm

RECOMMENDATION:

Use the default

CD_MAX_DISK

Sets the amount of disk space (in words) available for integral transforms

VARIABLE:

INTEGER

DEFAULT:

60,000,000 (60 MW)

OPTIONS:

User-defined

N_FROZEN_CORE

Controls the number of frozen core orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0 No frozen core orbitals

OPTIONS:

FC Frozen core approximation (all core orbitals frozen)
n Freeze *n* core orbitals

N_FROZEN_VIRTUAL

Controls the number of frozen virtual orbitals in a post-Hartree-Fock calculation

VARIABLE:

INTEGER

DEFAULT:

0 No frozen virtual orbitals

OPTIONS:

n Freeze *n* virtual orbitals

C.7 EXCITED STATES

See also *JOBTYPE*, *BASIS*, *EXCHANGE* and *CORRELATION*

CC_DCONVERGENCE

Convergence criterion for the RMS residuals of excited state vectors [$10^{**}(-n)$].

VARIABLE:

INTEGER

DEFAULT:

5

OPTIONS:

n $10^{**}(-n)$ convergence criterion

RECOMMENDATIONS:

Should be consistent with *CC_DTHRESHOLD*.

CC_DO_DISCONNECTED

Are disconnected terms included in the excited state coupled cluster equations?

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATIONS:

Inclusion of disconnected terms has very small effects and is not necessary.

CC_DTHRESHOLD

Specifies threshold for including a new expansion vector in the iterative Davidson diagonalization. Their norm must be above this threshold.

VARIABLE:

DOUBLE Integer code *abcde* is mapped to $abc * 10^{**}(-de)$

DEFAULT:

0.00001

RECOMMENDATIONS:

Should be consistent with *CC_DCONVERGENCE*.

CC_DMAXITER

Maximum number of iteration allowed for Davidson diagonalization procedure.

VARIABLE:

INTEGER

DEFAULT:

30

CC_HOW_DO_DOUBLES

Distinguishes different implementations of active space coupled cluster excited state (VOD) calculations.

VARIABLE:

INTEGER

DEFAULT:

2

OPTIONS:

0

Double excitations for excited states are defined in the whole orbital space. This model is unbalanced and should be used for testing purposes only.

1

Double excitations for the excited states are restricted to an active (valence) orbital space. This approximates linear response CASCCF, with no orbital relaxation.

2

Approximate treatment of orbital relaxation by including all internal and semi-internal excited state double excitations.

CC_NGUESS_DOUBLES

Specifies number of excited state guess vectors which are double excitations.

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

n

Include n guess vectors that are double excitations

RECOMMENDATIONS:

This should be set to the expected number of doubly excited states (see also *CC_PRECONV_DOUBLES*), otherwise they may not be found.

CC_NGUESS_SINGLES

Specifies number of excited state guess vectors that are single excitations.

VARIABLE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

n

Include n guess vectors that are single excitations

RECOMMENDATIONS:

Should be greater or equal than the number of excited states requested.

CC_NHIGHSPIN

Sets the number of coupled cluster high-spin excited state roots to find. Works only for singlet ground state and triplet excited states. The program will increase this number if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations

VARIABLE:

INTEGER

DEFAULT:

0

Do not look for any coupled cluster excited states

OPTIONS:

$n > 0$

Find n CC excited states (after an OD ground state job)

CC_NLOWSPIN

Sets the number of coupled cluster excited state roots to find with the same multiplicity as the ground state. For a spin-unrestricted ground state, (e.g., doublet radicals), this is the total number of states of all multiplicities. The program will increase this number if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations.

VARIABLE:

INTEGER

DEFAULT:

0

Do not look for any coupled cluster excited states

OPTIONS:

$n > 0$

Find n CC excited states (after an OD ground state job)

CC_NVEC_PER_ROOT

Specifies maximum number of vectors per root in Davidson' diagonalization.

VARIABLE:

INTEGER

DEFAULT:

6

OPTIONS:

n

Up to n vectors per root before the subspace is reset

RECOMMENDATIONS:

Larger values increase storage but speed convergence.

CC_PRECONV_DOUBLES

When TRUE, doubly-excited vectors are converged prior to a full excited states calculation.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATIONS:

Occasionally necessary to ensure a doubly excited state is found.

CC_PRECONV_SINGLES

When TRUE, singly-excited vectors are converged prior to a full excited states calculation.

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

CD_ALGORITHM

Determines the algorithm for integral transformations

VARIABLE:

STRING

DEFAULT:

Program-determined

OPTIONS:

DIRECT

Uses fully direct algorithm

SEMI_DIRECT

Uses disk-based sortless semi-direct algorithm

RECOMMENDATION:

Use the default

CD_MAX_DISK

Sets the amount of disk space (in words) available for integral transforms

VARIABLE:

INTEGER

DEFAULT:

60,000,000 (60 MW)

OPTIONS:

User-defined

CIS_CONVERGENCE

CIS is considered converged when error is less than $10^{-\text{CIS_CONVERGENCE}}$

VARIABLE:

INTEGER

DEFAULT:

6 CIS convergence threshold $\equiv 10^{-6}$

OPTIONS:

User-defined

CIS_GUESS_DISK

Read the CIS guess from disk (previous calculation).

VARIABLE:

LOGICAL

DEFAULT:

False Create a new guess

OPTIONS:

False Create a new guess

True Read the guess from disk

RECOMMENDATIONS:

Requires a guess from previous calculation

CIS_GUESS_DISK_TYPE

Determines the type of guesses to be read from disk

VARIABLE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATIONS:

Must be specified if *CIS_GUESS_DISK* is *TRUE*

CIS_N_ROOTS

Sets the number of CI-Singles (CIS) excited state roots to find

VARIABLE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

$n > 0$ Looks for n CIS excited states

CIS_RELAXED_DENSITY

Use the relaxed CIS density for attachment/detachment density analysis

VARIABLE:

LOGICAL

DEFAULT:

False Do not use the relaxed CIS density in analysis

OPTIONS:

False Do not use the relaxed CIS density in analysis

True Use the relaxed CIS density in analysis

CIS_SINGLET

Solve for singlet excited states in RCIS calculations (ignored for UCIS)

VARIABLE:

LOGICAL

DEFAULT:

True Solve for singlet states

OPTIONS:

True Solve for singlet states

False Do not solve for singlet states

CIS_STATE_DERIV

Sets which state to determine CIS gradient for excited state optimizations

VARIABLE:

INTEGER

DEFAULT:

0 Does not select any of the excited states

OPTIONS:

n Select the n^{th} state

CIS_TRIPLETS

Solve for triplet excited states in RCIS calculations (ignored for UCIS)

VARIABLE:

LOGICAL

DEFAULT:

True Solve for triplet states

OPTIONS:

True Solve for triplet states

False Do not solve for triplet states

MAX_CIS_CYCLES

Maximum number of CIS iterative cycles allowed

VARIABLE:

INTEGER

DEFAULT:

30

OPTIONS:

User-defined

RECOMMENDATIONS:

Default is usually sufficient

N_FROZEN_CORE

Controls the number of frozen core orbitals

VARIABLE:

INTEGER

DEFAULT:

0 No frozen core orbitals

OPTIONS:

FC Frozen core approximation

n Freeze *n* core orbitals

N_FROZEN_VIRTUAL

Controls the number of frozen virtual orbitals

VARIABLE:

INTEGER

DEFAULT:

0 No frozen virtual orbitals

OPTIONS:

n Freeze *n* virtual orbitals

RPA

Do an RPA calculation in addition to a CIS calculation

VARIABLE:

LOGICAL

DEFAULT:

False Do not do an RPA calculation

OPTIONS:

False Do not do an RPA calculation

True Do an RPA calculation

XCIS

Do an XCIS calculation in addition to a CIS calculation

VARIABLE:

LOGICAL

DEFAULT:

False Do not do an XCIS calculation

OPTIONS:

False Do not do an XCIS calculation

True Do an XCIS calculation (requires ROHF ground state)

C.8 MOLECULAR GEOMETRY CRITICAL POINTS

See also *JOBTYPE*, *BASIS*, *EXCHANGE* and *CORRELATION*

GEOM_OPT_COORDS

Controls the type of optimization coordinates

VARIABLE

INTEGER

DEFAULT:

-1 Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue

OPTIONS:

0 Optimize in Cartesian coordinates
 1 Generate and optimize in internal coordinates, if this fails abort
 -1 Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue
 2 Optimize in Z-matrix coordinates, if this fails abort
 -2 Optimize in Z-matrix coordinates, if this fails during any stage of the optimization switch to Cartesians and continue

RECOMMENDATION:

Use the default; delocalized internals are more efficient

GEOM_OPT_DMAX

Maximum allowed step size. Value supplied is multiplied by 10^{-3}

VARIABLE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

n User-defined cutoff

RECOMMENDATION:

Use default

GEOM_OPT_HESSIAN

Hessian status

VARIABLE:

INTEGER

DEFAULT:

DIAGONAL Set up (default) diagonal Hessian

OPTIONS:

DIAGONAL Set up (default) diagonal Hessian

READ Have exact or initial Hessian. Use as is if Cartesian or transform if internals

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees)

VARIABLE:

INTEGER

DEFAULT:

165 degrees

OPTIONS:

n User-defined level

RECOMMENDATION:

Use default

GEOM_OPT_MAX_CYCLES

Maximum number of optimisation cycles

VARIABLE:

INTEGER

DEFAULT:

20

OPTIONS:

User defined positive integer

RECOMMENDATION:

Use the default, increase for difficult cases

GEOM_OPT_MAX_DIIS

Controls maximum size of subspace for GDIIS

VARIABLE:

INTEGER

DEFAULT:

0 Do not use GDIIS

OPTIONS:

0 Do not use GDIIS

-1 Default size = min(NDEG, NATOMS, 4) NDEG = number of
molecular degrees of freedom

n Size specified by user

RECOMMENDATION:

Use default or do not set *n* too large

GEOM_OPT_MODE

Determines Hessian mode followed during TS search

VARIABLE:

INTEGER

DEFAULT:

0 Mode following off

OPTIONS:

0 Mode following off

n Maximise along mode *n*

RECOMMENDATION:

Use default

GEOM_OPT_TOL_DISPLACEMENT

Convergence on maximum atomic displacement

VARIABLE:

INTEGER

DEFAULT:

1200 $\equiv 1200 \times 10^{-6}$ tolerance on maximum atomic displacement

OPTIONS:

Integer value (tolerance = value $\times 10^{-6}$)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_TOL_ENERGY

Convergence on energy change of successive optimisation cycles

VARIABLE:

INTEGER

DEFAULT:

100 $\equiv 100 \times 10^{-8}$ tolerance on maximum gradient component

OPTIONS:

Integer value (tolerance = value $\times 10^{-8}$)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_TOL_GRADIENT

Convergence on maximum gradient component

VARIABLE:

INTEGER

DEFAULT:

300 ≡ 300 x 10⁻⁶ tolerance on maximum gradient component

OPTIONS:

Integer value (tolerance = value x 10⁻⁶)

RECOMMENDATION:

Use the default. To converge *GEOM_OPT_TOL_GRADIENT* and one of *GEOM_OPT_TOL_DISPLACEMENT* and *GEOM_OPT_TOL_ENERGY* must be satisfied

GEOM_OPT_SYMFLAG

Controls the use of symmetry in OPTIMIZE

VARIABLE:

INTEGER

DEFAULT:

1 Make use of point group symmetry

OPTIONS:

1 Make use of point group symmetry

0 Do not make use of point group symmetry

RECOMMENDATION:

Use default

GEOM_OPT_UPDATE

Controls the Hessian update algorithm

VARIABLE:

INTEGER

DEFAULT:

-1 Use the default update algorithm

OPTIONS:

-1 Use the default update algorithm

0 Do not update the Hessian (not recommended)

1 Murtagh-Sargent update

2 Powell update

3 Powell-Murtagh-Sargent update (TS default)

4 BFGS update (OPT default)

5 BFGS with safeguards to ensure retention of positive definiteness (GDISS default)

RECOMMENDATION:

Use default

C.9 MOLECULAR PROPERTIES AND ANALYSIS

CHEMSOL

Controls the use of ChemSol in Q-Chem

VARIABLE:

INTEGER

DEFAULT:

0 Do not use ChemSol

OPTIONS:

1 Perform a ChemSol calculation

CHEMSOL_EFIELD

Determines how the solute charge distribution is approximated in evaluating the electrostatic field of the solute.

VARIABLE:

INTEGER

DEFAULT:

1 Exact solute charge distribution is used.

OPTION:

0 Solute charge distribution is approximated by Mulliken atomic charges. This is a faster, but less rigorous procedure.

CHEMSOL_NN

Sets the number of grids used to calculate the average hydration free energy.

VARIABLE:

INTEGER

DEFAULT:

5 ΔG_{hydr} will be averaged over 5 different grids

OPTIONS:

n number of different grids (Max = 20).

DORAMAN

Controls calculation of Raman intensities. Requires *JOBTYPE* to be set to *FREQ*

VARIABLE:

LOGICAL

DEFAULT:

FALSE Do not calculate Raman intensities

OPTIONS:

FALSE Do not calculate Raman intensities

TRUE Do calculate Raman intensities

IGDESP

Controls evaluation of the electrostatic potential on a grid of points. If enabled, the output is in an ACSII file, plot.esp, in the format x,y,z,esp for each point.

VARIABLE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1 read grid input via the *\$plots* section of the input deck
 +*n* read *n* grid points from the ACSII file ESPGrid

INTRACULE

Controls whether intracule properties are calculated (see also *\$intracule*)

VARIABLE:

LOGICAL

DEFAULT:

FALSE No intracule properties

OPTIONS:

TRUE Evaluate intracule properties

MULTIPOLE_ORDER

Determines highest order to of multipole moments to print if wave function analysis requested

VARIABLE:

INTEGER

DEFAULT:

4

OPTIONS:

n Calculate moments to *n*th order

NBO

Controls the use of the NBO package

VARIABLE:

LOGICAL

DEFAULT:

FALSE Do not invoke the NBO package

OPTIONS:

FALSE Do not invoke the NBO package
 TRUE Do invoke the NBO package

POP_MULLIKEN

Controls running of Mulliken population analysis

VARIABLE:

LOGICAL/INTEGER

DEFAULT:

TRUE (1) Calculate Mulliken population

OPTIONS:

FALSE (0) Do not calculate Mulliken Population

TRUE (1) Calculate Mulliken population

2 Also calculate shell populations for each occupied orbital.

RECOMMENDATIONS:

TRUE. Trivial additional calculation

READ_VDW

Controls the input of user-defined atomic radii for ChemSol calculation

VARIABLE:

LOGICAL

DEFAULT:

FALSE Use default ChemSol parameters

OPTIONS:

TRUE Read from the *\$van_der_waals* section of the input file

SOLUTE_RADIUS

Sets the Onsager solvent model cavity radius

VARIABLE:

INTEGER $a_o = \text{SOLUTE_RADIUS}/10000$

DEFAULT:

No default

OPTIONS:

User-defined

RECOMMENDATION:

Use equation (8.1)

SOLVENT_DIELECTRIC

Sets the dielectric constant of the Onsager solvent continuum

VARIABLE:

INTEGER $\epsilon = \text{SOLVENT_DIELECTRIC}/10000$

DEFAULT:

No default

OPTIONS:

User-defined

RECOMMENDATION:

As per required solvent

STEWART

Controls whether a Stewart atom analysis is performed (see also *\$stewart* and *\$stewart_auxbasis*)

VARIABLE:

LOGICAL

DEFAULT:

FALSE No Stewart analysis

OPTIONS:

TRUE Perform Stewart analysis

SYMMETRY_DECOMPOSITION

Determines symmetry decompositions to calculate

VARIABLE:

INTEGER

DEFAULT:

1 Calculate MO eigenvalues and symmetry (if available)

OPTIONS:

0 No symmetry decomposition

1 Calculate MO eigenvalues and symmetry (if available)

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices

WAVEFUNCTION_ANALYSIS

Controls the running of the default wave function analysis

VARIABLE:

LOGICAL

DEFAULT:

TRUE Perform default wave function analysis

OPTIONS:

TRUE Perform default wave function analysis.

FALSE Do not perform default wave function analysis

C.10 PRINTING**CC_PRINT**

Controls the output from post-MP2 coupled cluster module of Q-Chem

VARIABLE:

INTEGER

DEFAULT:

1

OPTIONS:

0 → 7 Higher values can lead to deforestation...

CHEMSOL_PRINT

Controls printing in the ChemSol part of the Q-Chem output file.

VARIABLE:

INTEGER

DEFAULT:

0

Limited printout

OPTIONS:

1

Full printout.

DIIS_PRINT

Controls the output from DIIS SCF optimization

VARIABLE:

INTEGER

DEFAULT:

0

OPTIONS:

0

1

Chosen method and DIIS coefficients & solutions

2

Level 1 plus changes in multipole moments

3

Level 2 plus Multipole moments

4

Level 3 plus extrapolated Fock matrices

MOM_PRINT

Switches printing on within the MOM procedure

VARIABLE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE

Printing is turned off

TRUE

Printing is turned on

GEOM_OPT_PRINT

Amount of OPTIMIZE print output

VARIABLE:

INTEGER

DEFAULT:

3 Error messages, summary, warning, standard information and
gradient print out

OPTIONS:

0 Error messages only
1 Level 0 plus summary and warning print out
2 Level 1 plus standard information
3 Level 2 plus gradient print out
4 Level 3 plus hessian print out
5 Level 4 plus iterative print out
6 Level 5 plus internal generation print out
7 Debug print out

RECOMMENDATION:

Use the default

PRINT_GENERAL_BASIS

Controls print out of built in basis sets in input format

VARIABLE:

LOGICAL

DEFAULT:

False Do not print out standard basis set information

OPTIONS:

TRUE Print out standard basis set information
FALSE Do not print out standard basis set information

RECOMMENDATIONS:

Useful for modification of standard basis sets

PRINT_ORBITALS

Prints orbital coefficients with atom labels in analysis part of output.

VARIABLE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Prints occupied orbitals plus 5 virtuals.
NVIRT Number of virtuals to print.

RECOMMENDATION:

Use TRUE unless more virtuals are desired.

SCF_FINAL_PRINT

Controls level of output from SCF procedure to Q-Chem output file at the end of the SCF

VARIABLE:

INTEGER

DEFAULT:

0 No extra print out

OPTIONS:

0 No extra print out

1 Orbital Energies only

2 Level 1 plus MOs

3 Level 2 plus Fock and density matrices

SCF_GUESS_PRINT

Controls printing of guess MOs, Fock and density matrices

VARIABLE:

INTEGER

DEFAULT:

0 Do not print guesses

OPTIONS:

0 Do not print guesses

SAD

1 Atomic density matrices and molecular matrix

2 Level 1 plus density matrices

CORE and GWH

1 No extra output

2 Level 1 plus Fock and density matrices and, MO coefficients and eigenvalues

READ

1 No extra output

2 Level 1 plus density matrices, MO coefficients and eigenvalues

SCF_PRINT

Controls level of output from SCF procedure to Q-Chem output file

VARIABLE:

INTEGER

DEFAULT:

0 Minimal, concise, useful and necessary output

OPTIONS:

0 Minimal, concise, useful and necessary output

1 Level 0 plus component breakdown of SCF electronic energy

2 Level 1 plus density, Fock and MO matrices on each cycle

3 Level 2 plus two-electron Fock matrix components (Coulomb, HF exchange and DFT exchange-correlation matrices) on each cycle

RECOMMENDATION:

Proceed with care; can result in *extremely* large output files at level 2 or higher

VIBMAN_PRINT

Controls level of extra print out for vibrational analysis

VARIABLE:

INTEGER

DEFAULT:

1 Standard full information print out

OPTIONS:

1 Standard full information print out

3 Level 1 plus vibrational frequencies in atomic units

4 Level 3 plus mass weighted Hessian matrix, projected mass-weighted Hessian matrix

6 Level 4 plus vectors for translations and rotations, projection matrix

RECOMMENDATION:

Use default