

# Q-Chem User's Manual

Version 3.1  
March 2007

# Version 3.1

March 2007

## Q-Chem User's Guide

### This edition edited by:

Dr Andrew Gilbert

### Contributions from:

Greg Beran	(Coupled-cluster active space methods)
Prof. Dan Chipman and Dr Shawn T. Brown	(SS(V)PE solvation model)
Dr Laszlo Fusti-Molnar	(Fourier Transform Coulomb Method)
Prof. Martin Head-Gordon	(Auxiliary bases, SOS MP2, perfect and imperfect pairing)
Dr John Herbert	( <i>Ab initio</i> dynamics, Born-Oppenheimer dynamics)
Dr Jing Kong	(Fast XC calculations)
Prof. Anna Krylov	(EOM methods)
Dr Joerg Kussman and Prof. Dr Christian Ochsenfeld	(Linear scaling NMR and optical properties)
Dr Ching Yeh Lin	(Anharmonic Corrections)
Rohini Lochan	(SOS and MOS-MP2)
Prof. Vitaly Rassolov	(Geminal Models)
Ryan Steele	(Dual basis methods)
Dr Yihan Shao	(Integral algorithm improvements, QM-MM and improved TS finder)

### This is a revised and expanded version of the previous (2.1) edition, written by:

Dr Jeremy Dombroski  
Prof. Martin Head-Gordon  
Dr Andrew Gilbert

### Published by:

Q-Chem, Inc.  
5001 Baum Blvd  
Suite 690  
Pittsburgh, PA 15213

### Customer Support:

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



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	About this Manual . . . . .	1
1.2	Chapter Summaries . . . . .	1
1.3	Contact Information . . . . .	2
1.3.1	Customer Support . . . . .	2
1.4	Q-CHEM, Inc. . . . .	2
1.5	Company Mission . . . . .	2
1.6	Q-CHEM Features . . . . .	3
1.6.1	New Features in Q-CHEM 3.0 . . . . .	3
1.6.2	New Features in Q-CHEM 3.1 . . . . .	5
1.6.3	Summary of Existing Methods and Features . . . . .	5
1.7	Highlighted Features . . . . .	7
1.7.1	COLD PRISM . . . . .	7
1.7.2	Continuous Fast Multipole Method (CFMM) . . . . .	7
1.7.3	Parallel Computing . . . . .	7
1.7.4	Local MP2 . . . . .	8
1.7.5	High Level Coupled Cluster Methods . . . . .	8
1.7.6	Continuum Solvation Models . . . . .	8
1.7.7	OPTIMIZE . . . . .	8
1.7.8	SPARTAN . . . . .	9
1.8	Current Development and Future Releases . . . . .	9
1.9	Citing Q-CHEM . . . . .	9
<b>2</b>	<b>Installation</b>	<b>10</b>
2.1	Q-CHEM Installation Requirements . . . . .	10
2.1.1	Execution Environment . . . . .	10
2.1.2	Hardware Platforms and Operating Systems . . . . .	10
2.1.3	Memory and Hard Disk . . . . .	10

---

2.2	Installing Q-CHEM . . . . .	11
2.3	Environment Variables . . . . .	12
2.4	User Account Adjustments . . . . .	12
2.4.1	Example <i>.login</i> File Modifications . . . . .	12
2.5	The <i>qchem.setup</i> File . . . . .	13
2.6	Running Q-CHEM . . . . .	13
2.6.1	Serial Q-CHEM . . . . .	14
2.6.2	Parallel Q-CHEM . . . . .	14
2.7	Testing and Exploring Q-CHEM . . . . .	15
<b>3</b>	<b>Q-Chem Inputs</b> . . . . .	<b>16</b>
3.1	General Form . . . . .	16
3.2	Molecular Coordinate Input ( <i>\$molecule</i> ) . . . . .	17
3.2.1	Reading Molecular Coordinates From a Previous Calculation . . . . .	18
3.2.2	Reading molecular Coordinates from another file . . . . .	19
3.3	Cartesian Coordinates . . . . .	19
3.3.1	Examples . . . . .	19
3.4	Z-matrix Coordinates . . . . .	20
3.4.1	Dummy atoms . . . . .	22
3.5	Job Specification: The <i>\$rem</i> Array Concept . . . . .	22
3.6	<i>\$rem</i> Array Format in Q-CHEM Input . . . . .	23
3.7	Minimum <i>\$rem</i> Array Requirements . . . . .	23
3.8	User-defined basis set ( <i>\$basis</i> ) . . . . .	24
3.9	Comments ( <i>\$comment</i> ) . . . . .	24
3.10	User-defined Pseudopotentials ( <i>\$ecp</i> ) . . . . .	24
3.11	Addition of External Charges ( <i>\$external_charges</i> ) . . . . .	24
3.12	Intracules ( <i>\$intracule</i> ) . . . . .	24
3.13	Isotopic substitutions ( <i>\$isotopes</i> ) . . . . .	24
3.14	Applying a Multipole Field ( <i>\$multipole_field</i> ) . . . . .	25
3.15	Natural Bond Orbital Package ( <i>\$nbo</i> ) . . . . .	25
3.16	User-defined occupied guess orbitals ( <i>\$occupied</i> ) . . . . .	25
3.17	Geometry Optimization with General Constraints ( <i>\$opt</i> ) . . . . .	25
3.18	SS(V)PE Solvation Modeling ( <i>\$svp</i> and <i>\$svpirf</i> ) . . . . .	25
3.19	Orbitals, Densities and ESPs On a Mesh ( <i>\$plots</i> ) . . . . .	26
3.20	User-defined Van der Waals Radii ( <i>\$van_der_waals</i> ) . . . . .	26
3.21	User-defined exchange-correlation Density Functionals ( <i>\$xc_functional</i> ) . . . . .	26

---

3.22	Multiple Jobs in a Single File: Q-CHEM Batch Job Files . . . . .	26
3.23	Q-CHEM Output File . . . . .	28
3.24	Q-CHEM Scratch Files . . . . .	29
<b>4</b>	<b>Self-Consistent Field Ground State Methods</b>	<b>30</b>
4.1	Introduction . . . . .	30
4.1.1	Overview of Chapter . . . . .	30
4.1.2	Theoretical Background . . . . .	31
4.2	Hartree-Fock Calculations . . . . .	33
4.2.1	The Hartree-Fock Equations . . . . .	33
4.2.2	Wavefunction Stability Analysis . . . . .	35
4.2.3	Basic Hartree-Fock Job Control . . . . .	36
4.2.4	Additional Hartree-Fock Job Control Options . . . . .	39
4.2.5	Examples . . . . .	41
4.2.6	Symmetry . . . . .	42
4.3	Density Functional Theory . . . . .	43
4.3.1	Introduction . . . . .	43
4.3.2	Kohn-Sham Density Functional Theory . . . . .	44
4.3.3	Exchange-Correlation Functionals . . . . .	45
4.3.4	DFT Numerical Quadrature . . . . .	47
4.3.5	Angular Grids . . . . .	48
4.3.6	Standard Quadrature Grids . . . . .	48
4.3.7	Consistency Check and Cutoffs for Numerical Integration . . . . .	49
4.3.8	Basic DFT Job Control . . . . .	50
4.3.9	User-Defined Density Functionals . . . . .	53
4.3.10	Example . . . . .	53
4.4	Large Molecules and Linear Scaling Methods . . . . .	54
4.4.1	Introduction . . . . .	54
4.4.2	Continuous Fast Multipole Method (CFMM) . . . . .	54
4.4.3	Linear Scaling Exchange (LinK) Matrix Evaluation . . . . .	56
4.4.4	Incremental and Variable Thresh Fock Matrix Building . . . . .	57
4.4.5	Incremental DFT . . . . .	58
4.4.6	Fourier Transform Coulomb Method . . . . .	60
4.4.7	Examples . . . . .	61
4.5	SCF Initial Guess . . . . .	62
4.5.1	Introduction . . . . .	62

---

4.5.2	Simple Initial Guesses . . . . .	63
4.5.3	Reading MOs from Disk . . . . .	64
4.5.4	Modifying the Occupied Molecular Orbitals . . . . .	65
4.5.5	Basis Set Projection . . . . .	66
4.5.6	Examples . . . . .	67
4.6	Converging SCF Calculations . . . . .	69
4.6.1	Introduction . . . . .	69
4.6.2	Basic Convergence Control Options . . . . .	69
4.6.3	Direct Inversion in the Iterative Subspace (DIIS) . . . . .	70
4.6.4	Geometric Direct Minimization (GDM) . . . . .	72
4.6.5	Direct Minimization (DM) . . . . .	74
4.6.6	Maximum Overlap Method (MOM) . . . . .	74
4.6.7	Examples . . . . .	75
4.7	Dual-Basis Self-Consistent Field Calculations . . . . .	77
4.7.1	Examples . . . . .	78
4.8	Unconventional SCF Calculations . . . . .	78
4.8.1	CASE Approximation . . . . .	78
4.8.2	Polarized Atomic Orbital (PAO) Calculations . . . . .	79
4.9	Ground State Method Summary . . . . .	81
<b>5</b>	<b>Wavefunction-based Correlation Methods</b>	<b>86</b>
5.1	Introduction . . . . .	86
5.2	Møller-Plesset Perturbation Theory . . . . .	88
5.2.1	Introduction . . . . .	88
5.2.2	Theoretical Background . . . . .	88
5.3	Exact MP2 Methods . . . . .	89
5.3.1	Algorithm . . . . .	89
5.3.2	The Definition of Core Electron . . . . .	91
5.3.3	Algorithm Control and Customization . . . . .	91
5.3.4	Example . . . . .	94
5.4	Local MP2 Methods . . . . .	94
5.4.1	Local Triatomics in Molecules (TRIM) Model . . . . .	94
5.4.2	EPAO Evaluation Options . . . . .	96
5.4.3	Algorithm Control and Customization . . . . .	97
5.4.4	Examples . . . . .	98
5.5	Auxiliary Basis Set (Resolution of the Identity) MP2 Methods. . . . .	99

---

5.5.1	RI-MP2 energies and gradients. . . . .	100
5.5.2	Example . . . . .	100
5.5.3	Opposite spin (SOS-MP2 and MOS-MP2) energies and gradients. . . . .	101
5.5.4	Examples . . . . .	102
5.5.5	RI-TRIM MP2 Energies . . . . .	103
5.6	Self-Consistent Pair Correlation Methods . . . . .	103
5.6.1	Coupled Cluster Singles and Doubles (CCSD) . . . . .	104
5.6.2	Quadratic Configuration Interaction (QCISD) . . . . .	105
5.6.3	Optimized Orbital Coupled Cluster Doubles (OD) . . . . .	105
5.6.4	Quadratic Coupled Cluster Doubles (QCCD) . . . . .	106
5.6.5	Job Control Options . . . . .	107
5.6.6	Example . . . . .	109
5.7	Non-iterative Corrections to Coupled Cluster Energies . . . . .	110
5.7.1	(T) Triples Corrections . . . . .	110
5.7.2	(2) Triples and Quadruples Corrections . . . . .	110
5.7.3	Job Control Options . . . . .	111
5.7.4	Example . . . . .	111
5.8	Coupled Cluster Active Space Methods . . . . .	112
5.8.1	Introduction . . . . .	112
5.8.2	VOD and VOD(2) Methods . . . . .	113
5.8.3	VQCCD . . . . .	114
5.8.4	Convergence Strategies and More Advanced Options . . . . .	114
5.8.5	Examples . . . . .	116
5.9	Simplified Coupled-Cluster Methods Based on a Perfect Pairing Active Space. . .	118
5.10	Geminal models . . . . .	124
5.10.1	Reference wavefunction . . . . .	124
5.10.2	Perturbative corrections . . . . .	126
<b>6</b>	<b>Open-Shell and Excited State Methods</b>	<b>130</b>
6.1	General Excited State Features . . . . .	130
6.2	Non-Correlated Wavefunction Methods . . . . .	132
6.2.1	Single Excitation Configuration Interaction (CIS) . . . . .	132
6.2.2	Random Phase Approximation (RPA) . . . . .	133
6.2.3	Extended CIS (XCIS) . . . . .	133
6.2.4	Basic Job Control Options . . . . .	135
6.2.5	Customization . . . . .	136



---

6.2.6	CIS Analytical Derivatives . . . . .	138
6.2.7	Examples . . . . .	139
6.3	Time-Dependent Density Functional Theory (TDDFT) . . . . .	141
6.3.1	A Brief Introduction to TDDFT . . . . .	141
6.3.2	TDDFT within a Reduced Single Excitation Space . . . . .	142
6.3.3	Job Control for TDDFT . . . . .	142
6.3.4	Example . . . . .	144
6.4	Correlated Excited State Methods, CIS(D) . . . . .	146
6.4.1	CIS(D) Job Control . . . . .	148
6.4.2	Examples . . . . .	150
6.4.3	Resolution of the Identity CIS(D) Methods . . . . .	150
6.4.4	Example . . . . .	151
6.5	Coupled-Cluster Excited-State and Open-Shell Methods . . . . .	152
6.5.1	Excited states by EOM-EE-CCSD and EOM-EE-OD . . . . .	152
6.5.2	EOM-XX-CCSD suit of methods . . . . .	153
6.5.3	Spin-Flip Methods for Di- and Triradicals . . . . .	153
6.5.4	EOM-DIP-CCSD . . . . .	155
6.5.5	Equation-of-Motion Coupled-Cluster Job Control . . . . .	155
6.5.6	Examples . . . . .	160
6.5.7	Analytic gradients for the CCSD and EOM-XX-CCSD methods . . . . .	161
6.5.8	Properties for CCSD and EOM-CCSD wavefunctions . . . . .	162
6.5.9	Equation-of-Motion Coupled-Cluster Optimization and Properties Job Control	162
6.5.10	Examples . . . . .	167
6.5.11	EOM(2,3) methods for higher accuracy and problematic situations . . . . .	169
6.5.12	Active space EOM-CC(2,3) . . . . .	170
6.5.13	Job Control for EOM-(2,3) . . . . .	171
6.5.14	Examples . . . . .	173
6.6	Potential Energy Surface Crossing Minimization . . . . .	175
6.6.1	Job Control Options . . . . .	175
6.6.2	Examples . . . . .	176
6.7	Dyson Orbitals for Ionization from the ground and electronically excited states within EOM-CCSD formalism . . . . .	177
6.7.1	Dyson Orbitals Job Control . . . . .	178
6.7.2	Examples . . . . .	179
6.8	Attachment/Detachment Density Analysis . . . . .	180

---

<b>7</b>	<b>Basis Sets</b>	<b>184</b>
7.1	Introduction . . . . .	184
7.2	Built-in Basis sets . . . . .	185
7.3	Basis Set Symbolic Representation . . . . .	185
7.3.1	Customization . . . . .	188
7.4	User-defined Basis Sets ( <i>\$basis</i> ) . . . . .	189
7.4.1	Introduction . . . . .	189
7.4.2	Job Control . . . . .	189
7.4.3	Format For User-defined Basis Sets . . . . .	189
7.4.4	Example . . . . .	190
7.5	Mixed Basis Sets . . . . .	191
7.5.1	Examples . . . . .	191
7.6	Basis Set Superposition Error (BSSE) . . . . .	193
<b>8</b>	<b>Effective Core Potentials</b>	<b>196</b>
8.1	Introduction . . . . .	196
8.2	Built-In Pseudopotentials . . . . .	197
8.2.1	Overview . . . . .	197
8.2.2	Combining Pseudopotentials . . . . .	197
8.2.3	Examples . . . . .	198
8.3	User-Defined Pseudopotentials . . . . .	199
8.3.1	Job Control for User-Defined ECP's . . . . .	199
8.3.2	Example . . . . .	199
8.4	Pseudopotentials and Density Functional Theory . . . . .	200
8.4.1	Example . . . . .	200
8.5	Pseudopotentials and Electron Correlation . . . . .	201
8.5.1	Example . . . . .	201
8.6	Pseudopotentials and Vibrational Frequencies . . . . .	202
8.6.1	Example . . . . .	202
8.6.2	A Brief Guide to Q-CHEM's Built-in ECP's . . . . .	203
8.6.3	The HWMB Pseudopotential at a Glance . . . . .	204
8.6.4	The LANL2DZ Pseudopotential at a Glance . . . . .	205
8.6.5	The SBKJC Pseudopotential at a Glance . . . . .	206
8.6.6	The CRENBS Pseudopotential at a Glance . . . . .	207
8.6.7	The CRENBL Pseudopotential at a Glance . . . . .	207
8.6.8	The SRLC Pseudopotential at a Glance . . . . .	209

---

8.6.9	The SRSC Pseudopotential at a Glance . . . . .	211
<b>9</b>	<b>Molecular Geometry Critical Points</b>	<b>213</b>
9.1	Equilibrium Geometries and Transition Structures . . . . .	213
9.2	User-controllable Parameters . . . . .	214
9.2.1	Features . . . . .	214
9.2.2	Job Control . . . . .	214
9.2.3	Customization . . . . .	217
9.2.4	Example . . . . .	219
9.3	Constrained Optimization . . . . .	220
9.3.1	Introduction . . . . .	220
9.3.2	Geometry Optimization with General Constraints . . . . .	220
9.3.3	Frozen Atoms . . . . .	221
9.3.4	Dummy Atoms . . . . .	221
9.3.5	Dummy Atom Placement in Dihedral Constraints . . . . .	222
9.3.6	Additional Atom Connectivity . . . . .	223
9.3.7	Example . . . . .	223
9.3.8	Summary . . . . .	224
9.4	Intrinsic Reaction Coordinates . . . . .	224
9.4.1	Job control . . . . .	225
9.4.2	Example . . . . .	226
9.5	The Growing String Method . . . . .	227
9.6	Improved Dimer Method . . . . .	227
9.7	<i>Ab initio</i> Molecular Dynamics . . . . .	228
9.7.1	Examples . . . . .	233
9.8	Quantum Mechanics/Molecular Mechanics . . . . .	235
<b>10</b>	<b>Molecular Properties and Analysis</b>	<b>238</b>
10.1	Introduction . . . . .	238
10.2	Chemical Solvent Models . . . . .	238
10.2.1	Onsager Dipole Continuum Solvent . . . . .	239
10.2.2	Surface and Simulation of Volume Polarization for Electrostatics (SS(V)PE)	239
10.2.3	The SVP Section Variables . . . . .	242
10.2.4	Langevin Dipoles Solvation Model . . . . .	246
10.2.5	Customizing Langevin Dipoles Solvation Calculations . . . . .	248
10.2.6	Example . . . . .	249

---

10.3	Wavefunction Analysis . . . . .	249
10.3.1	Multipole Moments . . . . .	250
10.3.2	Symmetry Decomposition . . . . .	251
10.4	Visualization using MOLDEN . . . . .	251
10.4.1	Examples . . . . .	252
10.5	Intracules . . . . .	252
10.5.1	Position Intracules . . . . .	253
10.5.2	Momentum Intracules . . . . .	254
10.5.3	Wigner Intracules . . . . .	255
10.5.4	Intracule Job Control . . . . .	256
10.5.5	Format for the <i>\$intracule</i> Section . . . . .	258
10.5.6	Examples . . . . .	259
10.6	Vibrational Analysis . . . . .	260
10.6.1	Job Control . . . . .	260
10.6.2	Example . . . . .	262
10.7	Anharmonic Vibrational Frequency . . . . .	262
10.7.1	Vibration Configuration Interaction Theory . . . . .	263
10.7.2	Vibrational Perturbation Theory . . . . .	264
10.7.3	Transition–Optimized Shifted Hermite Theory . . . . .	265
10.8	Job Control . . . . .	266
10.8.1	Examples . . . . .	267
10.8.2	Isotopic Substitutions . . . . .	268
10.8.3	Example . . . . .	269
10.9	Interface to the NBO Package . . . . .	270
10.9.1	Job Control . . . . .	270
10.10	Plotting Densities and Orbitals . . . . .	270
10.11	Electrostatic Potentials . . . . .	272
10.12	Spin and charge densities at the nuclei . . . . .	274
10.13	NMR Shielding Tensors . . . . .	275
10.13.1	Job Control . . . . .	275
10.14	Linear–Scaling NMR chemical shifts: GIAO–HF and GIAO–DFT . . . . .	277
10.15	Linear–Scaling Computation of Electric Properties . . . . .	278
10.15.1	Examples for section <i>\$dppfreq</i> . . . . .	279
10.15.2	Features of mopropman . . . . .	280
10.15.3	Job Control . . . . .	280

---

10.16	Atoms in Molecules . . . . .	283
<b>11</b>	<b>Extended Customization</b>	<b>287</b>
11.1	User-dependent and Machine-dependent Customization . . . . .	287
11.1.1	<i>.qchemrc</i> and Preferences File Format . . . . .	288
11.1.2	Recommendations . . . . .	288
11.2	Q-CHEM Auxiliary files ( <i>\$QCAUX</i> ) . . . . .	288
11.3	Additional Q-CHEM Output . . . . .	289
11.3.1	Third Party FCHK File . . . . .	289
	<b>Bibliography</b>	<b>289</b>
<b>A</b>	<b>Geometry Optimization with Q-Chem</b>	<b>290</b>
A.1	Introduction . . . . .	290
A.2	Theoretical Background . . . . .	291
A.3	The Eigenvector Following (EF) Algorithm . . . . .	293
A.4	Delocalized Internal Coordinates . . . . .	295
A.5	Constrained Optimization . . . . .	298
A.6	Delocalized internal coordinates . . . . .	300
A.7	GDIIS . . . . .	302
<b>B</b>	<b>AOINTS</b>	<b>306</b>
B.1	Introduction . . . . .	306
B.2	Historical Perspective . . . . .	306
B.3	AOINTS: Calculating ERIs with Q-CHEM . . . . .	307
B.4	Shell-Pair Data . . . . .	308
B.5	Shell-Quartets and Integral Classes . . . . .	308
B.6	Fundamental ERI . . . . .	309
B.7	Angular Momentum Problem . . . . .	309
B.8	Contraction Problem . . . . .	310
B.9	Quadratic Scaling . . . . .	310
B.10	Algorithm Selection . . . . .	311
B.11	More Efficient Hartree-Fock Gradient and Hessian Evaluations . . . . .	311
B.12	User Controllable Variables . . . . .	312

---

<b>C</b>	<b>Q-Chem Quick Reference</b>	<b>315</b>
C.1	Q-CHEM Text Input Summary	315
C.1.1	Keyword: <i>\$molecule</i>	315
C.1.2	Keyword: <i>\$rem</i>	316
C.1.3	Keyword: <i>\$basis</i>	316
C.1.4	Keyword: <i>\$comment</i>	317
C.1.5	Keyword: <i>\$ecp</i>	317
C.1.6	Keyword: <i>\$external_charges</i>	317
C.1.7	Keyword: <i>\$intracule</i>	318
C.1.8	Keyword: <i>\$isotopes</i>	318
C.1.9	Keyword: <i>\$multipole_field</i>	318
C.1.10	Keyword: <i>\$nbo</i>	318
C.1.11	Keyword: <i>\$occupied</i>	319
C.1.12	Keyword: <i>\$opt</i>	319
C.1.13	Keyword: <i>\$svp</i>	319
C.1.14	Keyword: <i>\$svpirf</i>	320
C.1.15	Keyword: <i>\$plots</i>	320
C.1.16	Keyword <i>\$van_der_waals</i>	320
C.1.17	Keyword: <i>\$xc_functional</i>	321
C.2	Geometry Optimization with General Constraints	321
C.2.1	Frozen Atoms	321
C.3	<i>\$rem</i> Variable List	322
C.3.1	General	322
C.3.2	SCF Control	322
C.3.3	DFT Options	323
C.3.4	Large Molecules	323
C.3.5	Correlated Methods	323
C.3.6	Excited States	323
C.3.7	Geometry Optimizations	324
C.3.8	Vibrational Analysis	324
C.3.9	Reaction Coordinate Following	324
C.3.10	NMR Calculations	324
C.3.11	Wavefunction Analysis and Molecular Properties	324
C.3.12	Symmetry	325
C.3.13	Printing Options	325

---

C.3.14 Resource Control . . . . .	325
C.3.15 Alphabetical Listing . . . . .	326

# 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 3 and 4. 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 energy 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: Quick reference section.

## 1.3 Contact Information

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:

<b>Address:</b>	Q-CHEM, Inc. 5001 Baum Blvd Suite 690 Pittsburgh PA 15213	<b>Telephone:</b>	(724) 325-9969
		<b>Fax:</b>	(724) 325-9560
		<b>email:</b>	sales@q-chem.com support@q-chem.com info@q-chem.com

### 1.3.1 Customer Support

Full customer support is promptly provided through 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 please see Q-CHEM's home page (<http://www.q-chem.com>).

## 1.4 Q-Chem, Inc.

Q-CHEM, Inc. is based in Pittsburgh, Pennsylvania and was founded in 1993. Q-CHEM's scientific contributors and board members includes leading quantum chemistry software developers — Professors Martin Head-Gordon (Berkeley), Peter Gill (Canberra), Fritz Schaefer (Georgia), Anna Krylov (USC) and Dr Jing Kong (Pittsburgh). 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 100 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 60 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 New Features in Q-Chem 3.0

Q-CHEM 3.0 includes many new features, along with many enhancements in performance and robustness over previous versions. Below is a list of some of the main additions, and who is primarily to thank for implementing them. Further details and references can be found in the official citation for Q-CHEM (see Section ).

- Improved two-electron integrals package (Dr Yihan Shao):
  - Code for the Head-Gordon-Pople algorithm rewritten to avoid cache misses and to take advantage of modern computer architectures.
  - Overall increased in performance, especially for computing derivatives.
- Fourier Transform Coulomb method (Dr Laszlo Fusti-Molnar):
  - Highly efficient implementation for the calculation of Coulomb matrices and forces for DFT calculations.
  - Linear scaling regime is attained earlier than previous linear algorithms.
  - Present implementation works well for basis sets with high angular momentum and diffuse functions.
- Improved DFT quadrature evaluation:
  - Incremental DFT method avoids calculating negligible contributions from grid points in later SCF cycles (Dr Shawn Brown).
  - Highly efficient SG-0 quadrature grid with approximately half the accuracy and number of grid points as the SG-1 grid (Siu Hung Chien).
- Dual basis self-consistent field calculations (Dr Jing Kong, Ryan Steele):
  - Two stage SCF calculations can reduce computational cost by an order of magnitude.
  - Customized basis subsets designed for optimal projection into larger bases.
- Linear scaling diagonalization replacements (Dr Yihan Shao):
  - Block strategy avoids sparse-matrix manipulation overhead.
  - Effective for one-dimensional systems with  $> 100$  atoms.
- Auxiliary basis expansions for MP2 calculations:
  - RI-MP2 and SOS-MP2 energies (Dr. Yousung Jung) and gradients (Robert A. DiStasio Jr.).
  - RI-TRIM MP2 energies (Robert A. DiStasio Jr.).
  - Scaled opposite spin energies and gradients.

- Enhancements to the correlation package including:
  - Most extensive range of EOM-CCSD methods available including EOM-SF-CCSD, EOM-EE-CCSD, EOM-DIP-CCSD, EOM-IP/EA-CCSD (Prof. Anna Krylov).
  - Available for RHF/UHF/ROHF references.
  - Analytic gradients and properties calculations (permanent and transition dipoles *etc.*).
  - Full use of abelian point-group symmetry.
  - Singlet strongly orthogonal geminal (SSG) methods (Dr Vitaly Rassolov).
- Coupled-cluster perfect-pairing methods (Prof. Martin Head-Gordon):
  - Perfect pairing (PP), imperfect pairing (IP) and restricted pairing (RP) models.
  - PP(2) Corrects for some of the worst failures of MP2 theory.
  - Useful in the study of singlet molecules with diradicaloid character.
  - Applicable to systems with more than 100 active electrons.
- Hybrid quantum mechanics – molecular mechanics (QMMM) methods:
  - Fixed point-charge model based on the AMBER force field.
  - Two-layer ONIOM model (Dr Yihan Shao).
  - Integration with the MOLARIS simulation package.
  - Q-CHEM/CHARMM interface (Dr Lee Woodcock)
- *Ab Initio* Molecular Dynamics (Dr John Herbert):
  - Both direct Born-Oppenheimer molecular dynamics (BOMD) and extended Lagrangian *ab initio* molecular dynamics (ELMD). have been implemented.
  - Available for SCF ground and excited states.
- New continuum solvation models (Dr Shawn Brown):
  - Surface and Simulation of Volume Polarization for Electrostatics (SS(V)PE) model.
  - Available for HF and DFT calculations.
- New transition structure search algorithms (Andreas Heyden and Dr Baron Peters):
  - Growing string method for finding transition states.
  - Dimer Method which does not use the Hessian and is therefore useful for large systems.
- New reaction path finding algorithms (Dr Yihan Shao):
  - The string method.
  - Nudged elastic band method.
- Direct dynamics (Dr John Herbert):
  - Available for SCF wavefunctions (HF, DFT).
  - Direct Born-Oppenheimer molecular dynamics (BOMD).
  - Extended Lagrangian *ab initio* molecular dynamics (ELMD).
- Linear scaling properties for large systems (Jörg Kussmann and Prof. Dr. Christian Ochsenfeld):
  - NMR chemical shifts.
  - Static and dynamic polarizabilities.

- Static hyperpolarizabilities, optical rectification and electro-optical Pockels effect.
- Anharmonic frequencies (Dr Ching Yeh Lin):
  - Efficient implementation of high-order derivatives
  - Corrections via perturbation theory (VPT) or configuration interaction (VCI).
  - New transition optimized shifted Hermite (TOSH) method.
- Wavefunction analysis tools:
  - Spin densities at the nuclei (Dr Vitaly Rassolov).
  - Efficient calculation of localized orbitals.
  - Optimal atomic point-charge models for densities (Andrew Simmonett).
  - Calculation of position, momentum and Wigner intracules (Dr Nick Besley and Dr Darragh O’Neill).
- Graphical user interface options:
  - Seamless integration with the SPARTAN package (see [www.wavefun.com](http://www.wavefun.com)).
  - Support for the public domain version of WEBMO (see [www.webmo.net](http://www.webmo.net)).
  - Support the MOLDEN molecular orbital viewer (see [www.cmbi.ru.nl/molden](http://www.cmbi.ru.nl/molden)).
  - Support the JMOL package.

### 1.6.2 New Features in Q-Chem 3.1

Q-CHEM 3.1 provides the following important upgrades:

- Several new DFT functional options:
  - The nonempirical GGA functional PBE (from the open DF Repository distributed by the QCG CCLRC Daresbury Lab., implemented in Q-Chem 3.1 by Dr E. Proynov).
  - M05 and M06 suites of meta-GGA functionals for more accurate predictions of various types of reactions and systems (Dr Yan Zhao, Dr Nathan E. Schultz, Prof Don Truhlar).
- A faster correlated excited state method: RI-CIS(D) (Dr Young Min Rhee).
- Potential energy surface crossing minimization with CCSD and EOM-CCSD methods (Dr Evgeny Epifanovsky).
- Dyson orbitals for ionization from the ground and excited states within CCSD and EOM-CCSD methods (Dr Melania Oana).

### 1.6.3 Summary of Existing Methods and Features

- Efficient algorithms for large-molecule density functional calculations:
  - Second generation J-engine and J-force engine (Dr Yihan Shao).
  - LinK for exchange energies and forces.
  - CFMM for linear scaling Coulomb interactions (energies and gradients).
  - Linear scaling DFT 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 (Dr Ross Adamson).
  - B3LYP, B3P and user–definable hybrid functionals.
  - Analytical gradients and analytical frequencies.
  - SG–0 Standard quadrature grid (Siu–Hung Chien).
  - Lebedev grids up to 5294 points (Dr Shawn Brown).
  - High level wavefunction–based electron correlation methods (Chapter 5):
    - Efficient semi–direct MP2 energies and gradients.
    - MP3, MP4, QCISD, CCSD energies.
    - OD and QCCD energies and analytical gradients.
    - Triples corrections (QCISD(T), CCSD(T) and OD(T) energies).
    - CCSD(2) and OD(2) energies.
    - Active space coupled cluster methods: VOD, VQCCD, VOD(2).
    - Local second order Møller–Plesset (MP2) methods (DIM and TRIM).
    - Improved definitions of core electrons for post–HF correlation (Dr Vitaly Rassolov).
  - 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 (Prof. Anna Krylov).
    - Coupled–cluster excited–state geometry optimizations.
    - Coupled–cluster property calculations (dipoles, transition dipoles).
    - Spin–flip calculations for CCSD and TDDFT excited states (Prof. Anna Krylov and Dr Yihan Shao).
  - High performance geometry and transition structure optimization (Jon Baker):
    - 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.
    - Intrinsic reaction coordinate (IRC) following code.
  - Evaluation and visualization of molecular properties
    - Onsager, SS(V)PE and Langevin dipoles solvation models.
    - 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.
    - Isotopic substitution for frequency calculations (Robert Doerksen).
    - NMR chemical shifts (Joerg Kussmann).

- Atoms in Molecules (AIMPAC) support (Jim Ritchie).
- Stability analysis of SCF wavefunctions (Yihan Shao).
- Calculation of position and momentum molecular intracules (Aaron Lee, Nick Besley and Darragh O’Neill).
- Flexible basis set and effective core potential (ECP) functionality: (Ross Adamson and Peter Gill)
  - Wide range of built-in basis sets and ECPs.
  - Basis set superposition error correction.
  - Support for mixed and user-defined basis sets.
  - Effective core potentials for energies and gradients.
  - Highly efficient PRISM-based algorithms to evaluate ECP matrix elements.
  - Faster and more accurate ECP second derivatives for frequencies.

## 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 COLD PRISM

The COLD PRISM is the latest in a number of high performance two-electron integral algorithms developed by Peter Gill, Terry Adams and Ross Adamson. 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 generalized scheme, for the construction of two-electron matrix elements from shell-pair data. The implementation within Q-CHEM has been adapted to permit highly efficient evaluation of the matrix elements associated with effective core potentials.

### 1.7.2 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. 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. Q-CHEM also includes an improved treatment of the short-range interactions, developed by Yihan Shao, that significantly speeds up energy evaluation and dramatically speeds up force evaluation, with no loss of accuracy.

### 1.7.3 Parallel Computing

HF and DFT calculations, up to second derivatives, are parallelized in Q-CHEM. Dynamic load-balancing is employed to make fine distribution of analytic and numerical integral evaluations. The memory usage is shared for the solutions of the coupled-perturbed SCF equation, so that

one can afford frequency calculations on large structures by utilizing the large aggregated memory on a parallel computer. The parallelization is implemented with MPI, ensuring availability on all UNIX platforms, including Linux/PC clusters.

#### 1.7.4 Local MP2

Q-CHEM's local MP2 methods are unique, and were developed by Michael Lee, Paul Maslen and Martin Head-Gordon. 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.

#### 1.7.5 High Level Coupled Cluster Methods

Q-CHEM's coupled cluster capabilities have been developed 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 and CCSD(T), Q-CHEM also contains novel optimized orbital coupled cluster methods developed by Krylov, Sherrill and Ed Byrd, 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. Q-CHEM's coupled-cluster package included the ability to perform excited state calculations.

#### 1.7.6 Continuum Solvation Models

The previous version of Q-CHEM already contained continuum solvation capabilities in the form of the simple spherical cavity Onsager reaction field model and the more sophisticated Langevin dipoles model of aqueous solvation that naturally includes dielectric saturation effects. The spherical cavity model has been extended to include analytical SCF gradients, as well as to include higher order multipoles, via the Kirkwood treatment, and also to treat solvent with dissolved salts, via the Debye-Huckel approach. In addition, Q-CHEM 3.0 also contains an additional polarizable continuum solvation model developed by Chipman. This model defines the dielectric cavity as an iso-density contour, and solves the Surface and Simulation of Volume Polarization for Electrostatics (SS(V)PE) equations that take careful account of electrostatic effects associated with solute charge outside the cavity. This model is available for self-consistent reaction field energy evaluation with HF and DFT calculations.

#### 1.7.7 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 use of delocalized internal coordinates. Dr Baker wrote the optimization algorithms in the SPARTAN package and the optimization code in the Biosym-distributed versions of DMOL, TURBOMOL and ZINDO. The OPTIMIZE package in Q-CHEM includes support for intrinsic reaction coordinate following, which allows for the study of reaction pathways.

### 1.7.8 Spartan

Under joint development between Wavefunction and Q-CHEM, SPARTAN is a fully integrated front-end for the Q-CHEM package. It combines the ease of use of Wavefunction's graphical interface with the power of a full version of Q-CHEM as a computational back-end, for electronic structure calculations. Versions are available for the Windows (Windows 98 or higher), Macintosh (Mac OS 10.2 or higher), Linux and IRIX operating systems. Full details of the latest release and supported platforms can be found at the Wavefunction web site: <http://www.wavefun.com>.

## 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 citation for version 3 releases of Q-CHEM is a journal article that has been written describing the main technical features of the program. The full citation for this article is:

“Advances in quantum chemical methods and algorithms in the Q-CHEM 3.0 program package”,

Yihan Shao, Laszlo Fusti-Molnar, Yousung Jung, Jürg Kussmann, Christian Ochsenfeld, Shawn T. Brown, Andrew T.B. Gilbert, Lyudmila V. Slipchenko, Sergey V. Levchenko, Darragh P. O'Neill, Robert A. DiStasio Jr., Rohini C. Lochan, Tao Wang, Gregory J.O. Beran, Nicholas A. Besley, John M. Herbert, Ching Yeh Lin, Troy Van Voorhis, Siu Hung Chien, Alex Sodt, Ryan P. Steele, Vitaly A. Rassolov, Paul E. Maslen, Prakashan P. Korambath, Ross D. Adamson, Brian Austin, Jon Baker, Edward F. C. Byrd, Holger Daschel, Robert J. Doerksen, Andreas Dreuw, Barry D. Dunietz, Anthony D. Dutoi, Thomas R. Furlani, Steven R. Gwaltney, Andreas Heyden, So Hirata, Chao-Ping Hsu, Gary Kedziora, Rustam Z. Khaliulin, Phil Klunzinger, Aaron M. Lee, Michael S. Lee, WanZhen Liang, Itay Lotan, Nikhil Nair, Baron Peters, Emil I. Proynov, Piotr A. Pieniazek, Young Min Rhee, Jim Ritchie, Edina Rosta, C. David Sherrill, Andrew C. Simmonett, Joseph E. Subotnik, H. Lee Woodcock III, Weimin Zhang, Alexis T. Bell, Arup K. Chakraborty, Daniel M. Chipman, Frerich J. Keil, Arieh Warshel, Warren J. Hehre, Henry F. Schaefer III, Jing Kong, Anna I. Krylov, Peter M.W. Gill and Martin Head-Gordon. *Phys. Chem. Chem. Phys.* in press.



# Chapter 2

## Installation

### 2.1 Q-Chem Installation Requirements

#### 2.1.1 Execution Environment

Q-CHEM is shipped as a single executable along with several scripts for the computer system you will run Q-CHEM on. No compilation is required. Once the package is installed, it is ready to run. Please refer to the notes on the CD cover for instructions on installing the software on your particular platform. The system software required to run Q-CHEM on your platform is minimal, and includes:

- A suitable operating system.
- Run-time libraries (usually provided with your operating system).
- Perl, version 5.
- BLAS and LAPACK libraries.
- Vendor implementation of MPI or MPICH libraries (parallel version only).

Please check the Q-CHEM website, or contact Q-CHEM support (email: [support@q-chem.com](mailto:support@q-chem.com)) if further details are required.

#### 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 32 MB RAM, we recommend 128 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 calculation you will be able to perform.

Q-CHEM uses two types of memory: a chunk of static memory that is used by multiple data sets and managed by the code, and the dynamical memory allocation using system calls. The size of the static memory is specified by the user through the *\$rem* word MEM\_STATIC and has a default value of 64 MB. The *\$rem* word MEM\_TOTAL specifies the limit of the total memory the user's job can use and is related to the total memory of the system. Its default value is effectively unlimited for most machines. The limit for the dynamic memory allocation is given by (MEM\_TOTAL-MEM\_STATIC). The amount of MEM\_STATIC needed depends on the size of the user's particular job. Please note that one should not specify an excessively large value for MEM\_STATIC, otherwise it will reduce the available memory for dynamic allocation. The use of *\$rem* words will be discussed in the next chapter.

### **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 KBs. 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 automatically deleted on normal 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. Wavefunction-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.

Although the size of any one of the Q-CHEM temporary files will not exceed 2Gb, a user's job will not be limited by this. Q-CHEM writes large temporary data sets to multiple files so that it is not bounded by the 2Gb file size limitation on some operating systems.

## **2.2 Installing Q-Chem**

Users are referred to the guide on the CD cover for installation instructions pertinent to the release and platform. An encrypted license file, *qchem.license.dat*, must be obtained from your vendor before you will be able to use Q-CHEM. This file should be placed in the directory *\$QCAUX/license* and must be able to be read by all users of the software. This file is node-locked, *i.e.*, it will only operate correctly on the machine for which it was generated. Further details about obtaining this file, can be found on the CD cover.

**Do not alter the license file unless directed by Q-Chem Inc.**

## 2.3 Environment Variables

Q-CHEM requires four 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. If not explicitly set by the user, this defaults to <i>\$QC/aux</i> .
QCSCRATCH	Defines the directory in which Q-CHEM will store temporary files. Q-CHEM will usually remove these files on successful completion of the job, but they can be saved, if so wished. Therefore, <i>\$QCSCRATCH</i> should not reside in a directory that will be automatically removed at the end of a job, if the files are to be kept for further calculations. Note that many of these files can be very large, and it should be ensured that the volume that contains this directory has sufficient disk space available. The <i>\$QCSCRATCH</i> directory should be periodically checked for scratch files remaining from abnormally terminated jobs. <i>\$QCSCRATCH</i> defaults to the working directory if not explicitly set. Please see section 2.6 for details on saving temporary files and consult your systems administrator.
QCLOCALSCR	On certain platforms, such as Linux clusters, it is sometimes preferable to write the temporary files to a disk local to the node. <i>\$QCLOCALSCR</i> specifies this directory. The temporary files will be copied to <i>\$QCSCRATCH</i> at the end of the job, unless the job is terminated abnormally. In such cases Q-CHEM will attempt to remove the files in <i>\$QCLOCALSCR</i> , but may not be able to due to access restrictions. Please specify this variable only if required.

## 2.4 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).

### 2.4.1 Example *.login* File Modifications

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

```
# ***** Start qchem Configuration *****
setenv QC          qchem_root_directory_name
setenv QCAUX      $QC/aux
setenv QCSCRATCH  scratch_directory_name
if (-e $QC/bin/qchem.setup) source $QC/bin/qchem.setup
```

```
unset noclobber
# ***** End qchem Configuration *****
```

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

```
# ***** Start qchem Configuration *****
QC=qchem_root_directory_name;
export QC
QCAUX=$QC/aux;
export QCAUX
QCSCRATCH=scratch_directory_name;
export QCSCRATCH
noclobber=""
if [ -e $QC/bin/qchem.setup.sh ] ; then
    $QC/bin/qchem.setup.sh
fi
# ***** End qchem Configuration *****
```

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

## 2.5 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

## 2.6 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:

1. *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.
2. 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](http://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.

### 2.6.1 Serial Q-Chem

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. If the *savename* argument is not given, Q-CHEM deletes all temporary 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 and the current force constants (if available). 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. When *\$QCLOCALSCR* has been specified, the temporary files will be stored there and copied to *\$QCSCRATCH/savename/* at the end of normal termination.

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 maybe helpful to use the same job name 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/*.

### 2.6.2 Parallel Q-Chem

Running the parallel version of Q-CHEM interactively is the almost the same as running the serial version, except that an additional argument must be given that specifies the number of processors to use. The *qchem* shell script can be used in either of the following ways:

```
qchem -np n infile outfile
qchem -np n infile outfile savename
qchem -save -np n infile outfile savename
```

where  $n$  is the number of processors to use. If the `-np` switch is not given, Q-CHEM will default to running locally on a single processor.

When the additional argument *savename* is specified, the temporary files for parallel Q-CHEM are stored in `$QCSCRATCH/savename.0`. At the start of a job, any existing files will be copied into this directory, and on successful completion of the job, be copied to `$QCSCRATCH/savename/` for future use. If the job terminates abnormally, the files will not be copied.

To run parallel Q-CHEM using a batch scheduler such as PBS, users may have to modify the `mpirun` command in `$QC/bin/parallel.csh` depending on whether or not the MPI implementation requires the `-machinefile` option to be given. For further details users should read the `$QC/README.Parallel` file, and contact Q-CHEM if any problems are encountered (email: support@q-chem.com). Parallel users should also read the above section on using serial Q-CHEM.

## 2.7 Testing and Exploring Q-Chem

Q-CHEM is shipped with a small number of test jobs which are located 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. 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!

Please check the Q-CHEM web page (<http://www.q-chem.com>) and the README files in the `$QC/bin` directory for updated information

## 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 the *\$end* keyword. A short description of all Q-CHEM keywords is provided in Table C.1 and the following sections. 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 geometry 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 these differences (format requirements are summarized in Appendix C). However, because each keyword in the input file is sought out independently by the program, the overall format requirements of Q-CHEM input files are much less stringent. For example, the *\$molecule* section does not have to occur at the very start of the input file.

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

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

The order of preference is as shown, *i.e.*, 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

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

Table 3.1: Q-CHEM user input section keywords. See the *\$QC/samples* directory with your release for specific examples of Q-CHEM input using these keywords.

- Note:** (1) Users are able to enter keyword sections in any order.  
(2) Each keyword section must be terminated with the *\$end* keyword.  
(3) The *\$rem* and *\$molecule* sections must be included.  
(4) It is not necessary to have all keywords in an input file.  
(5) Each keyword section is described in Appendix C.  
(6) The entire Q-CHEM input is case-insensitive.

user overrides all defaults. Refer to Chapter 11 for details of *.qchemrc* and *preferences*. Currently, Q-CHEM only supports the *\$rem* keyword in *.qchemrc* and *preferences* files.

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.

## 3.2 Molecular Coordinate Input (*\$molecule*)

The *\$molecule* section communicates to the program the charge, spin multiplicity and geometry of the molecule being considered. The molecular coordinates input begins with two integers: the net charge and the spin multiplicity of the molecule. The net charge must be between -50 and 50, inclusive (0 for neutral molecules, 1 for cations, -1 for anions, *etc.*). The multiplicity must be between 1 and 10, inclusive (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 center 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. unless the INPUT\_BOHR *\$rem* variable is set to TRUE, in which case all lengths are assumed to be in bohr.

**Example 3.1** A molecule in *Z*-matrix coordinates. Note that the *\$molecule* input begins with the charge and multiplicity.

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

  distance = 1.0
  theta = 104.5
$end
```

### 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 the previous ones. For example, a geometry optimization at a low level of theory, followed by a 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:

- The *READ* variable is entered into the molecular coordinate input
- 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.

**Example 3.2** Reading a geometry from a prior calculation.

```
$molecule
  READ
$end

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

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).

### 3.2.2 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.

**Example 3.3** Reading molecular coordinates from another file. *filename* may be given either as the full file path, or path relative to the working directory.

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

## 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

**Example 3.4** Atomic number Cartesian coordinate input for H<sub>2</sub>O.

```
$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 3.5** Atomic symbol Cartesian coordinate input for H<sub>2</sub>O.

```
$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
```

- Note:** (1) Atoms can be declared by either atomic number or symbol.  
(2) Coordinates can be entered either as variables/parameters or real numbers.  
(3) Variables/parameters can be declared in any order.  
(4) 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 summarized 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)

**Example 3.6** Z-matrix for hydrogen peroxide

```
O1
O2  O1  oo
H1  O1  ho  O2  hoo
H2  O2  ho  O1  hoo  H1  hooh
```

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.

**Example 3.7** Using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```
$molecule
  0 1
  01
  02 01 oo
  H1 01 ho 02 hoo
  H2 02 ho 01 hoo H1 hooh

  oo = 1.5
  oh = 1.0
  hoo = 120.0
  hooh = 180.0
$end
```

**Example 3.8** Not using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```
$molecule
  0 1
  01
  02 01 1.5
  H1 01 1.0 02 120.0
  H2 02 1.0 01 120.0 H1 180.0
$end
```

**Example 3.9** Using parameters to define bond lengths and angles, and referring to atom connectivities by line number.

```
$molecule
  0 1
  8
  8 1 oo
  1 1 ho 2 hoo
  1 2 ho 1 hoo 3 hooh

  oo = 1.5
  oh = 1.0
  hoo = 120.0
  hooh = 180.0
$end
```

**Example 3.10** Referring to atom connectivities by line number, and entering bond length and angles directly.

```
$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
```

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 centered 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 shown in the following example shown in the following example:

**Example 3.11** 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.

```
REM_VARIABLE    VALUE    [comment]
```

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 JOBTYP is a single point energy, SP). Thus, to perform a single point energy calculation, the user does **not** need to set the  $\$rem$  variable JOBTYP to SP. However, to perform an optimization, for example, it is necessary to override the program default by setting JOBTYP 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 11) 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 Appendix C. 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

A short description of what the variable controls.

#### TYPE:

The type of variable, *i.e.* either INTEGER, LOGICAL or STRING

#### DEFAULT:

The default value, if any.

#### OPTIONS:

A list of the options available to the user.

#### RECOMMENDATION:

A quick recommendation, where appropriate.

**Example 3.12** General format of the *\$rem* section of the text input file.

```
$rem
  REM_VARIABLE  value  [ user_comment ]
  REM_VARIABLE  value  [ user_comment ]
  ...
$end
```

**Note:** (1) Erroneous lines will terminate the calculation.  
 (2) Tab stops can be used to format input.  
 (3) A line prefixed with an exclamation mark ‘!’ is treated as a comment and will be ignored by the program.

### 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). If a wavefunction-based correlation treatment (such as MP2) is used, HF is taken as the default for exchange.

**Example 3.13** Example of minimum *\$rem* requirements to run an MP2/6-31G\* energy calculation.

```
$rem
  BASIS          6-31G*   Just a small basis set
  CORRELATION    mp2      MP2 energy
$end
```

### 3.8 User-defined basis set (*\$basis*)

The *\$rem* variable BASIS 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.

### 3.9 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. Comments can also be provided in a *\$comment* block, although currently the entire input deck is copied to the output file, rendering this redundant.

### 3.10 User-defined Pseudopotentials (*\$ecp*)

The *\$rem* variable ECP 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 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 are in atomic units, and coordinates are in angstroms. The external charges are rotated with the molecule into the standard nuclear orientation.

**Example 3.14** General format for incorporating a set of external charges.

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

### 3.12 Intracules (*\$intracule*)

The *\$intracule* section allows the user to enter options to customize the calculation of molecular intracules. The INTRACULE *\$rem* variable must also be set to TRUE before this section takes effect. For further details see section 10.5.

### 3.13 Isotopic substitutions (*\$isotopes*)

By default Q-CHEM uses atomic masses that correspond to the most abundant naturally occurring isotopes. Alternative masses for any or all of the atoms in a molecule can be specified using the *\$isotopes* keyword. The ISOTOPES *\$rem* variable must be set to TRUE for this section to take effect. See section 10.8.2 for details.

### 3.14 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.

**Example 3.15** General format for imposing a multipole field.

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

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

### 3.15 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*.

### 3.16 User-defined occupied guess orbitals (*\$occupied*)

It is sometimes useful for the occupied guess orbitals to be other than the lowest  $N_\alpha$  (or  $N_\beta$ ) 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 section 4.5.4).

### 3.17 Geometry Optimization with General Constraints (*\$opt*)

When a user defines the JOBTYP 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 9 for details.

### 3.18 SS(V)PE Solvation Modeling (*\$svp* and *\$svpirf*)

The *\$svp* section is available to specify special parameters to the solvation module such as cavity grid parameters and modifications to the numerical integration procedure. The *\$svpirf* section allows the user to specify an initial guess for the solution of the cavity charges. For more details, see section 10.2.



### 3.19 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 section 10.10.

### 3.20 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 section 10.2.

### 3.21 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
$end
```

**Note:** Coefficients are real numbers.

### 3.22 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:

- To use information from a prior job in a later job
- 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 optimization 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 H<sub>2</sub> 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.

**Example 3.16** Example of using information from previous jobs in a single input file.

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

$molecule
  0 1
  H
  H 1 r

  r = 1.1
$end

$rem
  JOBTYP      opt      Optimize the bond length
  EXCHANGE    hf
  CORRELATION none
  BASIS        6-31G*
$end

@@@

$comment
  Now calculate the frequency of H-H at the same level of theory.
$end

$molecule
  read
$end

$rem
  JOBTYP      freq      Calculate vibrational frequency
```

```
    EXCHANGE      hf
    CORRELATION   none
    BASIS         6-31G*
    SCF_GUESS     read    Read the MOs from disk
$end

@@@

$comment
  Now a single point calculation at at MP2/6-311G(d,p)//HF/6-31G*
$end

$molecule
  read
$end

$rem
  EXCHANGE      hf
  CORRELATION   mp2
  BASIS         6-311G(d,p)
$end
```

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

### 3.23 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 optimization *etc.*) and the *\$rem* variable print levels. The general and default form is as follows:

- Q-CHEM citation
- User input
- 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.

## 3.24 Q-Chem Scratch Files

The directory set 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.

The 32-bit architecture on some platforms means there can be problems associated with files larger than about 2 GB. Q-CHEM handles this issue by splitting scratch files that are larger than this into several files, each of which is smaller than the 2 GB limit. The maximum number of these files (which in turn limits the maximum total file size) is determined by the following `$rem` variable:

### **MAX\_SUB\_FILE\_NUM**

Sets the maximum number of sub files allowed.

TYPE:

INTEGER

DEFAULT:

16 Corresponding to a total of 32Gb for a given file.

OPTIONS:

*n* User-defined number of gigabytes.

RECOMMENDATION:

Leave as default, or adjust according to your system limits.

## Chapter 4

# Self-Consistent Field Ground State Methods

### 4.1 Introduction

#### 4.1.1 Overview of Chapter

Theoretical chemical models [3] 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 [3, 6, 7].

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 [8] combined the wave nature of the electron with the statistical knowledge of the electron *viz.* Heisenberg's Uncertainty Principle [9] 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

$$H(\mathbf{R}, \mathbf{r})\Psi(\mathbf{R}, \mathbf{r}) = E(\mathbf{R})\Psi(\mathbf{R}, \mathbf{r}) \quad (4.1)$$

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)

$$H = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \frac{1}{2} \sum_{A=1}^M \frac{1}{M_A} \nabla_A^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N \sum_{j>i}^N \frac{1}{r_{ij}} + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{R_{AB}} \quad (4.2)$$

In equation (4.2)  $\nabla^2$  is the Laplacian operator

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (4.3)$$

$Z$  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  $A^{\text{th}}$  and  $B^{\text{th}}$  nucleus,  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$  is the distance between the  $i^{\text{th}}$  and  $j^{\text{th}}$  electrons,  $r_{iA} = |\mathbf{r}_i - \mathbf{R}_A|$  is the distance between the  $i^{\text{th}}$  electron and  $A^{\text{th}}$  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  $\Psi$ , is an eigenfunction of  $H$ .

Separating the motions of the electrons from that of the nuclei, an idea originally due to Born and Oppenheimer [10], yields the electronic Hamiltonian operator.

$$H_{\text{elec}} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N \sum_{j>i}^N \frac{1}{r_{ij}} \quad (4.4)$$

The solution of the corresponding electronic Schrödinger equation

$$H_{\text{elec}}\Psi_{\text{elec}} = E_{\text{elec}}\Psi_{\text{elec}} \quad (4.5)$$

gives the total electronic energy,  $E_{\text{elec}}$ , and electronic wave function,  $\Psi_{\text{elec}}$ , 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 eq. (4.2)) to the total electronic energy

$$E_{\text{tot}} = E_{\text{elec}} + E_{\text{nuc}} \quad (4.6)$$

Solving the eigenvalue problem (4.5) yields a set of eigenfunctions ( $\Psi_0, \Psi_1, \Psi_2 \dots$ ) with corresponding eigenvalues ( $E_0, E_1, E_2 \dots$ ) where  $E_0 \leq E_1 \leq E_2 \leq \dots$

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 (4.5) 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 (4.5) 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 [11, 12] determinant wavefunction

$$\Psi = \frac{1}{\sqrt{n!}} \begin{vmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_n(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_n(2) \\ \vdots & \vdots & & \vdots \\ \chi_1(n) & \chi_2(n) & \cdots & \chi_n(n) \end{vmatrix} \quad (4.7)$$

where  $\chi_i$ , a spin orbital, is the product of a molecular orbital  $\psi_i$  and a spin function ( $\alpha$  or  $\beta$ ).

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 4.3). All SCF methods lead to equations of the form

$$f(i)\chi(\mathbf{x}_i) = \varepsilon\chi(\mathbf{x}_i) \quad (4.8)$$

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

$$f(i) = -\frac{1}{2}\nabla_i^2 + v^{\text{eff}}(i) \quad (4.9)$$

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

The second approximation usually introduced when solving (4.5), is the introduction of an Atomic Orbital (AO) basis. AOs ( $\phi_\mu$ ) 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 (4.8) and introducing a finite basis,

$$\psi_i = \sum_{\mu} c_{\mu i} \phi_{\mu} \quad (4.10)$$

(4.8) reduces to the Roothaan-Hall matrix equation

$$\mathbf{FC} = \varepsilon\mathbf{SC} \quad (4.11)$$

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

$$S_{\mu\nu} = \int \phi_{\mu}(\mathbf{r})\phi_{\nu}(\mathbf{r})d\mathbf{r} \quad (4.12)$$

and  $\varepsilon$  is a diagonal matrix of the orbital energies. Generalizing to an unrestricted formalism by introducing separate spatial orbitals for  $\alpha$  and  $\beta$  spin in (4.7) yields the Pople-Nesbet [13] equations

$$\begin{aligned} \mathbf{F}^{\alpha}\mathbf{C}^{\alpha} &= \varepsilon^{\alpha}\mathbf{S}\mathbf{C}^{\alpha} \\ \mathbf{F}^{\beta}\mathbf{C}^{\beta} &= \varepsilon^{\beta}\mathbf{S}\mathbf{C}^{\beta} \end{aligned} \quad (4.13)$$

Solving (4.11) or (4.13) 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. 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  $\mathcal{O}(N^4)$ , 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  $\mathcal{O}(N)$ , an improvement that now allows SCF methods to be applied to molecules previously considered beyond the scope of *ab initio* 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 4.1.2.

Calculation	<i>\$rem</i> Variable JOBTYP
Single point energy (default)	SINGLE_POINT, SP
Force	FORCE
Equilibrium Structure Search	OPTIMIZATION, OPT
Transition Structure Search	TS
Intrinsic reaction pathway	RPATH
Frequency	FREQUENCY, FREQ
NMR Chemical Shift	NMR

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 The 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, (4.11) or the Pople–Nesbet equations (4.13), which can be traced back to the integro–differential equation (4.8) where the effective potential  $v^{\text{eff}}$  depends on the SCF methodology. In a restricted HF (RHF) formalism, the effective potential can be written as

$$v^{\text{eff}} = \sum_a^{N/2} [2J_a(1) - K_a(1)] - \sum_{A=1}^M \frac{Z_A}{r_{1A}} \quad (4.14)$$

where the Coulomb and exchange operators are defined as

$$J_a(1) = \int \psi_a^*(2) \frac{1}{r_{12}} \psi_a(2) d\mathbf{r}_2 \quad (4.15)$$

$$K_a(1)\psi_i(1) = \left[ \int \psi_a^*(2) \frac{1}{r_{12}} \psi_i(2) d\mathbf{r}_2 \right] \psi_a(1) \quad (4.16)$$

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

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - K_{\mu\nu} \quad (4.17)$$

where the core Hamiltonian matrix elements

$$H_{\mu\nu}^{\text{core}} = T_{\mu\nu} + V_{\mu\nu} \quad (4.18)$$

consist of kinetic energy elements

$$T_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left[ -\frac{1}{2}\nabla^2 \right] \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (4.19)$$

and nuclear attraction elements

$$V_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left[ -\sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} \right] \phi_\nu(\mathbf{r}) d\mathbf{r} \quad (4.20)$$

The Coulomb and Exchange elements are given by

$$J_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} (\mu\nu|\lambda\sigma) \quad (4.21)$$

$$K_{\mu\nu} = \frac{1}{2} \sum_{\lambda\sigma} P_{\lambda\sigma} (\mu\lambda|\nu\sigma) \quad (4.22)$$

where the density matrix elements are

$$P_{\mu\nu} = 2 \sum_{a=1}^{N/2} C_{\mu a} C_{\nu a} \quad (4.23)$$

and the two electron integrals are

$$(\mu\nu|\lambda\sigma) = \int \int \phi_\mu(\mathbf{r}_1) \phi_\nu(\mathbf{r}_1) \left[ \frac{1}{r_{12}} \right] \phi_\lambda(\mathbf{r}_2) \phi_\sigma(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad (4.24)$$

**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 (4.17) back into the Roothaan-Hall equations (4.11) 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

$$\begin{aligned} P_{\mu\nu}^{\alpha} &= \sum_{a=1}^{n_{\alpha}} C_{\mu a}^{\alpha} C_{\nu a}^{\alpha} \\ P_{\mu\nu}^{\beta} &= \sum_{a=1}^{n_{\beta}} C_{\mu a}^{\beta} C_{\nu a}^{\beta} \end{aligned} \quad (4.25)$$

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

$$F_{\mu\nu}^{\alpha} = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - K_{\mu\nu}^{\alpha} \quad (4.26)$$

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

$$K_{\mu\nu}^{\alpha} = \sum_{\lambda}^N \sum_{\sigma}^N P_{\lambda\sigma}^{\alpha} (\mu\lambda|\nu\sigma) \quad (4.27)$$

## 4.2.2 Wavefunction Stability Analysis

At convergence, the SCF energy will be at a stationary point with respect to changes in the MO coefficients. However, this stationary point is not guaranteed to be an energy minimum, and in cases where it is not, the wavefunction is said to be unstable. Even if the wavefunction is at a minimum, this minimum may be an artifact of the constraints placed on the form of the wavefunction. For example, an unrestricted calculation will usually give a lower energy than the corresponding restricted calculation, and this can give rise to a RHF→UHF instability.

To understand what instabilities can occur, it is useful to consider the most general form possible for the spin orbitals

$$\chi_i(\mathbf{r}, \zeta) = \psi_i^{\alpha}(\mathbf{r})\alpha(\zeta) + \psi_i^{\beta}(\mathbf{r})\beta(\zeta) \quad (4.28)$$

where the  $\psi$ 's are complex functions of the Cartesian coordinates  $\mathbf{r}$ , and  $\alpha$  and  $\beta$  are spin eigenfunctions of the spin-variable  $\zeta$ . The first constraint that is almost universally applied is to assume the spin orbitals depend only on one or other of the spin-functions  $\alpha$  or  $\beta$ . Thus the spin-functions take the form

$$\chi_i(\mathbf{r}, \zeta) = \psi_i^{\alpha}(\mathbf{r})\alpha(\zeta) \quad \text{or} \quad \chi_i(\mathbf{r}, \zeta) = \psi_i^{\beta}(\mathbf{r})\beta(\zeta) \quad (4.29)$$

where the  $\psi$ 's are still complex functions. Most SCF packages, including Q-CHEM's, deal only with real functions, and this places an additional constraint on the form of the wavefunction. If

there exists a complex solution to the SCF equations that has a lower energy, the wavefunction will exhibit either a RHF→CRHF or a UHF→CUHF instability. The final constraint that is commonly placed on the spin-functions is that  $\psi_i^\alpha = \psi_i^\beta$ , *i.e.*, the spatial parts of the spin-up and spin-down orbitals are the same. This gives the familiar restricted formalism and can lead to a RHF→UHF instability as mentioned above. Further details about the possible instabilities can be found in [14].

Wavefunction instabilities can arise for several reasons, but frequently occur if

- There exists a singlet biradical at a lower energy than the closed-shell singlet state.
- There exists a triplet state at a lower energy than the lowest singlet state.
- There are multiple solutions to the SCF equations, and the calculation has not found the lowest energy solution.

If a wavefunction exhibits an instability, the seriousness of it can be judged from the magnitude of the negative eigenvalues of the stability matrices. These matrices and eigenvalues are computed by Q-CHEM's Stability Analysis package, which was implemented by Dr Yihan Shao. The package is invoked by setting the STABILITY\_ANALYSIS *\$rem* variable is set to TRUE. In order to compute these stability matrices Q-CHEM must first perform a CIS calculation. This will be performed automatically, and does not require any further input from the user. By default Q-CHEM computes only the lowest eigenvalue of the stability matrix. This is usually sufficient to determine if there is a negative eigenvalue, and therefore an instability. Users wishing to calculate additional eigenvalues can do so by setting the CIS\_N\_ROOTS *\$rem* variable to a number larger than 1.

Q-CHEM's Stability Analysis package also seeks to correct internal instabilities (RHF→RHF or UHF→UHF). Then, if such an instability is detected, Q-CHEM automatically performs a unitary transformation of the molecular orbitals following the directions of the lowest eigenvector, and writes a new set of MOs to disk. One can read in these MOs as an initial guess in a second SCF calculation (set the SCF\_GUESS *\$rem* variable to READ), it might also be desirable to set the SCF\_ALGORITHM to GDM. In cases where the lowest-energy SCF solution breaks the molecular point-group symmetry, the SYM\_IGNORE *\$rem* should be set to TRUE.

**Note:** The stability analysis package can be used to analyze both DFT and HF wavefunctions.

### 4.2.3 Basic Hartree–Fock Job Control

In brief, Q-CHEM supports the three 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 two *\$rem* variables are required in order to run Hartree–Fock (HF) calculations:

- EXCHANGE must be set to HF.
- 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 type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP Single point energy.

OPT Geometry Minimization.

TS Transition Structure Search.

FREQ Frequency Calculation.

FORCE Analytical Force calculation.

RPATH Intrinsic Reaction Coordinate calculation.

NMR NMR chemical shift calculation.

RECOMMENDATION:

Job dependent

**EXCHANGE**

Specifies the exchange level of theory.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

HF Exact (Hartree–Fock).

RECOMMENDATION:

Use HF for Hartree–Fock calculations.

**BASIS**

Specifies the basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen User defined (*\$basis* keyword required).

Symbol Use standard basis sets as per Chapter 7.

Mixed Use a mixture of basis sets (see Chapter 7).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

**PRINT\_ORBITALS**

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

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print any orbitals.

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.  $10^{-\text{THRESH}}$  ( $\text{THRESH} \leq 14$ ).

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

$n$  for a threshold of  $10^{-n}$ .

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. Note that in Q-CHEM 3.0 the DIIS error is measured by the maximum error rather than the RMS error as in previous versions.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations.

7 For geometry optimizations and vibrational analysis.

8 For SSG calculations, see Chapter 5.

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.

TYPE:

LOGICAL

DEFAULT:

FALSE (Restricted) Closed-shell systems.

TRUE (Unrestricted) Open-shell systems.

OPTIONS:

TRUE (Unrestricted) Open-shell systems.

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 Section 4.5 on initial guesses).

#### 4.2.4 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.

**INTEGRALS\_BUFFER**

Controls the size of in-core integral storage buffer.

TYPE:

INTEGER

DEFAULT:

15 15 Megabytes.

OPTIONS:

User defined size.

RECOMMENDATION:

Use the default, or consult your systems administrator for hardware limits.

**DIRECT\_SCF**

Controls direct SCF.

TYPE:

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.

TYPE:

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.

**STABILITY\_ANALYSIS**

Performs stability analysis for a HF or DFT solution.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform stability analysis.

FALSE Do not perform stability analysis.

RECOMMENDATION:

Set to TRUE when a HF or DFT solution is suspected to be unstable.

**SCF\_PRINT**

Controls level of output from SCF procedure to Q-CHEM output file.

TYPE:

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.

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.

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

0 No extra print out.

1 Orbital energies and break-down of SCF energy.

2 Level 1 plus MOs and density matrices.

3 Level 2 plus Fock and density matrices.

RECOMMENDATION:

The break-down of energies is often useful (level 1).

### 4.2.5 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.

**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 1
  0
  H1 0 oh
  H2 0 oh H1 hoh

  oh = 1.2
  hoh = 120.0
$end

$rem
  JOBTYP      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 4.2** 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
  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.

```

$molecule
  0,2
  3
$end

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

```

**Example 4.4** RHF/6-31G stability analysis calculation on the singlet state of the oxygen molecule. The wavefunction is RHF→UHF unstable.



```
$molecule
  0 1
  0
  0 1 1.165
$end

$rem
  EXCHANGE          hf          Hartree--Fock
  UNRESTRICTED      false       Restricted MOs
  BASIS              6-31G(d)    Basis set
  STABILITY_ANALYSIS true       Perform a stability analysis
$end
```

## 4.2.6 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 [15–17]. 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.

In some cases it may be desirable to turn off symmetry altogether, for example if you do not want Q-CHEM to reorient the molecule into the standard nuclear orientation. If the SYMIGNORE *\$rem* is set to TRUE then the coordinates will not be altered from the input, and the point group will be set to  $C_1$ .

### SYMMETRY

Controls the efficiency through the use of point group symmetry for calculating integrals.

TYPE:

LOGICAL

DEFAULT:

TRUE Use symmetry for computing integrals (disabled for RIMP2 jobs).

OPTIONS:

TRUE Use symmetry when available.

FALSE Do not use symmetry. This is always the case for RIMP2 jobs

RECOMMENDATION:

Use default unless benchmarking. Note that symmetry usage is disabled for RIMP2 jobs.

**SYM\_IGNORE**

Controls whether or not Q-CHEM determines the point group of the molecule.

TYPE:

LOGICAL

DEFAULT:

FALSE Do determine the point group (disabled for RIMP2 jobs).

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use default unless you do not want the molecule to be reoriented. Note that symmetry usage is disabled for RIMP2 jobs.

**SYM\_TOL**

Controls the tolerance for determining point group symmetry. Differences in atom locations less than  $10^{-\text{SYM\_TOL}}$  are treated as zero.

TYPE:

INTEGER

DEFAULT:

5 corresponding to  $10^{-5}$ .

OPTIONS:

User defined.

RECOMMENDATION:

Use the default unless the molecule has high symmetry which is not being correctly identified. Note that relaxing this tolerance too much may introduce errors into the calculation.

## 4.3 Density Functional Theory

### 4.3.1 Introduction

In recent years, Density Functional Theory [6, 18, 19] 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 [21, 22] stems from the original work of Dirac [23], 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 [22], the ground state electronic energy,  $E$ , can be written as

$$E = E_T + E_V + E_J + E_{XC} \quad (4.30)$$

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

$$\begin{aligned} \rho_\alpha(\mathbf{r}) &= \sum_{i=1}^{n_\alpha} |\psi_i^\alpha|^2 \\ \rho_\beta(\mathbf{r}) &= \sum_{i=1}^{n_\beta} |\psi_i^\beta|^2 \end{aligned} \quad (4.31)$$

where  $n_\alpha$  and  $n_\beta$  are the number of alpha and beta electron respectively and,  $\psi_i$  are the Kohn–Sham orbitals. Thus, the total electron density is

$$\rho(\mathbf{r}) = \rho_\alpha(\mathbf{r}) + \rho_\beta(\mathbf{r}) \quad (4.32)$$

which within a finite basis [24] is represented by

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}^T \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \quad (4.33)$$

The components of (4.28) can now be written as

$$\begin{aligned}
E_T &= \sum_{i=1}^{n_\alpha} \left\langle \psi_i^\alpha \left| -\frac{1}{2} \nabla^2 \right| \psi_i^\alpha \right\rangle + \sum_{i=1}^{n_\beta} \left\langle \psi_i^\beta \left| -\frac{1}{2} \nabla^2 \right| \psi_i^\beta \right\rangle \\
&= \sum_{\mu\nu} P_{\mu\nu}^T \left\langle \phi_\mu(\mathbf{r}) \left| -\frac{1}{2} \nabla^2 \right| \phi_\nu(\mathbf{r}) \right\rangle
\end{aligned} \tag{4.34}$$

$$\begin{aligned}
E_V &= - \sum_{A=1}^M Z_A \frac{\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{R}_A|} d\mathbf{r} \\
&= - \sum_{\mu\nu} P_{\mu\nu}^T \sum_A \left\langle \phi_\mu(\mathbf{r}) \left| \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \right| \phi_\nu(\mathbf{r}) \right\rangle
\end{aligned} \tag{4.35}$$

$$\begin{aligned}
E_J &= \frac{1}{2} \left\langle \rho(\mathbf{r}_1) \left| \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right| \rho(\mathbf{r}_2) \right\rangle \\
&= \frac{1}{2} \sum_{\mu\nu} \sum_{\lambda\sigma} P_{\mu\nu}^T P_{\lambda\sigma}^T (\mu\nu|\lambda\sigma)
\end{aligned} \tag{4.36}$$

$$E_{XC} = \int f[\rho(\mathbf{r}), \nabla\rho(\mathbf{r}), \dots] d\mathbf{r} \tag{4.37}$$

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

$$\mathbf{F}^\alpha \mathbf{C}^\alpha = \varepsilon^\alpha \mathbf{S} \mathbf{C}^\alpha \tag{4.38}$$

$$\mathbf{F}^\beta \mathbf{C}^\beta = \varepsilon^\beta \mathbf{S} \mathbf{C}^\beta \tag{4.39}$$

where the Fock matrix elements are generalized to

$$F_{\mu\nu}^\alpha = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - F_{\mu\nu}^{\text{XC}\alpha} \tag{4.40}$$

$$F_{\mu\nu}^\beta = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - F_{\mu\nu}^{\text{XC}\beta} \tag{4.41}$$

where  $F_{\mu\nu}^{\text{XC}\alpha}$  and  $F_{\mu\nu}^{\text{XC}\beta}$  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

$$F_{\mu\nu}^{\text{XC}\alpha} = K_{\mu\nu}^\alpha \tag{4.42}$$

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 three basic types of functionals: those based on the local spin density approximation (LSDA), those based on generalized gradient approximations (GGA), and those that include not only gradient corrections (as in the GGA functionals), but also a dependence on the spin kinetic energy density. Furthermore, as discussed later in this subsection, each of these types of functionals can be used

with a certain amount of Hartree-Fock nonlocal exchange replacing some of the local exchange. When a nonzero amount of Hartree-Fock exchange is used, the functional is called hybrid. 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) [23]
- Vokso-Wilk-Nusair (Correlation) [25]
- Perdew-Zunger (Correlation) [26]
- Wigner (Correlation) [27]

the following GGA functionals:

- Becke88 (Exchange) [28]
- Gill96 (Exchange) [29]
- Gilbert-Gill99 (Exchange) [30]
- Lee-Yang-Parr (Correlation) [31]
- Perdew86 (Correlation) [32]
- GGA91 (Exchange and correlation) [33]
- PBE (Exchange and correlation) [34, 35]
- Becke97 (Exchange and correlation within a hybrid scheme) [35, 36]
- Becke97-1 (Exchange and correlation within a hybrid scheme) [35, 37]
- Becke97-2 (Exchange and correlation within a hybrid scheme) [35, 38]

and the following meta-GGA functionals containing the kinetic energy density ( $\tau$ ):

- BMK (Exchange and Correlation within a hybrid scheme) [39]
- M05 (Exchange and Correlation within a hybrid scheme) [40, 45]
- M05-2X (Exchange and Correlation within a hybrid scheme) [41, 45]
- M06-L (Exchange and Correlation within a hybrid scheme) [42, 45]
- M06-HF (Exchange and Correlation within a hybrid scheme) [43, 45]
- M06 (Exchange and Correlation within a hybrid scheme) [44, 45]
- M06-2X (Exchange and Correlation within a hybrid scheme) [44, 45]

In addition to the established density functionals, Q-CHEM contains the recent Empirical Density Functional 1 (EDF1), developed by Adamson, Gill and Pople [46]. EDF1 is a combined exchange and 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 this basis set, 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

(non-local) 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.

EDF2, another Empirical Density Functional, was developed by Ching Yeh Lin and Peter Gill [86] in a similar vein to EDF1, but is specially designed for harmonic frequency calculations. It was optimized using the cc-pVTZ basis set by fitting into experimental *harmonic* frequencies and is designed to describe the potential energy curvature well. Fortuitously, it also performs better than B3LYP for thermochemical properties.

Hybrid exchange–correlation functionals [47], 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 [48] and B3PW91 [49], with the additional option for users to define their own hybrid functionals via the `$xc-functional` keyword (see user-defined functionals in Section 4.3.9, below).

In addition, Q-Chem now includes the M05 and M06 suites of density functionals. These functionals are designed to be used only with certain definite percentages of Hartree-fock exchange. In particular, M06-L [42] is designed to be used with no Hartree-fock exchange (which reduces the cost for large molecules), and M05 [40], M05-2X [41], M06, and M06-2X [44] are designed to be used with 28% used with 100% density functional exchange because the Hartree-Fock nonlocal exchange replaces only some of the local exchange.

**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 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 [49], though without the atomic size adjustments”. The atomic integrals are then evaluated through standard one-center numerical techniques.

Thus, the exchange–correlation energy  $E_{XC}$  is obtained as

$$E_{XC} = \sum_A \sum_i w_{Ai} f(\mathbf{r}_{Ai}) \quad (4.43)$$

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

$$\mathbf{r}_{Ai} = \mathbf{R}_A + \mathbf{r}_i \quad (4.44)$$

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

The single-center integrations are further separated into radial and angular integrations. Within Q-CHEM, the radial part is usually treated by the Euler–Maclaurin scheme proposed by Murry *et al.* [50]. This scheme maps the semi-infinite domain  $[0, \infty) \rightarrow [0, 1)$  and applies the extended trapezoidal rule to the transformed integrand. Recently Gill and Chien [51] proposed a radial scheme based on a Gaussian quadrature on the interval  $[0, 1]$  with weight function  $\ln^2 x$ . This scheme is exact for integrands that are a linear combination of a geometric sequence of exponential functions, and is therefore well suited to evaluating atomic integrals. The authors refer to this scheme as MultiExp.

### 4.3.5 Angular Grids

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 [12, 13, 53] based on the octahedral group. Lebedev grids of the following degrees are available:

Degree	3rd	5th	7th	9th	11th	15th	17th	19th	23rd	29th
Points	6	18	26	38	50	86	110	146	194	302

Additional grids with the following number of points are also available: 74, 170, 230, 266, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, 5294. 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  $\theta$  and  $\phi$ . Integration in the  $\theta$  dimension is carried out with a Gaussian quadrature rule derived from the Legendre polynomials (orthogonal on  $[-1, 1]$  with weight function unity), while the  $\phi$  integration is done with equally spaced points.

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

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.

Combining these radial and angular schemes yields an intimidating selection of three-dimensional quadratures. In practice, is it useful to standardize the grids used in order to facilitate the comparison of calculations at different levels of theory.

### 4.3.6 Standard Quadrature Grids

Both the SG-0 [55] and SG-1 [56] standard quadrature grids were 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-0 grid was derived in this fashion from a MultiExp-Lebedev-(23,170), (*i.e.*, 23 radial points and 170 angular points per radial point). This grid was pruned whilst ensuring the error in the computed exchange energies for the atoms and a selection of small molecules was not larger than the corresponding error associated with SG-1. In our evaluation, the RMS error associated with the atomization energies for the molecules in the G1 data set is 72 microhartrees. While relative energies are expected to be reproduced well by this scheme, if absolute energies are being sought, a larger grid is recommended.

The SG-0 grid is implemented in Q-CHEM from H to micro Hartrees, excepted He and Na; in this scheme, each atom has around 1400-point, and SG-1 is used for those their SG-0 grids have not been defined. It should be noted that, since the SG-0 grid used for H has been re-optimized in this version of Q-CHEM (version 3.0), quantities calculated in this scheme may not reproduce those generated by the last version (version 2.1).

The SG-1 grid is derived 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.

Both the SG-0 and SG-1 grids were optimized so that the error in the energy when using the grid did not exceed a target threshold. For single point calculations this criterion is appropriate. However, derivatives of the energy can be more sensitive to the quality of the integration grid, and it is recommended that a larger grid be used when calculating these. Special care is required when performing DFT vibrational calculations as imaginary frequencies can be reported if the grid is inadequate. This is more of a problem with low-frequency vibrations. If imaginary frequencies are found, or if there is some doubt about the frequencies reported by Q-CHEM, the recommended procedure is to perform the calculation again with a larger grid and check for convergence of the frequencies. Of course the geometry must be re-optimized, but if the existing geometry is used as an initial guess, the geometry optimization should converge in only a few cycles.

#### 4.3.7 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 [57], 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  $\mathcal{O}(N)$  work for large molecules, where  $N$  is the size of the system. This is a significant improvement over the formal  $\mathcal{O}(N^3)$  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 signaled 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 maybe 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 [58] to ensure that the computed XC energy is the same for any orientation of the molecule in any Cartesian coordinate system.

### 4.3.8 Basic DFT Job Control

Three *\$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 4.2.3, and the extended options discussed in Section 4.2.4 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 hybrid.

TYPE:

STRING

DEFAULT:

No default exchange functional

OPTIONS:

HF	Fock exchange
Slater, S	Slater (Dirac 1930)
Becke, B	Becke 1988
Gill96, Gill	Gill 1996
GG99	Gilbert and Gill, 1999
Becke(EDF1), B(EDF1)	Becke (uses EDF1 parameters)
PW91, PW	Perdew
PBE	Perdew-Burke-Ernzerhof 1996
B97	Becke97 XC hybrid 1997
B97-1	Becke97 re-optimized by Hamprecht et al. 1998
B97-2	Becke97-1 optimized further by Wilson et al. 2001
B3PW91, Becke3PW91, B3P	B3PW91 hybrid
B3LYP, Becke3LYP	B3LYP hybrid
B3LYP5	B3LYP based on correlation functional #5 of Vosko, Wilk, and Nusair rather than their functional #3
EDF1	EDF1
EDF2	EDF2
BMK	BMK hybrid
M05	M05 hybrid
M05-2X	M05-2X hybrid
M06-L	M06-L hybrid
M06-HF	M06-HF hybrid
M06	M06 hybrid
M06-2X	M06-2X hybrid
General, Gen	User defined combination of K, X and C (refer next section).

RECOMMENDATION:

Consult the literature to guide your selection.

**CORRELATION**

Specifies the correlation functional.

TYPE:

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
PBE	Perdew–Burke–Ernzerhof 1996
Wigner	Wigner

RECOMMENDATION:

Consult the literature to guide your selection.

**FAST\_XC**

Controls direct variable thresholds to accelerate exchange correlation (XC) in DFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Turn FAST_XC on.
FALSE	Do not use FAST_XC.

RECOMMENDATION:

Caution: FAST\_XC improves the speed of a DFT calculation, but may occasionally cause the SCF calculation to diverge.

**XC\_GRID**

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

0 SG–0/SG–1 hybrid

OPTIONS:

0	Use SG–0 for H, C, N, and O, SG–1 for all other atoms.
1	Use SG–1 for all atoms.
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 4.3.4.
$-mn$	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 $n = 2N^2$ .

RECOMMENDATION:

Use default unless numerical integration problems arise. Larger grids may be required for optimization and frequency calculations.

**XC\_SMART\_GRID**

Uses SG-0 (where available) for early SCF cycles, and switches to the (larger) grid specified by XC\_GRID (which defaults to SG-1, if not otherwise specified) for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.

**4.3.9 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
```

**Note:** (1) Coefficients are real.

(2) A user-defined functional does not require all *X*, *C* and *K* components.

**4.3.10 Example**

**Example 4.5** Q-CHEM input for a DFT single point energy calculation on water.

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

$molecule
  O  1
  O
  H1 0 oh
  H2 0 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
```

## 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 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:

- Electron-electron Coulomb interactions, for which Q-CHEM incorporates the Continuous Fast Multipole Method (CFMM) discussed in section 4.4.2
- 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.
- 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 by the program based on 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 two 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 now at the Australian National University, were the first to develop the generalization of Greengard's Fast Multipole Method (FMM) [59] 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 led to an increasing number of linear scaling alternatives and analogies, 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 [14] the original Fast Multipole Method before generalizing it to the CFMM [61]. 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 [29, 63] then implemented it efficiently [64], 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 [65]. 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 [66] and forces [67] 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 [3].

The CFMM is controlled by the following input parameters:

#### **CFMM\_ORDER**

Controls the order of the multipole expansions in CFMM calculation.

TYPE:

INTEGER

DEFAULT:

15 For single point SCF accuracy

25 For tighter convergence (optimizations)

OPTIONS:

$n$  Use multipole expansions of order  $n$

RECOMMENDATION:

Use default.

#### **GRAIN**

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

TYPE:

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

RECOMMENDATION:

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 [69]. 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 [69] to evaluate both exchange energies and their gradients [56] 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 [56,69]. 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.

TYPE:

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 threshold used is computed as

$$tmp\_thresh = varthresh \times DIIS\_error \quad (4.45)$$

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 [18],  $\mathbf{F}$  is computed recursively as

$$\mathbf{F}^m = \mathbf{F}^{m-1} + \Delta\mathbf{J}^{m-1} - \frac{1}{2}\Delta\mathbf{K}^{m-1} \quad (4.46)$$

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

$$\Delta\mathbf{P}^m = \mathbf{P}^m - \mathbf{P}^{m-1} \quad (4.47)$$



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  
TYPE:  
INTEGER  
DEFAULT:  
1 Start INCFOCK after iteration number 1  
OPTIONS:  
User-defined (0 switches INCFOCK off)  
RECOMMENDATION:  
May be necessary to allow several iterations before switching on INCFOCK.

#### VARTHRESH

Controls the temporary integral cut-off threshold.  $tmp\_thresh = 10^{-\text{VARTHRESH}} \times DIIS\_error$   
TYPE:  
INTEGER  
DEFAULT:  
0 Turns VARTHRESH off  
OPTIONS:  
 $n$  User-defined threshold  
RECOMMENDATION:  
3 has been found to be a practical level, and can slightly speed up SCF evaluation.

### 4.4.5 Incremental DFT

Incremental DFT (IncDFT) uses the difference density and functional values to improve the performance of the DFT quadrature procedure by providing a better screening of negligible values. Using this option will yield improved efficiency at each successive iteration due to more effective screening.

#### INCDFT

Toggles the use of the IncDFT procedure for DFT energy calculations.  
TYPE:  
LOGICAL  
DEFAULT:  
TRUE  
OPTIONS:  
FALSE Do not use IncDFT  
TRUE Use IncDFT  
RECOMMENDATION:  
Turning this option on can lead to faster SCF calculations, particularly towards the end of the SCF. Please note that for some systems use of this option may lead to convergence problems.

**INCDFT\_DENDIFF\_THRESH**

Sets the threshold for screening density matrix values in the IncDFT procedure.

TYPE:

INTEGER

DEFAULT:

SCF\_CONVERGENCE + 3

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

**INCDFT\_GRIDDIFF\_THRESH**

Sets the threshold for screening functional values in the IncDFT procedure

TYPE:

INTEGER

DEFAULT:

SCF\_CONVERGENCE + 3

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

**INCDFT\_DENDIFF\_VARTHRESH**

Sets the lower bound for the variable threshold for screening density matrix values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT\_DENDIFF\_THRESH is reached. This means this value must be set lower than INCDFT\_DENDIFF\_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

**INCDFT\_GRIDDIFF\_VARTHRESH**

Sets the lower bound for the variable threshold for screening the functional values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT\_GRIDDIFF\_THRESH is reached. This means that this value must be set lower than INCDFT\_GRIDDIFF\_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

#### 4.4.6 Fourier Transform Coulomb Method

The Coulomb part of the DFT calculations using ‘ordinary’ Gaussian representations can be sped up dramatically using plane waves as a secondary basis set by replacing the most costly analytical electron repulsion integrals with numerical integration techniques. The main advantages to keeping the Gaussians as the primary basis set is that the diagonalization step is much faster than using plane waves as the primary basis set, and all electron calculations can be performed analytically.

The Fourier Transform Coulomb (FTC) technique [82,83] is precise and tunable and all results are practically identical with the traditional analytical integral calculations. The FTC technique is at least 2–3 orders of magnitude more accurate than other popular plane wave based methods using the same energy cutoff. It is also at least 2–3 orders of magnitude more accurate than the density fitting (resolution of identity) technique. Recently, an efficient way to implement the forces of the Coulomb energy was introduced [84], and a new technique to localize filtered core functions. Both of these features have been implemented within Q-CHEM and contribute to the efficiency of the method.

The FTC method achieves these spectacular results by replacing the analytical integral calculations, whose computational costs scales as  $\mathcal{O}(N^4)$  (where  $N$  is the number of basis function) with procedures that scale as only  $\mathcal{O}(N^2)$ . The asymptotic scaling of computational costs with system size is linear versus the analytical integral evaluation which is quadratic. Research at Q-CHEM Inc. has yielded a new, general, and very efficient implementation of the FTC method which work in tandem with the J-engine and the CFMM (Continuous Fast Multipole Method) techniques [85].

In the current implementation the speed-ups arising from the FTC technique are moderate when small or medium Pople basis sets are used. The reason is that the J-matrix engine and CFMM techniques provide an already highly efficient solution to the Coulomb problem. However, increasing the number of polarization functions and, particularly, the number of diffuse functions allows the FTC to come into its own and gives the most significant improvements. For instance, using the 6-311G+(df,pd) basis set for a medium-big sized molecule is more affordable today than before. We found also significant speed ups when non-Pople basis sets are used such as cc-pvTZ. The FTC energy and gradients calculations are implemented to use up to  $f$ -type basis functions.

##### FTC

Controls the overall use of the FTC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use FTC in the Coulomb part

1 Use FTC in the Coulomb part

RECOMMENDATION:

Use FTC when bigger and/or diffuse basis sets are used.

**FTC\_SMALLMOL**

Controls whether or not the operator is evaluated on a large grid and stored in memory to speed up the calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a big pre-calculated array to speed up the FTC calculations

0 Use this option to save some memory

RECOMMENDATION:

Use the default if possible and use 0 (or buy some more memory) when needed.

**FTC\_CLASS\_THRESH\_ORDER**

Together with FTC\_CLASS\_THRESH\_MULT, determines the cutoff threshold for included a shell-pair in the *dd* class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Logarithmic part of the FTC classification threshold. Corresponds to  $10^{-5}$

OPTIONS:

*n* User specified

RECOMMENDATION:

Use the default.

**FTC\_CLASS\_THRESH\_MULT**

Together with FTC\_CLASS\_THRESH\_ORDER, determines the cutoff threshold for included a shell-pair in the *dd* class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Multiplicative part of the FTC classification threshold. Together with the default value of the FTC\_CLASS\_THRESH\_ORDER this leads to the  $5 \times 10^{-5}$  threshold value.

OPTIONS:

*n* User specified.

RECOMMENDATION:

Use the default. If diffuse basis sets are used and the molecule is relatively big then tighter FTC classification threshold has to be used. According to our experiments using Pople-type diffuse basis sets, the default  $5 \times 10^{-5}$  value provides accurate result for an alanine5 molecule while  $1 \times 10^{-5}$  threshold value for alanine10 and  $5 \times 10^{-6}$  value for alanine15 has to be used.

#### 4.4.7 Examples

**Example 4.6** 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
  LIN_K      TRUE    Calculate K using LinK
$end
```

**Example 4.7** 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.

```
$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
```

## 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:

- 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.
- 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 (CORE)
- Generalized Wolfsberg–Helmholtz (GWH)
- Reading previously obtained MOs from disk. (READ)
- Basis set projection (BASIS2)

The first three of these guesses are built-in, and are briefly described in Section 4.5.2. The option of reading MOs from disk is described in Section 4.5.3. The initial guess MOs 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:
  - (a) 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,
  - (b) The SAD guess is not available for general (read-in) basis sets. All internal basis sets support the SAD guess.
  - (c) 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 [73] uses a combination of the overlap matrix elements (4.12), and the diagonal elements of the Core Hamiltonian matrix (4.18). 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.

$$H_{\mu\nu} = c_x S_{\mu\nu} (H_{\mu\mu} + H_{\nu\nu}) / 2 \quad (4.48)$$

3. **Core Hamiltonian:** The core Hamiltonian guess simply obtains the guess MO coefficients by diagonalizing the core Hamiltonian matrix (4.18). 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.

TYPE:

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. Note that because the density is made spherical, this may favor an undesired state for atomic systems, especially transition metals.

**SCF\_GUESS\_ALWAYS**

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

TYPE:

LOGICAL

DEFAULT:

False

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.

RECOMMENDATION:

Use default unless SCF convergence issues arise

**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
```

**Note:** (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.

**Note:** (1) 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).

**Note:** It is up to the user to make sure that the basis sets match between the two jobs. There is no internal checking for this, although the occupied orbitals are re-orthogonalized 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  $N_\alpha$  (or  $N_\beta$ ) 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 two 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.

**Example 4.8** Format for modifying occupied guess orbitals.

```
$occupied
  1 2 3 4 ... nalpha
  1 2 3 4 ... nbeta
$end
```

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

#### SCF\_GUESS\_PRINT

Controls printing of guess MOs, Fock and density matrices.

TYPE:

INTEGER

DEFAULT:

0

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.

RECOMMENDATION:

None



**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.

TYPE:

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 Inc. 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:

- A simple DFT calculation is performed in the small basis, BASIS2, yielding a converged density matrix in this basis.
- 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.
- By diagonalizing this matrix, an accurate initial guess for the density matrix in the large basis is obtained, and the target SCF calculation commences.

Two different methods of projection are available and can be set using the BASISPROJTYPE *\$rem*. The OVPROJECTION option expands the MOs from the BASIS2 calculation in the larger basis, while the FOPPROJECTION option constructs the Fock matrix in the larger basis using the density matrix from the initial, smaller basis set calculation. 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.

TYPE:

STRING

DEFAULT:

No second basis set default.

OPTIONS:

Symbol Use standard basis sets as per Chapter 7.

RECOMMENDATION:

BASIS2 should be smaller than BASIS. There is little advantage to using a basis larger than a minimal basis.

**BASISPROJTYPE**

Determines which method to use when projecting the density matrix of BASIS2  
 TYPE:  
 STRING  
 DEFAULT:  
 FOPPROJECTION  
 OPTIONS:  
 FOPPROJECTION Construct the Fock matrix in the second basis  
 OVPROJECTION Projects MO's from BASIS2 to BASIS.  
 RECOMMENDATION:  
 None

**4.5.6 Examples**

**Example 4.9** 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.

```

$molecule
  0 1
  0
  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
  0 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 4.10** 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.

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

$molecule
  1 1
  O 0.000 0.000 0.000
  H 0.000 0.000 1.000
$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 0.000 0.000 0.000
  H 0.000 0.000 1.000
$end

$rem
  BASIS          = 6-311++G(2df)
  EXCHANGE       = hf
  UNRESTRICTED  = false
  SCF_ALGORITHM = dm
  SCF_CONVERGENCE = 7
  SCF_GUESS     = read
  THRESH        = 10
$end
```

**Example 4.11** Input for an unrestricted HF calculation on H<sub>2</sub> 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 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).

```
$molecule
  0 1
  H 0.000 0.000 0.0
  H 0.000 0.000 -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
```

## 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, except for restricted open-shell SCF calculations.
- 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. GDM is the default algorithm for restricted open-shell SCF calculations.
- The older and less robust direct minimization method (DM). 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 the SCF convergence criterion and the integral threshold must be set in a compatible manner, (this usually means THRESH should be set to at least 3 higher than SCF\_CONVERGENCE).

#### MAX\_SCF\_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

**SCF\_ALGORITHM**

Algorithm used for converging the SCF.

TYPE:

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 geometric direct minimization is recommended. 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. Note that in Q-CHEM 3.0 the DIIS error is measured by the maximum error rather than the RMS error.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations.

7 For geometry optimizations and vibrational analysis.

8 For SSG calculations, see Chapter 5.

OPTIONS:

$n$  Corresponding to  $10^{-n}$

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 [6, 65] uses the property of an SCF solution which requires the density matrix to commute with the Fock matrix

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} \quad (4.49)$$

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

$$\mathbf{SP}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i\mathbf{S} = \mathbf{e}_i \quad (4.50)$$

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

$$\hat{\mathbf{F}}_k = \sum_{j=1}^{k-1} c_j \mathbf{F}_j \quad (4.51)$$

The DIIS coefficients  $c_k$ , are obtained by a least-squares constrained minimization of the error vectors, *viz*

$$Z = \left( \sum_k c_k \mathbf{e}_k \right) \cdot \left( \sum_k c_k \mathbf{e}_k \right) \quad (4.52)$$

where the constraint

$$\sum_k c_k = 1 \quad (4.53)$$

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

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_1 \cdot \mathbf{e}_N & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_N \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_N \cdot \mathbf{e}_N & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (4.54)$$

Convergence criteria requires the largest element of the  $N^{\text{th}}$  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

$$\hat{\mathbf{F}}_k = \sum_{j=k-(L+1)}^{k-1} c_j \mathbf{F}_j \quad (4.55)$$

where  $L$  is the size of the DIIS subspace. As the Fock matrix nears self-consistency the linear matrix equations (4.54) 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.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

RECOMMENDATION:

None

**DIIS\_PRINT**

Controls the output from DIIS SCF optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal print out.

1 Chosen method and DIIS coefficients and solutions.

2 Level 1 plus changes in multipole moments.

3 Level 2 plus Multipole moments.

4 Level 3 plus extrapolated Fock matrices.

RECOMMENDATION:

Use default

**Note:** In Q-CHEM 3.0 the DIIS error is determined by the maximum error rather than the RMS error. For backward compatibility the RMS error can be forced by using the following *\$rem*

**DIIS\_ERR\_RMS**

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Use default, the maximum error provides a more reliable criterion.

#### 4.6.4 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 hyper-spherical 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 [76]. GDM is a good alternative to DIIS for SCF jobs that exhibit convergence difficulties with DIIS.

Recently, Barry Dunietz, also working at Berkely with Martin Head-Gordon, has extended the GDM approach to restricted open-shell SCF calculations. Their results indicate that GDM is much more efficient than the older direct minimization method (DM).

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 SCF\_ALGORITHM which should be set to either GDM or DIIS\_GDM and the following:

#### **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.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM

*n* *n* DIIS iterations before switching to (G)DM.

RECOMMENDATION:

None

#### **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

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None



### 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 hyper-spherical 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.

#### PSEUDO\_CANONICAL

When SCF\_ALGORITHM = DM, this controls the way the initial step, and steps after subspace resets are taken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use Roothaan steps when (re)initializing

TRUE Use a steepest descent step when (re)initializing

RECOMMENDATION:

The default is usually more efficient, but choosing TRUE sometimes avoids problems with orbital reordering.

### 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. Q-CHEM contains the maximum overlap method (MOM) [77], developed by Andrew Gilbert and Peter Gill now at the Australian National University.

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 SCF 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\_PRINT

Switches printing on within the MOM procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Printing is turned off

TRUE Printing is turned on.

RECOMMENDATION:

None

**MOM\_START**

Determines when MOM is switched on to stabilize DIIS iterations.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE) MOM is not used

*n* MOM begins on cycle *n*.

RECOMMENDATION:

Set to 1 if preservation of initial orbitals is desired. If MOM is to be used to aid convergence, an SCF without MOM should be run to determine when the SCF starts oscillating. MOM should be set to start just before the oscillations.

**4.6.7 Examples**

**Example 4.12** 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
  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 = 2.672986
  rc3 = 1.354498
  tc3 = 62.851505
  rc5 = 1.372904
  ac5 = 116.454370
  rh1 = 1.085735
  rh2 = 1.085342
  ah2 = 122.157328
  rh4 = 1.087216
  ah4 = 119.523496
$end

$rem
  BASIS = 6-31G*
  EXCHANGE = hf
  INTSBUFFERSIZE = 15000000
  SCF_ALGORITHM = diis_gdm
  SCF_CONVERGENCE = 7
  THRESH = 10
$end

```

**Example 4.13** An example showing how to converge a ROHF calculation on the  $^3A_2$  state of

DMX. Note the use of reading in orbitals from a previous closed-shell calculation and the use of MOM to maintain the orbital occupancies. The  $^3B_1$  is obtained if MOM is not used.

```

$molecule
+1 1
  C      0.000000    0.000000    0.990770
  H      0.000000    0.000000    2.081970
  C     -1.233954    0.000000    0.290926
  C     -2.444677    0.000000    1.001437
  H     -2.464545    0.000000    2.089088
  H     -3.400657    0.000000    0.486785
  C     -1.175344    0.000000   -1.151599
  H     -2.151707    0.000000   -1.649364
  C      0.000000    0.000000   -1.928130
  C      1.175344    0.000000   -1.151599
  H      2.151707    0.000000   -1.649364
  C      1.233954    0.000000    0.290926
  C      2.444677    0.000000    1.001437
  H      2.464545    0.000000    2.089088
  H      3.400657    0.000000    0.486785
$end

$rem
  UNRESTRICTED    false
  EXCHANGE        hf
  BASIS           6-31+G*
  SCF_GUESS      core
$end

@@@

$molecule
  read
$end

$rem
  UNRESTRICTED    false
  EXCHANGE        hf
  BASIS           6-31+G*
  SCF_GUESS      read
  MOM_START      1
$end

$occupied
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28
$end

@@@

$molecule
-1 3
  ... <as above> ...
$end

$rem
  UNRESTRICTED    false
  EXCHANGE        hf
  BASIS           6-31+G*
  SCF_GUESS      read
$end

```

## 4.7 Dual-Basis Self-Consistent Field Calculations

The Dual-Basis approximation to self-consistent field energies provides an efficient means for approximating large basis set effects at vastly less cost than a full SCF calculation in a large basis set. The procedure is as follows [87]. First, a full SCF (HF or DFT) calculation is performed in a chosen small basis (specified by BASIS2). Second, a single SCF-like step in the large (target) basis (specified by BASIS) is used to perturbatively approximate the large basis energy. This correction amounts to a first-order approximation in the change in density matrix, after the single large-basis step:

$$E_{\text{total}} = E_{\text{small basis}} + \text{Tr}[(\Delta\mathbf{P})\mathbf{F}]_{\text{large basis}} \quad (4.56)$$

where  $\mathbf{F}$  (in the large basis) is built from the converged (small basis) density matrix. Thus, only a single Fock build is required in the large basis set. Currently, HF and DFT energies, as well as analytic first derivatives of HF energies (FORCE or OPT), are available.

In addition, the Dual-Basis approximation can be used for the reference energy of a correlated second order Møller-Plesset (MP2) calculation [88]. When activated, a Dual-Basis HF energy is first calculated; subsequently, the MO coefficients and orbital energies are used to calculate the correlation energy in the large basis. This technique is particularly effective for RI-MP2 calculations (see Section 5.5), in which the cost of the underlying SCF calculation often dominates.

Although any small/large basis set pairing can be handled by the code, we recommend using only proper subsets of the target basis. This choice not only produces more accurate results; it also leads to more efficient integral screening in both the energy and gradient. Subsets for cc-pVTZ and cc-pVQZ have been developed for this purpose [88] and are called “rcc-pVTZ” and “rcc-pVQZ”. In addition, 6-311+G\*\* can be used as a subset for 6-311++G(3df,3pd), the largest Pople-style set currently available.

Across the G3 set of 223 molecules, using cc-pVQZ, errors for MP2 are 0.27 kcal/mol (energy) and 0.08 kcal/mol (atomization energy per bond). Dual-Basis errors for B3LYP are 0.04 kcal/mol (energy) and 0.03 kcal/mol (atomization energy per bond).

Dual-Basis calculations are controlled with the following *\$rem*'s. DUAL\_BASIS\_ENERGY turns on the Dual-Basis approximation. Note that use of BASIS2 without DUAL\_BASIS\_ENERGY *only* uses basis set projection to generate the initial guess and does not invoke the Dual-Basis approximation. OVPROJECTION is used as the default projection mechanism for Dual-Basis calculations; it is not recommended that this be changed. In its current implementation, specification of SCF variables (e.g., THRESH) will apply to calculations in both basis sets.

### DUAL\_BASIS\_ENERGY

Activates dual-basis SCF (HF or DFT) energy correction.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Analytic first derivative available for HF (see JOBTYPE)

Can be used in conjunction with MP2 or RI-MP2

See BASIS, BASIS2, BASISPROJTYPE

RECOMMENDATION:

Use Dual-Basis to capture large-basis effects at smaller basis cost. Particularly useful with RI-MP2, in which HF often dominates. Use only proper subsets for small-basis calculation.

### 4.7.1 Examples

**Example 4.14** Input for a Dual-Basis B3LYP calculation.

```

$molecule
  0 1
  H
  H 1 0.75
$end

$rem
  JOBTYP      sp
  EXCHANGE    b3lyp
  BASIS        6-311+g**
  BASIS2       6-311g*
  DUAL_BASIS_ENERGY true
$end

```

**Example 4.15** Input for a Dual-Basis RI-MP2 calculation.

```

$molecule
  0 1
  H
  H 1 0.75
$end

$rem
  JOBTYP      sp
  EXCHANGE    hf
  CORRELATION  rimp2
  AUX_BASIS   rimp2-cc-pVQZ
  PURECART    11111 !for auxiliary set
  BASIS        cc-pVQZ
  BASIS2       rcc-pVQZ !special subset for cc-pVQZ
  DUAL_BASIS_ENERGY true
$end

```

## 4.8 Unconventional SCF Calculations

### 4.8.1 CASE Approximation

The Coulomb Attenuated Schrödinger Equation (CASE) [15] approximation follows from the KWIK [16] algorithm in which the Coulomb operator is separated into two pieces

$$\frac{1}{r_{12}} \equiv \frac{\operatorname{erfc}(\omega r_{12})}{r_{12}} + \frac{\operatorname{erf}(\omega r_{12})}{r_{12}} \quad (4.57)$$

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 (4.2) by neglecting the long-range portion of the identity in (4.57). The parameter  $\omega$  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  $\omega$  is chosen.

By virtue of the exponential decay of the attenuated operator, ERIs can be neglected on a proximity basis yielding a rigorous  $\mathcal{O}(N)$  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.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

$n$  Corresponding to  $\omega = n/1000$

RECOMMENDATION:

None

#### INTEGRAL\_2E\_OPR

Determines the two-electron operator.

TYPE:

INTEGER

DEFAULT:

-2 Coulomb Operator.

OPTIONS:

-1 Apply the CASE approximation.

-2 Coulomb Operator.

RECOMMENDATION:

Use default unless the CASE operator is desired.

### 4.8.2 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 (PAOs) is used [80]. 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. This deviation can be reduced to only 0.5 kcal/mol with the use of a simple non-iterative second order correction for “beyond-minimal basis” effects [35]. 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 PAOs, to permit structure optimization. For additional discussion of the PAO-SCF method and its uses, see the references cited above.

Calculations with PAOs 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 optimize polarized atomic orbitals (see PAO\_METHOD)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use efficient (and riskier) strategy to converge PAOs.

1 Use conservative (and slower) strategy to converge PAOs.

RECOMMENDATION:

None

**PAO\_METHOD**

Controls evaluation of polarized atomic orbitals (PAOs).

TYPE:

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.

RECOMMENDATION:

None

## 4.9 Ground State Method Summary

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

- Input a molecular geometry (*\$molecule* keyword)
  - Cartesian
  - *Z*-matrix
  - Read from prior calculations
- Declare the job specification (*\$rem*keyword)
  - JOBTYPED
    - \* Single point
    - \* Optimization
    - \* Frequency
    - \* See Table 4.1.2 for further options
  - 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 wavefunction-based 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.
- 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, PAOs.



# References and Further Reading

- [1] Basis sets (Chapter 7) and Effective Core Potentials (Chapter 8)
- [2] Molecular Geometry Critical Points (Chapter 9)
- [3] Molecular Properties analysis (Chapter 10)
- [4] AOINTS (Appendix B).
- [5] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York, 1986).
- [6] 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).
- [7] F. Jensen, *Introduction to Computational Chemistry* (Wiley, New York, 1999).
- [8] E. Schrödinger, *Ann. Physik.*, **79**, 361, (1926).
- [9] W. Heisenberg, *Z. Physik.*, **39**, 499, (1926).
- [10] M. Born and J. R. Oppenheimer, *Ann. Physik.*, **84**, 457, (1927).
- [11] J. C. Slater, *Phys. Rev.*, **34**, 1293, (1929).
- [12] J. C. Slater, *Phys. Rev.*, **35**, 509, (1930).
- [13] J. A. Pople and R. K. Nesbet, *J. Chem. Phys.*, **22**, 571, (1954).
- [14] R. Seeger and J. A. Pople, *J. Chem. Phys.*, **66**, 3045, (1977).
- [15] T. Takada, M. Dupuis and H. F. King, *J. Chem. Phys.*, **75**, 332, (1981).
- [16] M. Dupuis and H. F. King, *Int. J. Quantum Chem.*, **11**, 613, (1977).
- [17] M. Dupuis and H. F. King, *J. Chem. Phys.*, **68**, 3998, (1978).
- [18] R. G. Parr and W. Yang, *Density-Functional theory of atoms and molecules* (Oxford University Press: New York, 1989)
- [19] W. Kohn, A. D. Becke and R. G. Parr, *J. Phys. Chem.*, **100**, 12974, (1996).
- [20] See for example, T.Ziegler, *Chem. Rev.* **91**, 651, (1991); *Chemical Applications of Density Functional Methods*, edited by B.B.Laird, R.B.Ross, and T.Ziegler (American Chemical Society, Washington, 1996).
- [21] P. Hohenberg and W. Kohn, *Phys. Rev. B*, **136**, 864, (1964).
- [22] W. Kohn and L. J. Sham, *Phys. Rev. A*, **140**, 1133, (1965).
- [23] P. A. M. Dirac, *Proc. Cam. Phil. Soc.*, **26**, 376, (1930).

- [24] J. A. Pople, P. M. W. Gill and B. G. Johnson, *Chem. Phys. Lett.*, **199**, 557, (1992).
- [25] S. H. Vosko, L. Wilk and M. Nusair, *Can. J. Phys.*, **58**, 1200, (1980).
- [26] J. P. Perdew and A. Zunger, *Phys. Rev. B*, **23**, 5048, (1981).
- [27] E. P. Wigner, *Trans. Faraday Soc.*, **34**, 678, (1938).
- [28] A. D. Becke, *Phys. Rev. A*, **38**, 3098, (1988).
- [29] P. M. W. Gill, *Mol. Phys.*, **89**, 443, (1996).
- [30] A. T. B. Gilbert and P. M. W. Gill, *Chem. Phys. Lett.*, **312**, 511, (1999).
- [31] C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B*, **37**, 785, (1988).
- [32] J. P. Perdew, *Phys. Rev. B*, **33**, 8822, (1986).
- [33] J. R. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, *Phys. Rev. B*, **46**, 6671, (1992).
- [34] J.P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.*, **77**, 3865, (1996).
- [35] The PBE exchange and correlation functionals were obtained from the Density Functional Repository as developed and distributed by the Quantum Chemistry Group, CCLRC Daresbury Laboratory, Daresbury, Cheshire, WA4 4AD United Kingdom.
- [36] A. D. Becke, *J. Chem. Phys.*, **107**, 8554, (1997).
- [37] F. A. Hampercht, A. J. Cohen, D. J. Tozer and N. C. Handy, *J. Chem. Phys.*, **109**, 6264, (1998).
- [38] P. J. Wilson, T. J. Bradley and D. J. Tozer, *J. Chem. Phys.*, **115**, 9233, (2001).
- [39] A. Daniel Boese and Jan M. L. Martin, *J. Chem. Phys.*, **121**, 3405, (2004).
- [40] Y. Zhao, N. E. Schultz and D. G. Truhlar, *J. Chem. Phys.*, **123**, 161103, (2005).
- [41] Y. Zhao, N. E. Schultz and D. G. Truhlar, *J. Chem. Theory Comput.*, **2**, 364, (2006).
- [42] Y. Zhao and D. G. Truhlar, *J. Chem. Phys.*, **125**, 194101, (2006).
- [43] Y. Zhao and D. G. Truhlar, *J. Phys. Chem. A*, **110**, 13126, (2006).
- [44] Y. Zhao and D. G. Truhlar, *Theor. Chem. Acc.*, to be published.
- [45] Among M05-series and M06-series functionals, Zhao and Truhlar recommend M06-2X and M05-2X for main-group thermochemistry and kinetics, M06-L, M06 and M05 for organometallic and inorganometallic thermochemistry, M06-2X, M05-2X, M06-HF, and M06 for non-covalent interactions, and M06-HF for long-range charge transfer via a TDDFT approach.
- [46] R. D. Adamson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.*, **284**, 6, (1998).
- [47] A. D. Becke, *J. Chem. Phys.*, **98**, 1372, (1993).
- [48] P. J. Stephens, F. J. Devlin, C. F. Chabalowski and M. J. Frisch, *J. Phys. Chem.*, **98**, 11623, (1994).
- [49] A. D. Becke, *J. Chem. Phys.*, **88**, 2547, (1988).
- [50] C. W. Murray, N. C. Handy and G. J. Laming, *Mol. Phys.*, **78**, 997, (1993).
- [51] S.-H. Chien and P. M. W. Gill, *J. Comp. Chem.*, **24**, 732, (2003).

- [52] Lebedev, *Sibirsk. Mat. Zh.*, **18**, 132, (1977).
- [53] Lebedev, *Zh. Vychisl. Mat. Mat. Fiz.*, **15**, 48, (1975).
- [54] Lebedev, *Zh. Vychisl. Mat. Mat. Fiz.*, **16**, 293, (1976).
- [55] S.-H. Chien and P. M. W. Gill, *J. Comput. Chem.*, **27**, 730, (2006).
- [56] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Chem. Phys. Lett.*, **209**, 506, (1993).
- [57] A. A. Jarecki and E. R. Davidson, *Chem. Phys. Lett.*, **300**, 44, (1999).
- [58] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.*, **220**, 377, (1994).
- [59] L. Greengard, *The rapid evaluation of potential fields in particle systems*; The MIT Press: London, (1987).
- [60] C. A. White and M. Head-Gordon, *J. Chem. Phys.*, **101**, 6593, (1994).
- [61] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.*, **230**, 8, (1994).
- [62] C. A. White and M. Head-Gordon, *J. Chem. Phys.*, **105**, 5061, (1996).
- [63] C. A. White and M. Head-Gordon, *Chem. Phys. Lett.*, **257**, 647, (1996).
- [64] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head-Gordon, *Chem. Phys. Lett.*, **253**, 268, (1996).
- [65] C. A. White and M. Head-Gordon, *J. Chem. Phys.*, **104**, 2620, (1996).
- [66] Y. Shao and M. Head-Gordon, *Chem. Phys. Lett.*, **323**, 425, (2000).
- [67] Y. Shao and M. Head-Gordon, *J. Chem. Phys.*, **114**, 6572, (2001).
- [68] T. R. Adams, R. D. Adamson, and P. M. W. Gill, *J. Chem. Phys.*, **107**, 124, (1997).
- [69] E. Schwegler, M. Challacombe and M. Head-Gordon, *J. Chem. Phys.*, **106**, 9708, (1997).
- [70] C. Ochsenfeld, C. A. White and M. Head-Gordon, *J. Chem. Phys.*, **109**, 1663, (1998).
- [71] C. Ochsenfeld, *Chem. Phys. Lett.*, **327**, 216, (2000).
- [72] E. Schwegler and M. Challacombe, *J. Chem. Phys.*, **106**, 9708, (1996).
- [73] M. Wolfsberg and L. Helmholz, *J. Chem. Phys.*, **20**, 837, (1952).
- [74] P. Pulay, *J. Comput. Chem.*, **3**, 556, (1982).
- [75] P. Pulay, *Chem. Phys. Lett.*, **73**, 393, (1980).
- [76] T. Van Voorhis and M. Head-Gordon, *Mol. Phys.*, **100**, 1713, (2002).
- [77] A.T.B.Gilbert, G.B.W.Gill, and P.M.W.Gill, (in preparation).
- [78] R. D. Adamson, J. P. Dombroski and P. M. W. Gill, *Chem. Phys. Lett.*, **254**, 329, (1996).
- [79] J. P. Dombroski, S. W. Taylor and P. M. W. Gill, *J. Phys. Chem.*, **100**, 6272, (1996).
- [80] M.S.Lee and M.Head-Gordon, *J. Chem. Phys.*, **107**, 9085, (1997).
- [81] M.S.Lee and M.Head-Gordon, *Comp. Chem.*, **24**, 295, (2000).
- [82] L. Fusti-Molnar and P. Pulay, *J. Chem. Phys.*, **117**, 7827, (2002).
- [83] L. Fusti-Molnar and P. Pulay, *J. Chem. Phys.*, **116**, 7795, (2002).

- 
- [84] L. Fusti-Molnar *J. Chem. Phys.*, **119**, 11080, (2003).
- [85] L. Fusti-Molnar and Jing Kong, *J. Chem. Phys.*, **122**, 074108, (2005).
- [86] C.Y. Lin, M.W. George, and P.M.W. Gill, *Aust. J. Chem.*, **57**, 365, (2004).
- [87] W.Z. Liang, and M. Head-Gordon, *J. Phys. Chem. A*, **108**, 3206, (2004).
- [88] R.P. Steele, Y. Shao, R.A. DiStasio Jr, J. Kong and M. Head-Gordon, *J. Chem. Phys.* (in press).

## 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. Wavefunction-based approaches for describing electron correlation [5, 6] 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 Schrödinger equation. In order of increasing accuracy, and also increasing cost, they are:

1. Perturbative treatment of pair correlations between electrons, typically capable of recovering 80% or so of the correlation energy in stable molecules.
2. Self-consistent treatment of pair correlations between electrons, capable of recovering on the order of 95% or so of the correlation energy.
3. Non-iterative corrections for higher than double substitutions, which can 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 [7] and G3 methods [8].

These methods are discussed in the following three subsections.

There is also a fourth 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 section 5.8. Finally, equation of motion (EOM) methods provide tools for describing open-shell and electronically excited species.

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

- BASIS to specify the basis set (see Chapter 7)
- CORRELATION method for treating Correlation (defaults to NONE)
- N\_FROZEN\_CORE frozen core electrons (0 default, optionally FC, or *n*)

Additionally, for EOM calculations the number of target states in each irreducible representation (irrep) should also be specified (see section 6.5.5).

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.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

MP2	Sections 5.2 and 5.3
Local_MP2	Section 5.4
RILMP2	Section 5.5.1
ZAPT2	A more efficient restricted open-shell MP2 method [9].
MP3	Section 5.2
MP4SDQ	Section 5.2
MP4	Section 5.2
CCD	Section 5.6
CCD(2)	Section 5.7
CCSD	Section 5.6
CCSD(T)	Section 5.7
CCSD(2)	Section 5.7
QCISD	Section 5.6
QCISD(T)	Section 5.7
OD	Section 5.6
OD(T)	Section 5.7
OD(2)	Section 5.7
VOD	Section 5.8
VOD(2)	Section 5.8
QCCD	Section 5.6
VQCCD	Section 5.8

RECOMMENDATION:

Consult the literature for guidance.

## 5.2 Møller-Plesset Perturbation Theory

### 5.2.1 Introduction

Møller–Plesset Perturbation Theory [10] 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 [7, 8]. 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  $\Psi_0$  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  $\Psi_0$  and energy  $E_0$  lie near the exact wave function  $\Psi$  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  $\lambda$  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  $\lambda^n$  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  $\Psi_0$  and integrating over all space yields the following expression for the  $n^{\text{th}}$  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

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

of which the first term is the MP2 energy.

It can be shown that the MP2 energy can be written (in terms of spin-orbitals) as

$$E_0^{(2)} = -\frac{1}{4} \sum_{ab}^{\text{virt}} \sum_{ij}^{\text{occ}} \frac{|\langle ab || ij \rangle|^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \quad (5.13)$$

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 5.6 while the computational cost of MP4 is similar to the (T) corrections discussed in Section 5.7.

## 5.3 Exact MP2 Methods

### 5.3.1 Algorithm

Second order Møller–Plesset theory (MP2) [10] probably the simplest useful wave function-based electron correlation method. Revived in the mid-1970s, it remains highly popular today, because it offers systematic improvement in optimized geometries and other molecular properties relative to Hartree–Fock (HF) theory [3]. Indeed, in a recent comparative study of small closed-shell molecules [12], 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 [13]. 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 3<sup>rd</sup> and 5<sup>th</sup> 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 [8, 15], 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:

- Direct algorithm (energies only).
- Disk-based sortless semi-direct algorithm (energies and gradients).
- 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 four times. The `AO2MO_DISK` option is also ignored in this algorithm, which requires up to  $O^2VN$  megabytes of disk space.

There are three important options 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).

- MEM\_STATIC: The value specified for this  $\$rem$  variable must be sufficient to permit efficient integral evaluation (10-80Mb) and to hold a large temporary array whose size is  $OVN$ , the product of the number of occupied, virtual and total numbers of orbitals.
- AO2MO\_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.
- 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 The Definition of Core Electron

The number of core electrons in an atom is relatively well defined, and consists of certain atomic shells, (note that ECPs are available in ‘small-core’ and ‘large-core’ varieties, see Chapter 8 for further details). For example, in phosphorus the core consists of  $1s$ ,  $2s$ , and  $2p$  shells, for a total of ten electrons. In molecular systems, the core electrons are usually chosen as those occupying the  $n/2$  lowest energy orbitals, where  $n$  is the number of core electrons in the constituent atoms. In some cases, particularly in the lower parts of the periodic table, this definition is inappropriate and can lead to significant errors in the correlation energy. Vitaly Rassolov has implemented an alternative definition of core electrons within Q-CHEM which is based on a Mulliken population analysis, and which addresses this problem [16].

The current implementation is restricted to  $n-kl$  type basis sets such as 3-21 or 6-31, and related bases such as 6-31+G(d). There are essentially two cases to consider, the outermost 6G functions (or 3G in the case of the 3-21G basis set) for Na, Mg, K and Ca, and the 3d functions for the elements Ga–Kr. Whether or not these are treated as core or valence is determined by the CORE\_CHARACTER  $\$rem$ , as summarized in Table 5.3.2.

CORE_CHARACTER	Outermost 6G (3G) for Na, Mg, K, Ca	3d (Ga–Kr)
1	valence	valence
2	valence	core
3	core	core
4	core	valence

Table 5.1: A summary of the effects of different core definitions

### 5.3.3 Algorithm Control and Customization

The direct and semi-direct 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  $\$rem$  variables are:

**MEM\_STATIC**

Sets the memory for individual program modules.

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb.

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

For direct and semi-direct MP2 calculations, this must exceed  $OVN +$  requirements for AO integral evaluation (32–160 Mb), as discussed above.

**MEM\_TOTAL**

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 (2 Gb)

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

Use default, or set to the physical memory of your machine. Note that if more than 1GB is specified for a Coupled Cluster job, the memory is allocated as follows

12% MEM\_STATIC  
3% CC\_TMPBUFFSIZE  
50% CC\_CC\_BLK\_TNSR\_BUFFSIZE  
35% Other memory requirements:

**AO2MO\_DISK**

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

TYPE:

INTEGER

DEFAULT:

2000 Corresponding to 2000 Mb.

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

Should be set as large as possible, discussed in Section 5.3.1.

**CD\_ALGORITHM**

Determines the algorithm for MP2 integral transformations.

TYPE:

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:

Semi-direct 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.

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

*n* Freeze *n* virtual orbitals.

RECOMMENDATION:

None

**CORE\_CHARACTER**

Selects how the core orbitals are determined in the frozen-core approximation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy-based definition.

1-4 Use Muliken-based definition (see Table 5.3.2 for details).

RECOMMENDATION:

Use default, unless performing calculations on molecules with heavy elements.

**PRINT\_CORE\_CHARACTER**

Determines the print level for the CORE\_CHARACTER option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No additional output is printed.

1 Prints core characters of occupied MOs.

2 Print level 1, plus prints the core character of AOs.

RECOMMENDATION:

Use default, unless you are uncertain about what the core character is.

### 5.3.4 Example

**Example 5.1** Example of an MP2/6-31G\* calculation employing the frozen core approximation. Note that the EXCHANGE *\$rem* variable will default to HF

```
$molecule
  0 1
  0
  H1 0 oh
  H2 0 oh H1 hoh

  oh = 1.01
  hoh = 105
$end

$rem
  CORRELATION      mp2
  BASIS            6-31g*
  N_FROZEN_CORE    fc
$end
```

## 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 re-formulations 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

- *Size-consistent*: meaning that the energy of a super-system of two non-interacting molecules should be the sum of the energy obtained from individual calculations on each molecule.
- *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.
- *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 [18, 35] 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 four 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 three 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 anti-bonding 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* (EPAOs) [19]. 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* (PAOs) (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 two. 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  $VNO$ . The number of double substitutions in such a redundant representation will be typically three to five 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:

$$E_{\text{DIM MP2}} = -\frac{1}{2} \sum_{\bar{P}\bar{Q}} \frac{(\bar{P}|\bar{Q})(\bar{P}||\bar{Q})}{\Delta_{\bar{P}} + \Delta_{\bar{Q}}} \quad (5.17)$$

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:

$$E_{\text{TRIM MP2}} = - \sum_{\bar{P}bj} \frac{(\bar{P}|jb)(\bar{P}||jb)}{\Delta_{\bar{P}} + \varepsilon_b - \varepsilon_j} - E_{\text{DIM MP2}} \quad (5.18)$$

where the sum is now mixed between atomic substitutions  $\bar{P}$ , and nonlocal occupied  $j$  to virtual  $b$  substitutions. See references [18,35] for a full derivation and discussion.

The accuracy of the local TRIM and DIM models has been tested in a series of calculations [18,35]. In particular, the TRIM model has been shown to be quite faithful to full MP2 theory via the following tests:

- 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 [20]. The DIM model recovers around 95% of the correlation energy.
- The performance of the TRIM model for relative energies is very robust, as shown in ref. [35] 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.
- 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.
- For calculations of the MP2 contribution to the G3 and G3(MP2) energies with the larger molecules in the G3–99 database [21], 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 (EPAOs) that spans the occupied space. There are three valid choices for this basis, controlled by the PAO\_METHOD and EPAO\_ITERATE keywords described below.

- *Uniterated EPAOs*: The initial guess EPAOs 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. [19] 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 [35] 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 EPAOs.
- *EPAOs*: EPAOs are defined by minimizing a localization functional as described in ref. [19]. 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.

- *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.

TYPE:

STRING

DEFAULT:

`EPAO` For local MP2, EPAOs are chosen by default.

OPTIONS:

`EPAO` Find EPAOs by minimizing delocalization function.

`PAO` Do SCF in a molecule-optimized minimal basis.

RECOMMENDATION:

None

#### **EPAO\_ITERATE**

Controls iterations for EPAO calculations (see `PAO_METHOD`).

TYPE:

INTEGER

DEFAULT:

0 Use uniterated EPAOs based on atomic blocks of SPS.

OPTIONS:

$n$  Optimize the EPAOs 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 optimized EPAOs in which case a value of  $n = 500$  is reasonable.

#### **EPAO\_WEIGHTS**

Controls algorithm and weights for EPAO calculations (see `PAO_METHOD`).

TYPE:

INTEGER

DEFAULT:

115 Standard weights, use 1<sup>st</sup> and 2<sup>nd</sup> order optimization

OPTIONS:

15 Standard weights, with 1<sup>st</sup> order optimization 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 EPAOs 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 [22]. 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*: 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 governed by a threshold. This is a very large reduction from the case of a full MP2 calculation, where, in the case of four integral evaluations,  $OVN^2/4$  disk space is required. As the local MP2 disk space requirement is not adjustable, the AO2MO\_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 MEM\_STATIC (80 to 160 Mb) should be specified to permit efficient AO basis two-electron integral evaluation, but all large scratch arrays are allocated from MEM\_TOTAL.

#### 5.4.4 Examples

**Example 5.2** 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.

```

$molecule
O 1
C
C 1 1.32095
C 2 1.47845 1 121.19
O 3 1.18974 2 123.83 1 180.00
H 1 1.07686 2 121.50 3 0.00
H 1 1.07450 2 122.09 3 180.00
H 2 1.07549 1 122.34 3 180.00
H 3 1.09486 2 115.27 4 180.00
$end

$rem
CORRELATION local_mp2
BASIS 6-311g**
$end

@@@

$molecule
O 1
C

```

```

C 1 1.31656
C 2 1.49838 1 123.44
O 3 1.18747 2 123.81 1 92.28
H 1 1.07631 2 122.03 3 -0.31
H 1 1.07484 2 121.43 3 180.28
H 2 1.07813 1 120.96 3 180.34
H 3 1.09387 2 115.87 4 179.07
$end

$rem
CORRELATION local_mp2
BASIS 6-311g**
$end

```

## 5.5 Auxiliary Basis Set (Resolution of the Identity) MP2 Methods.

For a molecule of fixed size, increasing the number of basis functions *per atom*,  $n$ , leads to  $\mathcal{O}(n^4)$  growth in the number of significant four-center two-electron integrals, since the number of non-negligible product charge distributions,  $|\mu\nu\rangle$ , grows as  $\mathcal{O}(n^2)$ . As a result, the use of large (high-quality) basis expansions is computationally costly. Perhaps the most practical way around this “basis set quality” bottleneck is the use of auxiliary basis expansions [23,25,28]. The ability to use auxiliary basis sets to accelerate a variety of electron correlation methods, including both energies and analytical gradients, is one of the major new features of Q-CHEM 3.0.

The auxiliary basis  $\{|K\rangle\}$  is used to approximate products of Gaussian basis functions:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C_{\mu\nu}^K \quad (5.19)$$

Auxiliary basis expansions were introduced long ago, and are now widely recognized as an effective and powerful approach, which is sometimes synonymously called resolution of the identity (RI) or density fitting (DF). When using auxiliary basis expansions, the rate of growth of computational cost of large-scale electronic structure calculations with  $n$  is reduced to approximately  $n^3$ .

If  $n$  is fixed and molecule size increases, auxiliary basis expansions reduce the pre-factor associated with the computation, while not altering the scaling. The important point is that the prefactor can be reduced by 5 or 10 times or more. Such large speedups are possible because the number of auxiliary functions required to obtain reasonable accuracy,  $X$ , has been shown to be only about 3 or 4 times larger than  $N$ .

The auxiliary basis expansion coefficients,  $\mathbf{C}$ , are determined by minimizing the deviation between the fitted distribution and the actual distribution,  $\langle\mu\nu - \widetilde{\mu\nu}|\mu\nu - \widetilde{\mu\nu}\rangle$ , which leads to the following set of linear equations:

$$\sum_L \langle K|L\rangle C_{\mu\nu}^L = \langle K|\mu\nu\rangle \quad (5.20)$$

Evidently solution of the fit equations requires only two- and three-center integrals, and as a result the (four-center) two-electron integrals can be approximated as the following optimal expression for a given choice of auxiliary basis set:

$$\langle\mu\nu|\lambda\sigma\rangle \approx \langle\widetilde{\mu\nu}|\widetilde{\lambda\sigma}\rangle = \sum_K LC_{\mu}^L \langle L|K\rangle C_{\lambda\sigma}^K \quad (5.21)$$

In the limit where the auxiliary basis is complete (*i.e.* all products of AOs are included), the fitting procedure described above will be exact. However, the auxiliary basis is invariably incomplete

(as mentioned above,  $X \approx 3N$ ) because this is essential for obtaining increased computational efficiency. Standardized auxiliary basis sets have been developed by the Karlsruhe group for second order perturbation (MP2) calculations [26,27] of the correlation energy. With these basis sets, small absolute errors (e.g. below 60  $\mu$ -Hartree per atom in MP2) and even smaller relative errors in computed energies are found, while the speed-up can be 3–30 fold. This development has made the routine use of auxiliary basis sets for electron correlation calculations possible.

Correlation calculations that can take advantage of auxiliary basis expansions are described in the remainder of this section (MP2, and MP2-like methods) and in Section 5.9 (simplified active space coupled cluster methods such as PP, PP(2), IP, RP). These methods automatically employ auxiliary basis expansions when a valid choice of auxiliary basis set is supplied using the AUX\_BASIS keyword which is used in the same way as the BASIS keyword. The PURECART  $\$rem$  is no longer needed here, even if using a auxiliary basis that does not have a predefined value. There is a built-in automatic procedure that provides the effect of the PURECART  $\$rem$  in these cases by default.

### 5.5.1 RI-MP2 energies and gradients.

Following common convention, the MP2 energy evaluated approximately using an auxiliary basis is referred to as “resolution of the identity” MP2, or RI-MP2 for short. RIMP2 energy and gradient calculations are enabled simply by specifying the AUX\_BASIS keyword discussed above. As discussed above, RI-MP2 energies [28] and gradients [29,30] are significantly faster than the best conventional MP2 energies and gradients, and cause negligible loss of accuracy, when an appropriate standardized auxiliary basis set is employed. Therefore they are recommended for jobs where turnaround time is an issue. Disk requirements are very modest – one merely needs to hold various 3-index arrays. Memory requirements grow more slowly than our conventional MP2 algorithms – only quadratically with molecular size. The minimum memory requirement is approximately  $3X^2$ , where  $X$  is the number of auxiliary basis functions, for both energy and analytical gradient evaluations, with some additional memory being necessary for integral evaluation and other small arrays.

In fact, for molecules that are not too large (perhaps no more than 20 or 30 heavy atoms) the RI-MP2 treatment of electron correlation is so efficient that the computation is dominated by the initial Hartree-Fock calculation. This is despite the fact that as a function of molecule size, the cost of the RI-MP2 treatment still scales more steeply with molecule size (it is just that the pre-factor is so much smaller with the RI approach). Its scaling remains 5<sup>th</sup> order with the size of the molecule, which only dominates the initial SCF calculation for larger molecules. Thus, for RI-MP2 energy evaluation on moderate size molecules (particularly in large basis sets), it is desirable to use the dual basis HF method to further improve execution times (see Section 4.7).

### 5.5.2 Example

**Example 5.3** Q-CHEM input for an RIMP2 geometry optimization.

```
$molecule
O 1
O
H 1 0.9
F 1 1.4 2 100.
$end

$rem
JOBTYPE      opt
CORRELATION  rimp2
```

```
BASIS          cc-pvtz
AUX_BASIS     rimp2-cc-pvtz
% PURECART     1111
SYMMETRY      false
$end
```

For the size of required memory, the followings need to be considered.

#### MEM\_STATIC

Sets the memory for AO-integral evaluations and their transformations.

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb.

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

For RI-MP2 calculations,  $150(ON + V)$  of MEM\_STATIC is required. Because a number of matrices with  $N^2$  size also need to be stored, 32–160 Mb of additional MEM\_STATIC is needed.

#### MEM\_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 (2 Gb)

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

Use default, or set to the physical memory of your machine. The minimum requirement is  $3X^2$ .

### 5.5.3 Opposite spin (SOS-MP2 and MOS-MP2) energies and gradients.

The accuracy of MP2 calculations can be significantly improved by semi-empirically scaling the opposite-spin and same-spin correlation components with separate scaling factors, as shown by Grimme [31]. Results of similar quality can be obtained by just scaling the opposite spin correlation (by 1.3), as was recently demonstrated [32]. Furthermore this SOS-MP2 energy can be evaluated using the RI approximation together with a Laplace transform technique, in effort that scales only with the 4<sup>th</sup> power of molecular size. Efficient algorithms for the energy [32] and the analytical gradient [33] of this method are available in Q-CHEM 3.0, and offer advantages in speed over MP2 for larger molecules, as well as statistically significant improvements in accuracy.

However, we note that the SOS-MP2 method does systematically underestimate long-range dispersion (for which the appropriate scaling factor is 2 rather than 1.3) but this can be accounted for by making the scaling factor distance-dependent, which is done in the modified opposite spin variant (MOS-MP2) that has recently been proposed and tested [34]. The MOS-MP2 energy and analytical gradient are also available in Q-CHEM 3.0 at a cost that is essentially identical with SOS-MP2. Timings show that the 4<sup>th</sup> order implementation of SOS-MP2 and MOS-MP2 yields substantial speedups over RIMP2 for molecules in the 40 heavy atom regime and larger. It is also possible to customize the scale factors for particular applications, such as weak interactions, if required.

A fourth order scaling SOS-MP2/MOS-MP2 energy calculation can be invoked by setting the CORRELATION keyword to either SOSMP2 or MOSMP2. MOS-MP2 further requires the specification of the *\$rem* variable OMEGA, which tunes the level of attenuation of the MOS operator:

$$= \frac{1}{r_{12}} + c_{\text{MOS}} \cdot \frac{\text{erf}(\omega r_{12})}{r_{12}} \quad (5.22)$$

The recommended OMEGA value is  $\omega = 0.6 \text{ a.u.}$  [34]. The fast algorithm makes use of auxiliary basis expansions and therefore, the keyword AUX\_BASIS should be set consistently with the user's choice of BASIS. Fourth-order scaling analytical gradient for both SOS-MP2 and MOS-MP2 are also available and is automatically invoked when JOBTYP E is set to OPT or FORCE. The minimum memory requirement is  $3X^2$ , where  $X$  = the number of auxiliary basis functions, for both energy and analytical gradient evaluations. Disk space requirement for closed shell calculations is  $\sim 2OVX$  for energy evaluation and  $\sim 4OVX$  for analytical gradient evaluation.

Summary of key *\$rem* variables to be specified:

CORRELATION	SOSMP2 MOSMP2
JOBTYP E	sp (default) single point energy evaluation opt geometry optimization with analytical gradient force force evaluation with analytical gradient
BASIS	user's choice (standard or user-defined: general/mixed)
AUX_BASIS	corresponding auxiliary basis (standard or user-defined: aux_general/aux_mixed)
OMEGA	no default n $\omega = n/1000$ 600 recommended value $\omega = 0.6 \text{ a.u.}$
N_FROZEN_CORE	Optional
N_FROZEN_VIRTUAL	Optional

### 5.5.4 Examples

**Example 5.4** Example of SOS-MP2 geometry optimization

```

$molecule
  0 3
  C1
  H1  C1  1.07726
  H2  C1  1.07726  H1  131.60824
$end

$rem
  JOBTYP E      opt
  CORRELATION   sosmp2
  BASIS         cc-pvdz
  AUX_BASIS     rimp2-cc-pvdz
%  PURECART    1111
  UNRESTRICTED true
  SYMMETRY     false
$end

```

**Example 5.5** Example of MOS-MP2 energy evaluation with frozen core approximation

```

$molecule
  0 1
  C1
  C1 1 2.05
$end

$rem

```

```

JOBTYPE          sp
CORRELATION      mosmp2
OMEGA            600
BASIS            cc-pVTZ
AUX_BASIS        rimp2-cc-pVTZ
% PURECART        1111
N_FROZEN_CORE    fc
THRESH           12
SCF_CONVERGENCE  8
$end

```

### 5.5.5 RI-TRIM MP2 Energies

The triatomics in molecules (TRIM) local correlation approximation to MP2 theory [35] was described in detail in Section 5.4.1 which also discussed our implementation of this approach based on conventional four-center two-electron integrals. Q-CHEM 3.0 also includes an auxiliary basis implementation of the TRIM model. The new RI-TRIM MP2 energy algorithm [36] greatly accelerates these local correlation calculations (often by an order of magnitude or more for the correlation part), which scale with the 4<sup>th</sup> power of molecule size. The electron correlation part of the calculation is speeded up over normal RI-MP2 by a factor proportional to the number of atoms in the molecule. For a hexadecapeptide, for instance, the speedup is approximately a factor of 4 [36]. The TRIM model can also be applied to the scaled opposite spin models discussed above. As for the other RI-based models discussed in this section, we recommend using RI-TRIM MP2 instead of the conventional TRIM MP2 code whenever run-time of the job is a significant issue. As for RI-MP2 itself, TRIM MP2 is invoked by adding AUX\_BASIS *\$rem*s to the input deck, in addition to requesting CORRELATION = RILMP2.

**Example 5.6** Example of RI-TRIM MP2 energy evaluation

```

$molecule
  0 3
  C1
  H1  C1  1.07726
  H2  C1  1.07726  H1  131.60824
$end

$rem
CORRELATION      rilmp2
BASIS            cc-pVDZ
AUX_BASIS        rimp2-cc-pVDZ
% PURECART        1111
UNRESTRICTED     true
SYMMETRY         false
$end

```

## 5.6 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 Professors Anna Krylov and 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 Prof. Steve Gwaltney at Berkeley, while the extensions to QCCD were implemented by Ed Byrd at Berkeley.

### 5.6.1 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 [37], 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.

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:

$$|\Psi_{\text{CCSD}}\rangle = \exp(\hat{T}_1 + \hat{T}_2)|\Phi_0\rangle \quad (5.23)$$

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):

$$\hat{T}_1|\Phi_0\rangle = \sum_i^{\text{occ}} \sum_a^{\text{virt}} t_i^a |\Phi_i^a\rangle \quad (5.24)$$

$$\hat{T}_2|\Phi_0\rangle = \frac{1}{4} \sum_{ij}^{\text{occ}} \sum_{ab}^{\text{virt}} t_{ij}^{ab} |\Phi_{ij}^{ab}\rangle \quad (5.25)$$

It is unfeasible to determine the CCSD energy by variational minimization of  $\langle E \rangle_{\text{CCSD}}$  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:

$$\begin{aligned} E_{\text{CCSD}} &= \langle \Phi_0 | \hat{H} | \Psi_{\text{CCSD}} \rangle \\ &= \left\langle \Phi_0 \left| \hat{H} \left| \left( 1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2 \right) \Phi_0 \right\rangle_C \right. \end{aligned} \quad (5.26)$$

$$\begin{aligned} 0 &= \left\langle \Phi_i^a \left| \hat{H} - E_{\text{CCSD}} \right| \Psi_{\text{CCSD}} \right\rangle \\ &= \left\langle \Phi_i^a \left| \hat{H} \left| \left( 1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 + \frac{1}{3!} \hat{T}_1^3 \right) \Phi_0 \right\rangle_C \right. \end{aligned} \quad (5.27)$$

$$\begin{aligned} 0 &= \left\langle \Phi_{ij}^{ab} \left| \hat{H} - E_{\text{CCSD}} \right| \Psi_{\text{CCSD}} \right\rangle \\ &= \left\langle \Phi_{ij}^{ab} \left| \hat{H} \left| \left( 1 + \hat{T}_1 + \frac{1}{2} \hat{T}_1^2 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 + \frac{1}{3!} \hat{T}_1^3 \right. \right. \right. \\ &\quad \left. \left. \left. + \frac{1}{2} \hat{T}_2^2 + \frac{1}{2} \hat{T}_1^2 \hat{T}_2 + \frac{1}{4!} \hat{T}_1^4 \right) \Phi_0 \right\rangle_C \right. \end{aligned} \quad (5.28)$$

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 6<sup>th</sup> power of the molecular size, for fixed choice of basis set. Disk storage scales with the 4<sup>th</sup> 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 non feasibility) of a CCSD calculation on a larger molecule (or with a larger basis set) given that a smaller trial calculation is first performed. Q-CHEM supports both energies and analytic gradients for CCSD for RHF and UHF references (including frozen-core). For ROHF, only energies and properties are available.

### 5.6.2 Quadratic Configuration Interaction (QCISD)

Quadratic configuration interaction with singles and doubles (QCISD) [38] 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 non variational. 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:

$$E_{QCISD} = \langle \Phi_0 | \hat{H} | (1 + \hat{T}_2) \Phi_0 \rangle_C \quad (5.29)$$

$$0 = \langle \Phi_i^a | \hat{H} | (\hat{T}_1 + \hat{T}_2 + \hat{T}_1 \hat{T}_2) \Phi_0 \rangle_C \quad (5.30)$$

$$0 = \langle \Phi_{ij}^{ab} | \hat{H} | \left(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2\right) \Phi_0 \rangle_C \quad (5.31)$$

QCISD energies are available in Q-CHEM, and are requested with the QCISD keyword. As discussed in Section 5.7, the non iterative QCISD(T) correction to the QCISD solution is also available to approximately incorporate the effect of higher substitutions.

### 5.6.3 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:

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

$$0 = \langle \Phi_{ij}^{ab} | \hat{H} | \left(1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2\right) \Phi_0 \rangle_C \quad (5.33)$$

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) [39]. 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:

$$\frac{\partial E_{\text{CCD}}}{\partial \theta_i^a} = 0 \quad (5.34)$$

where the rotation angle  $\theta_i^a$  mixes the  $i^{\text{th}}$  occupied orbital with the  $a^{\text{th}}$  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 artificial symmetry breaking, or non-convergence), 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 5.8.

#### 5.6.4 Quadratic Coupled Cluster Doubles (QCCD)

The non variational 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 [40]. The Quadratic Coupled Cluster Doubles (QCCD) method [41] 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:

$$E_{\text{QCCD}} = \left\langle \Phi_0 \left( 1 + \hat{\Lambda}_2 + \frac{1}{2} \hat{\Lambda}_2^2 \right) \middle| \hat{H} \middle| \exp \left( \hat{T}_2 \right) \Phi_0 \right\rangle_C \quad (5.35)$$

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:

$$E_{\text{CCD}} = \left\langle \Phi_0 \left( 1 + \hat{\Lambda}_2 \right) \left| \hat{H} \right| \exp \left( \hat{T}_2 \right) \Phi_0 \right\rangle_C \quad (5.36)$$

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, (5.35) and (5.36), 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 [40] 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 [41], and the follow-up paper describing the full implementation [42]. We note that these improvements carry a computational price. While QCCD scales formally with the 6<sup>th</sup> 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. Q-CHEM also contains some configuration interaction models (CISD and CISDT). The CI methods are inferior to CC due to size-consistency issues, however, these models may be useful for benchmarking and development purposes.

### 5.6.5 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 and EOM methods discussed later in Section 5.8. The more common options relating to convergence control are discussed there, in Section 5.8.4. Below we list the options that one should be aware of for routine calculations.

#### MEM\_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 (2 Gb)

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

Use default, or set to the physical memory of your machine. Note that if more than 1GB is specified for a Coupled Cluster job, the memory is allocated as follows

12% MEM\_STATIC

3% CC\_TMPBUFFSIZE

50% CC\_CC\_BLK\_TNSR\_BUFFSIZE

35% Other memory requirements:

**CC\_DO\_CISDT**

Controls the calculation of CISDT

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute triples (do CISD only)

1 Do compute the triples correction (full CISDT)

RECOMMENDATION:

None

**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) [ $10^{-n}$ ].

TYPE:

INTEGER

DEFAULT:

8 Energies.

8 Gradients.

OPTIONS:

$n$  Corresponding to  $10^{-n}$  convergence criterion.

RECOMMENDATION:

None

**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.

TYPE:

INTEGER

DEFAULT:

2502 Corresponding to 0.25

OPTIONS:

$abcde$  Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

Increase to 0.5 or 0.75 for non convergent coupled-cluster calculations.

**CC\_MAXITER**

Maximum number of iterations to optimize the coupled-cluster energy.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

$n$  up to  $n$  iterations to achieve convergence.

RECOMMENDATION:

None

**CC\_PRINT**

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

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

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

RECOMMENDATION:

Increase if you need more output and don't like trees

**5.6.6 Example**

**Example 5.7** A series of jobs evaluating the correlation energy (with core orbitals frozen) of the ground state of the NH<sub>2</sub> radical with three methods of coupled-cluster singles and doubles type: CCSD itself, OD, and QCCD.

```
$molecule
  0 2
  N
  H1 N 1.02805
  H2 N 1.02805 H1 103.34
$end

$rem
  CORRELATION  ccsd
  BASIS        6-31g*
  N_FROZEN_CORE fc
$end

@@@

$molecule
  read
$end

$rem
  CORRELATION  od
  BASIS        6-31g*
  N_FROZEN_CORE fc
$end

@@@

$molecule
  read
$end

$rem
  CORRELATION  qccd
  BASIS        6-31g*
  N_FROZEN_CORE fc
$end
```

## 5.7 Non-iterative Corrections to Coupled Cluster Energies

### 5.7.1 (T) Triples Corrections

To approach chemical accuracy in reaction energies and related properties, it is necessary to account for electron correlation effects that involve three 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) [43] and QCISD(T) methods [38]. The accuracy of these methods is well-documented for many cases [44], 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 7<sup>th</sup> power of molecule size (or the 4<sup>th</sup> 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 and properties are not currently available for any of these (T) corrections.

### 5.7.2 (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 4<sup>th</sup> and 5<sup>th</sup> order Møller-Plesset perturbation theory, as well as higher order terms that result from the fact that the converged cluster amplitudes are employed to evaluate the 4<sup>th</sup> and 5<sup>th</sup> 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 non iterative correction that offers the prospect of improved accuracy in problem cases of the types identified above [45]. 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 two of the cluster methods discussed above, as OD(2) and CCSD(2) [45, 46]. 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 5<sup>th</sup> order Møller-Plesset theory [47], 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 7<sup>th</sup> 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 [45]. 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.

### 5.7.3 Job Control Options

The evaluation of a non iterative (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 non iterative (2) corrections, such as OD(2) and CCSD(2).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

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).

### 5.7.4 Example

**Example 5.8** 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.

```
    O
    H 0 0.97907
$end

$rem
    CORRELATION    ccsd(t)
    BASIS          cc-pvtz
    N_FROZEN_CORE  fc
$end

@@@

$molecule
    read
$end

$rem
    CORRELATION    ccsd(2)
    BASIS          cc-pvtz
    N_FROZEN_CORE  fc
    CC_INCL_CORE_CORR  false
$end
```

## 5.8 Coupled Cluster Active Space Methods

### 5.8.1 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 more 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 Schrödinger 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) [48], or equivalently, the fully optimized reaction space (FORS) method [49].

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 [50]. They are active space generalizations of the OD and QCCD methods discussed previously in Sections 5.6.3 and 5.6.4, 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 6<sup>th</sup> 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:

$$\frac{\partial E_{\text{CCD}}}{\partial \theta_i^{j'}} = 0 \quad (5.37)$$

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.

$$\frac{\partial E_{\text{CCD}}}{\partial \theta_a^{b'}} = 0 \quad (5.38)$$

## 5.8.2 VOD and VOD(2) Methods

The VOD method is the active space version of the OD method described earlier in Section 5.6.3. 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 non variational 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.

### 5.8.3 VQCCD

The VQCCD method is the active space version of the QCCD method described earlier in Section 5.6.3. 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 non variational 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 tens of kcal/mol. The computational cost scales in the same manner with system size as the VOD method, albeit with a significantly larger prefactor.

### 5.8.4 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.

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Activates procedure 2 initially, and procedure 1 when gradients are smaller than DIIS12\_SWITCH.

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.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

$n$  User-defined

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.

TYPE:

INTEGER

DEFAULT:

2502 Corresponding to 0.25

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

Increase to 0.5 or 0.75 for non convergent 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.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

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 01001 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.

TYPE:

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.

### 5.8.5 Examples

**Example 5.9** 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
  O 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 5.10** 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!

```
$molecule
  O 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
```

## 5.9 Simplified Coupled-Cluster Methods Based on a Perfect Pairing Active Space.

Molecules where electron correlations are strong are characterized by small energy gaps between the nominally occupied orbitals (that would comprise the Hartree-Fock wavefunction, for example) and nominally empty orbitals. Examples include so-called diradicaloid molecules [51], or molecules with partly broken chemical bonds (and thus some transition structures). Because the energy gap is small, other electron configurations apart from the reference determinant contribute to the molecular wavefunction with considerable amplitude, and it is a significant error to omit them from the wavefunction. Including all possible configurations is vast overkill, and it is common to restrict the configurations that one generates to constructed not from all molecular orbitals, but rather just from orbitals that are either “core” or “active”. In this section, we consider just one type of active space, which is composed of two orbitals to represent each electron pair, one nominally occupied (bonding or lone pair in character) and the other nominally empty, or correlating (it is typically antibonding in character). This is usually called the perfect pairing active space, and it clearly is well-suited to represent the bond-antibond correlations that are associated with bond-breaking.

The exact quantum chemistry in this (or any other) active space is given by a Complete Active Space SCF (CASSCF) calculation, whose exponential cost growth with molecule size makes it prohibitive for systems with more than about 14 active orbitals. One well-defined coupled cluster approximation to CASSCF is to include only double substitutions in the valence space, whose orbitals are optimized. In the framework of conventional CC theory, this defines the valence optimized doubles (VOD) model [50], which scales as  $\mathcal{O}(N^6)$  (see Section 5.8.2). This is still too expensive to readily apply to large molecules.

The methods described in this section bridge the gap between sophisticated but expensive coupled cluster methods and inexpensive methods such as DFT, HF and MP2 theory that may be (and indeed often are) inadequate for describing molecules that exhibit strong electron correlations such as diradicals. The coupled cluster perfect pairing (PP) [52, 53], imperfect pairing (IP) [54] and restricted coupled cluster (RCC) [55] models are local approximations to VOD that include only a linear and quadratic number of double substitution amplitudes respectively. They are close in spirit to generalized valence bond (GVB) type wavefunctions [56], because in fact they are all coupled cluster models for GVB which share the same perfect pairing active space. We shall therefore sometimes collectively refer to PP, IP and RCC as GVB methods in the remainder of this section.

To be specific, the coupled cluster PP wavefunction is written as:

$$|\Psi\rangle = \exp\left(\sum_{i=1}^{n_{\text{active}}} t_i \hat{a}_{i*}^\dagger \hat{a}_{i*}^\dagger \hat{a}_{\bar{i}} \hat{a}_i\right) |\Phi\rangle \quad (5.39)$$

where  $n_{\text{active}}$  is the number of active electrons, and the  $t_i$  are the linear number of unknown cluster amplitudes, corresponding to exciting the two electrons in the  $i$ th electron pair from their bonding orbital pair to their antibonding orbital pair. In addition to  $t_i$ , the core and active orbitals are optimized as well to minimize the PP energy. The algorithm used for this is a slight modification of the GDM method, described for SCF calculations in Section 4.6.4. Despite the simplicity of the PP wavefunction, with only a linear number of correlation amplitudes, it is still a useful theoretical model chemistry with which to explore highly correlated systems. This is because it is exact for a single electron pair in the PP active space, and thus it is also exact for a collection of non-interacting electron pairs in this active space. And molecules, after all, are simply collections of interacting electron pairs! In practice PP on molecules recovers between 60% and 80% of the correlation energy in its active space.

Cases where PP needs improvement include molecules with several strongly correlated electron pairs that are all localized in the same region of space, and therefore involve significant inter-pair, as well as intra-pair correlations. For this purpose, we have the IP and RCC wavefunctions. The corresponding wavefunction expression for the IP and RCC wavefunctions includes an additional quadratic number of cluster amplitudes,  $t_{ij}$  that describes the correlation of an electron in the  $i^{\text{th}}$  pair with an electron in the  $j^{\text{th}}$  pair. IP and RCC are virtually identical physically. Generally IP should be used unless bonds are being completely broken with restricted orbitals, in which case RCC is preferred as it has been constructed to eliminate the tendency of restricted coupled cluster methods to become non-variational in the dissociation limit. IP and RCC typically recover between 80% and 95% of the correlation energy in their perfect pairing active spaces.

In Q-CHEM 3.0, PP, IP and RCC are implemented with a new resolution of the identity (RI) algorithm that makes them computationally very efficient [57, 58]. The PP model is available with restricted and unrestricted orbitals, while the IP and RCC models are only available with restricted orbitals (work on an unrestricted implementation is in progress). They can be applied to systems with more than 100 active electrons, and both energies and analytical gradients are available. These methods are requested via the standard CORRELATION keyword, according to CORRELATION = PP, CORRELATION = IP, and CORRELATION = RCC respectively. If AUX\_BASIS is not specified, the calculation uses four-center two-electron integrals by default. Much faster auxiliary basis algorithms (see 5.5 for an introduction), which are used for the correlation energy (not the reference SCF energy), can be enabled by specifying a valid string for AUX\_BASIS. The example below illustrates a simple IP calculation.

**Example 5.11** Imperfect pairing with auxiliary basis set for geometry optimization.

```

$molecule
  O 1
  H
  F 1 1.0
$end

$rem
  JOBTYP  opt
  CORRELATION  ip
  BASIS  cc-pVDZ
  AUX_BASIS  rimp2-cc-pVDZ
% PURECART  11111
$end

```

There are often considerable challenges in converging the orbital optimization associated with these GVB-type calculations. The situation is somewhat analogous to SCF calculations but more severe because there are more orbital degrees of freedom that affect the energy (for instance, mixing occupied active orbitals amongst each other, mixing active virtuals with each other, mixing core and active occupied, mixing active virtual and inactive virtual). Furthermore the energy changes associated with many of these new orbital degrees of freedom are rather small and delicate. As a consequence, in cases where the correlations are strong, these GVB-type jobs often require many more iterations than the corresponding GDM calculations at the SCF level. This is a reflection of the correlation model itself. To deal with convergence issues, a number of REM values are available to customize the calculations, as listed below.

**GVB\_ORB\_MAX\_ITER**

Controls the number of orbital iterations allowed in GVB-CC calculations. Some jobs, particularly unrestricted PP jobs can require 500–1000 iterations.

TYPE:

INTEGER

DEFAULT:

256

OPTIONS:

User-defined number of iterations.

RECOMMENDATION:

Default is typically adequate, but some jobs, particularly UPP jobs, can require 500–1000 iterations if converged tightly.

**GVB\_ORB\_CONV**

The GVB-CC wave function is considered converged when the root-mean-square orbital gradient and orbital step sizes are less than  $10^{-\text{GVB\_ORB\_CONV}}$ . Adjust THRESH simultaneously.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

$n$  User-defined

RECOMMENDATION:

Use 6 for PP(2) jobs or geometry optimizations. Tighter convergence (*i.e.* 7 or higher) cannot always be reliably achieved.

**GVB\_ORB\_SCALE**

Scales the default orbital step size by  $n/1000$ .

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

$n$  User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but for some stretched geometries it can help with convergence to use smaller values.

**GVB\_AMP\_SCALE**

Scales the default orbital amplitude iteration step size by  $n/1000$  for IP/RCC. PP amplitude equations are solved analytically, so this parameter does not affect PP.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

$n$  User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but in some highly-correlated systems it can help with convergence to use smaller values.

**GVB\_RESTART**

Restart a job from previously-converged GVB-CC orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Useful when trying to converge to the same GVB solution at slightly different geometries, for example.

Another issue that a user of these methods should be aware of is the fact that there is a multiple minimum challenge associated with PP,IP and RCC calculations. In SCF calculations it is sometimes possible to converge to more than one set of orbitals that satisfy the SCF equations at a given geometry. The same problem can arise in GVB calculations, and based on our experience to date, the problem in fact is more commonly encountered in GVB calculations than in SCF calculations. A user may therefore want to (or have to!) tinker with the initial guess used for the calculations. One way is to set `GVB_RESTART = TRUE` (see above), to replace the default initial guess (the converged SCF orbitals which are then localized). Another way is to change the localized orbitals that are used in the initial guess, which is controlled by the `GVB_LOCAL` variable, described below. Sometimes different localization criteria, and thus different initial guesses, lead to different converged solutions.

**GVB\_LOCAL**

Sets the localization scheme used in the initial guess wave function.

TYPE:

INTEGER

DEFAULT:

2 Pipek-Mezey orbitals

OPTIONS:

1 Boys localized orbitals

2 Pipek-Mezey orbitals

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. It can be helpful to try both to ensure the global minimum has been found.

If the calculation is perfect pairing (`CORRELATION = PP`), it is possible to look for unrestricted solutions in addition to restricted ones. Indeed there is no implementation of restricted open shell orbitals for PP in Q-CHEM 3.0. Unrestricted orbitals are the default for molecules with odd



numbers of electrons, but can also be specified for molecules with even numbers of electrons. This is accomplished by setting `GVB_UNRESTRICTED = TRUE`. Given a restricted guess, this will, however usually converge to a restricted solution anyway, so additional REM variables should be specified to ensure an initial guess that has broken spin symmetry. This can be accomplished by using an unrestricted SCF solution as the initial guess, using the techniques described in Chapter 4. Alternatively a restricted set of guess orbitals can be explicitly symmetry broken just before the calculation starts by using `GVB_GUESS_MIX`, which is described below.

### **GVB\_UNRESTRICTED**

Controls restricted versus unrestricted PP jobs. Usually handled automatically.

TYPE:

LOGICAL

DEFAULT:

same value as UNRESTRICTED

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set this variable explicitly only to do a UPP job from an RHF or ROHF initial guess.

### **GVB\_GUESS\_MIX**

Similar to `SCF_GUESS_MIX`, it breaks alpha-beta symmetry for UPP by mixing the alpha HOMO and LUMO orbitals according to the user-defined fraction of LUMO to add the HOMO. 100 corresponds to a 1:1 ratio of HOMO and LUMO in the mixed orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  User-defined,  $0 \leq n \leq 100$

RECOMMENDATION:

25 often works well to break symmetry without overly impeding convergence.

Other REM variables relevant to GVB calculations are given below. It is possible to explicitly set the number of active electron pairs using the `GVB_N_PAIRS` variable. The default is to make all valence electrons active. Other reasonable choices are certainly possible. For instance all electron pairs could be active ( $n_{\text{active}} = n_{\beta}$ ). Or alternatively one could make only formal bonding electron pairs active ( $n_{\text{active}} = N_{\text{STO-3G}} - n_{\alpha}$ ). Or in some cases, one might want only the most reactive electron pair to be active ( $n_{\text{active}} = 1$ ). Clearly making physically appropriate choices for this variable is essential for obtaining physically appropriate results!

**GVB\_N\_PAIRS**

Alternative to CC\_REST\_OCC and CC\_REST\_VIR for setting active space size in GVB and valence coupled cluster methods.

TYPE:

INTEGER

DEFAULT:

PP active space (1 occ and 1 virt for each valence electron pair)

OPTIONS:

*n* user-defined

RECOMMENDATION:

Use default unless one wants to study a special active space. When using small active spaces, it is important to ensure that the proper orbitals are incorporated in the active space. If not, use the *\$reorder\_mo* feature to adjust the SCF orbitals appropriately.

**GVB\_PRINT**

Controls the amount of information printed during a GVB-CC job.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

*n* User-defined

RECOMMENDATION:

Should never need to go above 0 or 1.

The PP and IP models are potential replacements for HF theory as a zero order description of electronic structure and can be used as a starting point for perturbation theory. They neglect all correlations that involve electron configurations with one or more orbitals that are outside the active space. Physically this means that so-called “dynamical correlations”, which correspond to atomic-like correlations involving high angular momentum virtual levels are neglected. This means that the GVB models cannot be very accurate for describing energy differences which are sensitive to this neglected correlation energy – for instance atomization energies. It is desirable to correct them for this neglected correlation in a way that is similar to how the HF reference is corrected via MP2 perturbation theory.

For this purpose, the leading (second order) correction to the PP model, termed PP(2) [59], has been formulated and efficiently implemented for restricted and unrestricted orbitals (energy only). PP(2) improves upon many of the worst failures of MP2 theory (to which it is analogous), such as for open shell radicals. PP(2) also greatly improves relative energies relative to PP itself. PP(2) is implemented using a resolution of the identity (RI) approach to keep the computational cost manageable. This cost scales in the same 5<sup>th</sup> order way with molecular size as RI-MP2, but with a pre-factor that is about 5 times larger. It is therefore vastly cheaper than CCSD or CCSD(T) calculations which scale with the 6<sup>th</sup> and 7<sup>th</sup> powers of system size respectively. PP(2) calculations are requested with CORRELATION = PP(2). Since the only available algorithm uses auxiliary basis sets, it is essential to also provide a valid value for AUX.BASIS to have a complete input file.

The example below shows a PP(2) input file for the challenging case of the N<sub>2</sub> molecule with a stretched bond. For this reason a number of the non-standard options outlined earlier for orbital convergence are enabled here. First, this case is an unrestricted calculation on a molecule with an even number of electrons, and so it is essential to break the alpha-beta spin symmetry in order to find an unrestricted solution. Second, we have chosen to leave the lone pairs uncorrelated, which

is accomplished by specifying GVB\_N\_PAIRS.

**Example 5.12** A non-standard PP(2) calculation. UPP(2) for stretched N<sub>2</sub> with only 3 correlating pairs Try Boys localization scheme for initial guess.

```
$molecule
  0 1
  N
  N 1 1.65
$end

$rem
  UNRESTRICTED true
  CORRELATION pp(2)
  EXCHANGE hf
  BASIS cc-pvdz
  AUX_BASIS rimp2-cc-pvdz must use RI with PP(2)
% PURECART 11111
  SCF_GUESS_MIX 10 mix SCF guess 100{\%}
  GVB_GUESS_MIX 25 mix GVB guess 25{\%} also!
  GVB_N_PAIRS 3 correlate only 3 pairs
  GVB_ORB_CONV 6 tighter convergence
  GVB_LOCAL 1 use Boys initial guess
$end
```

We have already mentioned a few issues associated with GVB calculations: the neglect of dynamic correlation (which can be remedied with PP(2)), the convergence challenges and the multiple minimum issues. Another weakness of these GVB methods is occasional symmetry-breaking artifacts that are a consequence of the limited number of retained pair correlation amplitudes. For example, benzene in the PP approximation prefers D<sub>3h</sub> symmetry over D<sub>6h</sub> by 3 kcal/mol (with a 2° distortion), while in IP, this difference is reduced to 0.5 kcal/mol and less than 1° [54]. Likewise the allyl radical breaks symmetry in the unrestricted PP model [53], although to a lesser extent than in restricted open shell HF. Another occasional weakness is the limitation to the perfect pairing active space, which is not necessarily appropriate for molecules with expanded valence shells, such as in some transition metal compounds (*e.g.* expansion from 4s3d into 4s4p3d) or possibly hypervalent molecules (expansion from 3s3p into 3s3p3d). The singlet strongly orthogonal geminal method (see the next Section) is capable of dealing with expanded valence shells and could be used for such cases. The perfect pairing active space is satisfactory for most organic and first row inorganic molecules.

To summarize, while these GVB methods are powerful and can yield much insight when used properly, they do have enough pitfalls that they should not be considered true “black box” methods.

## 5.10 Geminal models

### 5.10.1 Reference wavefunction

Computational models that use single reference wavefunction describe molecules in terms of independent electrons interacting via mean Coulomb and exchange fields. It is natural to improve this description by using correlated electron pairs, or *geminals*, as building blocks for molecular wavefunctions. Requirements of computational efficiency and size consistency constrain geminals to have  $S_z = 0$  [60], with each geminal spanning its own subspace of molecular orbitals [61]. Geminal wavefunctions were introduced into computational chemistry by Hurley, Lennard-Jones, and Pople [62]. An excellent review of the history and properties of geminal wavefunctions is given by Surjan [63].

We implement a size consistent model chemistry based on Singlet type Strongly orthogonal Geminals (SSG). In SSG, the number of molecular orbitals in each singlet electron pair is an adjustable parameter chosen to minimize total energy. Open shell orbitals remain uncorrelated. The SSG wavefunction is computed by setting SSG \$rem variable to 1. Both spin-restricted (RSSG) and spin-unrestricted (USSG) versions are available, chosen by the UNRESTRICTED \$rem variable.

The wavefunction has the form

$$\begin{aligned}\Psi_{\text{SSG}} &= \hat{A} [\psi_1(\mathbf{r}_1, \mathbf{r}_2) \dots \psi_{n_\beta}(\mathbf{r}_{2n_\beta-1}, \mathbf{r}_{2n_\beta}) \phi_i(\mathbf{r}_{2n_\beta+1}) \dots \phi_j(\mathbf{r}_{n_\beta+n_\alpha})] \\ \psi_a(\mathbf{r}_1, \mathbf{r}_2) &= \sum_{k \in A} \frac{D_i^A}{\sqrt{2}} [\phi_k(\mathbf{r}_1) \bar{\phi}_k(\mathbf{r}_2) - \phi_k(\mathbf{r}_2) \bar{\phi}_k(\mathbf{r}_1)] \\ \phi_k(\mathbf{r}_1) &= \sum_{\lambda} C_{\lambda}^k \chi_{\lambda}(\mathbf{r}_1) \\ \bar{\phi}_k(\mathbf{r}_1) &= \sum_{\lambda} \bar{C}_{\lambda}^k \chi_{\lambda}(\mathbf{r}_1)\end{aligned}\tag{5.40}$$

with the coefficients  $C$ ,  $D$ , and subspaces  $A$  chosen to minimize the energy

$$E_{\text{SSG}} = \frac{\langle \Psi_{\text{SSG}} | \hat{H} | \Psi_{\text{SSG}} \rangle}{\langle \Psi_{\text{SSG}} | \Psi_{\text{SSG}} \rangle}\tag{5.41}$$

evaluated with the exact hamiltonian  $\hat{H}$ . A constraint  $\bar{C}_{\lambda}^k = C_{\lambda}^k$  for all MO coefficients yields a spin-restricted version of SSG.

SSG model can use any orbital-based initial guess. It is often advantageous to compute Hartree-Fock orbitals and then read them as initial guess for SSG. The program distinguishes Hartree-Fock and SSG initial guess wavefunctions, and in former case makes preliminary assignment of individual orbital pairs into geminals. The verification of orbital assignments is performed every ten wavefunction optimization steps, and the orbital pair is reassigned if total energy is lowered.

The convergence algorithm consists of combination of three types of minimization steps. The direct minimization steps [64] seeks a minimum along the gradient direction, rescaled by the quantity analogous to the orbital energy differences in SCF theory [60]. If the orbitals are nearly degenerate or inverted, a perturbative re-optimization of single geminal is performed. Finally, new set of the coefficients  $C$  and  $D$  is formed from a linear combination of previous iterations, in a manner similar to DIIS algorithm [6, 65]. The size of iterative subspace is controlled by the DIIS\_SUBSPACE\_SIZE keyword.

After convergence is achieved, SSG reorders geminals based on geminal energy. The energy, along with geminal expansion coefficients, is printed for each geminal. Presence of any but the leading coefficient with large absolute value (value of 0.1 is often used for the definition of “large”) indicates the importance of electron correlation in the system. The Mulliken population analysis is also performed for each geminal, which enables easy assignment of geminals into such chemical objects as core electron pairs, chemical bonds, and lone electron pairs.

As an example, consider the sample calculation of ScH molecule with 6-31G basis set at the experimental bond distance of 1.776 Å. In its singlet ground state the molecule has 11 geminals. Nine of them form core electrons on Sc. Two remaining geminals are:

*Geminal 10*  $E = -1.342609$   
0.99128 -0.12578 -0.03563 -0.01149 -0.01133 -0.00398

*Geminal 11*  $E = -0.757086$   
0.96142 -0.17446 -0.16872 -0.12414 -0.03187 -0.01227 -0.01204 -0.00435 -0.00416 -0.00098

Mulliken population analysis shows that geminal 10 is delocalized between Sc and H, indicating a bond. It is moderately correlated, with second expansion coefficient of a magnitude 0.126. The geminal of highest energy is localized on Sc. It represents  $4s^2$  electrons and describes their excitation into  $3d$  orbitals. Presence of three large expansion coefficients show that this effect cannot be described within GVB framework [67].

### 5.10.2 Perturbative corrections

The SSG description of molecular electronic structure can be improved by perturbative description of missing inter-geminal correlation effects. We have implemented Epstein–Nesbet form of perturbation theory [68, 69] that permits a balanced description of one- and two-electron contributions to excited states' energies in SSG model. This form of perturbation theory is especially accurate for calculation of weak intermolecular forces. Also, two-electron  $[i\bar{j}, j\bar{i}]$  integrals are included in the reference Hamiltonian in addition to intra-geminal  $[i\bar{j}, i\bar{j}]$  integrals that are needed for reference wavefunction to be an eigenfunction of the reference Hamiltonian [70].

All perturbative contributions to the SSG(EN2) energy (second-order Epstein–Nesbet perturbation theory of SSG wavefunction) are analyzed in terms of largest numerators, smallest denominators, and total energy contributions by the type of excitation. All excited states are subdivided into dispersion-like with correlated excitation within one geminal coupled to the excitation within another geminal, single, and double electron charge transfer. This analysis permits careful assessment of the quality of SSG reference wavefunction. Formally, the SSG(EN2) correction can be applied both to RSSG and USSG wavefunctions. Experience shows that molecules with broken or nearly broken bonds may have divergent RSSG(EN2) corrections. USSG(EN2) theory is balanced, with largest perturbative corrections to the wavefunction rarely exceeding 0.1 in magnitude.

#### SSG

Controls the calculation of the SSG wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute the SSG wavefunction

1 Do compute the SSG wavefunction

RECOMMENDATION:

See also the UNRESTRICTED and DIIS\_SUBSPACE\_SIZE *\$rem* variables.

# References and Further Reading

- [1] Self-consistent field methods (Chapter 4)
- [2] Excited state calculations (Chapter 6)
- [3] Basis sets (Chapter 7)
- [4] Effective core potentials (Chapter 8)
- [5] 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).
- [6] 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).
- [7] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, *J. Chem. Phys.*, **94**, 722, (1991).
- [8] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople, *J. Chem. Phys.*, **109**, 7764, (1998).
- [9] D. Jayatilaka and T.J. Lee, *Chem. Phys. Lett.*, **199**, 211, (1992).
- [10] C. Møller and M. S. Plesset, *Phys. Rev.*, **46**, 618, (1934).
- [11] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York 1986).
- [12] T. Helgaker, J. Gauss, P. Jorgensen, and J. Olsen, *J. Chem. Phys.*, **106**, 6430, (1997).
- [13] M. Head-Gordon, *Mol. Phys.*, **96**, 673, (1999).
- [14] M. Head-Gordon, J. A. Pople and M. J. Frisch, *Chem. Phys. Lett.*, **153**, 503, (1988).
- [15] M. J. Frisch, M. Head-Gordon, and J. A. Pople, *Chem. Phys. Lett.*, **166**, 275, (1990).
- [16] V.A. Rassolov, J.A. Pople, P. C. Redfern and L.A. Curtiss, *Chem. Phys. Lett.*, **350**, 573, (2001).
- [17] M. S. Lee, P. E. Maslen, and M. Head-Gordon, *J. Chem. Phys.* **112**, 3592, (2000).
- [18] 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.
- [19] M. S. Lee and M. Head-Gordon, *Int. J. Quant. Chem.*, **76**, 169, (2000).
- [20] S. Saebo and P.Pulay, *Ann. Rev. Phys. Chem.*, **44**, 213, (1993).

- [21] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, *J. Chem. Phys.*, **112**, 7374, (2000).
- [22] M. S. Lee, Ph. D. Thesis, University of California, Berkeley, (2000).
- [23] B. I. Dunlap, *Phys. Chem. Chem. Phys.*, **2**, 2113, (2000).
- [24] M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.*, **208**, 359, (1993).
- [25] Y. Jung, A. Sodt, P.M.W. Gill, and M. Head-Gordon, *Proc. Natl. Acad. Sci. U.S.A.*, **102**, 6692, (2005).
- [26] F. Weigend, M. Haser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.*, **294**, 143, (1998).
- [27] F. Weigend, A. Kohn, and C. Hattig, *J. Chem. Phys.*, **116**, 3175, (2002).
- [28] M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.*, **208**, 359, (1993).
- [29] F. Weigend, and M. Haser, *Theor. Chem. Acc.*, **97**, 331, (1997).
- [30] R.A. DiStasio Jr, P. Steele, Y.M. Rhee, Y. Shao and M. Head-Gordon *J. Comp. Chem.* (submitted).
- [31] S. Grimme, *J. Chem. Phys.*, **118**, 9095, (2003).
- [32] Y.S. Jung, R.C. Lochan, A.D. Dutoi, and M. Head-Gordon, *J. Chem. Phys.*, **121**, 9793, (2004).
- [33] R.C. Lochan and M. Head-Gordon (in preparation).
- [34] R.C. Lochan, Y. Jung, and M. Head-Gordon, *J. Phys. Chem. A*, **109**, 7598, (2005).
- [35] M.S. Lee, P.E. Maslen, and M. Head-Gordon, *J. chem. Phys.*, **112**, 3592, (2000).
- [36] R.A. DiStasio, Y.S. Jung, and M. Head-Gordon, *J. Chem. Theor. Comput.*, **1**, 862, (2005).
- [37] G. D. Purvis and R. J. Bartlett, *J. Chem. Phys.*, **76**, 1910, (1982).
- [38] J. A. Pople, M. Head-Gordon and K. Raghavachari, *J. Chem. Phys.*, **87**, 5968, (1987).
- [39] C. D. Sherrill, A. I. Krylov, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.*, **109**, 4171, (1998).
- [40] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.*, **113**, 8873, (2000).
- [41] T. Van Voorhis and M. Head-Gordon, *Chem. Phys. Lett.*, **330**, 585, (2000).
- [42] E.F.C. Byrd, T. Van Voorhis and M. Head-Gordon *J. Phys. Chem. B*, **106**, 8070, (2002).
- [43] K. Raghavachari, G. W. Trucks, J. A. Pople and M. Head-Gordon, *Chem. Phys. Lett.*, **157**, 479, (1989).
- [44] 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.
- [45] S. R. Gwaltney and M. Head-Gordon, *Chem. Phys. Lett.*, **323**, 21, (2000).
- [46] S. R. Gwaltney and M. Head-Gordon *J. Chem. Phys.*, **115**, 5033, (2001).
- [47] S. A. Kucharski and R. J. Bartlett, *J. Chem. Phys.*, **108**, 5243, (1998).
- [48] For an introduction to the CASSCF method, see, B. O. Roos, *Adv. Chem. Phys.*, **69**, 399, (1987).

- [49] See for example, K. Ruedenberg, M. W. Schmidt, M. M. Gilbert, and S. T. Elbert, *Chem. Phys.*, **71**, 49, (1982).
- [50] A. I. Krylov, C. D. Sherrill, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.*, **109**, 10669, (1998).
- [51] Y.S. Jung, and M. Head-Gordon, *ChemPhysChem*, **4**, 522, (2003).
- [52] J. Cullen, *Chem. Phys.*, **202**, 217, (1996).
- [53] G.J.O. Beran, B. Austin, A. Sodt, and M. Head-Gordon, *J. Phys. Chem. A*, **109**, 9183, (2005).
- [54] T. Van Voorhis, and M. Head-Gordon, *Chem. Phys. Lett.*, **317**, 575, (2000).
- [55] T. Van Voorhis, and M. Head-Gordon, *J. Chem. Phys.*, **115**, 7814, (2001).
- [56] W.A. Goddard, and L.B. Harding, *Annu. Rev. Phys. Chem.*, **29**, 363, (1978).
- [57] T. Van Voorhis, and M. Head-Gordon, *J. Chem. Phys.*, **117**, 9190, (2002).
- [58] A. Sodt, G.J.O. Beran, Y. Jung, B. Austin, and M. Head-Gordon, *J. Chem. Theor. Comput.* (in press, 2006).
- [59] G.J.O. Beran, S.R. Gwaltney, and M. Head-Gordon, *J. Chem. Phys.* (in press).
- [60] V. A. Rassolov, *J. Chem. Phys.*, **117**, 5978, (2002).
- [61] T. Arai, *J. Chem. Phys.*, **33**, 95, (1960).
- [62] A. C. Hurley, J. E. Lennard-Jones, and J. A. Pople, *Proc. R. Soc. London, Ser. A*, **220**, 446, (1953).
- [63] P. R. Surján, *Topics in Current Chemistry*, 1999,
- [64] R. Seeger and J. A. Pople, *J. Chem. Phys.*, **65**, 265, (1976).
- [65] P. Pulay, *Chem. Phys. Lett.*, **73**, 393, (1980).
- [66] P. Pulay, *J. Comp. Chem.*, **3**, 556, (1982).
- [67] F. W. Bobrowicz and W. A. Goddard III, *Methods of Electronic Structure Theory*, edited by H. F. Schaefer (Plenum, New York, 1977), vol. 3, pp. 79–127.
- [68] P. S. Epstein, *Phys. Rev.*, **28**, 695, (1926).
- [69] R. K. Nesbet, *Proc. Roy. Soc., Ser. A*, **230**, 312, (1995).
- [70] V. A. Rassolov, F. Xu, and S. Garashchuk, *J. Chem. Phys.*, **120**, 10385, (2004).
- [71] T. Kato, *Commun. Pure Appl. Math.*, **10**, 151, (1957).
- [72] R. T. Pack and W. Byers Brown, *J. Chem. Phys.*, **45**, 556, (1966).
- [73] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **104**, 9908, (1996).
- [74] D. M. Chipman, *Theor. Chim. Acta*, **76**, 73, (1989).
- [75] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **105**, 1470, (1996).
- [76] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **105**, 1479, (1996).
- [77] J. O. Hirschfelder, *J. Chem. Phys.*, **33**, 1462, (1960).
- [78] B. Wang, J. Baker, and P. Pulay, *Phys. Chem. Chem. Phys.*, **2**, 2131, (2000).



## Chapter 6

# Open-Shell and 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 two-electron excitations, although even here reasonable results can sometimes be obtained. Moreover, some of the excited state methods can be extended to treat open-shell wavefunctions, *e.g.* diradicals, ionized and electron attachment states and more.

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 three 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. The spin-flip variant of CIS extends it to diradicals.
- 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. TDDFT can be extended to treat di- and tri-radicals and bond-breaking by adopting the spin-flip approach (see section 6.3.1 for details).
- Wavefunction-based electron correlation treatments (Sections 6.4 and 6.5). 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 significantly more computationally expensive. These methods can also be extended to describe certain multi-configurational wavefunctions, for example, problematic doublet radicals, diradicals, triradicals, and more.

In general, a basis set appropriate for a ground state density functional theory or a Hartree–Fock calculation will be appropriate for describing valence excited states. However, many excited states involve significant contributions from 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. To describe true Rydberg excited states, Q-CHEM allows the user to add two or more sets of diffuse functions (see Chapter 7). For example the 6-311(2+)G\* basis includes two sets of diffuse functions on heavy atoms and is generally adequate for description of both valence and Rydberg excited states.

Q-CHEM supports four 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, CIS(D), EOM-CCSD and EOM-OD, each of which will be discussed in turn.

- **Visualization**

It is possible to visualize the excited states either by attachment–detachment density analysis (available for CIS only) or by plotting the transition density (see *plots* descriptions in Chapters 3 and 10). Transition densities can be calculated for CIS and EOM-CCSD methods. The theoretical basis of the attachment–detachment density analysis is discussed in Section 6.4 of this Chapter. In addition Dyson orbitals can be calculated and plotted for the ionization from the ground and electronically excited states for the CCSD and EOM-CCSD wavefunctions.

- **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. Analytic first derivatives are available for UCIS, RCIS and EOM-CCSD, EOM-OD excited state optimizations may also be performed using finite difference methods, however, these can be very time-consuming to compute.

- **Optimization of the crossings between potential energy surfaces**

Seams between potential energy surfaces can be located and optimized by using analytic gradients within CCSD and EOM-CCSD formalisms.

- **Properties**

Properties such as transition dipoles, dipole moments, spatial extent of electron densities and  $\langle S^2 \rangle$  values can be computed for EOM-CCSD and EOM-OD wavefunctions.

- **Excited state vibrational analysis**

Given an optimized 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  $n^{\text{th}}$ -order saddle points. Analytic excited state vibrational analyses can only be performed using the UCIS and RCIS methods, for which efficient analytical second derivatives are available. EOM-CCSD frequencies are also available using analytic first derivatives and second derivatives obtained from finite difference methods. EOM-OD frequencies are only available through finite difference calculations.

## 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 [3,4] energy and wave function begins by selecting the HF single-determinant wave function as reference for the ground state of the system

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

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

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

where  $\mathbf{S}$  is the overlap matrix,  $\mathbf{C}$  is the matrix of molecular orbital coefficients,  $\varepsilon$  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 anti-symmetrized two-electron integrals

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

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

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

where  $P^{\text{HF}}$  is the HF density matrix and  $V_{\text{nuc}}$  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

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

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

where we have introduced the anti-symmetrized 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:

- do not yield pure spin states
- resemble more closely ionization rather than excitation
- 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

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

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

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

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) [3,4], unrestricted (UCIS), or restricted open shell (ROCIS) [5] spin orbitals.

## 6.2.2 Random Phase Approximation (RPA)

The Random Phase Approximation (RPA) [6,7] 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) [8] 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 [9], 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  $\alpha$  spin electron from the singly occupied orbital and the promotion of a  $\beta$  spin electron into the singly occupied orbital. In particular, the spin adapted configurations

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

(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

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

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  $\alpha$  and  $\beta$  space

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

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,  $\beta$  unoccupied orbitals, through which the spin–flip double excitation proceeds, may not match the half–occupied  $\alpha$  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

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

$n$   $n > 0$  Looks for  $n$  CIS excited states

RECOMMENDATION:

None

#### **CIS\_SINGLET**

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

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for singlet states

FALSE Do not solve for singlet states.

RECOMMENDATION:

None

#### **CIS\_TRIPLET**

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

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for triplet states

FALSE Do not solve for triplet states.

RECOMMENDATION:

None

#### **RPA**

Do an RPA calculation in addition to a CIS calculation

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not do an RPA calculation

True Do an RPA calculation.

RECOMMENDATION:

None

**XCIS**

Do an XCIS calculation in addition to a CIS calculation

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not do an XCIS calculation

True Do an XCIS calculation (requires ROHF ground state).

RECOMMENDATION:

None

**6.2.5 Customization****N\_FROZEN\_CORE**

Controls the number of frozen core orbitals

TYPE:

INTEGER

DEFAULT:

0 No frozen core orbitals

OPTIONS:

FC Frozen core approximation

*n* Freeze *n* core orbitals

RECOMMENDATION:

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.

TYPE:

INTEGER

DEFAULT:

0 No frozen virtual orbitals

OPTIONS:

*n* Freeze *n* virtual orbitals

RECOMMENDATION:

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

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

*n* User-defined number of cycles

RECOMMENDATION:

Default is usually sufficient.

**CIS\_CONVERGENCE**

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

TYPE:

INTEGER

DEFAULT:

6 CIS convergence threshold  $10^{-6}$

OPTIONS:

$n$  Corresponding to  $10^{-n}$

RECOMMENDATION:

None

**CIS\_RELAXED\_DENSITY**

Use the relaxed CIS density for attachment/detachment density analysis

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the relaxed CIS density in analysis

True Use the relaxed CIS density in analysis.

RECOMMENDATION:

None

**CIS\_GUESS\_DISK**

Read the CIS guess from disk (previous calculation)

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Create a new guess

True Read the guess from disk

RECOMMENDATION:

Requires a guess from previous calculation.

**CIS\_GUESS\_DISK\_TYPE**

Determines the type of guesses to be read from disk

TYPE:

INTEGER

DEFAULT:

Nil

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATION:

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 [10–12]. 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 [13, 14], to yield analytical gradients, excited state vibrational frequencies, force constants, polarizabilities, and infrared intensities. Their evaluation is controlled by two *\$rem* variables, 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

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP Single point energy

FORCE Analytical Force calculation

OPT Geometry Minimization

TS Transition Structure Search

FREQ Frequency Calculation

RECOMMENDATION:

None

#### **CIS\_STATE\_DERIV**

Sets CIS state for excited state optimizations and vibrational analysis

TYPE:

INTEGER

DEFAULT:

0 Does not select any of the excited states

OPTIONS:

*n* Select the *n*<sup>th</sup> state.

RECOMMENDATION:

Check to see that the states do no change order during an optimization

The semi-direct method [8] 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

$$\text{“C” memory} = \text{MEM\_TOTAL} - \text{MEM\_STATIC}$$

“C” memory must be large enough to contain a number of arrays whose size is  $3 \times N_{\text{Atoms}} \times N_{\text{Basis}}^2$  ( $N_{\text{Atoms}}$  is the number of atoms and  $N_{\text{Basis}}$  refers to the number of basis functions). The value of the *\$rem* variable MEM\_STATIC 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 80 Mb is sufficient, and by default MEM\_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

**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.

```

$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 6.2** 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
  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

  rh1 = 1.08574
  rh2 = 1.08534
  rc2 = 2.67299
  rc3 = 1.35450
  rh4 = 1.08722
  rc5 = 1.37290
  tc3 = 62.85
  ah2 = 122.16
  ah4 = 119.52
  ac5 = 116.45
$end

$rem
  basis = 6-31+G*
  exchange = hf
  mem_static = 80
  intsbuffersize = 15000000
  scf_convergence = 8
  cis_n_roots = 5
  xcis = true
$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 non-planar structure with zero forces, and all frequencies real.

```

$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
```

## 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 [15, 16], 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 [17], together with the CIS-like Tamm-Dancoff approximation [18]. 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 0.1-0.3 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 [19, 20]. 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. [21]. 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 [17].

Standard TDDFT also does not yield a good description of static correlation effects (see section 5.8), because it is based on a single reference configuration of Kohn-Sham orbitals. Recently, a new variation of TDDFT called spin-flip density functional theory (SFDF) was developed by Yihan Shao, Martin Head-Gordon and Anna Krylov to address this issue [26]. SFDF is different from standard TDDFT in two ways:

- The reference is a high-spin triplet (quartet) for a system with an even (odd) number of electrons;
- One electron is spin-flipped from an alpha Kohn-Sham orbital to a beta orbital during the excitation.

SFDF can describe the ground state as well as a few low-lying excited states, and has been applied to bond-breaking processes, and di- and tri-radicals with degenerate or near-degenerate frontier orbitals. See also section 6.5.8 for details on wavefunction-based spin-flip models.

### 6.3.2 TDDFT within a Reduced Single Excitation Space

Much of chemistry and biology occurs in solution or on surfaces. The molecular environment can have a large effect on electronic structure and may change chemical behavior. Q-CHEM is able to compute excited states within a local region of a system through performing the TDDFT (or CIS) calculation with a reduced single excitation subspace [22]. This allows the excited states of a solute molecule to be studied with a large number of solvent molecules reducing the rapid rise in computational cost. The success of this approach relies on there being only weak mixing between the electronic excitations of interest and those omitted from the single excitation space. For systems in which there are strong hydrogen bonds between solute and solvent, it is advisable to include excitations associated with the neighboring solvent molecule(s) within the reduced excitation space.

The reduced single excitation space is constructed from excitations between a subset of occupied and virtual orbitals. These can be selected from an analysis based on Mulliken populations and molecular orbital coefficients. For this approach the atoms that constitute the solvent needs to be defined. Alternatively, the orbitals can be defined directly. The atoms or orbitals are specified within a *\$solute* block. These approach is implemented within the TDA and has been used to study the excited states of formamide in solution [23], CO on the Pt(111) surface [24] and the tryptophan chromophore within proteins [25].

### 6.3.3 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 several 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 RPA = FALSE, which leads to a calculation employing the Tamm-Dancoff approximation (TDA), which is usually a good approximation to full TDDFT.

- If `CC_SPIN_FLIP` is set to `TRUE` when performing a TDDFT calculation, a SFDFFT calculation will also be performed. At present, SFDFFT is only implemented within TDDFT/TDA so RPA must be set to `FALSE`. Remember to set the spin multiplicity to 3 for systems with an even-number of electrons (e.g., diradicals), and 4 for odd-number electron systems (e.g., triradicals).

TDDFT and TDDFT/TDA are both available only for excitation energies at present.

**TRNSS**

Controls whether reduced single excitation space is used

TYPE:

LOGICAL

DEFAULT:

FALSE Use full excitation space

OPTIONS:

TRUE Use reduced excitation space

RECOMMENDATION:

None

**TRTYPE**

Controls how reduced subspace is specified

TYPE:

INTEGER

DEFAULT:

1 Select orbitals localized on a set of atoms

OPTIONS:

2 Specify a set of orbitals

RECOMMENDATION:

None

**N\_SOL**

Specifies number of atoms or orbitals in *\$solute*

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

**CISTR\_PRINT**

Controls level of output

TYPE:

LOGICAL

DEFAULT:

FALSE Minimal output

OPTIONS:

TRUE Increase output level

RECOMMENDATION:

None

**CUTOCC**

Specifies occupied orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTOCC/100

DEFAULT:

50

OPTIONS:

0-200

RECOMMENDATION:

None

**CUTVIR**

Specifies virtual orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTVIR/100

DEFAULT:

0 No truncation

OPTIONS:

0-100

RECOMMENDATION:

None

### 6.3.4 Example

**Example 6.4** This example shows two jobs which request variants of time-dependent density functional theory calculations. The first job, using the default value of `RPA = FALSE`, performs TDDFT in the Tamm-Dancoff approximation (TDA). The second job, with `RPA = TRUE` performs a both TDA and full TDDFT calculations.

```
$comment
  methyl peroxy radical
  TDDFT/TDA and full TDDFT with 6-31+G*
$end

$molecule
  0 2
  C 1.00412 -0.18045 0.00000
  O -0.24600 0.59615 0.00000
  O -1.31237 -0.23026 0.00000
  H 1.81077 0.56720 0.00000
  H 1.03665 -0.80545 -0.90480
  H 1.03665 -0.80545 0.90480
$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 6.5** This example shows a calculation of the excited states of a formamide–water complex within a reduced excitation space of the orbitals located on formamide

```

$comment
  formamide--water
  TDDFT/TDA in reduced excitation space
$end

$molecule
  0 1
  H  1.13  0.49 -0.75
  C  0.31  0.50 -0.03
  N -0.28 -0.71  0.08
  H -1.09 -0.75  0.67
  H  0.23 -1.62 -0.22
  O -0.21  1.51  0.47
  O -2.69  1.94 -0.59
  H -2.59  2.08 -1.53
  H -1.83  1.63 -0.30
$end

$rem
  EXCHANGE          b3lyp
  CIS_N_ROOTS       10
  BASIS              6-31++G**
  TRNSS             TRUE
  TRTYPE            1
  CUTOCC            60
  CUTVIR            40
  CISTR_PRINT       TRUE
$end

$solute
  1
  2
  3
  4
  5
  6
$end

```

**Example 6.6** SF-TDDFT SP calculation of the 6 lowest states of the TMM diradical using recommended 50-50 functional

```

$molecule
  0 3
  C
  C 1 CC1
  C 1 CC2  2 A2
  C 1 CC2  2 A2  3 180.0

```



```

H 2 C2H 1 C2CH 3 0.0
H 2 C2H 1 C2CH 4 0.0
H 3 C3Hu 1 C3CHu 2 0.0
H 3 C3Hd 1 C3CHd 4 0.0
H 4 C3Hu 1 C3CHu 2 0.0
H 4 C3Hd 1 C3CHd 3 0.0

CC1 = 1.35
CC2 = 1.47
C2H = 1.083
C3Hu = 1.08
C3Hd = 1.08
C2CH = 121.2
C3CHu = 120.3
C3CHd = 121.3
A2 = 121.0
$end

$rem
jobtype SP
EXCHANGE GENERAL Exact exchange
BASIS 6-31G*
SCF_GUESS CORE
SCF_CONVERGENCE 10
MAX_SCF_CYCLES 100
CC_SPIN_FLIP 1
CIS_N_ROOTS 6
CC_DCONVERGENCE 10
CIS_CONVERGENCE 10
MAX_CIS_CYCLES = 100
$end

$xc_functional
X HF 0.5
X S 0.08
X B 0.42
C VWN 0.19
C LYP 0.81
$end

```

## 6.4 Correlated Excited State Methods, CIS(D)

CIS(D) [27, 28] 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) and 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

$$|\Psi\rangle = \exp(T_1 + T_2)|\Psi_0\rangle \quad (6.15)$$

Determining the singles and doubles amplitudes requires solving the two equations

$$\langle \Psi_i^a | H - E \left| \left( 1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3 \right) \Psi_0 \right\rangle = 0 \quad (6.16)$$

$$\langle \Psi_{ij}^{ab} | H - E \left| \left( 1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3 + \frac{1}{2}T_2^2 + \frac{1}{2}T_1^2T_2 + \frac{1}{4!}T_1^4 \right) \Psi_0 \right\rangle = 0 \quad (6.17)$$

which lead to the CCSD excited state equations, which can be written

$$\langle \Psi_i^a | H - E \left| \left( U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \frac{1}{2}T_1^2U_1 \right) \Psi_0 \right\rangle = \omega b_i^a \quad (6.18)$$

$$\langle \Psi_i^a | H - E \left| \left( U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \frac{1}{2}T_1^2U_1 + T_2U_2 + \frac{1}{2}T_1^2U_2 + T_1T_2U_1 + \frac{1}{3!}T_1^3U_1 \right) \Psi_0 \right\rangle = \omega b_{ij}^{ab} \quad (6.19)$$

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

$$\omega^{(2)} = b^{(0)\dagger} \mathbf{A}^{(1)} b^{(1)} + b^{(0)\dagger} \mathbf{A}^{(2)} b^{(0)} \quad (6.20)$$

or in the alternative form

$$\begin{aligned} \omega^{(2)} &= \omega^{\text{CIS(D)}} \\ &= E^{\text{CIS(D)}} - E^{\text{MP2}} \end{aligned} \quad (6.21)$$

where

$$E^{\text{CIS(D)}} = \langle \Psi^{\text{CIS}} | V | U_2 \Psi^{\text{HF}} \rangle + \langle \Psi^{\text{CIS}} | V | T_2 U_1 \Psi^{\text{HF}} \rangle \quad (6.22)$$

and

$$E^{\text{MP2}} = \langle \Psi^{\text{HF}} | V | T_2 \Psi^{\text{HF}} \rangle \quad (6.23)$$

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” [29]. 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.

### 6.4.1 CIS(D) Job Control

The CIS(D) algorithm which is currently used in Q-CHEM has much in common with Q-CHEM's coupled-cluster package and share many of the *\$rem* options. As with all post-HF calculations, it is important to ensure there are sufficient resources available for the necessary integral calculations and transformations. For CIS(D), these resources are controlled using the *\$rem* variables CC\_TMPBUFSIZE, CC\_BLK\_TNSR\_BUFSIZE, MEM\_STATIC and MEM\_TOTAL. To request a CIS(D) calculation the CORRELATION *\$rem* should be set to CIS(D) and the number of excited states to calculate should be specified by CC\_NLOWSPIN and CC\_NHIGHSPIN variables.

#### CC\_NHIGHSPIN

Sets the number of high-spin excited state roots to find. Works only for singlet reference states and triplet excited states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

[*i, j, k . . .*] Find *i* excited states in the first irrep, *j* states in the second irrep *etc.*

RECOMMENDATION:

None

#### CC\_NLOWSPIN

Sets the number of low-spin excited state roots to find. In the case of closed-shell reference states, excited singlet states will be found. For any other reference state, all states (*e.g.* both singlet and triplet) will be calculated.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

[*i, j, k . . .*] Find *i* excited states in the first irrep, *j* states in the second irrep *etc.*

RECOMMENDATION:

None

**Note:** It is a symmetry of a *transition* rather than that of a *target state* which is specified in excited state calculations. The symmetry of the target state is a product of the symmetry of the reference state and the transition. For closed-shell molecules, the former is fully symmetric and the symmetry of the target state is the same as that of transition, however, for open-shell references this is not so.

#### CC\_STATE\_DERIV

Selects which EOM or CIS(D) state is to be considered for optimization or property calculations.

TYPE:

INTEGER

DEFAULT:

-1 Turns off optimization/property calculations

OPTIONS:

*n* Optimize the *n*<sup>th</sup> excited state.

RECOMMENDATION:

None

**Note:** Should be smaller or equal to the number of excited states calculated in the corresponding irrep. The symmetry is turned off for finite difference calculations

**CC\_TMPBUFFSIZE**

Maximum size, in Mb, of additional buffers for temporary arrays used to work with individual blocks or matrices.

TYPE:

INTEGER

DEFAULT:

3% of MEM\_TOTAL

OPTIONS:

*n* Integer number of Mb

RECOMMENDATION:

Should not be smaller than the size of the largest possible block.

**CC\_BLK\_TNSR\_BUFFSIZE**

Specifies the maximum size, in Mb, of the buffers for in-core storage of block-tensors.

TYPE:

INTEGER

DEFAULT:

50% of MEM\_TOTAL

OPTIONS:

*n* Integer number of Mb

RECOMMENDATION:

Larger values can give better I/O performance and are recommended for systems with large memory (add to your *.qchemrc* file)

**MEM\_STATIC**

Sets the memory for individual fortran program modules

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb

OPTIONS:

*n* User-defined number of megabytes.

RECOMMENDATION:

For direct and semi-direct MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (32-160 Mb), as discussed above.

**MEM\_TOTAL**

Sets the total memory available to Q-CHEM

TYPE:

INTEGER

DEFAULT:

2000 2 Gb

OPTIONS:

*n* User-defined number of megabytes

RECOMMENDATION:

Use default, or set to about 80% of the physical memory of your machine.

## 6.4.2 Examples

**Example 6.7** Q-CHEM CIS(D) SP calculations for ozone at the experimental ground state geometry  $C_{2v}$

```

$molecule
0 1
0
0 1 RE
0 2 RE 1 A

RE=1.272
A=116.8
$end

$rem
jobtype          SP
BASIS            6-31G*
N_FROZEN_CORE    3          use frozen core
correlation      CIS(D)
CC_NLOWSPIN      [2,2,2,2] find 2 lowest singlets in each irrep.
CC_NHIGHSPIN     [2,2,2,2] find two lowest triplets in each irrep.
$end

```

**Example 6.8** Q-CHEM CIS(D) geometry optimization for the lowest triplet state of water. The symmetry is automatically turned off for finite difference calculations

```

$molecule
0 1
o
h 1 r
h 1 r 2 a

r 0.95
a 104.0
$end

$rem
jobtype          opt
basis            3-21g
correlation      cis(d)
scf_guess        read
cc_nhighspin     1  calculate one lowest triplet
cc_state_deriv   1  optimize the geometry of the lowest state.
$end

```

## 6.4.3 Resolution of the Identity CIS(D) Methods

Because of algorithmic similarity with MP2 calculation, the “resolution of the identity” approximation can also be used in CIS(D). In fact, RI-CIS(D) is orders of magnitudes more efficient than previously explained CIS(D) algorithms for effectively all molecules with more than a few atoms. Like in MP2, this is achieved by reducing the prefactor of the computational load, while keeping the overall cost scaling still with the fifth power of the system size. Basically, RI-CIS(D) calculations are enabled by setting the CORRELATION *\$rem* to RICIS(D) with the CIS calculation explained in 6.2.1. MEM\_STATIC, AUX\_BASIS, and PURECART need to be set by following the same guide as in RI-MP2 (Sec. 5.5). In addition, a few other *\$rem* variables have to be set for efficiency.

**MEM\_STATIC**

Sets the memory for individual program modules

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

For RI-CIS(D) calculations,  $150(N^2 + N)$  of MEM\_STATIC is required. Because a number of matrices with  $N^2$  size also need to be stored, 32–160 Mb of additional MEM\_STATIC is needed.

**MEM\_TOTAL**

Sets the total memory available to Q-CHEM

TYPE:

INTEGER

DEFAULT:

2000 2 Gb

OPTIONS:

$n$  User-defined number of megabytes

RECOMMENDATION:

The minimum memory requirement of RI-CIS(D) is approximately  $\text{MEM\_STATIC} + \max(3SVX, 3X^2)$  ( $S$ : number of excited states,  $X$ : number of auxiliary basis functions). However, because RI-CIS(D) uses a batching scheme for efficient evaluations of electron repulsion integrals, more memory will significantly speed up the calculation. Put as much memory as possible if you are not sure about what to use (but never put any more than available). The actual memory size used in the calculation will be printed out in the output file.

**AO2MO\_DISK**

Sets the scratch space size for individual program modules

TYPE:

INTEGER

DEFAULT:

2000 2 Gb

OPTIONS:

$n$  User-defined number of megabytes.

RECOMMENDATION:

The minimum disk requirement of RI-CIS(D) is approximately  $3SOVX$ . Again, the batching scheme will become more efficient with more available disk space. Put as much disk space as possible if you are not sure about what to use (but never put any more than available). The size of the actually used disk space will also be printed out in the output file.

Presently in Q-CHEM, RI approximation is supported for closed-shell restricted CIS(D) and open-shell unrestricted UCIS(D) energy calculations.

**6.4.4 Example**

**Example 6.9** Q-CHEM input for an RI-CIS(D) calculation.

\$molecule

```
0 1
C      0.667472    0.000000    0.000000
C     -0.667472    0.000000    0.000000
H      1.237553    0.922911    0.000000
H      1.237553   -0.922911    0.000000
H     -1.237553    0.922911    0.000000
H     -1.237553   -0.922911    0.000000
$end

$rem
  jobtype          sp
  exchange         hf
  basis            6-311(2+,2+)G**
  mem_total       1000
  mem_static      100
  ao2mo_disk      1000
  aux_basis       rimp2-aug-cc-pVTZ
  purecart        1111
  correlation     ricis(d)
  cis_n_roots     10
  cis_singlets    true
  cis_triplets    false
$end
```

## 6.5 Coupled-Cluster Excited-State and Open-Shell Methods

### 6.5.1 Excited states by EOM-EE-CCSD and EOM-EE-OD

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 [30] or equations of motion methods [31]. 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 the 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. The description of these states can be improved by including triple excitations, as in the EOM(2,3) method.

Q-CHEM includes coupled-cluster methods for excited states based on the optimized orbital coupled-cluster doubles (OD), and the coupled cluster singles and doubles (CCSD) methods, described earlier. OD excitation energies have been shown to be essentially identical in numerical performance to CCSD excited states [32].

These methods, while far more computationally expensive than TDDFT, are nevertheless useful as proven high accuracy methods for the study of excited states of small molecules. Moreover, they are capable of describing both valence and Rydberg excited states. 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 includes extensions of EOM methods to treat ionized or electron attachment systems, as well as di- and tri-radicals.

### 6.5.2 EOM–XX–CCSD suit of methods

Q-CHEM features the most complete set of EOM-CCSD models that enables accurate, robust, and efficient calculations of electronically excited states (EOM–EE–CCSD) [38–40]; ground and excited states of diradicals and triradicals (EOM–SF–CCSD [41, 42], available only in Q-CHEM); ionization potentials and electron attachment energies as well as problematic doublet radicals, cation or anion radicals, (EOM–IP/EA–CCSD) [43–45]. Conceptually, EOM is very similar to configuration interaction (CI): target EOM states are found by diagonalizing the similarity transformed Hamiltonian  $\bar{H} = e^{-T} H e^T$ :

$$\bar{H}R = ER, \quad (6.24)$$

where  $T$  and  $R$  are general excitation operators with respect to the reference determinant  $|\Phi_0\rangle$ . In the EOM–CCSD models,  $T$  and  $R$  are truncated at single and double excitations, and the amplitudes  $T$  satisfy the CC equations for the reference state  $|\Phi_0\rangle$ :

$$\begin{aligned} \langle \Phi_i^a | \bar{H} | \Phi_0 \rangle \\ \langle \Phi_{ij}^{ab} | \bar{H} | \Phi_0 \rangle \end{aligned} \quad (6.25)$$

The computational scaling of EOM–CCSD and CISD methods is identical, *i.e.*,  $\mathcal{O}(N^6)$ , however EOM–CCSD is numerically superior to CISD because correlation effects are “folded in” in the transformed Hamiltonian, and because EOM–CCSD is rigorously size-extensive.

By combining different types of excitation operators and references  $|\Phi_0\rangle$ , different groups of target states can be accessed as explained in Fig. 6.1. For example, electronically excited states can be described when the reference  $|\Phi_0\rangle$  corresponds to the ground state wave function, and operators  $R$  conserve the number of electrons and a total spin [40]. In the ionized/electron attached EOM models [43, 44], operators  $R$  are not electron conserving (*i.e.*, include different number of creation and annihilation operators) — these models can accurately treat ground and excited states of doublet radicals and some other open-shell systems. For example, singly ionized EOM methods, *i.e.*, EOM–IP–CCSD and EOM–EA–CCSD, have proven very useful for doublet radicals whose theoretical treatment is often plagued by symmetry breaking. Finally, the EOM–SF method [41, 42] in which the excitation operators include spin-flip allows one to access diradicals, triradicals, and bond-breaking.

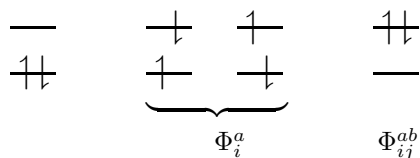
Q-CHEM features EOM–EE/SF/IP/EA–CCSD methods for both closed and open-shell references (RHF/UHF/ROHF), including frozen core/virtual options. All EOM models take the full advantage of molecular point group symmetry. Analytic gradients are available for RHF and UHF references, for the full orbital space, and with frozen core/virtual orbitals [46]. Properties calculations (permanent and transition dipole moments,  $\langle S^2 \rangle$ ,  $\langle R^2 \rangle$ , *etc.*) are also available. The current implementation of the EOM–XX–CCSD methods enables calculations of medium-size molecules, *e.g.*, up to 10–14 heavy atoms.

### 6.5.3 Spin–Flip Methods for Di- and Triradicals

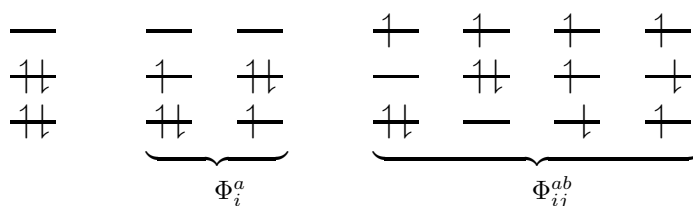
The Spin–Flip method [33, 34, 48] was developed as a method to address the bond-breaking problem associated with a single-determinant description of the wavefunction. Both closed and open shell



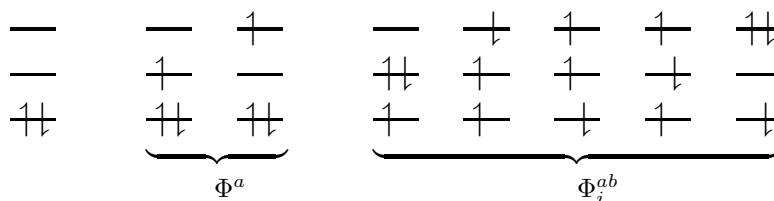
$$\text{EOM-EE} \quad \Psi(M_s = 0) = R(M_s = 0)\Psi_0(M_s = 0)$$



$$\text{EOM-IP} \quad \Psi(N) = R(-1)\Psi_0(N + 1)$$



$$\text{EOM-EA} \quad \Psi(N) = R(+1)\Psi_0(N - 1)$$



$$\text{EOM-SF} \quad \Psi(M_s = 0) = R(M_s = -1)\Psi_0(M_s = 1)$$

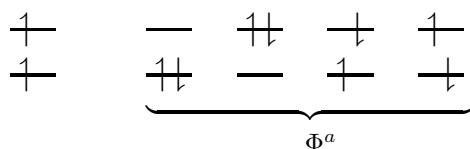


Figure 6.1: In the EOM formalism, target states  $\Psi$  are described as excitations from a reference state  $\Psi_0$ :  $\Psi = R\Psi_0$ , where  $R$  is a general excitation operator. Different EOM models are defined by choosing the reference and the form of the operator  $R$ . In the EOM models for electronically excited states (EOM-EE, upper panel), the reference is the closed-shell ground state Hartree-Fock determinant, and the operator  $R$  conserves the number of  $\alpha$  and  $\beta$  electrons. Note that two-configurational open-shell singlets can be correctly described by EOM-EE since both leading determinants appear as single electron excitations. The second and third panels present the EOM-IP/EA models. The reference states for EOM-IP/EA are determinants for  $N + 1/N - 1$  electron states, and the excitation operator  $R$  is ionizing or electron-attaching, respectively. Note that both the EOM-IP and EOM-EA sets of determinants are spin-complete and balanced with respect to the target multi-configurational ground and excited states of doublet radicals. Finally, the EOM-SF method (the lowest panel) employs the high-spin triplet state as a reference, and the operator  $R$  includes spin-flip, *i.e.*, does not conserve the number of  $\alpha$  and  $\beta$  electrons. All the determinants present in the target low-spin states appear as single excitations, which ensures their balanced treatment both in the limit of large and small HOMO-LUMO gaps.

singlet states are described within a single reference as spin-flipping, (*e.g.*,  $\alpha \rightarrow \beta$  excitations from the triplet reference state, for which both dynamical and non-dynamical correlation effects are smaller than for the corresponding singlet state. This is because the exchange hole, which arises from the Pauli exclusion between same-spin electrons, partially compensates for the poor

description of the coulomb hole by the mean-field Hartree-Fock model. Furthermore, because two  $\alpha$  electrons cannot form a bond, no bond breaking occurs as the internuclear distance is stretched, and the triplet wavefunction remains essentially single-reference in character. The spin-flip model has also proved useful in the description of di- and tri-radicals.

The spin-flip method is available for the CIS, CIS(D), CISD, CISDT, OD, CCSD, and EOM-(2,3) levels of theory. For the OD and CCSD models, the following non-relaxed properties are also available: dipoles, transition dipoles, eigenvalues of the spin-squared operator ( $\langle S^2 \rangle$ ), and densities. Analytic gradients are also for SF-CIS and EOM-SF-CCSD methods. To invoke a spin-flip calculation the `CC_SPIN_FLIP $rem` should be set to `TRUE`, along with the associated `$rem` settings for the chosen level of correlation. Note that the high multiplicity triplet or quartet reference states should be used.

#### 6.5.4 EOM-DIP-CCSD

Double-ionization potential (DIP) is another non-electron-conserving variant of the EOM approach [37]. In DIP, target states are found by detaching two electrons from the reference state:

$$\Psi_k(N) = R_{N-2}(k)\Psi_0(N+2), \quad (6.26)$$

and the excitation operator  $R(k)$  has the following form:

$$R(k) = R_1(k) + R_2(k), \quad (6.27)$$

$$R_1(k) = 1/2 \sum_{ij} r_{ij} j i, \quad R_2(k) = 1/6 \sum_{ijka} r_{ijk}^a a^\dagger k j i. \quad (6.28)$$

As a reference state in the EOM-DIP calculations one usually takes the well-described closed-shell state. Therefore, EOM-DIP is a useful tool for describing molecules with electronic degeneracies of the type “ $2n - 2$  electrons on  $n$  degenerate orbitals”. The simplest examples of such systems are diradicals with two-electrons-on-two-orbitals pattern.

Accuracy of the EOM-DIP-CCSD method in description of the corresponding systems is similar to accuracy of other EOM-CCSD models, *i.e.*, 0.1–0.3 eV. The scaling of EOM-DIP-CCSD is  $\mathcal{O}(N^6)$ , analogous to that of other EOM-CCSD methods. However, its computational cost is less compared to, *e.g.*, EOM-EE-CCSD, and it increases more slowly with increasing basis set size.

#### 6.5.5 Equation-of-Motion Coupled-Cluster Job Control

There is a rich range of input control options for coupled-cluster excited state or other EOM calculations. The minimal requirement is the input for the reference state CCSD or OD calculation (see Chapter 5), plus specification of the number of *target states* requested through `CC_NLOWSPIN` and `CC_NHIGHSPIN`. Users must be aware of the point group symmetry of the system being studied and also the symmetry of the initial and target states of interest, as well as symmetry of transition. It is possible to turn off the use of symmetry using the `CC_SYMMETRY`. If set to `FALSE` the molecule will be treated as having  $C_1$  symmetry and all states will be of  $A$  symmetry. The full range of input options that are directly relevant to EOM-CC module follows:

**CC\_NHIGHSPIN**

Sets the number of high-spin excited state roots to find. Works only for singlet RHF reference states and triplet excited states, is ignored for UHF or ROHF references

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any coupled-cluster excited states

OPTIONS:

[*i*, *j*, *k* . . .] Find *i* CC excited states in the irrep symmetry, *j* excited states in the second irrep, *etc.*

RECOMMENDATION:

None

**CC\_NLOWSPIN**

Sets the number of coupled-cluster excited state roots to find with the same multiplicity as the ground state in the case of a closed-shell singlet RHF reference. For a spin-unrestricted ground state, (*e.g.*, doublet radicals or high-spin triplet or quartets), this is the total number of states of all multiplicities. It also specifies the target states in EOM-IP/EA/SF

TYPE:

INTEGER

DEFAULT:

0 Do not look for any EOM target states (*e.g.* excited states)

OPTIONS:

[*i*, *j*, *k* . . .] Find *i* CC excited states in the first irrep, *j* excited states in the second irrep, *etc.*

RECOMMENDATION:

None

**Note:** It is a symmetry of a *transition* rather than that of a *target state* which is specified in excited state calculations. The symmetry of the target state is a product of the symmetry of the reference state and the transition. For closed-shell molecules, the former is fully symmetric and the symmetry of the target state is the same as that of transition, however, for open-shell references this is not so.

**Note:** For the CC\_NHIGHSPIN and CC\_NLOWSPIN options, Q-CHEM will increase the number of roots if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations.

**CC\_DCONVERGENCE**

Convergence criterion for the RMS residuals of excited state vectors

TYPE:

INTEGER

DEFAULT:

5 Corresponding to  $10^{-5}$

OPTIONS:

*n* Corresponding to  $10^{-n}$  convergence criterion

RECOMMENDATION:

Use default. Should normally be set to the same value as CC\_DTHRESHOLD.

**CC\_DO\_DISCONNECTED**

Determines whether disconnected terms included in the EOM-OD equations

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

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.

TYPE:

INTEGER

DEFAULT:

00105 Corresponding to 0.00001

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

Use default unless converge problems are encountered. Should normally be set to the same values as CC\_DCONVERGENCE, if convergence problems arise try setting to a value less than CC\_DCONVERGENCE.

**CC\_DMAXITER**

Maximum number of iteration allowed for Davidson diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

*n* User-defined number of iterations

RECOMMENDATION:

Default is usually sufficient

**CC\_NGUESS\_DOUBLES**

Specifies number of excited state guess vectors which are double excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

*n* Include *n* guess vectors that are double excitations

RECOMMENDATION:

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.

TYPE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

$n$  Include  $n$  guess vectors that are single excitations

RECOMMENDATION:

Should be greater or equal than the number of excited states requested.

**CC\_DMAXVECTORS**

Specifies maximum number of vectors in the subspace for the Davidson diagonalization.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

$n$  Up to  $n$  vectors per root before the subspace is reset

RECOMMENDATION:

Larger values increase disk storage but accelerate and stabilize convergence.

**CC\_PRECONV\_DOUBLES**

When TRUE, doubly-excited vectors are converged prior to a full excited states calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

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.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

**CC\_SPIN\_FLIP**

Selects whether do perform a standard excited state calculation, or a spin-flip calculation. Spin multiplicity should be set to 3 for systems with an even number of electrons, and 4 for systems with an odd number of electrons.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

**CC\_IP**

If TRUE, calculates EOM-IP-CCSD excitation energies and properties (upon request).

TYPE:

LOGICAL

DEFAULT:

FALSE (no EOM-IP-CCSD states will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

**CC\_EA**

If TRUE, calculates EOM-EA-CCSD excitation energies and properties (upon request).

TYPE:

LOGICAL

DEFAULT:

FALSE (no EOM-EA-CCSD states will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

**Note:** When CC\_IP is set to TRUE, it can change the convergence of Hartree-Fock iterations compared to the same job with CC\_IP=FALSE, because a very diffuse basis function is added to a center of symmetry *before* the Hartree-Fock iterations start. For the same reason, BASIS2 keyword is incompatible with CC\_IP=TRUE. In order to read Hartree-Fock guess from a previous job, you must specify CC\_IP=TRUE (even if you do not request for any correlation or excited states) in that previous job. Currently, the second moments of electron density are incorrect for the EOM-IP-CCSD target states. CC\_NLOWSPIN should also be set to specify the number of target EOM-IP states.

**CC\_DIP**

Initializes a EOM-DIP-CCSD calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform an EOM-DIP-CCSD calculation

1 Do perform an EOM-DIP-CCSD calculation

RECOMMENDATION:

None

**CC\_SPIN\_FLIP\_MS**

This option is only used in EOM-SF using quintet references and including triple excitations. By default, SF flips the spin of one  $\alpha$  electron. One can ask to flip the spins of two  $\alpha$  electrons by specifying `CC_SPIN_FLIP_MS = 1`

TYPE:

INTEGER

DEFAULT:

0 For  $\alpha \rightarrow \alpha$  excitation

1 For  $\alpha \rightarrow \beta$  excitations

OPTIONS:

2 Allows double spin-flip calculations, *i.e.*, two electrons are flipped in the excitation operator:  $\alpha\alpha \rightarrow \beta\beta$ .

RECOMMENDATION:

This option can be useful when starting from quintet references - though this is not typical for EOM-SF.

**6.5.6 Examples**

**Example 6.10** EOM-EE-OD and EOM-EE-CCSD calculations of the singlet excited states of formaldehyde

```

$molecule
0 1
0
C,1,R1
H,2,R2,1,A
H,2,R2,1,A,3,180.

R1=1.4
R2=1.0
A=120.
$end

$rem
correlation od
basis 6-31+g
cc_nlowspin [2,2,2,2]
$end

@@@
$molecule
read
$end

$rem
```

```

correlation  ccscd
basis        6-31+g
cc_nlowspin  [2,2,2,2]
$end

```

**Example 6.11** EOM-SF-CCSD calculations for methylene from high-spin  $^3B_2$  reference

```

$molecule
0 3
C
H 1 rCH
H 1 rCH 2 aHCH

rCH   = 1.1167
aHCH  = 102.07
$end

$rem
jobtype          SP
CORRELATION      CCSD
BASIS            6-31G*
SCF_GUESS        CORE
CC_NGUESS_SINGLES 4
CC_SPIN_FLIP     1
CC_NLOWSPIN      [2,0,0,2]   Get two singlet A1 states and singlet and triplet B2 states
$end

```

**Example 6.12** EOM-IP-CCSD calculations for  $NO_3$  using closed-shell anion reference

```

$molecule
-1 1
N
O 1 r1
O 1 r2  2 A2
O 1 r2  2 A2  3 180.0

r1   = 1.237
r2   = 1.237
A2   = 120.00
$end

$rem
jobtype          SP           single point
LEVCOR           CCSD
BASIS            6-31G*
basis_lin_dep_thresh 14
CC_NLOWSPIN = [1,1,2,1]   ground and excited states of the radical
CC_IP = true
$end

```

### 6.5.7 Analytic gradients for the CCSD and EOM-XX-CCSD methods

Analytic gradients are available for the CCSD and all EOM-CCSD methods for both closed- and open-shell references (UHF and RHF only), including frozen core/virtuals functionality [46]. Practical computational studies of spectroscopic and dynamical properties usually focus on stationary points of PESs and thus require energy derivatives. For example, first derivatives allow one to characterize molecular equilibrium geometries, transition structures, and to calculate minimum energy paths and intrinsic reaction coordinates. In principle, derivatives of any order can be computed numerically from total energies by a finite difference procedure. Since such calculations



require only total energies, they can be performed for any electronic structure method. However, this universality of the numerical derivatives is their only advantage. The numerical evaluation of energy gradient for a system with  $N$  degrees of freedom requires at least  $2N$  single point energy calculations. Moreover, the finite difference procedure often encounters numerical problems, such as poor convergence, numerical noise, *etc.*. Calculation of analytic gradients is free of numerical instabilities, and can be performed approximately at a cost of a single point energy calculation, which results in significant time savings for polyatomic molecules.

Application limit: same as for the single-point CCSD or EOM-CCSD calculations.

Limitations: ROHF-CCSD and ROHF-EOM-CCSD gradients are not yet available.

### 6.5.8 Properties for CCSD and EOM-CCSD wavefunctions

For direct comparisons with experiments, calculation of molecular properties is extremely important. For the CCSD and EOM-CCSD wave functions, Q-CHEM currently can calculate permanent and transition dipole moments, oscillator strengths,  $\langle R^2 \rangle$  (as well as XX, YY and ZZ components separately, which is useful for assigning different Rydberg states, *e.g.*,  $3p_x$  vs.  $3s$ , *etc.*), and the  $\langle S^2 \rangle$  values. Interface of the CCSD and EOM-CCSD codes with the NBO 5.0 package is also available.

The coupled-cluster package in Q-CHEM can calculate properties of excited states including transition dipoles and geometry optimizations. The excited state of interest is selected using a combination of the *\$rem* variables CC.STATE\_DERIV and CC.REFSYM. The former of these determines the state, whilst the latter determines the symmetry of the state. Alternatively, both these can be specified using the single STATE\_TO\_OPT *\$rem*. Users must be aware of the point group symmetry of the system being studied and also the symmetry of the excited state of interest. It is possible to turn off the use of symmetry using the CC.SYMMETRY. If set to FALSE the molecule will be treated as having  $C_1$  symmetry and all states will be of  $A$  symmetry.

### 6.5.9 Equation-of-Motion Coupled-Cluster Optimization and Properties Job Control

#### CC.STATE\_DERIV

Together with CC.REFSYM, selects which EOM state is to be considered for optimization or property calculations. When transition properties are requested by CC.TRANS\_PROP, the transition properties will be calculated between this state and all other EOM states.

TYPE:

INTEGER

DEFAULT:

-1 Turns off optimization/property calculations

OPTIONS:

$n$  Optimize the  $n^{\text{th}}$  excited state.

RECOMMENDATION:

None

**Note:** Should be smaller or equal to the number of excited states calculated in the corresponding irrep.

**CC\_REFSYM**

Together with CC\_STATE\_DERIV, selects which EOM state is to be considered for optimization or property calculations. When transition properties are requested, the transition properties will be calculated between this state and all other EOM states.

TYPE:

INTEGER

DEFAULT:

-1 Turns off optimization/property calculations

OPTIONS:

$n$  Optimize the requested state in  $n^{\text{th}}$  irrep.

RECOMMENDATION:

None

**STATE\_TO\_OPT**

An alternative to the above two *\$rem* variables which allows the state and symmetry to be specified using a single variable

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i,j]$  optimize the  $j$ th state of the  $i$ th irrep.

RECOMMENDATION:

Do not use in conjunction with CC\_REFSYM and CC\_STATE\_DERIV

**CC\_PROP**

Whether or not the non-relaxed (expectation value) one-particle CCSD properties will be calculated. The properties currently include permanent dipole moment, the second moments  $\langle X^2 \rangle$ ,  $\langle Y^2 \rangle$ , and  $\langle Z^2 \rangle$  of electron density, and the total  $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$  (in atomic units). Incompatible with JOBTYP=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (lambda CCSD equations) for properties with the cost approximately the same as CCSD equations. Use default if you do not need properties. The cost of the properties calculation itself is low. The CCSD one-particle density can be analyzed with NBO package by specifying NBO=TRUE, CC\_PROP=TRUE and JOBTYP=FORCE.

**CC\_TWOPART\_PROP**

Request for calculation of non-relaxed two-particle CCSD properties. The two-particle properties currently include  $\langle S^2 \rangle$ . The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of  $\langle S^2 \rangle$ . The variable CC\_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are extremely computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

**CC\_EXSTATES\_PROP**

Whether or not the non-relaxed (expectation value) one-particle EOM-CCSD target state properties will be calculated. The properties currently include permanent dipole moment, the second moments  $\langle X^2 \rangle$ ,  $\langle Y^2 \rangle$ , and  $\langle Z^2 \rangle$  of electron density, and the total  $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$  (in atomic units). Incompatible with JOBTYP=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (EOM-CCSD equations for the left eigenvectors) need to be solved for properties, approximately doubling the cost of calculation for each irrep. Sometimes the equations for left and right eigenvectors converge to different sets of target states. In this case, the simultaneous iterations of left and right vectors will diverge, and the properties for several or all the target states may be incorrect! The problem can be solved by varying the number of requested states, specified with CC\_NLOWSPIN and CC\_NHIGHSPIN, or the number of guess vectors (CC\_NGUESS\_SINGLES). The cost of the one-particle properties calculation itself is low. The one-particle density of an EOM-CCSD target state can be analyzed with NBO package by specifying the state with CC\_REFSYM and CC\_STATE\_DERIV and requesting NBO=TRUE and CC\_EXSTATES\_PROP=TRUE.

**CC\_TRANS\_PROP**

Whether or not the transition dipole moment (in atomic units) and oscillator strength for the EOM-CCSD target states will be calculated. By default, the transition dipole moment is calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment between a set of EOM-CCSD states and another EOM-CCSD state, the CC\_REFSYM and CC\_STATE\_DERIV must be specified for this state.

TYPE:

LOGICAL

DEFAULT:

FALSE (no transition dipole and oscillator strength will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case if transition properties between the CCSD reference and EOM-CCSD target states are requested) need to be solved for transition properties, approximately doubling the computational cost. The cost of the transition properties calculation itself is low.

**CC\_EOM\_TWOPART\_PROP**

Request for calculation of non-relaxed two-particle EOM-CCSD target state properties. The two-particle properties currently include  $\langle S^2 \rangle$ . The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of  $\langle S^2 \rangle$ . The variable CC\_EXSTATES\_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are extremely computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation for each state). Do not request the two-particle properties unless you really need them.

**CC\_SYMMETRY**

Controls the use of symmetry in coupled-cluster calculations

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Use the point group symmetry of the molecule

FALSE Do not use point group symmetry (all states will be of  $A$  symmetry).

RECOMMENDATION:

It is automatically turned off for any finite difference calculations, *e.g.* second derivatives.

**Note:** The keywords below are useful mostly for the developers

**CC\_AMPL\_RESP**

If set to TRUE, adds amplitude response terms to one-particle and two-particle CCSD density matrices before calculation of properties. CC\_PROP must be set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no amplitude response terms will be added to density matrices)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost is always about the cost of an analytic gradient calculation, independent of whether or not the two-particle properties are requested. Besides, adding amplitude response terms without orbital response will unlikely improve the quality of the properties. However, it can be used for debugging purposes.

**CC\_FULL\_RESP**

If set to TRUE, adds both amplitude and orbital response terms to one- and two-particle CCSD density matrices before calculation of the properties. CC\_PROP must be set to TRUE. If both CC\_AMPL\_RESP=TRUE and CC\_FULL\_RESP=TRUE, the CC\_AMPL\_RESP=TRUE will be ignored.

TYPE:

LOGICAL

DEFAULT:

FALSE No orbital response terms will be added to density matrices, and no amplitude response will be calculated if CC\_AMPL\_RESP = FALSE

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost for the full response properties calculation is about the same as the cost of the analytic gradient. Adding full response terms improves quality of calculated properties, but usually it is a small but expensive correction. Use it only if you really need accurate properties.

**CC\_EOM\_AMPL\_RESP**

If set to TRUE, adds amplitude response terms to one-particle and two-particle EOM-CCSD density matrices before calculation of properties. CC\_EXSTATES\_PROP must be set to TRUE.

TYPE:

INTEGER

DEFAULT:

FALSE (no amplitude response terms will be added to density matrices)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost is always about the cost of an analytic gradient calculation for each state, independent of whether or not the two-particle properties are requested. Besides, adding amplitude response terms without orbital response will unlikely improve the quality of the properties. However, it can be used for debugging purposes.

**CC\_EOM\_FULL\_RESP**

If set to TRUE, adds both amplitude and orbital response terms to one- and two-particle EOM-CCSD density matrices before calculation of the properties. CC\_EXSTATES\_PROP must be set to TRUE. If both CC\_EOM\_AMPL\_RESP=TRUE and CC\_EOM\_FULL\_RESP=TRUE, the CC\_EOM\_AMPL\_RESP=TRUE will be ignored.

TYPE:

LOGICAL

DEFAULT:

FALSE No orbital response terms will be added to density matrices, and no amplitude response will be calculated if CC\_EOM\_AMPL\_RESP = FALSE

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost for the full response properties calculation is about the same as the cost of the analytic gradient for each state. Adding full response terms improves quality of calculated properties, but usually it is a small but expensive correction. Use it only if you really need accurate properties.

**6.5.10 Examples**

**Example 6.13** Geometry optimization for the excited open-shell singlet state,  $^1B_2$ , of methylene followed by the calculations of the fully relaxed one-electron properties using EOM-EE-CCSD

```

$molecule
0 1
C
H 1 rCH
H 1 rCH 2 aHCH

rCH    = 1.083
aHCH   = 145.
$end

$rem
jobtype      OPT
CORRELATION  CCSD
BASIS        cc-pVTZ
SCF_GUESS    CORE
SCF_CONVERGENCE 9
CC_NLOWSPIN  [0,0,0,1]
CC_NGUESS_SINGLES 2
cc_state_to_opt [4,1]
CC_DCONVERGENCE 9      use tighter convergence for EOM amplitudes
CC_T_CONV      9      use tighter convergence for CCSD amplitudes
$end

@@@
$molecule
READ
$end

$rem
jobtype      SP
CORRELATION  CCSD
BASIS        cc-pVTZ
SCF_GUESS    READ
CC_NLOWSPIN  [0,0,0,1]
CC_NGUESS_SINGLES 2

```

```

CC_EXSTATES_PROP 1 calculate properties for EOM states
CC_EOM_FULL_RESP 1 use fully relaxed properties
$end

```

**Example 6.14** Property and transition property calculation on the lowest singlet state of CH<sub>2</sub> using EOM-SF-CCSD

```

$molecule
  0 3
  C
  H 1 rch
  H 1 rch 2 ahch

  rch = 1.1167
  ahch = 102.07
$end

$rem
  CORRELATION      ccsd
  EXCHANGE         hf
  BASIS            cc-pvtz
  SCF_GUESS        core
  SCF_CONVERGENCE  9
  CC_NLOWSPIN      [2,0,0,3]  Get three 1^B2 and two 1^A1
  CC_SPIN_FLIP     1  Do SF. Note the triplet reference.
  CC_EXSTATES_PROP 1
  CC_TRANS_PROP    1
  CC_STATE_DERIV   1  First EOM state in the REFSYM irrep
  CC_REFSYM        4  Calc. trans. prop. between 1^B2 and all other states
$end

```

**Example 6.15** Geometry optimization with tight convergence for the <sup>2</sup>A<sub>1</sub> excited state of CH<sub>2</sub>Cl, followed by calculation of non-relaxed and fully relaxed permanent dipole moment and  $\langle S^2 \rangle$ .

```

$molecule
  0 2
  H
  C 1 CH
  CL 2 CCL 1 CCLH
  H 2 CH 3 CCLH 1 DIH

  CH=1.096247
  CCL=2.158212
  CCLH=122.0
  DIH=180.0
$end

$rem
  JOBTYP E      OPT
  CORRELATION   CCSD
  BASIS         6-31G*  Basis Set
  SCF_GUESS     SAD
  CC_DCONVERGENCE 9  EOM amplitude convergence
  CC_T_CONV      9  CCSD amplitudes convergence
  CC_NLOWSPIN    [0,0,0,1]
  cc_state_to_opt [4,1]
  CC_NGUESS_SINGLES 2
  GEOM_OPT_TOL_GRADIENT 2
  GEOM_OPT_TOL_DISPLACEMENT 2
  GEOM_OPT_TOL_ENERGY 2
$end

```

```

@@@
$molecule
READ
$end

$rem
JOBTYPE          SP
CORRELATION      CCSD
BASIS            6-31G*  Basis Set
SCF_GUESS        READ
CC_NLOWSPIN      [0,0,0,1]
CC_NGUESS_SINGLES 2
CC_EXSTATES_PROP 1   calculate one-electron properties
CC_EOM_TWOPART_PROP 1 and two-electron properties (S^2)
$end

@@@
$molecule
READ
$end

$rem
JOBTYPE          SP
CORRELATION      CCSD
BASIS            6-31G*  Basis Set
SCF_GUESS        READ
CC_NLOWSPIN      [0,0,0,1]
CC_NGUESS_SINGLES 2
CC_EXSTATES_PROP 1
CC_EOM_TWOPART_PROP 1
CC_EOM_FULL_RESP 1  same as above, but do fully relaxed properties
$end

```

**Example 6.16** CCSD calculation on three  $A_2$  and one  $B_2$  state of formaldehyde. Transition properties will be calculated between the third  $A_2$  state and all other EOM states

```

$molecule
  O  1
  O
  C  1  1.4
  H  2  1.0  1  120
  H  3  1.0  1  120
$end

$rem
BASIS            6-31+G
CORRELATION      CCSD
CC_NLOWSPIN      [0,3,0,1]
STATE_TO_OPT     [1,2]
CC_STATE_DERIV   2
UNRESTRICTED     true
CC_TRANS_PROP    true
$end

```

### 6.5.11 EOM(2,3) methods for higher accuracy and problematic situations

In the EOM-CC(2,3) approach [47], the transformed Hamiltonian  $\bar{H}$  is diagonalized in the basis of the reference, singly, doubly, and triply excited determinants, *i.e.*, the excitation operator  $R$  is truncated at triple excitations. The excitation operator  $T$ , however, is truncated at double



excitation level, and its amplitudes are found from the CCSD equations, just like for EOM-CCSD (or EOM-CC(2,2)) method.

The accuracy of the EOM-CC(2,3) method closely follows that of full EOM-CCSDT (which can be also called EOM-CC(3,3)), whereas computational cost of the former model is less.

The inclusion of triple excitations is necessary for achieving chemical accuracy (1 kcal/mol) for ground state properties. It is even more so for excited states. In particular, triple excitations are crucial for doubly excited states [47], excited states of some radicals and SF calculations (diradicals, triradicals, bond-breaking) when a reference open-shell state is heavily spin-contaminated. Accuracy of EOM-CCSD and EOM-CC(2,3) is compared in Table 6.5.11.

System	EOM-CCSD	EOM-CC(2,3)
Singly-excited electronic states	0.1–0.2 eV	0.01 eV
Doubly-excited electronic states	~ 1 eV	0.1–0.2 eV
Severe spin-contamination of the reference	~ 0.5 eV	≤ 0.1 eV
Breaking single bond (EOM-SF)	0.1–0.2 eV	0.01 eV
Breaking double bond (EOM-2SF ?)	~ 1 eV	0.1–0.2 eV

Table 6.1: Performance of the EOM-CCSD and EOM-CC(2,3) methods

To extent the applicability of the EOM-CC(2,3) model to larger systems, the active-space variant of EOM-CC(2,3) was also implemented, in which triple excitations are restricted to semi-internal ones.

Since the computational scaling of EOM-CC(2,3) method is  $\mathcal{O}(N^8)$ , these calculations can be performed only for relatively small systems. Moderate size molecules (10 heavy atoms) can be tackled by either using the active space implementation or tiny basis sets. To achieve high accuracy for these systems, energy additivity schemes are used, as advocated by Gauss and coworkers, who were very successful in extrapolating EOM-CCSDT/large basis set values by combining large basis set EOM-CCSD calculations with small basis set EOM-CCSDT ones.

Along with EOM-CC(2,3), the CISDT method has been also implemented. It can be invoked by specifying `CC_MAXITER = 0` and `CC_DO_CISDT = 1`. Default for `CC_DO_CISDT` is 0. EE, SF, and double SF variants of CISDT are available. Note that double SF CISDT is a size-extensive model. The active space CISDT calculations can be performed similarly to the active space EOM-CC(2,3) ones.

Running the full EOM-CC(2,3) calculations is straightforward, however, the calculations are expensive with the bottlenecks being storage of the data on a hard drive and the CPU time. Calculations with around 80 basis functions are possible for a molecule consisting of four first row atoms (NO dimer). The number of basis functions can be larger for smaller systems.

**Note:** In EE calculations, one needs to always solve for at least one low-spin root in the first symmetry irrep in order to obtain the correlated EOM energy of the reference. The triples correction to the total reference energy must be used to evaluate EOM-(2,3) excitation energies.

### 6.5.12 Active space EOM-CC(2,3)

Active space calculations are less demanding with respect to the size of a hard drive. The main bottlenecks here are the memory usage and the CPU time. Both arise due to the increased number of orbital blocks in the active space calculations. In the current implementation, each block can contain from 0 up to 16 orbitals of the same symmetry irrep, occupancy, and spin-symmetry. For

example, for a typical molecule of  $C_{2v}$  symmetry, in a small/moderate basis set (*e.g.*, TMM in 6-31G\*), the number of blocks for each index is:

$$\begin{aligned} \text{occupied: } & (\alpha + \beta) \times (a_1 + a_2 + b_1 + b_2) = 2 \times 4 = 8 \\ \text{virtinals: } & (\alpha + \beta) \times (2a_1 + a_2 + b_1 + 2b_2) = 2 \times 6 = 12 \\ & \text{(usually there are more than 16 } a_1 \text{ and } b_2 \text{ virtual orbitals).} \end{aligned}$$

In EOM-CCSD, the total number of blocks is  $O^2V^2 = 8^2 \times 12^2 = 9216$ . In EOM-CC(2,3) the number of blocks in the EOM part is  $O^3V^3 = 8^3 \times 12^3 = 884736$ . In active space EOM-CC(2,3), additional fragmentation of blocks occurs to distinguish between the restricted and active orbitals. For example, if the active space includes occupied and virtual orbitals of all symmetry irreps (this will be a very large active space), the number of occupied and virtual blocks for each index is 16 and 20, respectively, and the total number of blocks increases to  $3.3 \times 10^7$ . Not all of the blocks contain real information, some blocks are zeros because of the spatial or spin-symmetry requirements. For the  $C_{2v}$  symmetry group, the number of non-zero blocks is about 10–12 times less than the total number of blocks, *i.e.*,  $3 \times 10^6$ . This is the number of non-zero blocks in *one* vector. Davidson diagonalization procedure requires  $(2*\text{MAX\_VECTORS} + 2*\text{NROOTS})$  vectors, where MAX\_VECTORS is the maximum number of vectors in the subspace, and NROOTS is the number of the roots to solve for. Taking NROOTS=2 and MAX\_VECTORS=20, we obtain 44 vectors with the total number of non-zero blocks being  $1.3 \times 10^8$ .

In our implementation, each block is a logical unit of information. Along with real data, which are kept on a hard drive at all the times except of their direct usage, each non-zero block contains an auxiliary information about its size, structure, relative position with respect to other blocks, location on a hard drive, and so on. The auxiliary information about blocks is *always* kept in memory. Currently, the approximate size of this auxiliary information is about 400 bytes per block. It means, that in order to keep information about one vector ( $3 \times 10^6$  blocks), 1.2 GB of memory is required! The information about 44 vectors amounts 53 GB. Moreover, the huge number of blocks significantly slows down the code.

To make the calculations of active space EOM-CC(2,3) feasible, we need to reduce the total number of blocks. One way to do this is to reduce the symmetry of the molecule to lower or  $C_1$  symmetry group (of course, this will result in more expensive calculation). For example, lowering the symmetry group from  $C_{2v}$  to  $C_s$  would results in reducing the total number of blocks in active space EOM-CC(2,3) calculations in about  $2^6 = 64$  times, and the number of non-zero blocks in about 30 times (the relative portion of non-zero blocks in  $C_s$  symmetry group is smaller compared to that in  $C_{2v}$ ).

Alternatively, one may keep the MAX\_VECTORS and NROOTS parameters of Davidson's diagonalization procedure as small as possible (this mainly concerns the MAX\_VECTORS parameter). For example, specifying MAX\_VECTORS = 12 instead of 20 would require 30% less memory.

One more trick concerns specifying the active space. In a desperate situation of a severe lack of memory, should the two previous options fail, one can try to modify (increase) the active space in such a way that the fragmentation of active and restricted orbitals would be less. For example, if there is one restricted occupied  $b_1$  orbital and one active occupied  $B_1$  orbital, adding the restricted  $b_1$  to the active space will reduce the number of blocks, by the price of increasing the number of FLOPS. In principle, adding extra orbital to the active space should increase the accuracy of calculations, however, a special care should be taken about the (near) degenerate pairs of orbitals, which should be handled in the same way, *i.e.*, both active or both restricted.

### 6.5.13 Job Control for EOM-(2,3)

The following options are available:

**CC\_DO\_TRIPLES**

This keyword initializes a EOM-CC(2,3) calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No EOM-CC(2,3) calculation

1 Perform a EOM-CC(2,3) calculation

RECOMMENDATION:

None

**CC\_DO\_SMALL\_TRIPLES**

Pre-converges the EOM-CCSD block of the Hamiltonian by doing several EOM-CCSD iterations before switching to EOM(2,3) using the EOM-CCSD vectors as a guess for EOM-CC(2,3)

TYPE:

INTEGER

DEFAULT:

0 Do not do any EOM-CCSD iterations

OPTIONS:

$n$  Do  $n$  iterations

RECOMMENDATION:

None

**CC\_PRECONV\_SD**

Solves the EOM-CCSD equations, prints energies, then uses EOM-CCSD vectors as initial vectors in EOM-CC(2,3). Very convenient for calculations using energy additivity schemes.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Turns the pre-converging on

RECOMMENDATION:

Turning this option on is recommended

**CC\_RESTR\_AMPL**

Forces the integrals,  $T$ , and  $R$  amplitudes to be determined in the full space even though the CC\_REST\_OCC and CC\_REST\_VIR keywords are used.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Do apply restrictions

1 Do not apply restrictions

RECOMMENDATION:

None

**CC\_RESTR\_TRIPLES**

Restricts  $R_3$  amplitudes to the active space, *i.e.*, one electron should be removed from the active occupied orbital and one electron should be added to the active virtual orbital.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Applies the restrictions

RECOMMENDATION:

None

**CC\_REST\_OCC**

Sets the number of restricted occupied orbitals including frozen occupied orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Restrict  $n$  occupied orbitals.

RECOMMENDATION:

None

**CC\_REST\_VIR**

Sets the number of restricted virtual orbitals including frozen virtual orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Restrict  $n$  virtual orbitals.

RECOMMENDATION:

None

To select the active space, orbitals can be reordered by specifying the new order in the *\$reorder\_mosection*. The section consists of two rows of numbers ( $\alpha$  and  $\beta$  sets), starting from 1, and ending with  $n$ , where  $n$  is the number of the last orbital specified.

**Example 6.17** Example *\$reorder\_mosection* with orbitals 16 and 17 swapped for both  $\alpha$  and  $\beta$  electrons

```
$reorder_mo
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 16
$end
```

**6.5.14 Examples**

**Example 6.18** EOM-SF(2,3) calculations of methylene.

```

$molecule
0 3
C
H 1 CH
H 1 CH 2 HCH

CH = 1.07
HCH = 111.0
$end

$rem
correlation      ccsd
cc_do_triples    1    do EOM-(2,3)
basis            6-31G
cc_nlowspin      [2,0,0,2]
n_frozen_core    1
n_frozen_virtual 1
cc_preconv_sd    1 Get EOM-CCSD energies first.
cc_spin_flip     1
$end

```

**Example 6.19** This is active-space EOM-SF(2,3) calculations for methane with an elongated CC bond. HF MOs should be reordered as specified in the *\$reorder\_mosection* such that active space for triples consists of sigma and sigma\* orbitals.

```

$molecule
0 3
C
H 1 CH
H 1 CHX 2 HCH
H 1 CH 2 HCH 3 A120
H 1 CH 2 HCH 4 A120

CH=1.086
HCH=109.4712206
A120=120.
CHX=1.8
$end

$rem
jobtype          sp
correlation      ccsd
exchange         hf
basis            6-31G*
cc_nlowspin      [1,0]
cc_spin_flip     1
n_frozen_core    1
cc_do_triples    1    does eom-2,3
cc_preconv_sd    1    does eom-ccsd first
cc_restr_triples 1    triples are restricted to the active space only
cc_restr_ampl    0    ccsd and eom singles and doubles are full-space
cc_rest_occ      4    specifies active space
cc_rest_vir      17   specifies active space
print_orbitals   10   (number of virtuals to print)
$end

$reorder_mo
1 2 5 4 3
1 2 3 4 5
$end

```

## 6.6 Potential Energy Surface Crossing Minimization

Potential energy surface crossing optimization procedure finds energy minima on crossing seams. On the seam, the potential surfaces are degenerated in the subspace perpendicular to the plane defined by two vectors – gradient difference

$$\mathbf{g} = \frac{\partial}{\partial \mathbf{q}}(E_1 - E_2)$$

and derivative coupling

$$\mathbf{h} = \left\langle \Psi_1 \left| \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right| \Psi_2 \right\rangle.$$

At the time Q-CHEM is unable to locate crossing minima for states which have non-zero derivative coupling. Fortunately, often this is not the case. Minima on the seams of conical intersections of states of different multiplicity can be found as their derivative coupling is zero. Minima on the seams of intersections of states of different point group symmetry can be located as well.

To run a PES crossing minimization, CCSD and EOM-CCSD methods must be employed for the ground and excited state calculations respectively.

### 6.6.1 Job Control Options

#### XOPT\_STATE\_1, XOPT\_STATE\_2

Specify two electronic states the intersection of which will be searched.

TYPE:

[INTEGER, INTEGER, INTEGER]

DEFAULT:

No default value (the option must be specified to run this calculation)

OPTIONS:

[spin, irrep, state]

spin = 0

Addresses states with low spin,  
see also CC\_NLOWSPIN.

spin = 1

Addresses states with high spin,  
see also CC\_NHIGHSPIN.

irrep

Specifies the irreducible representation to which  
the state belongs, for  $C_{2v}$  point group symmetry  
irrep = 1 for  $A_1$ , irrep = 2 for  $A_2$ ,  
irrep = 3 for  $B_1$ , irrep = 4 for  $B_2$ .

state

Specifies the state number within the irreducible  
representation, state = 1 means the lowest excited  
state, state = 2 is the second excited state, etc.

0, 0, -1

Ground state.

RECOMMENDATION:

Only intersections of states with different spin or symmetry can be calculated at this time.

**Note:** The spin can only be specified when using closed-shell RHF references. In the case of open-shell references all states are treated together, see also CC\_NHIGHSPIN.

**XOPT\_SEAM\_ONLY**

Orders an intersection seam search only, no minimization is to perform.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Find a point on the intersection seam and stop.

FALSE Perform a minimization of the intersection seam.

RECOMMENDATION:

In systems with a large number of degrees of freedom it might be useful to locate the seam first setting this option to TRUE and use that geometry as a starting point for the minimization.

**6.6.2 Examples**

**Example 6.20** Minimize the intersection of  $\tilde{A}^2A_1$  and  $\tilde{B}^2B_1$  states of the NO<sub>2</sub> molecule using EOM-IP-CCSD method

```

$molecule
-1 1
N1
O2 N1 rno
O3 N1 rno O2 aono

rno = 1.3040
aono = 106.7
$end

$rem
JOBTYPE          opt          Optimize the intersection seam
EXCHANGE         hf
UNRESTRICTED    true
CORRELATION      ccsd
BASIS            6-31g

CC_IP            true
CC_NLOWSPIN     [1,0,1,0]  C2v point group symmetry

XOPT_STATE_1    [0,1,1]   1A1 low spin state
XOPT_STATE_2    [0,3,1]   1B1 low spin state
GEOM_OPT_TOL_GRADIENT  30       Tighten gradient tolerance
$END

```

**Example 6.21** Minimize the intersection of  $\tilde{A}^1B_2$  and  $\tilde{B}^1A_2$  states of the N<sub>3</sub><sup>+</sup> ion using EOM-CCSD method

```

$molecule
1 1
N1
N2 N1 rnn
N3 N2 rnn N1 annn

rnn=1.46
annn=70.0
$end

$rem

```

```

JOBTYPE          opt
EXCHANGE         hf
CORRELATION      ccscd
BASIS            6-31g

CC_NLOWSPIN     [0,2,0,2]   C2v point group symmetry

XOPT_STATE_1    [0,4,1]    1B2 low spin state
XOPT_STATE_2    [0,2,2]    2A2 low spin state
XOPT_SEAM_ONLY  true      Find the seam only
GEOM_OPT_TOL_GRADIENT 100

$end

$opt
CONSTRAINT      Set constraints on the N-N bond lengths
  stre 1 2 1.46
  stre 2 3 1.46
ENDCONSTRAINT
$end

@@@

$molecule
  READ
$end

$rem
JOBTYPE          opt          Optimize the intersection seam
EXCHANGE         hf
CORRELATION      ccscd
BASIS            6-31g

CC_NLOWSPIN     [0,2,0,2]

XOPT_STATE_1    [0,4,1]
XOPT_STATE_2    [0,2,2]
GEOM_OPT_TOL_GRADIENT 30
$end

```

## 6.7 Dyson Orbitals for Ionization from the ground and electronically excited states within EOM-CCSD formalism

Dyson orbitals can be used to compute angular distribution of photoelectrons. For a general wavefunction, Dyson orbitals represent the overlap between the  $N$  electron molecular wavefunction and the  $N-1/N+1$  electron wavefunction of the corresponding cation/anion:

$$\phi^d(1) = \frac{1}{N-1} \int \Psi^N(1, \dots, n) \Psi^{N-1}(2, \dots, n) d2 \dots dn \quad (6.29)$$

$$\phi^d(1) = \frac{1}{N+1} \int \Psi^N(2, \dots, n+1) \Psi^{N+1}(1, \dots, n+1) d2 \dots d(n+1) \quad (6.30)$$

For Hartree-Fock wavefunctions and within Koopmans' approximation, these are just the canonical wavefunction of the ionized/attached electron. For correlated wavefunctions, Dyson orbitals can be decomposed into contributions from the reference molecular orbitals:



$$\phi^d = \sum_p \gamma_p \phi_p \quad (6.31)$$

$$\gamma_p = \langle \Psi^N | p^+ | \Psi^{N-1} \rangle \quad (6.32)$$

$$\gamma_p = \langle \Psi^N | p | \Psi^{N+1} \rangle \quad (6.33)$$

The calculation of Dyson orbitals is straightforward within the EOM-IP/EA-CCSD methods, where cation/anion and initial molecule states are defined with respect to the same MO basis. Since the left and right CC vectors are not the same, one can define correspondingly two Dyson orbitals (left-right and right-left):

$$\gamma_p^R = \langle \Phi_0 e^{T_1+T_2} L^{EE} | p^+ | R^{IP} e^{T_1+T_2} \Phi_0 \rangle \quad (6.34)$$

$$\gamma_p^L = \langle \Phi_0 e^{T_1+T_2} L^{IP} | p | R^{EE} e^{T_1+T_2} \Phi_0 \rangle \quad (6.35)$$

The norm of these orbitals is proportional to the one-electron character of the transition.

Dyson orbitals can be used as a tool that shows the difference between a molecular state and its cation/anion. For ionization transitions, these orbitals can be also interpreted as the wavefunction of the leaving electron before ionization.

### 6.7.1 Dyson Orbitals Job Control

The calculation of Dyson orbitals is implemented for two cases: ground (reference) and excited state ionization/electron attachment. To obtain the ground state Dyson orbitals one needs to run a regular EOM-IP/EA-CCSD calculation, request transition properties calculation by setting `CC_TRANS_PROP=TRUE` and `CC_DO_DYSON = TRUE`. The Dyson orbitals decomposition in the MO basis is printed in the output, for all transitions between the reference and all IP/EA states. At the end of the file, also the coefficients of these Dyson orbitals in the AO basis are available.

#### **CC\_DO\_DYSON**

Whether ground state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE (the option must be specified to run this calculation)

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

none

For calculating Dyson orbitals between excited states from the reference configuration and IP/EA states, `CC_TRANS_PROP=TRUE` and `CC_DO_DYSON_EE = TRUE` have to be added to the usual EOM-IP/EA-CCSD calculation. The `CC_NHIGHSPIN` keyword is used to specify the (initial) excited states (EE), and `CC_NLOWSPIN` specifies target ionized or attached states for the IP/EA calculation. The Dyson orbital decomposition in MO and AO bases is printed for each EE-IP/EA pair of states in the order: EE1 - IP/EA1, EE1 - IP/EA2, ..., EE2 - IP/EA1, EE2 - IP/EA2, ..., and so on.

**CC\_DO\_DYSON\_EE**

Whether excited state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE (the option must be specified to run this calculation)

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

none

Dyson orbitals can be also plotted using `IANLTY = 200` and the `$plots` utility. Only the sizes of the box need to be specified, followed by a line of zeros:

```
$plots
comment
  10  -2  2
  10  -2  2
  10  -2  2
  0   0  0  0
$plots
```

All Dyson orbitals on the xyz Cartesian grid will be written in the resulting plot.mo file. For RHF(UHF) reference, the columns order in plot.mo is:  $\phi_1^{lr}\alpha$  ( $\phi_1^{lr}\beta$ )  $\phi_1^{rl}\alpha$  ( $\phi_1^{rl}\beta$ )  $\phi_2^{lr}\alpha$  ( $\phi_2^{lr}\beta$ ) ...

**Note:** The meaning of the `CC_NHIGHSPIN` is different for Dyson orbital calculations!

## 6.7.2 Examples

**Example 6.22** Plotting grd-ex and ex-grd state Dyson orbitals for ionization of the oxygen molecule. The target states of the cation are  ${}^2A_g$  and  ${}^2B_{2u}$ .

```
$molecule
0 3
0 0.000 0.000 0.000
0 1.222 0.000 0.000
$end

$rem
jobtype      sp
basis        6-31G*
correlation  ccsd
cc_ip        true   request EOM-IP calcs
cc_nlowspin  [1,0,0,0,0,0,1,0] Target EOM-IP states
cc_trans_prop true   request transition OPDMs to be calculated
cc_do_dyson true   calculate Dyson orbitals
IANLTY 200
$end

$plots
plots excited states densities and trans densities
  10  -2  2
  10  -2  2
  10  -2  2
```

```

    0    0    0    0
$plots

```

**Example 6.23** Plotting ex-ex state Dyson orbitals between the 1st  $^2A_1$  excited state of the HO radical and the the 1st  $A_1$  and  $A_2$  excited states of  $\text{HO}^-$

```

$molecule
-1 1
H 0.000 0.000 0.000
O 1.000 0.000 0.000
$end

$rem
jobtype          SP
correlation      CCSD
BASIS            6-31G*
CC_NLOWSPIN     [1,0,0,0]    states of HO radical
CC_NHIGHSPIN    [1,1,0,0]    excited states of HO-
CC_IP true      do EOM-IP
CC_TRANS_PROP true    calculate transition properties
CC_DO_DYSON_EE true    calculate Dyson orbitals for ionization from excited states
IANLTY 200
$end

$plots
plot excited states densities and trans densities
  10  -2  2
  10  -2  2
  10  -2  2
  0   0  0  0
$plots

```

## 6.8 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 [9] which has proven successful in applications reported so far [9,35,36]. 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

$$\Delta = \mathbf{P}_1 - \mathbf{P}_2 \quad (6.36)$$

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

$$\mathbf{U}^t \Delta \mathbf{U} = \delta \quad (6.37)$$

The sum of the occupation numbers  $\delta_p$  of these orbitals is then

$$\text{tr}(\Delta) = \sum_{p=1}^N \delta_p = n \quad (6.38)$$

where  $n$  is the net gain or loss of electrons in the transition. The net gain in an electronic transition which does not involve ionization or electron attachment will obviously be zero.

The detachment density

$$\mathbf{D} = \mathbf{U}\mathbf{d}\mathbf{U}^t \quad (6.39)$$

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

$$d_p = -\min(\delta_p, 0) \quad (6.40)$$

The detachment density corresponds to the electron density associated with single particle levels vacated in an electronic transition or hole density.

The attachment density

$$\mathbf{A} = \mathbf{U}\mathbf{a}\mathbf{U}^t \quad (6.41)$$

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

$$a_p = \max(\delta_p, 0) \quad (6.42)$$

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

$$\Delta = \mathbf{A} - \mathbf{D} \quad (6.43)$$

# References and Further Reading

- [1] Basis sets (Chapters 7 and 8)
- [2] Ground state methods (Chapters 4 and 5)
- [3] J. E. Del Bene, R. Ditchfield and J. A. Pople, *J. Chem. Phys.*, **55**, 2236, (1971).
- [4] J. B. Foresman, M. Head-Gordon, J. A. Pople and M. J. Frisch, *J. Phys. Chem.*, **96**, 135, (1992).
- [5] D. Maurice and M. Head-Gordon, *Int. J. Quantum Chem.*, **29**, 361, (1995).
- [6] T. D. Bouman and A. E. Hansen, *Int. J. Quantum Chem. Sym*, **23**, 381, (1989).
- [7] A. E. Hansen, B. Voigt and S. Rettrup, *Int. J. Quantum Chem.*, **23**, 595, (1983).
- [8] D. Maurice and M. Head-Gordon, *J. Phys. Chem.*, **100**, 6131, (1996).
- [9] M. Head-Gordon, A. M. Grana, D. Maurice and C. A. White, *J. Phys. Chem.*, **99**, 14261, (1995).
- [10] J. F. Stanton, J. Gauss, N. Ishikawa and M. Head-Gordon, *J. Chem. Phys.*, **103**, 4160, (1995).
- [11] S. Zilberg and Y. Haas, *J. Chem. Phys.*, **103**, 20, (1995).
- [12] C. M. Gittins, E. A. Rohlfing and C. M. Rohlfing, *J. Chem. Phys.*, **105**, 7323, (1996).
- [13] D. Maurice and M. Head-Gordon, *Mol. Phys.*, **96**, 1533, (1999).
- [14] D. Maurice, Ph.D. Thesis. "Single Electron Theories of Excited States" University of California: Berkeley, 1998.
- [15] E. Runge and E. K. U. Gross, *Phys. Rev. Lett.*, **52**, 997, (1984).
- [16] M. E. Casida, in *Recent Advances in Density Functional Methods*, Part I, edited by D. P. Chong (World Scientific, Singapore, 1995), p 155.
- [17] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.*, **302**, 375, (1999).
- [18] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.*, **314**, 291, (1999).
- [19] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub, *J. Chem. Phys.*, **108**, 4439, (1998).
- [20] D. J. Tozer and N. C. Handy, *J. Chem. Phys.*, **109**, 10180, (1998).
- [21] S. Hirata, T. J. Lee, and M. Head-Gordon, *J. Chem. Phys.*, **111**, 8904, (1999).
- [22] N. A. Besley, *Chem. Phys. Lett.*, **390**, 124, (2004).

- [23] N. A. Besley, M. T. Oakley, A. J. Cowan and J. D. Hirst, *J. Am. Chem. Soc.*, **126**, 13502, (2004).
- [24] N. A. Besley, *J. Chem. Phys.*, **122**, 184706, (2005).
- [25] D. M. Rogers, N. A. Besley, P. O'Shea and J. D. Hirst, *J. Phys. Chem. B*, **109**, 23061, (2005).
- [26] Y. Shao, M. Head-Gordon and A. I. Krylov, *J. Chem. Phys.*, **188**, 4807, (2003).
- [27] M. Head-Gordon, R. J. Rico, M. Oumi and T. J. Lee, *Chem. Phys. Lett.*, **219**, 21, (1994).
- [28] M. Head-Gordon, D. Maurice and M. Oumi, *Chem. Phys. Lett.*, **246**, 114, (1995).
- [29] M. Oumi, D. Maurice, T. J. Lee and M. Head-Gordon, *Chem. Phys. Lett.*, **279**, 151, (1997).
- [30] See for example, H. Koch and P. Jorgensen, *J. Chem. Phys.*, **93**, 3333, (1990).
- [31] J.F. Stanton, and R.J. Bartlett, *J. Chem. Phys.*, **98**, 7029, (1993).
- [32] A. I. Krylov, C. D. Sherrill, and M. Head-Gordon, *J. Chem. Phys.*, **113**, 6509, (2000).
- [33] A. I. Krylov, *Chem. Phys. Lett.*, **338**, 375, (2001).
- [34] A. I. Krylov, *Chem. Phys. Lett.*, **350**, 522, (2001).
- [35] M. Oumi, D. Maurice and M. Head-Gordon, *Spectrochim. Acta A*, **55**, 525, (1999).
- [36] C.-P. Hsu, S. Hirata and M. Head-Gordon, *J. Phys. Chem. A*, **105**, 451, (2001).
- [37] M. Wladyslawski and M. Nooijen, *ACS Symposium Series*, **828**, 65, (2002).
- [38] H. Sekino and R.J. Bartlett, *Int. J. Quant. Chem. Symp.*, **18**, 255, (1984).
- [39] *J. Chem. Phys.*, **93**, 3345, (1990).
- [40] J.F. Stanton and R.J. Bartlett, *J. Chem. Phys.*, **98**, 7029, (1993).
- [41] A.I. Krylov, *Chem. Phys. Lett.*, **338**, 375, (2001).
- [42] S.V. Levchenko and A.I. Krylov, *J. Chem. Phys.*, **120**, 175, (2004).
- [43] J.F. Stanton and J. Gauss, *J. Chem. Phys.*, **101**, 8938, (1994).
- [44] M. Nooijen and R.J. Bartlett, *J. Chem. Phys.*, **102**, 3629, (1995).
- [45] D. Sinha, D. Mukhopadhyaya, R. Chaudhuri, and D. Mukherjee, *Chem. Phys. Lett.*, **154**, 544, (1989).
- [46] S.V. Levchenko, T. Wang, and A.I. Krylov, *J. Chem. Phys.*, **122**, 224106, (2005).
- [47] S. Hirata, M. Nooijen, and R.J. Bartlett, *Chem. Phys. Lett.*, **326**, 255, (2000).
- [48] A.I. Krylov, *Acc. Chem. Res.*, **39**, 83 (2006).

# 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.

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, the 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” [3], 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 behavior at the origin and in the asymptotic regions. However, the two-electron repulsion integrals (ERIs) encountered when using Slater basis functions are expensive and difficult to evaluate. 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. The number of basis functions used depends on 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 [3–7] to aid your selection and see the section “Further Reading” at the end of this chapter.

## 7.2 Built-in Basis sets

Q-CHEM is equipped with many standard basis sets [8], and allows the user to specify the required basis set by its standard symbolic representation. The available built-in basis sets are of four types:

- Pople basis sets
- Dunning basis sets
- Correlation consistent Dunning basis sets
- Ahlrichs basis sets

In addition, Q-CHEM supports the following features:

- Extra diffuse functions available for high quality excited state calculations.
- Standard polarization functions.
- Basis sets are requested by symbolic representation.
- *s*, *p*, *sp*, *d*, *f* and *g* angular momentum types of basis functions.
- Maximum number of shells per atom is 100.
- Pure and Cartesian basis functions.
- Mixed basis sets (see section 7.5).
- Basis set superposition error (BSSE) corrections.

The following *\$rem* keyword controls the basis set:

### **BASIS**

Sets the basis set to be used

### TYPE:

STRING

### DEFAULT:

No default basis set

### OPTIONS:

General, Gen User-defined. See section below

Symbol Use standard basis sets as in the table below

Mixed Use a combination of different basis sets

### RECOMMENDATION:

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 polarization functions; in parentheses with the light atom designation following heavy atom designation. (*i.e.*, *heavy*, *light*). Use a period (.) as a placeholder (see examples).



	$j$	$k$	$l$	$m$	$n$
STO- $j(k+, l+)G(m, n)$	2,3,6	$a$	$b$	$d$	$p$
$j-21(k+, l+)G(m, n)$	3	$a$	$b$	$2d$	$2p$
$j-31(k+, l+)G(m, n)$	4,6	$a$	$b$	$3d$	$3p$
$j-311(k+, l+)G(m, n)$	6	$a$	$b$	$df, 2df, 3df$	$pd, 2pd, 3pd$

Table 7.1: Summary of Pople type basis sets available in the Q-CHEM program.  $m$  and  $n$  refer to the polarization functions on heavy and light atoms respectively.  $a$   $k$  is the number of sets of diffuse functions on heavy  $b$   $l$  is the number of sets of diffuse functions on light atoms

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→Zn
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 7.2: Atoms supported for Pople basis sets available in Q-CHEM (see the Table below for specific examples).

Symbolic Name	Atoms Supported
3-21G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Xe, Cs
3-21+G	H, He, Na→Cl, Na→Ar, K, Ca, Ga→Kr
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 7.3: 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 light atom diffuse functions
$m$	# sets of d functions on heavy atoms
$n$	# sets of p functions on light atoms

Table 7.4: 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 7.5: Atoms supported for old Dunning basis sets available in Q-CHEM.

Symbolic Name	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 7.6: Examples of extended Dunning basis sets.

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 7.7: 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 7.8: 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 studying anions and excited states of molecules, and for the latter several sets of additional diffuse functions may be required. These extra diffuse functions can be generated from the standard diffuse functions by applying a scaling factor to the exponent of the original diffuse function. This yields a geometric series of exponents for the diffuse functions which includes the original standard functions along with more diffuse functions.

When using very large basis sets, especially those that include many diffuse functions, or if the system being studied is very large, linear dependence in the basis set may arise. This results in an over-complete description of the space spanned by the basis functions, and can cause a loss of uniqueness in the molecular orbital coefficients. Consequently, the SCF may be slow to converge or behave erratically. Q-CHEM will automatically check for linear dependence in the basis set, and will project out the near-degeneracies, if they exist. This will result in there being slightly fewer molecular orbitals than there are basis functions. Q-CHEM checks for linear-dependence by considering the eigenvalues of the overlap matrix. Very small eigenvalues are an indication that the basis set is close to being linearly dependent. The size at which the eigenvalues are considered to be too small is governed by the *\$rem* variable BASIS\_LIN\_DEP\_THRESH. By default this is set to 6, corresponding to a threshold of  $10^{-6}$ . This has been found to give reliable results, however, if you have a poorly behaved SCF, and you suspect there maybe linear dependence in you basis, the threshold should be increased.

#### PRINT\_GENERAL\_BASIS

Controls print out of built in basis sets in input format

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATION:

Useful for modification of standard basis sets.

#### BASIS\_LIN\_DEP\_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of  $10^{-6}$

OPTIONS:

$n$  Sets the threshold to  $10^{-n}$

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

## 7.4 User-defined Basis Sets (*\$basis*)

### 7.4.1 Introduction

Users may, on occasion, prefer to use non-standard basis, 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 or 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 Job Control

In order to use a user-defined basis set the BASIS *\$rem* must be set to GENERAL or GEN.

When using a non-standard basis set which incorporates  $d$  or higher angular momentum basis functions, 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 specify all  $g$ ,  $f$  and  $d$  basis functions as being in the pure form. 121 would indicate  $g$ - and  $d$ - functions are pure and  $f$ -functions Cartesian.

#### PURECART

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

2111 Cartesian  $h$ -functions and pure  $g$ ,  $f$ ,  $d$  functions

OPTIONS:

*hgf d* Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

In standard basis sets all functions are pure, except for the  $d$  functions in  $n - 21G$  type bases (e.g., 3-21G) and  $n - 31G$  (6-31G, 6-31G\*, 6-31+G\* ...). In particular, the 6-311G series uses pure functions (both  $d$  and  $f$ ).

### 7.4.3 Format For User-defined Basis Sets

The format for the user-defined basis section is as follows:

```

$basis
  X  0
  L  K    scale
  α1 C1Lmin C1Lmin+1 ... C1Lmax
  α2 C2Lmin C2Lmin+1 ... C2Lmax
  ⋮   ⋮       ⋮       ⋮       ⋮
  αK CKLmin CKLmin+1 ... CKLmax
****
$end

```

where

*X* Atomic symbol of the atom (atomic number not accepted)  
*L* Angular momentum symbol (S, P, SP, D, F, G)  
*K* Degree of contraction of the shell (integer)  
*scale* Scaling to be applied to exponents (default is 1.00)  
*a<sub>i</sub>* Gaussian primitive exponent (positive real number)  
*C<sub>i</sub><sup>L</sup>* 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. Note that more than one contraction coefficient per line is one required for compound shells like SP. As with all Q-CHEM input deck information, all input is case-insensitive.

#### 7.4.4 Example

**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.

```

$molecule
  0 1
  0
  H 0 oh
  H 0 oh 2 hoh

  oh = 1.2
  hoh = 110.0
$end

$rem
  EXCHANGE hf
  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 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
        0.39000    1.000000
F  1  1.00
        4.10000    1.000000
G  1  1.00
        3.35000    1.000000
****
$end

```

## 7.5 Mixed Basis Sets

In addition to defining a custom basis set, it is also possible to specify different standard basis sets for different atoms. For example, in a large alkene molecule the hydrogen atoms could be modeled by the STO-3G basis, while the carbon atoms have the larger 6-31G(d) basis. This can be specified within the *\$basis* block using the more familiar basis set labels.

- Note:** (1) It is not possible to augment a standard basis set in this way; the whole basis needs to be inserted as for a user-defined basis (angular momentum, exponents, contraction coefficients) and additional functions added. Standard basis set exponents and coefficients can be easily obtained by 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.

The user can also specify different basis sets for atoms of the same type, but in different parts of the molecule. This allows a larger basis set to be used for the active region of a system, and a smaller basis set to be used in the less important regions. To enable this the BASIS keyword must be set to MIXED and a *\$basis* section included in the input deck that gives a complete specification of the basis sets to be used. The format is exactly the same as for the user-defined basis, except that the atom number (as ordered in the *\$molecule* section) must be specified in the field after the atomic symbol. A basis set must be specified for every atom in the input, even if the same basis set is to be used for all atoms of a particular element. Custom basis sets can be entered, and the shorthand labeling of basis sets is also supported.

The use of different basis sets for a particular element means the global potential energy surface is no longer unique. The user should exercise caution when using this feature of mixed basis sets, especially during geometry optimizations and transition state searches.

### 7.5.1 Examples

**Example 7.2** 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 0 oh
  H 0 oh 2 hoh

  oh = 1.2
  hoh = 110.0
$end

```

```

$rem
  EXCHANGE  hf
  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 7.3** Example of using a mixed basis set for methanol. The user is able to specify different 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 a standard basis set so that only the non-defined atoms need have the exponents and contraction coefficients entered. Note that a basis set has to be specified for every atom in the molecule, even if the same basis is to be used on an atom type. Note also that the dummy atom is not counted.

```

$molecule
  O 1
  C
  O C rco
  H1 C rch1 0 h1co
  x C 1.0 0 xco1 h1 180.0
  H2 C rch2 x h2cx h1 90.0
  H3 C rch2 x h2cx h1 -90.0
  H4 0 roh C hoc h1 180.0

  rco = 1.421
  rch1 = 1.094
  rch2 = 1.094
  roh = 0.963
  h1co = 107.2
  xco = 129.9
  h2cx = 54.25
  hoc = 108.0
$end

$rem
  exchange  hf
  basis     mixed  user-defined mixed basis
$end

$basis
  C 1
  3-21G
  ****
  O 2
  S 3 1.00
    3.22037000E+02  5.92394000E-02
    4.84308000E+01  3.51500000E-01
    1.04206000E+01  7.07658000E-01
  SP 2 1.00
    7.40294000E+00 -4.04453000E-01  2.44586000E-01
    1.57620000E+00  1.22156000E+00  8.53955000E-01
  SP 1 1.00

```

```

    3.73684000E-01  1.00000000E+00  1.00000000E+00
SP 1 1.00
    8.45000000E-02  1.00000000E+00  1.00000000E+00
****
H 3
6-31(+,+)G(d,p)
****
H 4
sto-3g
****
H 5
sto-3g
****
H 6
sto-3g
****
$end

```

## 7.6 Basis Set Superposition Error (BSSE)

When calculating binding energies, the energies of the fragments are usually higher than they should be due to the smaller effective basis set used for the individual species. This leads to an overestimate of the binding energy called the basis set superposition error. The effects of this can be corrected for by performing the calculations on the individual species in the presence of the basis set associated with the other species. This requires basis functions to be placed at arbitrary points in space, not just those defined by the nuclear centers. This can be done within Q-CHEM by using ghost atoms. These atoms have zero nuclear charge, but can support a user defined basis set. Ghost atom locations are specified in the *\$molecule* section, as for any other atom, and the basis must be specified in a *\$basis* section in the same manner as for a mixed basis.

**Example 7.4** A calculation on a water monomer in the presence of the full dimmer basis set. The energy will be slightly lower than that without the ghost atom functions due to the greater flexibility of the basis set.

```

$molecule
O 1
O 1.68668 -0.00318 0.000000
H 1.09686 0.01288 -0.741096
H 1.09686 0.01288 0.741096
Gh -1.45451 0.01190 0.000000
Gh -2.02544 -0.04298 -0.754494
Gh -2.02544 -0.04298 0.754494
$end

$rem
EXCHANGE hf
CORRELATION mp2
BASIS mixed
$end

$basis
O 1
6-31G*
****
H 2
6-31G*
****
H 3
6-31G*

```



```
****  
O 4  
6-31G*  
****  
H 5  
6-31G*  
****  
H 6  
6-31G*  
****  
$end
```

# References and Further Reading

- [1] Ground State Methods (Chapters 4 and 5).
- [2] Effective Core Potentials (Chapter 8).
- [3] W. J. Hehre, L. Radom, P. v. R. Schleyer and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley, New York, 1986)
- [4] S. Huzinaga, *Comp. Phys. Rep.*, **2**, 279, (1985).
- [5] E. R. Davidson and D. Feller, *Chem. Rev.*, **86**, 681, (1986).
- [6] 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.
- [7] F. Jensen, *Introduction to Computational Chemistry* (Wiley, New York, 1999).
- [8] 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 [3–6].

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.*, [7] 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  $3d$  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 [8]

<http://www.emsl.pnl.gov/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 *\$rem* keyword entirely, Q-CHEM will automatically provide the matching one.

The following *\$rem* variable controls which ECP is used:

#### ECP

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No pseudopotential

OPTIONS:

General, Gen User defined. (*\$ecp* keyword required)

Symbol Use standard pseudopotentials discussed above.

RECOMMENDATION:

Pseudopotentials are recommended for first row transition metals and heavier elements. Consult the reviews for more details.

### 8.2.2 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. There is also a built-in combination that can be invoked specifying “ECP=LACVP”. It assigns automatically 6-31G\* or other suitable type basis sets for atoms H-Ar, while uses LANL2DZ for heavier atoms.

### 8.2.3 Examples

**Example 8.1** Computing the HF/LANL2DZ energy of AgCl at a bond length of 2.4 Å.

```
$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 8.2** Computing the HF geometry of CdBr<sub>2</sub> 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.

```
$molecule
  0 1
  Cd
  Br1 Cd r
  Br2 Cd r Br1 180

  r = 2.4
$end

$rem
  JOBTYP E 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
```

## 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,  $\hat{P}_0$ ,  $\hat{P}_1$ , ...,  $\hat{P}_{L-1}$

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*

**Note:** (1) All of the information in the *\$ecp* block is case-insensitive.

(2) The  $L$  value may not exceed 4. That is, nothing beyond  $G$  projectors is allowed.

(3) The power of  $r$  (which includes the Jacobian  $r^2$  factor) must be 0, 1 or 2.

### 8.3.2 Example

**Example 8.3** Optimizing the HF geometry of  $\text{AlH}_3$  using a user-defined ECP and basis set on Al and the 3-21G basis on H.

```

$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
0.90110 -0.30377 -0.07929
0.44950 0.13382 0.16540
0.14050 0.76037 0.53015
SP 1 1.00
0.04874 0.32232 0.47724
****
H
3-21G
****
$end

```

## 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 a SG-1/SG-0 hybrid (see section 4.3.4) which was originally defined only for the elements up to argon. In Q-CHEM 2.0 and above, 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

**Example 8.4** Optimization of the structure of  $\text{XeF}_5^+$  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.

```

$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
```

## 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

**Example 8.5** Optimization of the structure of Se<sub>8</sub> using HF/LANL2DZ, followed by a single-point energy calculation at the MP2/LANL2DZ level.

```
$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
  EXCHANGE    hf
  ECP          lanl2dz
$end

@@@

$molecule
  read
$end
```



```
$rem
  JOBTYP      sp      Single-point energy
  CORRELATION mp2     MP2 correlation energy
  ECP         lan12dz Hay-Wadt ECP and basis
  SCF_GUESS   read    Read in the MOs
$end
```

## 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 3.0 cannot calculate analytic second derivatives of the nuclear potential-energy term when ECP's are used, and must therefore resort to finite difference methods. However, for HF and DFT calculations, it can compute analytic second derivatives for all other terms in the Hamiltonian. The program takes full advantage of this by only computing the potential-energy derivatives numerically, and adding these to the analytically calculated second derivatives of the remaining energy terms.

There is a significant speed advantage associated with this approach as, at each finite-difference step, only the potential-energy term needs to be calculated. This term requires only three-center integrals, which are far fewer in number and much cheaper to evaluate than the four-center, two-electron integrals associated with the electron-electron interaction terms. Readers are referred to Table 9.1 for a full list of the analytic derivative capabilities of Q-CHEM.

### 8.6.1 Example

**Example 8.6** Structure and vibrational frequencies of TeO<sub>2</sub> 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. Note also that only the second derivatives of the potential-energy term will be calculated by finite-difference, all other terms will be calculated analytically.

```
$molecule
  0 1
  Te
  01 Te r
  02 Te r 01 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
```

### 8.6.2 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.

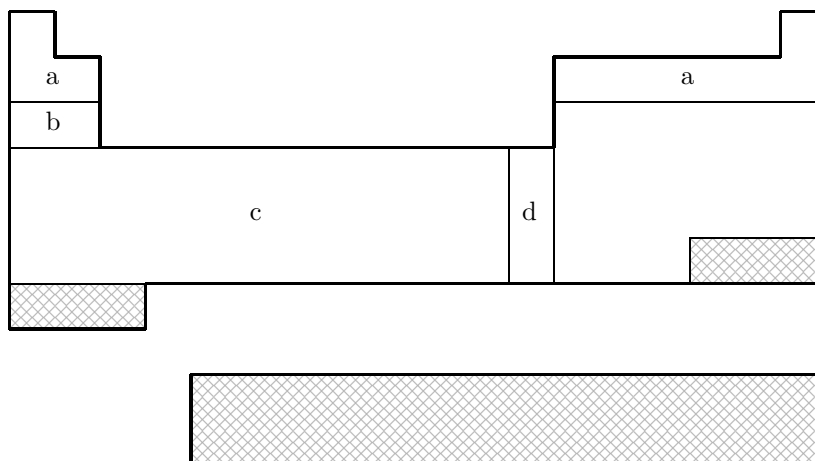
The following information is provided for each ECP:

- 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 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 the 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.

## 8.6.3 The HWMB Pseudopotential at a Glance

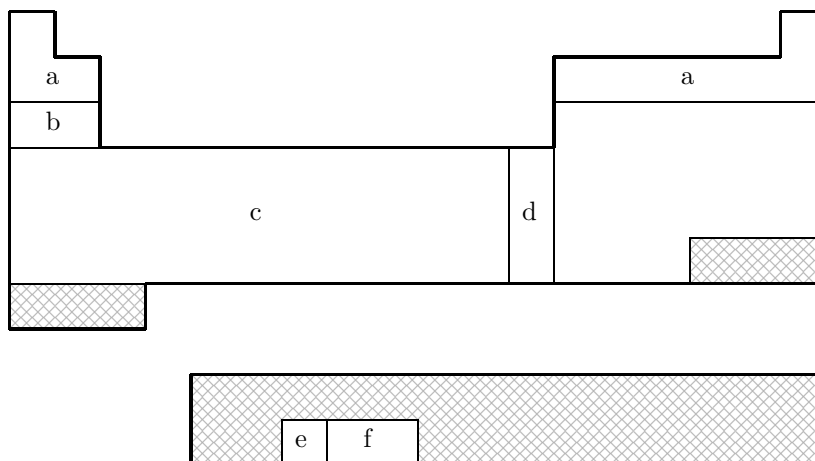


*HWMB is **not** available for shaded elements*

- (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]	<i>P</i>	(1s,1p)
K-Ca	[Ne]	<i>P</i>	(2s,1p)
Sc-Cu	[Ne]	<i>P</i>	(2s,1p,1d)
Zn	[Ar]	<i>D</i>	(1s,1p,1d)
Ga-Kr	[Ar]+3d	<i>D</i>	(1s,1p)
Rb-Sr	[Ar]+3d	<i>D</i>	(2s,1p)
Y-Ag	[Ar]+3d	<i>D</i>	(2s,1p,1d)
Cd	[Kr]	<i>D</i>	(1s,1p,1d)
In-Xe	[Kr]+4d	<i>D</i>	(1s,1p)
Cs-Ba	[Kr]+4d	<i>D</i>	(2s,1p)
La	[Kr]+4d	<i>D</i>	(2s,1p,1d)
Hf-Au	[Kr]+4d+4f	<i>F</i>	(2s,1p,1d)
Hg	[Xe]+4f	<i>F</i>	(1s,1p,1d)
Tl-Bi	[Xe]+4f+5d	<i>F</i>	(1s,1p)

## 8.6.4 The LANL2DZ Pseudopotential at a Glance

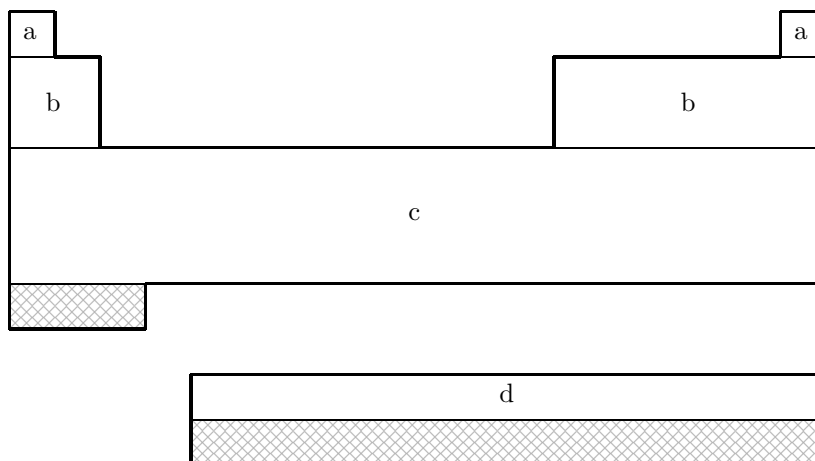


*LANL2DZ is **not** available for shaded elements*

- (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]	<i>P</i>	(2s,2p)
K–Ca	[Ne]	<i>P</i>	(3s,3p)
Sc–Cu	[Ne]	<i>P</i>	(3s,3p,2d)
Zn	[Ar]	<i>D</i>	(2s,2p,2d)
Ga–Kr	[Ar]+3d	<i>D</i>	(2s,2p)
Rb–Sr	[Ar]+3d	<i>D</i>	(3s,3p)
Y–Ag	[Ar]+3d	<i>D</i>	(3s,3p,2d)
Cd	[Kr]	<i>D</i>	(2s,2p,2d)
In–Xe	[Kr]+4d	<i>D</i>	(2s,2p)
Cs–Ba	[Kr]+4d	<i>D</i>	(3s,3p)
La	[Kr]+4d	<i>D</i>	(3s,3p,2d)
Hf–Au	[Kr]+4d+4f	<i>F</i>	(3s,3p,2d)
Hg	[Xe]+4f	<i>F</i>	(2s,2p,2d)
Tl	[Xe]+4f+5d	<i>F</i>	(2s,2p,2d)
Pb–Bi	[Xe]+4f+5d	<i>F</i>	(2s,2p)
U–Pu	[Xe]+4f+5d	<i>F</i>	(3s,3p,2d,2f)

## 8.6.5 The SBKJC Pseudopotential at a Glance

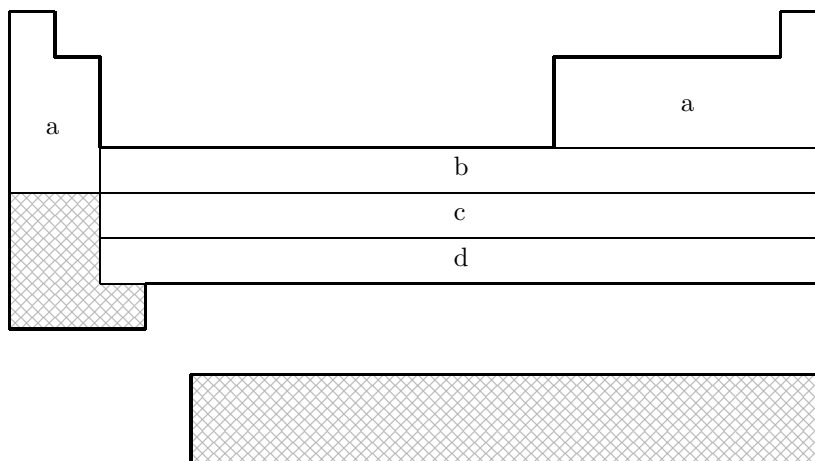


*SBKJC is **not** available for shaded elements*

- (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]	<i>S</i>	(2s,2p)
Na-Ar	[Ne]	<i>P</i>	(2s,2p)
K-Ca	[Ar]	<i>P</i>	(2s,2p)
Sc-Ga	[Ne]	<i>P</i>	(4s,4p,3d)
Ge-Kr	[Ar]+3d	<i>D</i>	(2s,2p)
Rb-Sr	[Kr]	<i>D</i>	(2s,2p)
Y-In	[Ar]+3d	<i>D</i>	(4s,4p,3d)
Sn-Xe	[Kr]+4d	<i>D</i>	(2s,2p)
Cs-Ba	[Xe]	<i>D</i>	(2s,2p)
La	[Kr]+4d	<i>F</i>	(4s,4p,3d)
Ce-Lu	[Kr]+4d	<i>D</i>	(4s,4p,1d,1f)
Hf-Tl	[Kr]+4d+4f	<i>F</i>	(4s,4p,3d)
Pb-Rn	[Xe]+4f+5d	<i>F</i>	(2s,2p)

## 8.6.6 The CRENBS Pseudopotential at a Glance

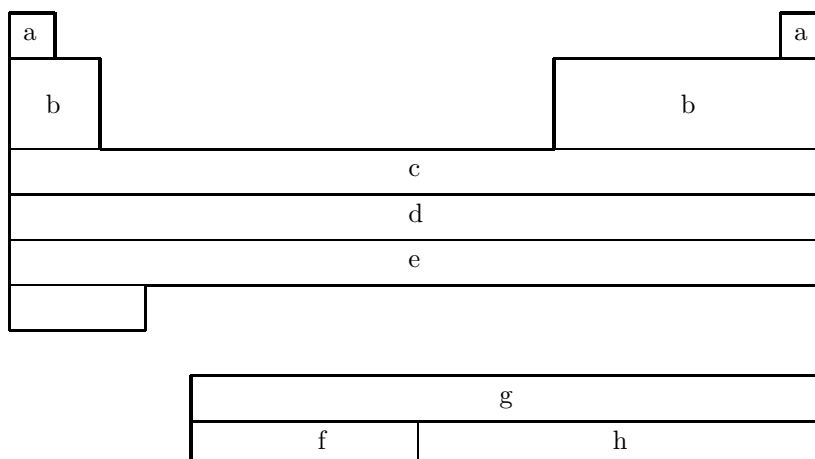


CRENBS is **not** available for shaded elements

- (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]	<i>P</i>	(1s,0p,1d)
Ga-Kr	[Ar]+3d	<i>D</i>	(1s,1p)
Y-Cd	[Kr]	<i>D</i>	(1s,1p,1d)
In-Xe	[Kr]+4d	<i>D</i>	(1s,1p)
La	[Xe]	<i>D</i>	(1s,1p,1d)
Hf-Hg	[Xe]+4f	<i>F</i>	(1s,1p,1d)
Tl-Rn	[Xe]+4f+5d	<i>F</i>	(1s,1p)

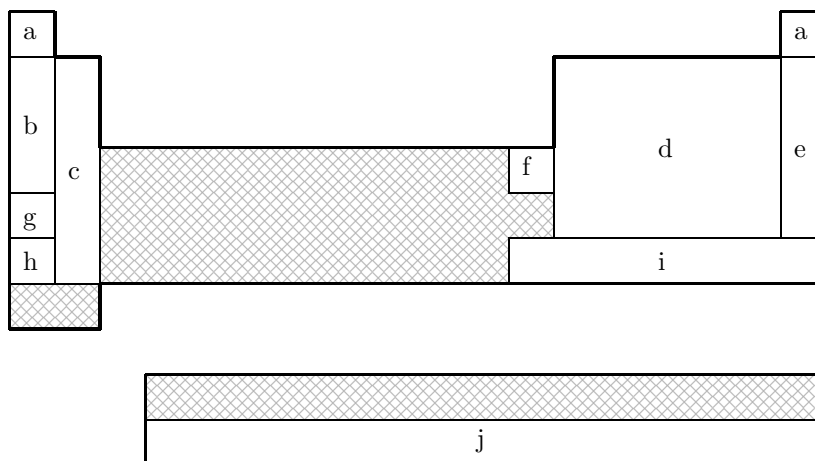
## 8.6.7 The CRENBL Pseudopotential at a Glance



- (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)

## 8.6.8 The SRLC Pseudopotential at a Glance



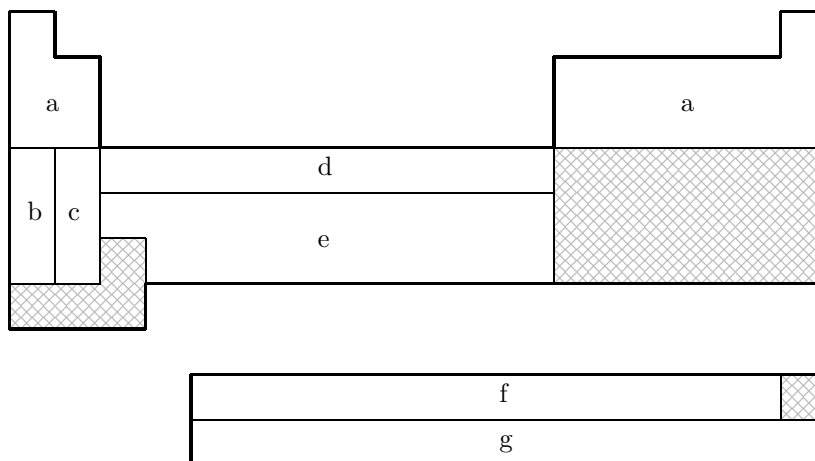
*SRLC is **not** available for shaded elements*

- (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]	<i>P</i>	(2s,2p)
B-N	[He]	<i>D</i>	(2s,2p)
O-F	[He]	<i>D</i>	(2s,3p)
Ne	[He]	<i>D</i>	(4s,4p,3d,1f)
Na-P	[Ne]	<i>D</i>	(2s,2p)
S-Cl	[Ne]	<i>D</i>	(2s,3p)
Ar	[Ne]	<i>F</i>	(4s,4p,3d,1f)
K-Ca	[Ar]	<i>D</i>	(2s,2p)
Zn	[Ar]+3d	<i>D</i>	(3s,2p)
Ga-As	[Ar]+3d	<i>F</i>	(2s,2p)
Se-Br	[Ar]+3d	<i>F</i>	(2s,3p)
Kr	[Ar]+3d	<i>G</i>	(4s,4p,3d,1f)
Rb-Sr	[Kr]	<i>D</i>	(2s,2p)
In-Sb	[Kr]+4d	<i>F</i>	(2s,2p)
Te-I	[Kr]+4d	<i>F</i>	(2s,3p)
Xe	[Kr]+4d	<i>G</i>	(4s,4p,3d,1f)
Cs-Ba	[Xe]	<i>D</i>	(2s,2p)
Hg-Bi	[Xe]+4f+5d	<i>G</i>	(2s,2p,1d)
Po-At	[Xe]+4f+5d	<i>G</i>	(2s,3p,1d)
Rn	[Xe]+4f+5d	<i>G</i>	(2s,2p,1d)
Ac-Lr	[Xe]+4f+5d	<i>G</i>	(5s,5p,4d,3f,2g)

## 8.6.9 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]	<i>F</i>	(5s,4p)
Ca	[Ne]	<i>F</i>	(4s,4p,2d)
Sc–Zn	[Ne]	<i>D</i>	(6s,5p,3d)
Rb	[Ar]+3d	<i>F</i>	(5s,4p)
Sr	[Ar]+3d	<i>F</i>	(4s,4p,2d)
Y–Cd	[Ar]+3d	<i>F</i>	(6s,5p,3d)
Cs	[Kr]+4d	<i>F</i>	(5s,4p)
Ba	[Kr]+4d	<i>F</i>	(3s,3p,2d,1f)
Ce–Yb	[Ar]+3d	<i>G</i>	(5s,5p,4d,3f)
Hf–Pt	[Kr]+4d+4f	<i>G</i>	(6s,5p,3d)
Au	[Kr]+4d+4f	<i>F</i>	(7s,3p,4d)
Hg	[Kr]+4d+4f	<i>G</i>	(6s,6p,4d)
Ac–Lr	[Kr]+4d+4f	<i>G</i>	(8s,7p,6d,4f)

# References and Further Reading

- [1] Ground State Methods (Chapters 4 and 5)
- [2] Basis Sets (Chapter 7)
- [3] P. A. Christiansen, W. C. Ermler and K. S. Pitzer, *Ann. Rev. Phys. Chem.*, **36**, 407, (1985).
- [4] P. Pyykko, *Chem. Rev.* **88**, 563, (1988).
- [5] M. S. Gordon, and T. R. Cundari, *Coord. Chem. Rev.*, **147**, 87, (1996).
- [6] See articles by G. Frenking et al, T. R. Cundari et al, and J. Almlöf and O. Gropen, in *Reviews in Computational Chemistry*, volume 8, edited by K. B. Lipkowitz and D. B. Boyd (Wiley-VCH, 1996).
- [7] L. R. Kahn and W. A. Goddard III, *J. Chem. Phys.*, **56**, 2685, (1972).
- [8] 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 for transition structures. Both equilibrium and transition structures are stationary points and therefore the energy gradients will vanish. Characterization 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 optimization features currently available through Jon Baker’s OPTIMIZE package (see Appendix A), 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 maximize the rate of convergence. Users are able to override these decisions, but in general, this is not recommended.

Another consideration when trying to minimize the optimization time concerns the quality of the gradient and Hessian. 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	✓	<i>h</i>	✓	<i>f</i>
HF	✓	<i>h</i>	✓	<i>f</i>
ROHF	✓	<i>h</i>	✗	
MP2	✓	<i>h</i>	✗	
(V)OD	✓	<i>h</i>	✗	
(V)QCCD	✓	<i>h</i>	✗	
CIS (except RO)	✓	<i>h</i>	✓	<i>f</i>
CFMM	✓	<i>h</i>	✗	

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

### 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
- Reaction pathways using intrinsic reaction coordinates (IRC)

### 9.2.2 Job Control

**Note:** Users input starting geometry through the *\$molecule* keyword.

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.

TYPE:

STRING

DEFAULT:

Default is single-point, which should be changed to one of the following options.

OPTIONS:

OPT      Equilibrium structure optimization.

TS        Transition structure optimization.

RPATH    Intrinsic reaction path following.

RECOMMENDATION:

Application-dependent.

**GEOM\_OPT\_HESSIAN**

Determines the initial Hessian status.

TYPE:

STRING

DEFAULT:

DIAGONAL

OPTIONS:

DIAGONAL    Set up diagonal Hessian.

READ        Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

RECOMMENDATION:

An accurate initial Hessian will improve the performance of the optimizer, but is expensive to compute.

**GEOM\_OPT\_COORDS**

Controls the type of optimization coordinates.

TYPE:

INTEGER

DEFAULT:

-1

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.

TYPE:

INTEGER

DEFAULT:

300  $\equiv 300 \times 10^{-6}$  tolerance on maximum gradient component.

OPTIONS:

$n$  Integer value (tolerance =  $n \times 10^{-6}$ ).

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.

TYPE:

INTEGER

DEFAULT:

1200  $\equiv 1200 \times 10^{-6}$  tolerance on maximum atomic displacement.

OPTIONS:

$n$  Integer value (tolerance =  $n \times 10^{-6}$ ).

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 optimization cycles.

TYPE:

INTEGER

DEFAULT:

100  $\equiv 100 \times 10^{-8}$  tolerance on maximum gradient component.

OPTIONS:

$n$  Integer value (tolerance = value  $n \times 10^{-8}$ ).

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 optimization cycles.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

$n$  User defined positive integer.

RECOMMENDATION:

The default should be sufficient for most cases. Increase if the initial guess geometry is poor, or for systems with shallow potential wells.

**GEOM\_OPT\_PRINT**

Controls the amount of OPTIMIZE print output.

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

1

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 a transition state search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Mode following off.

$n$  Maximize along mode  $n$ .

RECOMMENDATION:

Use default, for geometry optimizations.



**GEOM\_OPT\_MAX\_DIIS**

Controls maximum size of subspace for GDIIS.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use GDIIS.

-1 Default size =  $\min(\text{NDEG}, \text{NATOMS}, 4)$  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  $10^{-3}$ .

TYPE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

$n$  User-defined cutoff.

RECOMMENDATION:

Use default.

**GEOM\_OPT\_UPDATE**

Controls the Hessian update algorithm.

TYPE:

INTEGER

DEFAULT:

-1

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).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

*n* User-defined level.

RECOMMENDATION:

Use default.

**FDIFF\_STEPSIZE**

Displacement used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 0.001 Å.

OPTIONS:

*n* Use a step size of  $n \times 10^{-5}$ .

RECOMMENDATION:

Use default, unless on a very flat potential, in which case a larger value should be used.

## 9.2.4 Example

**Example 9.1** 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

```
$molecule
  0 1
  0
  H 1 oh
  H 1 oh 2 hoh

  oh = 1.1
  hoh = 104
$end

$rem
  JOBTYP  freq  Calculate an analytic Hessian
  EXCHANGE hf
  BASIS    6-31g(d)
$end

$comment
Now proceed with the Optimization making sure to read in the analytic
Hessian (use other available information too).
$end

@@@

$molecule
  read
$end
```

```

$rem
  JOBTYP      opt
  EXCHANGE    hf
  BASIS       6-31g(d)
  SCF_GUESS   read
  GEOM_OPT_HESSIAN  read   Have the initial Hessian
$end

```

## 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*

Values in angstroms;  $value > 0$ :

```
stre  atom1  atom2  value
```

#### *angles*

Values in degrees,  $0 \leq value \leq 180$ ; *atom2* is the middle atom of the bend:

```
bend  atom1  atom2  atom3  value
```

#### *out-of-plane-bends*

Values in degrees,  $-180 \leq value \leq 180$  *atom2*; angle between *atom4* and the *atom1-atom2-atom3* plane:

```
outp  atom1  atom2  atom3  atom4  value
```

#### *dihedral angles*

Values in degrees,  $-180 \leq value \leq 180$ ; angle the plane *atom1-atom2-atom3* makes with the plane *atom2-atom3-atom4*:

```
tors  atom1  atom2  atom3  atom4  value
```

***coplanar bends***

Values in degrees,  $-180 \leq \text{value} \leq 180$ ; bending of *atom1-atom2-atom3* in the plane *atom2-atom3-atom4*:

```
linc  atom1  atom2  atom3  atom4  value
```

***perpendicular bends***

Values in degrees,  $-180 \leq \text{value} \leq 180$ ; 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 up to seven real atoms in the defining list.
2. Positioned a unit distance along the normal to a plane defined by three atoms, centered 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 seven 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.

**Note:** 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  $0^\circ$  or  $\pm 180^\circ$ . This is because the corresponding constraint normals are zero vectors. Also, dihedral constraints near these two limiting values (within, say  $20^\circ$ ) 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  $180^\circ$  can be redefined to two constraints of  $90^\circ$  with respect to a suitably positioned dummy atom. The same thing can be done with a  $180^\circ$  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  $0^\circ$  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

**Example 9.2** Methanol geometry optimization with constraints.

```
$comment
  Methanol geom opt with constraints in bond length and bond angles.
$end

$molecule
  O  1
  C  0.14192  0.33268  0.00000
  O  0.14192 -1.08832  0.00000
  H  1.18699  0.65619  0.00000
  H -0.34843  0.74268  0.88786
  H -0.34843  0.74268 -0.88786
  H -0.77395 -1.38590  0.00000
$end

$rem
  GEOM_OPT_PRINT  6
  JOBTYP          opt
  EXCHANGE        hf
  BASIS           3-21g
$end

$opt
  CONSTRAINT
  stre 1 6 1.8
```

```
bend 2 1 4 110.0
bend 2 1 5 110.0
ENDCONSTRAINT
$end
```

### 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 Intrinsic Reaction Coordinates

The concept of a reaction path, although seemingly well-defined chemically (*i.e.*, how the atoms in the system move to get from reactants to products), is somewhat ambiguous mathematically because, using the usual definitions, it depends on the coordinate system. Stationary points on a potential energy surface are *independent* of coordinates, but the path connecting them is not, and so different coordinate systems will produce different reaction paths. There are even different definitions of what constitutes a “reaction path”; the one used in Q-CHEM is based on the intrinsic reaction coordinate (IRC), first defined in this context by Fukui [2]. This is essentially a series of steepest descent paths going downhill from the transition state.

The reaction path is most unlikely to be a straight line and so by taking a finite step length along the direction of the gradient you will leave the “true” path. A series of small steepest descent steps will zig-zag along the actual reaction path (this is known as “stitching”). Ishida *et al.* [3] developed a predictor-corrector algorithm, involving a second gradient calculation after the initial steepest descent step, followed by a line search along the gradient bisector to get back on the path; this was subsequently improved by Schmidt *et al.* [4], and is the method we have adopted. For the first step downhill from the transition state this approach cannot be used (as the gradient is zero); instead a step is taken along the Hessian mode corresponding to the imaginary frequency.

The reaction path can be defined and followed in Z-matrix coordinates, Cartesian coordinates or mass-weighted Cartesians. The latter represents the “true” IRC as defined by Fukui [2]. However, if the main reason for following the reaction path is simply to determine which minima a given transition state connects (perhaps the major use), then it doesn’t matter which coordinates are used. In order to use the IRC code the transition state geometry and the exact Hessian must be available. These must be computed via transition state (JOBTYPE = TS) and frequency calculation (JOBTYPE = FREQ) respectively.

### 9.4.1 Job control

An IRC calculation is invoked by setting the JOBTYPE *\$rem* to RPATH.

#### RPATH\_COORDS

Determines which coordinate system to use in the IRC search.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use mass-weighted coordinates.

2 Use Z-matrix coordinates.

RECOMMENDATION:

Use default.

#### RPATH\_DIRECTION

Determines the direction of the eigen mode to follow. This will not usually be known prior to the Hessian diagonalization.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Descend in the positive direction of the eigen mode.

-1 Descend in the negative direction of the eigen mode.

RECOMMENDATION:

It is usually not possible to determine in which direction to go *a priori*, and therefore both directions will need to be considered.



**RPATH\_MAX\_CYCLES**

Specifies the maximum number of points to find on the reaction path.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

$n$  User-defined number of cycles.

RECOMMENDATION:

Use more points if the minimum is desired, but not reached using the default.

**RPATH\_MAX\_STEPSIZE**

Specifies the maximum step size to be taken (in thousandths of *a.u.*).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 *a.u.*.

OPTIONS:

$n$  Step size =  $n/1000$ .

RECOMMENDATION:

None.

**RPATH\_TOL\_DISPLACEMENT**

Specifies the convergence threshold for the step. If a step size is chosen by the algorithm that is smaller than this, the path is deemed to have reached the minimum.

TYPE:

INTEGER

DEFAULT:

5000 Corresponding to 0.005 *a.u.*

OPTIONS:

$n$  User-defined. Tolerance =  $n/1000000$ .

RECOMMENDATION:

None.

**RPATH\_PRINT**

Specifies the print output level.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

$n$

RECOMMENDATION:

Use default, little additional information is printed at higher levels. Most of the output arises from the multiple single point calculations that are performed along the reaction pathway.

## 9.4.2 Example

### Example 9.3

```
$molecule
  O 1
  C
  H 1 1.20191
  N 1 1.22178 2 72.76337
$end

$rem
  JOBTYP E    freq
  BASIS      sto-3g
  EXCHANGE   hf
$end

@@@

$molecule
  read
$end

$rem
  JOBTYP E    rpath
  BASIS      sto-3g
  EXCHANGE   hf
  SCF_GUESS  read
  RPATH_MAX_CYCLES 30
$end
```

## 9.5 The Growing String Method

One of the difficulties associated with a saddle point search is to obtain a good initial guess for the starting configuration that can later be used for a local surface walking algorithm. This difficulty becomes especially relevant for large systems, where the search space dimensionality is high and the initial starting configuration is often far away from the final saddle point. A very promising method for finding a good guess of the saddle point configuration and the minimum energy pathway connecting reactant and product states are interpolation algorithms. For example, the nudged elastic band method [5,6] and the string method [7] start from a certain initial reaction pathway connecting the reactant and the product state, and then optimize in discretized path space towards the minimum energy pathway. The highest energy point on the so obtained approximation to the minimum energy pathway becomes a good initial guess for the saddle point configuration that can subsequently be used with any local surface walking algorithm.

Inevitably, the performance of an interpolation method heavily relies on the choice of the initial reaction pathway, and a poorly chosen initial pathway can cause slow convergence, or convergence to an incorrect pathway. The growing string method [8] offered an elegant solution to this problem, in which two string fragments (one from the reactant and the other from the product state) were grown until the two fragments join. The growing string method has been implemented within Q-CHEM.

## 9.6 Improved Dimer Method

Once a good approximation to the minimum energy pathway is obtained, *e.g.*, with the help of an interpolation algorithm such as the growing string method, local surface walking algorithms can be used to determine the exact location of the saddle point. Baker's partitioned rational function optimization (P-RFO) method, which utilizes an approximate or exact Hessian, has proven to be a very powerful method for this purpose.

The dimer method [9] on the other hand, is a mode following algorithm that utilizes only the curvature along one direction in configuration space (rather than the full Hessian) and requires only gradient evaluations. It is therefore especially applicable for large systems where a full Hessian calculation is very time consuming, or for saddle point searches where the eigenvector of the lowest Hessian eigenvalue of the starting configuration does not correspond to the reaction coordinate. A recent modification of this method has been developed [10,11] to significantly reduce the influence of numerical noise, as it is common in quantum chemical methods, on the performance of the dimer algorithm, and to significantly reduce its computational cost. This improved dimer method has recently been implemented within Q-CHEM.

## 9.7 *Ab initio* Molecular Dynamics

Q-CHEM can propagate classical molecular dynamics trajectories on the Born-Oppenheimer potential energy surface generated by a particular theoretical model chemistry (*e.g.*, B3LYP/6-31G\*). This procedure, in which the forces on the nuclei are evaluated on-the-fly, is known variously as “direct dynamics”, “*ab initio* molecular dynamics”, or “Born-Oppenheimer molecular dynamics” (BOMD). In its most straightforward form, a BOMD calculation consists of an energy + gradient calculation at each molecular dynamics time step, and thus each time step is comparable in cost to one geometry optimization step. A BOMD calculation may be requested using any SCF energy + gradient method available in Q-CHEM, including excited-state gradients; however, methods lacking analytic gradients will be prohibitively expensive, except for very small systems.

Initial Cartesian coordinates and velocities must be specified for the nuclei. Coordinates are specified in the *\$molecule* section as usual, while velocities can be specified using a *\$velocity* section with the form:

```
$velocity
vx,1 vy,1 vz,1
vx,2 vy,2 vz,2
vx,N vy,N vz,N
$end
```

Here  $v_{x,i}$ ,  $v_{y,i}$ , and  $v_{z,i}$  are the  $x$ ,  $y$ , and  $z$  Cartesian velocities of the  $i$ th nucleus, specified in atomic units (bohrs per *a.u.* of time, where 1 *a.u.* of time is approximately 0.0242 fs). The *\$velocity* section thus has the same form as the *\$molecule* section, but without atomic symbols and without the line specifying charge and multiplicity. The atoms must be ordered in the same manner in both the *\$velocity* and *\$molecule* sections.

As an alternative to a *\$velocity* section, initial nuclear velocities can be sampled from certain distributions (*e.g.*, Maxwell-Boltzmann), using the AIMD\_INIT\_VELOC variable described below.

Although the Q-CHEM output file dutifully records the progress of any *ab initio* molecular dynamics job, the most useful information is printed not to the main output file but rather to a directory called “AIMD” that is a subdirectory of the job’s scratch directory. (All *ab initio* molecular dynamics jobs should therefore use the “-save” option.) The AIMD directory consists of a set of files that record, in ASCII format, one line of information at each time step. Each file contains a few comment lines (indicated by “#”) that describe its contents and which we summarize in the list below.

- Cost: Records the number of SCF cycles, the total cpu time, and the total memory use at each dynamics step.
- EComponents: Records various components of the total energy (all in Hartrees).

- Energy: Records the total energy and fluctuations therein.
- MulMoments: If multipole moments are requested, they are printed here.
- NucCarts: Records the nuclear Cartesian coordinates  $x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N$  at each time step, in either bohrs or angstroms.
- NucForces: Records the Cartesian forces on the nuclei at each time step (same order as the coordinates, but given in atomic units).
- NucVeloc: Records the Cartesian velocities of the nuclei at each time step (same order as the coordinates, but given in atomic units).
- TandV: Records the kinetic and potential energy, as well as fluctuations in each.

For ELMD jobs, there are other output files as well:

- ChangeInF: Records the matrix norm and largest magnitude element of  $\Delta \mathbf{F} = \mathbf{F}(t + \delta t) - \mathbf{F}(t)$  in the basis of Cholesky-orthogonalized AOs. The files ChangeInP, ChangeInL, and ChangeInZ provide analogous information for the density matrix  $\mathbf{P}$  and the Cholesky orthogonalization matrices  $\mathbf{L}$  and  $\mathbf{Z}$  defined in [12].
- DeltaNorm: Records the norm and largest magnitude element of the curvy-steps rotation angle matrix  $\Delta$  defined in [12]. Matrix elements of  $\Delta$  are the dynamical variables representing the electronic degrees of freedom. The output file DeltaDotNorm provides the same information for the electronic velocity matrix  $d\Delta/dt$ .
- ElecGradNorm: Records the norm and largest magnitude element of the electronic gradient matrix  $\mathbf{F}\mathbf{P} - \mathbf{P}\mathbf{F}$  in the Cholesky basis.
- dTfict: Records the instantaneous time derivative of the fictitious kinetic energy at each time step, in atomic units.

*Ab initio* molecular dynamics jobs are requested by specifying `JOBTYPE = AIMD`. Initial velocities must be specified either using a *\$velocity* section or via the `AIMD_INIT_VELOC` keyword described below. In addition, the following *\$rem* variables must be specified for any *ab initio* molecular dynamics job:

#### **AIMD.METHOD**

Selects an *ab initio* molecular dynamics algorithm.

TYPE:

STRING

DEFAULT:

BOMD

OPTIONS:

BOMD Born-Oppenheimer molecular dynamics.

CURVY Curvy-steps Extended Lagrangian molecular dynamics.

RECOMMENDATION:

BOMD yields exact classical molecular dynamics, provided that the energy is tolerably conserved. ELMD is an approximation to exact classical dynamics whose validity should be tested for the properties of interest.

**TIME\_STEP**

Specifies the molecular dynamics time step, in atomic units (1 *a.u.* = 0.0242 fs).

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

Smaller time steps lead to better energy conservation; too large a time step may cause the job to fail entirely. Make the time step as large as possible, consistent with tolerable energy conservation.

**AIMD\_STEPS**

Specifies the requested number of molecular dynamics steps.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

None.

*Ab initio* molecular dynamics calculations can be quite expensive, and thus Q-CHEM includes several algorithms designed to accelerate such calculations. At the self-consistent field (Hartree-Fock and DFT) level, BOMD calculations can be greatly accelerated by using information from previous time steps to construct a good initial guess for the new molecular orbitals or Fock matrix, thus hastening SCF convergence. A Fock matrix extrapolation procedure [13], based on a suggestion by Pulay and Fogarasi [14], is available for this purpose.

The Fock matrix elements  $\mathbf{F}_{\mu\nu}$  in the atomic orbital basis are oscillatory functions of the time  $t$ , and Q-CHEM's extrapolation procedure fits these oscillations to a power series in  $t$ ,

$$\mathbf{F}_{\mu\nu}(t) = \sum_{n=0}^N c_n t^n \quad (9.1)$$

The  $N+1$  extrapolation coefficients  $c_n$  are determined by a fit to a set of  $M$  Fock matrices retained from previous time steps. Fock matrix extrapolation can significantly reduce the number of SCF iterations required at each time step, but for low-order extrapolations, or if SCF\_CONVERGENCE is set too small, a systematic drift in the total energy may be observed. Benchmark calculations testing the limits of energy conservation can be found in Ref. [13], and demonstrate that numerically exact classical dynamics (without energy drift) can be obtained at significantly reduced cost.

Fock matrix extrapolation is requested by specifying values for  $N$  and  $M$ , as in the form of the following two *\$rem* variables:

**FOCK\_EXTRAP\_ORDER**

Specifies the polynomial order  $N$  for Fock matrix extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

$N$  Extrapolate using an  $N$ th-order polynomial ( $N > 0$ ).

RECOMMENDATION:

None

**FOCK\_EXTRAP\_POINTS**

Specifies the number  $M$  of old Fock matrices that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

$M$  Save  $M$  Fock matrices for use in extrapolation ( $M > N$ )

RECOMMENDATION:

Higher-order extrapolations with more saved Fock matrices are faster and conserve energy better than low-order extrapolations, up to a point. In many cases, the scheme ( $N = 6$ ,  $M = 12$ ), in conjunction with `SCF_CONVERGENCE = 6`, is found to provide about a 50% savings in computational cost while still conserving energy.

Assuming decent conservation, a BOMD calculation represents exact classical dynamics on the Born–Oppenheimer potential energy surface. In contrast, so-called extended Lagrangian molecular dynamics (ELMD) methods make an approximation to exact classical dynamics in order to expedite the calculations. ELMD methods—of which the most famous is Car–Parrinello molecular dynamics—introduce a fictitious dynamics for the electronic (orbital) degrees of freedom, which are then *propagated* alongside the nuclear degrees of freedom, rather than *optimized* at each time step as they are in a BOMD calculation. The fictitious electronic dynamics is controlled by a fictitious mass parameter  $\mu$ , and the value of  $\mu$  controls both the accuracy and the efficiency of the method. In the limit of small  $\mu$  the nuclei and the orbitals propagate adiabatically, and ELMD mimics true classical dynamics. Larger values of  $\mu$  slow down the electronic dynamics, allowing for larger time steps (and more computationally efficient dynamics), at the expense of an ever-greater approximation to true classical dynamics.

Q-CHEM’s ELMD algorithm is based upon propagating the density matrix, expressed in a basis of atom-centered Gaussian orbitals, along shortest-distance paths (geodesics) of the manifold of allowed density matrices  $\mathbf{P}$ . Idempotency of  $\mathbf{P}$  is maintained at every time step, by construction, and thus our algorithm requires neither density matrix purification, nor iterative solution for Lagrange multipliers (to enforce orthogonality of the molecular orbitals). We call this procedure “curvy steps” ELMD [12], and in a sense it is a time-dependent implementation of the GDM algorithm (Section 4.6) for converging SCF single-point calculations.

The extent to which ELMD constitutes a significant approximation to BOMD continues to be debated. When assessing the accuracy of ELMD, the primary consideration is whether there exists a separation of time scales between nuclear oscillations, whose time scale  $\tau_{\text{nuc}}$  is set by the period of the fastest vibrational frequency, and electronic oscillations, whose time scale  $\tau_{\text{elec}}$  may be estimated according to [12]

$$\tau_{\text{elec}} \geq \sqrt{\mu/(\varepsilon_{\text{LUMO}} - \varepsilon_{\text{HOMO}})} \quad (9.2)$$

A conservative estimate, suggested in [12], is that essentially exact classical dynamics is attained when  $\tau_{\text{nuc}} > 10 \tau_{\text{elec}}$ . In practice, we recommend careful benchmarking to insure that ELMD faithfully reproduces the BOMD observables of interest.

Due to the existence of a fast time scale  $\tau_{\text{elec}}$ , ELMD requires smaller time steps than BOMD. When BOMD is combined with Fock matrix extrapolation to accelerate convergence, it is no longer clear that ELMD methods are substantially more efficient, at least in Gaussian basis sets [13, 14].

The following *\$rem* variables are required for ELMD jobs:

#### **AIMD\_FICT\_MASS**

Specifies the value of the fictitious electronic mass  $\mu$ , in atomic units, where  $\mu$  has dimensions of (energy)  $\times$  (time)<sup>2</sup>.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified

RECOMMENDATION:

Values in the range of 50–200 *a.u.* have been employed in test calculations; consult [12] for examples and discussion.

Additional job control variables for *ab initio* molecular dynamics.

#### **AIMD\_INIT\_VELOC**

Specifies the method for selecting initial nuclear velocities.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

**THERMAL** Random sampling of nuclear velocities from a Maxwell–Boltzmann distribution. The user must specify the temperature in Kelvin via the *\$rem* variable AIMD\_TEMP.

**ZPE** Choose velocities in order to put zero–point vibrational energy into each normal mode, with random signs. This option requires that a frequency job to be run beforehand.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *\$velocity* section.

#### **AIMD\_MOMENTS**

Requests that multipole moments be output at each time step.

TYPE:

INTEGER

DEFAULT:

0 Do not output multipole moments.

OPTIONS:

*n* Output the first *n* multipole moments.

RECOMMENDATION:

None

**AIMD\_TEMP**

Specifies a temperature (in Kelvin) for Maxwell–Boltzmann velocity sampling.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified number of Kelvin.

RECOMMENDATION:

This variable is only useful in conjunction with `AIMD_INIT_VELOC = THERMAL`.

Note that the simulations are run at constant energy, rather than constant temperature, so the mean nuclear kinetic energy will fluctuate in the course of the simulation.

**DEUTERATE**

Requests that all hydrogen atoms be replaced with deuterium.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not replace hydrogens.

OPTIONS:

TRUE Replace hydrogens with deuterium.

RECOMMENDATION:

Replacing hydrogen atoms reduces the fastest vibrational frequencies by a factor of 1.4, which allow for a larger fictitious mass and time step in ELMD calculations.

There is no reason to replace hydrogens in BOMD calculations.

**9.7.1 Examples**

**Example 9.4** Simulating thermal fluctuations of the water dimer at 298 K.

```

$molecule
  0 1
  O 1.386977 0.011218 0.109098
  H 1.748442 0.720970 -0.431026
  H 1.741280 -0.793653 -0.281811
  O -1.511955 -0.009629 -0.120521
  H -0.558095 0.008225 0.047352
  H -1.910308 0.077777 0.749067
$end

$rem
  JOBTYP  aimd
  AIMD_METHOD  bomd
  EXCHANGE  b3lyp
  BASIS  6-31g*
  TIME_STEP  20      (20 a.u. = 0.48 fs)
  AIMD_STEPS  1000
  AIMD_INIT_VELOC  thermal
  AIMD_TEMP  298
  FOCK_EXTRAP_ORDER  6      request Fock matrix extrapolation
  FOCK_EXTRAP_POINTS  12
$end

```

**Example 9.5** Propagating  $F^-(H_2O)_4$  on its first excited-state potential energy surface, calculated at the CIS level.



```

$molecule
-1 1
O -1.969902 -1.946636 0.714962
H -2.155172 -1.153127 1.216596
H -1.018352 -1.980061 0.682456
O -1.974264 0.720358 1.942703
H -2.153919 1.222737 1.148346
H -1.023012 0.684200 1.980531
O -1.962151 1.947857 -0.723321
H -2.143937 1.154349 -1.226245
H -1.010860 1.980414 -0.682958
O -1.957618 -0.718815 -1.950659
H -2.145835 -1.221322 -1.158379
H -1.005985 -0.682951 -1.978284
F 1.431477 0.000499 0.010220
$end

$rem
JOBTYPE      aimd
AIMD_METHOD  bomd
EXCHANGE     hf
BASIS        6-31+G*
ECP          SRLC
PURECART     1111
CIS_N_ROOTS  3
CIS_TRIPLETS false
CIS_STATE_DERIV 1          propagate on first excited state
AIMD_INIT_VELOC thermal
AIMD_TEMP     150
TIME_STEP     25
AIMD_STEPS    827          (500 fs)
$end

```

**Example 9.6** Simulating vibrations of the NaCl molecule using ELMD.

```

$molecule
0 1
Na 0.000000 0.000000 -1.742298
Cl 0.000000 0.000000 0.761479
$end

$rem
JOBTYPE      freq
EXCHANGE     b3lyp
ECP          sbkjc
$end

@@@

$molecule
read
$end

$rem
JOBTYPE      aimd
EXCHANGE     b3lyp
ECP          sbkjc
TIME_STEP    14
AIMD_STEPS   500
AIMD_METHOD  curvy
AIMD_FICT_MASS 360
AIMD_INIT_VELOC zpe
$end

```

## 9.8 Quantum Mechanics/Molecular Mechanics

Q-CHEM 3.0 introduces the capability of performing QM/MM calculations in conjunction with the CHARMM package. Both software packages are required to perform these calculations, but all the code required for communication between the programs is incorporated in the released versions. QM/MM jobs can be controlled using the following *\$rem* keywords:

### QM\_MM

Turns on the Q-CHEM/CHARMM interface.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do QM/MM calculation through the Q-CHEM/CHARMM interface.

FALSE Turn this feature off.

RECOMMENDATION:

Use default unless running calculations with CHARMM.

### QMMM\_PRINT

Controls the amount of output printed from a QM/MM job.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Limit molecule, point charge, and analysis printing.

FALSE Normal printing.

RECOMMENDATION:

Use default unless running calculations with CHARMM.

### QMMM\_CHARGES

Controls the printing of QM charges to file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Writes a charges.dat file with the Mulliken charges from the QM region.

FALSE No file written.

RECOMMENDATION:

Use default unless running calculations with CHARMM where charges on the QM region need to be saved.

**IGDEFIELD**

Triggers the calculation of the electrostatic potential and/or the electric field at the points given in the file ESPGrid.

TYPE:

INTEGER

DEFAULT:

UNDEFINED

OPTIONS:

- 0 Computes ESP.
- 1 Computes ESP and EFIELD.
- 2 Computes EFIELD.

RECOMMENDATION:

Must use this *\$rem* when IGDESP is specified.

**GEOM\_PRINT**

Controls the amount of geometric information printed at each step.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

- TRUE Prints out all geometric information; bond distances, angles, torsions.
- FALSE Normal printing of distance matrix.

RECOMMENDATION:

Use if you want to be able to quickly examine geometric parameters at the beginning and end of optimizations. Only prints in the beginning of single point energy calculations.

**Example 9.7** Do a basic QM/MM optimization of the water dimer. You need CHARMM to do this but the is the Q-CHEM file that is needed to test the QM/MM functionality. You also need a ESPGrid file with the MM atom positions. These are the bare necessities for a CHARMM/Q-CHEM QM/MM calculation.

```

$molecule
  0 1
  O   -0.91126   1.09227   1.02007
  H   -1.75684   1.51867   1.28260
  H   -0.55929   1.74495   0.36940
$end

$rem
  IGDESP      3          ! Number of MM atoms
  EXCHANGE    hf         ! HF Exchange
  BASIS       cc-pvdz    ! Correlation Consistent Basis
  QM_MM       true       ! Turn on QM/MM calculation
  JOBTYP      force     ! Need this for QM/MM optimizations
  SYMMETRY    off       ! Stop standard rotation of input
  SYM_IGNORE  true       ! Do not use symmetry
  PRINT_INPUT false     ! Turn off input echoing, important for performance
  QMMM_PRINT  true       ! Turn off some analysis that's not needed for QM/MM
$end

$external_charges
  1.20426     -0.64330   0.79922   -0.83400
  1.01723     -1.36906   1.39217   0.41700
  0.43830     -0.06644   0.91277   0.41700
$end

```

# References and Further Reading

- [1] Appendix A on geometry optimization.
- [2] K. Fukui, *J. Phys. Chem*, **74**, 4161, (1970).
- [3] K. Ishida, K. Morokuma and A. Komornicki, *J. Chem. Phys*, **66**, 2153, (1977).
- [4] M. W. Schmidt, M. S. Gordon and M. Dupuis, *J. Am. Chem. Soc*, **107**, 2585, (1985).
- [5] G. Mills and H. Jónsson, *Phys. Rev. Lett.*, **72**, 1124, (1994).
- [6] G. Henkelman and H. Jónsson, *J. Chem. Phys.*, **113**, 9978, (2000).
- [7] E. Weinan, W. Ren and E. Vanden-Eijnden, *Phys. Rev. B*, **66**, 052301, (2002).
- [8] B. Peters, A. Heyden, A. T. Bell and A. Chakraborty, *J. Chem. Phys.*, **120**, 7877, (2004).
- [9] G. Henkelman, H. Jónsson, *J. Chem. Phys.*, **111**, 7010, (1999).
- [10] A. Heyden, B. Peters, A. T. Bell, F. J. Keil, *J. Phys. Chem. B*, **109**, 1857, (2005).
- [11] A. Heyden, A.T. Bell and F.J. Keil, *J. Chem. Phys.*, **123**, 224101, (2005).
- [12] J.M. Herbert and M. Head-Gordon, *J. Chem. Phys.*, **121**, 11542, (2004).
- [13] J.M. Herbert and M. Head-Gordon, *Phys. Chem. Chem. Phys.*, **7**, 3269, (2005).
- [14] P. Pulay and G. Fogarasi, *Chem. Phys. Lett.*, **386**, 272, (2004).
- [15] H.L. Woodcock, M. Hodoscek, A.T.B. Gilbert, P.M.W. Gill, H.F. Schaefer and B.R. Brooks, (in preparation).

## Chapter 10

# Molecular Properties and Analysis

### 10.1 Introduction

Q-CHEM has incorporated a number of molecular properties and wavefunction analysis tools, summarized as follows:

- Chemical solvent models
- Population analysis
- Calculation of molecular intracules
- Vibrational analysis (including isotopic substitution)
- Interface to the Natural Bond Orbital package
- Molecular orbital symmetries
- Multipole moments
- Data generation for 2-D or 3-D plots
- NMR shielding tensors and chemical shifts

### 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 vapor to polar solutions. Although in principle it is possible to model solvation effects upon the solute properties by super-molecular (cluster) calculations (*e.g.*, by averaging over several possible configurations of the first solvation shell), these calculations are very demanding. Furthermore, the super-molecular 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 modeling of chemical reactions and ligand-receptor interactions in aqueous solution. Q-CHEM contains three 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 SS(V)PE model developed by Daniel Chipman from the University of Notre Dame, and the 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 [5] 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  $\varepsilon$ . 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) [6] 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 [7]. See the review by Tomasi and Perisco [8] 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.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

$n \quad a_0 = n \times 10^{-5}$ .

RECOMMENDATION:

Use equation (10.1).

#### SOLVENT\_DIELECTRIC

Sets the dielectric constant of the Onsager solvent continuum.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

$n \quad \varepsilon = n \times 10^{-5}$ .

RECOMMENDATION:

As per required solvent.

### 10.2.2 Surface and Simulation of Volume Polarization for Electrostatics (SS(V)PE)

The SS(V)PE model [9–11] treats solvent as a continuum dielectric, solving for apparent surface charges on the solute cavity surface by solving Poisson's equation. SS(V)PE is particularly designed to provide a good approximation to the volume polarization arising from “escaped charge”, which is that fraction of the wavefunction extending past the cavity boundary. In addition, the cavity construction and Poisson solver also receive careful numerical treatment. For example, the cavity may be chosen to be an iso-density contour surface, and the Lebedev grids for the Poisson

solver can be chosen very densely. For more information on the similarities and differences of the SS(V)PE model to other continuum methods, see Ref. [10].

Lebedev grids [12–14] are used as a single-center integration grid for the solution of the Poisson equation on the cavity surface. A Lebedev grid having octahedral symmetry on the initial unit sphere is projected onto the actual molecular surface, and so supports full symmetry for all Abelian point groups. The larger and/or the less spherical the solute molecule may be, the more points are needed to get satisfactory precision in the results. Further experience will be required to develop detailed recommendations for this parameter. Values as small as 110 points are usually sufficient for simple diatomics and triatomics. The default value of 1202 points has been found adequate to obtain the energy to within 0.1 kcal/mol for solutes the size of monosubstituted benzenes.

The SS(V)PE model treats the electrostatic interaction between a quantum solute and a classical dielectric continuum solvent. No treatment is yet implemented for cavitation, dispersion, or any of a variety of other specific solvation effects. Note that corrections for these latter effects that might be reported by other programs are generally not transferable. The reason is that they are usually parameterized to improve the ultimate agreement with experiment. In addition to providing corrections for the physical effects advertised, they therefore also inherently include contributions that help to make up for any deficiencies in the electrostatic description. Consequently, they are appropriate only for use with the particular electrostatic model in which they were originally developed.

Analytic nuclear gradients are not yet available for the SS(V)PE energy, but numerical differentiation will permit optimization of small solute molecules. Wavefunctions may be any of the RHF or UHF and their KS-DFT analogs.

Researchers that use this feature are asked to cite Ref. [9].

### **SVP**

Sets whether to perform the isodensity solvation procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform the SS(V)PE solvation procedure.

TRUE Perform the SS(V)PE solvation procedure.

RECOMMENDATION:

NONE

### **SVP\_MEMORY**

Specifies the amount of memory for use by the solvation module.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

*n* corresponds to the amount of memory in MB.

RECOMMENDATION:

The default should be fine for medium size molecules with the default Lebedev grid, only increase if needed.

**SVP\_PATH**

Specifies whether to run a gas phase computation prior to performing the solvation procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 runs a gas-phase calculation and after convergence runs the SS(V)PE computation.

1 does not run a gas-phase calculation.

RECOMMENDATION:

Running the gas-phase calculation provides a good guess to start the solvation stage and provides a more complete set of solvated properties.

**SVP\_CHARGE\_CONV**

Determines the convergence value for the charges on the cavity. When the change in charges fall below this value, if the electron density is converged, then the calculation is considered converged.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

$n$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

The default value unless convergence problems arise.

**SVP\_CAVITY\_CONV**

Determines the convergence value of the iterative isodensity cavity procedure.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

The default value unless convergence problems arise.

**SVP\_GUESS**

Specifies how and if the solvation module will use a given guess for the charges and cavity points.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No guessing.

1 Read a guess from a previous Q-CHEM solvation computation.

2 Use a guess specified by the \$svpirf section from the input

RECOMMENDATION:

It is helpful to also set SCF\_GUESS to READ when using a guess from a previous Q-CHEM run.



The format for the `$svpirf` section of the input is the following:

```
$svpirf
  <# point> <x point> <y point> <z point> <charge> <grid weight>
  <# point> <x normal> <y normal> <z normal>
$end
```

### 10.2.3 The SVP Section Variables

Now listed are a number of variables that directly access the solvation module and therefore must be specified in the context of a FORTRAN namelist. The format is as follows:

```
$svp
  <KEYWORD>=<VALUE>, <KEYWORD>=<VALUE>, ...
  <KEYWORD>=<VALUE>
$end
```

For example, the section may look like this:

```
$svp
  RHOISO=0.001, DIELST=78.39, NPTLEB=110
$end
```

The following keywords are supported in the `$svp` section:

#### **DIELST**

The static dielectric constant.

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

real number specifying the constant.

RECOMMENDATION:

The default value 78.39 is appropriate for water solvent.

#### **ISHAPE**

A flag to set the shape of the cavity surface.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 use the electronic isodensity surface.

1 use a spherical cavity surface.

RECOMMENDATION:

Use the default surface.

**RHOISO**

Value of the electronic isodensity contour used to specify the cavity surface. (Only relevant for ISHAPE = 0).

TYPE:

FLOAT

DEFAULT:

0.001

OPTIONS:

Real number specifying the density in electrons/bohr<sup>3</sup>.

RECOMMENDATION:

The default value is optimal for most situations. Increasing the value produces a smaller cavity which ordinarily increases the magnitude of the solvation energy.

**RADSPH**

Sphere radius used to specify the cavity surface (Only relevant for ISHAPE=1).

TYPE:

FLOAT

DEFAULT:

Half the distance between the outermost atoms plus 1.4 Angstroms.

OPTIONS:

Real number specifying the radius in bohr (if positive) or in Angstroms (if negative).

RECOMMENDATION:

Make sure that the cavity radius is larger than the length of the molecule.

**INTCAV**

A flag to select the surface integration method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Single center Lebedev integration.

1 Single center spherical polar integration.

RECOMMENDATION:

The Lebedev integration is by far the more efficient.

**NPTLEB**

The number of points used in the Lebedev grid for the single-center surface integration. (Only relevant if INTCAV=0).

TYPE:

INTEGER

DEFAULT:

1202

OPTIONS:

Valid choices are: 6, 18, 26, 38, 50, 86, 110, 146, 170, 194, 302, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, or 5294.

RECOMMENDATION:

The default value has been found adequate to obtain the energy to within 0.1 kcal/mol for solutes the size of monosubstituted benzenes.

**NPTTHE, NPTPHI**

The number of  $(\theta, \phi)$  points used for single-centered surface integration (relevant only if INTCAV=1).

TYPE:

INTEGER

DEFAULT:

8,16

OPTIONS:

$\theta, \phi$  specifying the number of points.

RECOMMENDATION:

These should be multiples of 2 and 4 respectively, to provide symmetry sufficient for all Abelian point groups. Defaults are too small for all but the tiniest and simplest solutes.

**LINEQ**

Flag to select the method for solving the linear equations that determine the apparent point charges on the cavity surface.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

- 0 use LU decomposition in memory if space permits, else switch to LINEQ=2
- 1 use conjugate gradient iterations in memory if space permits, else use LINEQ=2
- 2 use conjugate gradient iterations with the system matrix stored externally on disk.

RECOMMENDATION:

The default should be sufficient in most cases.

**CVGLIN**

Convergence criterion for solving linear equations by the conjugate gradient iterative method (relevant if LINEQ=1 or 2).

TYPE:

FLOAT

DEFAULT:

1.0E-7

OPTIONS:

Real number specifying the actual criterion.

RECOMMENDATION:

The default value should be used unless convergence problems arise.

The single-center surface integration approach may fail for certain highly non spherical molecular surfaces. The program will automatically check for this and bomb out with a warning message if need be. The single-center approach succeeds only for what is called a star surface, meaning that an observer sitting at the center has an unobstructed view of the entire surface. Said another way, for a star surface any ray emanating out from the center will pass through the surface only once. Some cases of failure may be fixed by simply moving to a new center with the ITRNGR parameter described below. But some surfaces are inherently nonstar surfaces and cannot be treated with this program until more sophisticated surface integration approaches are implemented.

**ITRNGR**

Translation of the cavity surface integration grid.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

- 0 No translation (*i.e.*, center of the cavity at the origin of the atomic coordinate system)
- 1 Translate to the center of nuclear mass.
- 2 Translate to the center of nuclear charge.
- 3 Translate to the midpoint of the outermost atoms.
- 4 Translate to midpoint of the outermost non-hydrogen atoms.
- 5 Translate to user-specified coordinates in Bohr.
- 6 Translate to user-specified coordinates in Angstroms.

RECOMMENDATION:

The default value is recommended unless the single-center integrations procedure fails.

**TRANX, TRANY, TRANZ**

$x$ ,  $y$ , and  $z$  value of user-specified translation (only relevant if ITRNGR is set to 5 or 6)

TYPE:

FLOAT

DEFAULT:

0, 0, 0

OPTIONS:

$x$ ,  $y$ , and  $z$  relative to the origin in the appropriate units.

RECOMMENDATION:

None.

**IROTGR**

Rotation of the cavity surface integration grid.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

- 0 No rotation.
- 1 Rotate initial  $xyz$  axes of the integration grid to coincide with principal moments of nuclear inertia (relevant if ITRNGR=1)
- 2 Rotate initial  $xyz$  axes of integration grid to coincide with principal moments of nuclear charge (relevant if ITRNGR=2)
- 3 Rotate initial  $xyz$  axes of the integration grid through user-specified Euler angles as defined by Wilson, Decius, and Cross.

RECOMMENDATION:

The default is recommended unless the knowledgeable user has good reason otherwise.

**ROTTHE ROTPHI ROTCHI**

Euler angles ( $\theta$ ,  $\phi$ ,  $\chi$ ) in degrees for user-specified rotation of the cavity surface.  
(relevant if IROTGR=3)

TYPE:

FLOAT

DEFAULT:

0,0,0

OPTIONS:

$\theta$ ,  $\phi$ ,  $\chi$  in degrees

RECOMMENDATION:

None.

**IOPPRD**

Specifies the choice of system operator form.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Symmetric form.

1 Non-symmetric form.

RECOMMENDATION:

The default uses more memory but is generally more efficient, we recommend its use unless there is shortage of memory available.

**10.2.4 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 [15,16], 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 Q-CHEM/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 parameterization of this model can be found in the original literature [15,17].

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).

<b>Iterative Langevin Dipoles (ILD)</b>	<b>Results (kcal/mol)</b>
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  $\Delta G$  is calculated by using the linear–response approximation [15] 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 keyword section.

#### **CHEMSOL**

Controls the use of CHEMSOL in Q-CHEM.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use CHEMSOL.

1 Perform a CHEMSOL calculation.

RECOMMENDATION:

None

#### **CHEMSOL\_EFIELD**

Determines how the solute charge distribution is approximated in evaluating the electrostatic field of the solute.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Exact solute charge distribution is used.

0 Solute charge distribution is approximated by Mulliken atomic charges.

This is a faster, but less rigorous procedure.

RECOMMENDATION:

None.

#### **CHEMSOL\_NN**

Sets the number of grids used to calculate the average hydration free energy.

TYPE:

INTEGER

DEFAULT:

5  $\Delta G_{\text{hydr}}$  will be averaged over 5 different grids.

OPTIONS:

*n* Number of different grids (Max = 20).

RECOMMENDATION:

None.

**CHEMSOL\_PRINT**

Controls printing in the CHEMSOL part of the Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Limited printout.

1 Full printout.

RECOMMENDATION:

None

## 10.2.5 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 [17]. 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.

**READ\_VDW**

Controls the input of user–defined atomic radii for CHEMSOL calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use default CHEMSOL parameters.

TRUE Read from the *\$van\_der\_waals* section of the input file.

RECOMMENDATION:

None.

Two different (mutually exclusive) formats can be used, as shown below.

```
$van_der_waals
1
atomic_number  radius
...
$end
```

```
$van_der_waals
2
sequential_atom_number  VdW_radius
...
$end
```

The purpose of the second format is to permit the user to customize the radius of specific atoms, in the order that they appear in the *\$molecule* section, 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*. All radii should be given in Å.

## 10.2.6 Example

**Example 10.1** A Langevin dipoles calculation on the methoxide anion. A customized value is specified for the radius of the C atom.

```

$molecule
-1 1
  C  0.0000  0.0000 -0.5274
  O  0.0000  0.0000  0.7831
  H  0.0000  1.0140 -1.0335
  H  0.8782 -0.5070 -1.0335
  H -0.8782 -0.5070 -1.0335
$end

$rem
  EXCHANGE          hf
  BASIS             6-31G
  SCF_CONVERGENCE  6
  CHEMSOL          1
  READ_VDW         true
$end

$van_der_waals
2
1 2.5
$end

```

## 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.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Perform default wavefunction analysis.

FALSE Do not perform default wavefunction analysis.

RECOMMENDATION:

None



**Note:** WAVEFUNCTION\_ANALYSIS has no effect on NBO, solvent models or vibrational analyses.

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  $\psi_i$  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  $\psi_a$  into (10.3) and obtain

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} = \sum_{\mu} (\mathbf{PS})_{\mu\mu} = \text{Tr}(\mathbf{PS}) \quad (10.4)$$

where we interpret  $(\mathbf{PS})_{\mu\mu}$  as the number of electrons associated with  $\phi_{\mu}$ . If the basis functions are atom-centered, 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 [18]. Q-CHEM performs a Mulliken population analysis by default.

### POP\_MULLIKEN

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE (or 0) Do not calculate Mulliken Population.

TRUE (1) Calculate Mulliken population

2 Also calculate shell populations for each occupied orbital.

RECOMMENDATION:

Leave as TRUE. Trivial additional calculation

### 10.3.1 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.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

$n$  Calculate moments to  $n^{\text{th}}$  order.

RECOMMENDATION:

Use default unless higher multipoles are required.

### 10.3.2 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.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No symmetry decomposition.

1 Calculate MO eigenvalues and symmetry (if available).

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices.

RECOMMENDATION:

None

## 10.4 Visualization using MolDen

Upon request, Q-CHEM will generate an input file for the MOLDEN program, a molecular visualization program that is freely available at <http://www.cmbi.ru.nl/molden/molden.html>. MOLDEN can be used to view ball-and-stick molecular models (including stepwise visualization of a geometry optimization), molecular orbitals, and vibrational normal modes. MOLDEN also contains a powerful  $Z$ -matrix editor. Orbital visualization is currently supported for  $s$ ,  $p$ , and  $d$  functions (pure or Cartesian), as well as pure  $f$  functions.

**MOLDEN\_FORMAT**

Requests a MOLDEN-formatted input file containing information from a Q-CHEM job.

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

True Append MOLDEN input file at the end of the Q-CHEM output file.

RECOMMENDATION:

None.

### 10.4.1 Examples

**Example 10.2** Generating a MOLDEN file for molecular orbital visualization.

```
$molecule
  0 1
  0
  H 1 0.95
  H 1 0.95 2 104.5
$end

$rem
  EXCHANGE      hf
  BASIS          cc-pvtz
  PRINT_ORBITALS true (default is to print 5 virtual orbitals)
  MOLDEN_FORMAT true
$end
```

**Example 10.3** Generating a MOLDEN file to step through a geometry optimization.

```
$molecule
  0 1
  0
  H 1 0.95
  H 1 0.95 2 104.5
$end

$rem
  JOBTYP      opt
  EXCHANGE    hf
  BASIS        6-31G*
  MOLDEN_FORMAT true
$end
```

## 10.5 Intracules

The many dimensions of electronic wave functions makes them difficult to analyze and interpret. It is often convenient to reduce this large number of dimensions, yielding simpler functions that can more readily provide chemical insight. The most familiar of these is the one-electron density  $\rho(\mathbf{r})$ , which gives the probability of an electron being found at the point  $\mathbf{r}$ . Analogously, the one-electron momentum density  $\pi(\mathbf{p})$  gives the probability that an electron will have a momentum of  $\mathbf{p}$ . However, the wavefunction is reduced to the one-electron density much information is lost.

In particular, it is often desirable to retain explicit two–electron information. Intracules are two–electron distribution functions and provide information about the *relative* position and momentum of electrons. A detailed account of the different type of intracules can be found in [19]. Q-CHEM’s intracule package was developed by Aaron Lee and Nick Besley, and can compute the following intracules for or HF wavefunctions:

- Position intracules,  $P(u)$ : describes the probability of finding two electrons separated by a distance  $u$ .
- Momentum intracules,  $M(v)$ : describes the probability of finding two electrons with relative momentum  $v$ .
- Wigner intracule,  $W(u, v)$ : describes the combined probability of finding two electrons separated by  $u$  and with relative momentum  $v$ .

### 10.5.1 Position Intracules

The intracule density  $I(\mathbf{u})$ , represents the probability for the inter-electronic vector  $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2$ .

$$I(\mathbf{u}) = \int \rho(\mathbf{r}_1 \mathbf{r}_2) \delta(\mathbf{r}_{12} - \mathbf{u}) d\mathbf{r}_1 d\mathbf{r}_2 \quad (10.6)$$

where  $\rho(\mathbf{r}_1, \mathbf{r}_2)$  is the two–electron density. A simpler quantity is the spherically averaged intracule density

$$P(u) = \int I(\mathbf{u}) d\Omega_{\mathbf{u}} \quad (10.7)$$

where  $\Omega_{\mathbf{u}}$  is the angular part of  $\mathbf{v}$ , measures the probability that two electrons are separated by a scalar distance  $u = |\mathbf{u}|$ . This intracule is called a position intracule [19]. If the molecular orbitals are expanded within a basis set

$$\psi_a(\mathbf{r}) = \sum_{\mu} c_{\mu a} \phi_{\mu}(\mathbf{r}) \quad (10.8)$$

$P(u)$  can be expressed as

$$P(u) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma} (\mu\nu\lambda\sigma)_{\text{P}} \quad (10.9)$$

where  $\Gamma_{\mu\nu\lambda\sigma}$  is the two–particle density matrix and  $(\mu\nu\lambda\sigma)_{\text{P}}$  is the position integral

$$(\mu\nu\lambda\sigma)_{\text{P}} = \int \phi_{\mu}^*(\mathbf{r}) \phi_{\nu}(\mathbf{r}) \phi_{\lambda}^*(\mathbf{r} + \mathbf{u}) \phi_{\sigma}(\mathbf{r} + \mathbf{u}) d\mathbf{r} d\Omega \quad (10.10)$$

and  $\phi_{\mu}(\mathbf{r})$ ,  $\phi_{\nu}(\mathbf{r})$ ,  $\phi_{\lambda}(\mathbf{r})$  and  $\phi_{\sigma}(\mathbf{r})$  are basis functions. For HF wavefunctions the position intracule can be decomposed into Coulomb  $P_{\text{J}}(u)$  and exchange  $P_{\text{K}}(u)$  components.

$$P_{\text{J}}(u) = \frac{1}{2} \sum_{\mu\nu\lambda\sigma} D_{\mu\nu}^{\alpha} D_{\lambda\sigma}^{\alpha} (\mu\nu\lambda\sigma)_{\text{P}} \quad (10.11)$$

$$P_{\text{K}}(u) = -\frac{1}{2} \sum_{\mu\nu\lambda\sigma} \left[ D_{\mu\lambda}^{\alpha} D_{\nu\sigma}^{\alpha} + D_{\mu\lambda}^{\beta} D_{\nu\sigma}^{\beta} \right] (\mu\nu\lambda\sigma)_{\text{P}} \quad (10.12)$$

where  $D_{\mu\nu}$  etc. are density matrix elements. The evaluation of  $P(u)$ ,  $P_J(u)$  and  $P_K(u)$  within Q-CHEM has been described in detail [20]. Some of the moments of  $P(u)$  are physically significant [21], for example

$$\int_0^{\infty} u^0 P(u) du = \frac{n(n-1)}{2} \quad (10.13)$$

$$\int_0^{\infty} u^0 P_J(u) du = \frac{n^2}{2} \quad (10.14)$$

$$\int_0^{\infty} u^2 P_J(u) du = nQ - \mu^2 \quad (10.15)$$

$$\int_0^{\infty} u^0 P_K(u) du = -\frac{n}{2} \quad (10.16)$$

where  $n$  is the number of electrons and,  $\mu$  is the electronic dipole moment and  $Q$  is the trace of the electronic quadrupole moment tensor. Q-CHEM can compute both moments and derivatives of position intracules.

### 10.5.2 Momentum Intracules

Analogous quantities can be defined in momentum space.  $\bar{I}(\mathbf{v})$  represents the probability density of a relative momentum  $\mathbf{v} = \mathbf{p}_1 - \mathbf{p}_2$

$$\bar{I}(\mathbf{v}) = \int \pi(\mathbf{p}_1, \mathbf{p}_2) \delta(\mathbf{p}_1 - \mathbf{v}) d\mathbf{p}_1 d\mathbf{p}_2 \quad (10.17)$$

where  $\pi(\mathbf{p}_1, \mathbf{p}_2)$  momentum two-electron density. Similarly, the spherically averaged intracule

$$M(v) = \int \bar{I}(\mathbf{v}) d\Omega_{\mathbf{v}} \quad (10.18)$$

where  $\Omega_{\mathbf{v}}$  is the angular part of  $\mathbf{v}$ , is a measure of relative momentum  $v = |\mathbf{v}|$  and is called the momentum intracule.  $M(v)$  can be written as

$$M(v) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma} (\mu\nu\lambda\sigma)_M \quad (10.19)$$

where  $\Gamma_{\mu\nu\lambda\sigma}$  is the two-particle density matrix and  $(\mu\nu\lambda\sigma)_M$  is the momentum integral [22]

$$(\mu\nu\lambda\sigma)_M = \frac{v^2}{2\pi^2} \int \phi_{\mu}^*(\mathbf{r}) \phi_{\nu}(\mathbf{r} + \mathbf{q}) \phi_{\lambda}^*(\mathbf{u} + \mathbf{q}) \phi_{\sigma}(\mathbf{u}) j_0(qv) d\mathbf{r} d\mathbf{q} d\mathbf{u} \quad (10.20)$$

The momentum integrals only possess four-fold permutational symmetry, *i.e.*,

$$(\mu\nu\lambda\sigma)_M = (\nu\mu\lambda\sigma)_M = (\sigma\lambda\nu\mu)_M = (\lambda\sigma\mu\nu)_M \quad (10.21)$$

$$(\nu\mu\lambda\sigma)_M = (\mu\nu\sigma\lambda)_M = (\lambda\sigma\nu\mu)_M = (\sigma\lambda\mu\nu)_M \quad (10.22)$$

and therefore generation of  $M(v)$  is roughly twice as expensive as  $P(u)$ . Momentum intracules can also be decomposed into Coulomb  $M_J(v)$  and exchange  $M_K(v)$  components.

$$M_J(v) = \frac{1}{2} \sum_{\mu\nu\lambda\sigma} D_{\mu\nu} D_{\lambda\sigma} (\mu\nu\lambda\sigma)_M \quad (10.23)$$

$$M_K(v) = -\frac{1}{2} \sum_{\mu\nu\lambda\sigma} \left[ D_{\mu\lambda}^\alpha D_{\nu\sigma}^\alpha + D_{\mu\lambda}^\beta D_{\nu\sigma}^\beta \right] (\mu\nu\lambda\sigma)_M \quad (10.24)$$

Again the even-order moments are physically significant [22]

$$\int_0^\infty v^0 M(v) dv = \frac{n(n-1)}{2} \quad (10.25)$$

$$\int_0^\infty u^0 M_J(v) dv = \frac{n^2}{2} \quad (10.26)$$

$$\int_0^\infty v^2 P_J(v) dv = 2nE_T \quad (10.27)$$

$$\int_0^\infty v^0 M_K(v) dv = -\frac{n}{2} \quad (10.28)$$

$$(10.29)$$

where  $n$  is the number of electrons and  $E_T$  is the total electronic kinetic energy. Currently, Q-CHEM can compute  $M(v)$ ,  $M_J(v)$  and  $M_K(v)$  using  $s$  and  $p$  basis functions only. Moments are generated using quadrature and consequently for accurate results  $M(v)$  must be computed over a large and closely spaced  $v$  range.

### 10.5.3 Wigner Intracules

$P(u)$  and  $M(v)$  provide a representation of an electron distribution in either position *or* momentum space but neither alone can provide a complete description. For a combined position *and* momentum description an intracule in phase space is required. Defining such an intracule is more difficult since there is no phase space second-order reduced density. However, the second-order Wigner distribution [23]

$$W_2(\mathbf{r}_1, \mathbf{p}_1, \mathbf{r}_2, \mathbf{p}_2) = \frac{1}{\pi^6} \int \rho_2(\mathbf{r}_1 + \mathbf{q}_1, \mathbf{r}_1 - \mathbf{q}_1, \mathbf{r}_2 + \mathbf{q}_2, \mathbf{r}_2 - \mathbf{q}_2) e^{-2i(\mathbf{p}_1 \cdot \mathbf{q}_1 + \mathbf{p}_2 \cdot \mathbf{q}_2)} d\mathbf{q}_1 d\mathbf{q}_2 \quad (10.30)$$

can be interpreted as giving the probability of finding an electron at  $\mathbf{r}_1$  with momentum  $\mathbf{p}_1$  and another electron at  $\mathbf{r}_2$  with momentum  $\mathbf{p}_2$ .  $W_2(\mathbf{r}_1, \mathbf{r}_2, \mathbf{p}_1, \mathbf{p}_2)$  is often referred to as ‘quasi-probability’ distribution’ since it is not positive everywhere.

The Wigner distribution can be used in an analogous way to the second order reduced densities to define a combined position and momentum intracule. This intracule is called a Wigner intracule, and is formally defined as

$$W(u, v) = \int W_2(\mathbf{r}_1, \mathbf{p}_1, \mathbf{r}_2, \mathbf{p}_2) \delta(\mathbf{r}_{12} - \mathbf{u}) \delta(\mathbf{p}_{12} - \mathbf{v}) d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{p}_1 d\mathbf{p}_2 d\Omega_{\mathbf{u}} d\Omega_{\mathbf{v}} \quad (10.31)$$

If the orbitals are expanded in a basis set,  $W(u, v)$  can be written as

$$W(u, v) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma} (\mu\nu\lambda\sigma)_W \quad (10.32)$$

where  $(\mu\nu\lambda\sigma)_W$  is the Wigner integral

$$(\mu\nu\lambda\sigma)_W = \frac{v^2}{2\pi^2} \int \int \phi_\mu^*(\mathbf{r}) \phi_\nu(\mathbf{r} + \mathbf{q}) \phi_\lambda^*(\mathbf{r} + \mathbf{q} + \mathbf{u}) \phi_\sigma(\mathbf{r} + \mathbf{u}) j_0(qv) d\mathbf{r} d\mathbf{q} d\Omega_{\mathbf{u}} \quad (10.33)$$

Wigner integrals are similar to momentum integrals and only have four-fold permutational symmetry. Evaluating Wigner integrals is considerably more difficult than their position or momentum counterparts. The fundamental  $[ssss]_W$  integral

$$[ssss]_W = \frac{u^2 v^2}{2\pi^2} \int \int \exp[-\alpha|\mathbf{r} - \mathbf{A}|^2 - \beta|\mathbf{r} + \mathbf{q} - \mathbf{B}|^2 - \gamma|\mathbf{r} + \mathbf{q} + \mathbf{u} - \mathbf{C}|^2 - \delta|\mathbf{r} + \mathbf{u} - \mathbf{D}|^2] \times j_0(qv) d\mathbf{r} d\mathbf{q} d\Omega_{\mathbf{u}} \quad (10.34)$$

can be expressed as

$$[ssss]_W = \frac{\pi u^2 v^2 e^{-(R+\lambda^2 u^2 + \mu^2 v^2)}}{2(\alpha + \delta)^{3/2} (\beta + \gamma)^{3/2}} \int e^{-\mathbf{P} \cdot \mathbf{u}} j_0(|\mathbf{Q} + \eta \mathbf{u}|v) d\Omega_{\mathbf{u}} \quad (10.35)$$

or

$$[ssss]_W = \frac{2\pi^2 u^2 v^2 e^{-(R+\lambda^2 u^2 + \mu^2 v^2)}}{(\alpha + \delta)^{3/2} (\beta + \gamma)^{3/2}} \sum_{n=0}^{\infty} (2n+1) i_n(Pu) j_n(\eta uv) j_n(Qv) P_n\left(\frac{\mathbf{P} \cdot \mathbf{Q}}{PQ}\right) \quad (10.36)$$

Two approaches for evaluating  $(\mu\nu\lambda\sigma)_W$  have been implemented in Q-CHEM, full details can be found in [24]. The first approach uses the first form of  $[ssss]_W$  and used Lebedev quadrature to perform the remaining integrations over  $\Omega_{\mathbf{u}}$ . For high accuracy large Lebedev grids [12–14] should be used, grids of up to 5294 points are available in Q-CHEM. Alternatively, the second form can be adopted and the integrals evaluated by summation of a series. Currently, both methods have been implemented within Q-CHEM for  $s$  and  $p$  basis functions only.

When computing intracules it is most efficient to locate the loop over  $u$  and/or  $v$  points within the loop over shell-quartets [25]. However, for  $W(u, v)$  this requires a large amount of memory to store all the integrals arising from each  $(u, v)$  point. Consequently, an additional scheme, in which the  $u$  and  $v$  points loop is outside the shell-quartet loop, is available. This scheme is less efficient, but substantially reduces the memory requirements.

#### 10.5.4 Intracule Job Control

The following *\$rem* variables can be used to control the calculation of intracules.

**INTRACULE**

Controls whether intracule properties are calculated (see also the *\$intracule* section).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE No intracule properties.

TRUE Evaluate intracule properties.

RECOMMENDATION:

None

**WIG\_MEM**

Reduce memory required in the evaluation of  $W(u, v)$ .

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use low memory option.

TRUE Use low memory option.

RECOMMENDATION:

The low memory option is slower, use default unless memory is limited.

**WIG\_LEB**

Use Lebedev quadrature to evaluate Wigner integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Evaluate Wigner integrals through series summation.

TRUE Use quadrature for Wigner integrals.

RECOMMENDATION:

None

**WIG\_GRID**

Specify angular Lebedev grid for Wigner intracule calculations.

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Lebedev grids up to 5810 points.

RECOMMENDATION:

Larger grids if high accuracy required.



**N\_WIG\_SERIES**

Sets summation limit for Wigner integrals.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n < 100$

RECOMMENDATION:

Increase  $n$  for greater accuracy.

**N\_I\_SERIES**

Sets summation limit for series expansion evaluation of  $i_n(x)$ .

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

**N\_J\_SERIES**

Sets summation limit for series expansion evaluation of  $j_n(x)$ .

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

**10.5.5 Format for the *\$intracule* Section**

<i>int_type</i>	0	Compute $P(u)$ only
	1	Compute $M(v)$ only
	2	Compute $W(u, v)$ only
	3	Compute $P(u)$ , $M(v)$ and $W(u, v)$
	4	Compute $P(u)$ and $M(v)$
	5	Compute $P(u)$ and $W(u, v)$
	6	Compute $M(v)$ and $W(u, v)$
<i>u_points</i>		Number of points, start, end.
<i>v_points</i>		Number of points, start, end.
<i>moments</i>	0–4	Order of moments to be computed ( $P(u)$ only).
<i>derivs</i>	0–4	order of derivatives to be computed ( $P(u)$ only).
<i>accuracy</i>	$n$	( $10^{-n}$ ) specify accuracy of intracule interpolation table ( $P(u)$ only).

### 10.5.6 Examples

**Example 10.4** Compute HF/STO-3G  $P(u)$ ,  $M(v)$  and  $W(u, v)$  for Ne, using Lebedev quadrature with 974 point grid.

```

$molecule
  0 1
  Ne
$end

$rem
  EXCHANGE    hf
  BASIS       sto-3g
  INTRACULE   true
  WIG_LEB     true
  WIG_GRID    974
$end

$intracule
  int_type    3
  u_points    10  0.0  10.0
  v_points    8   0.0   8.0
  moments     4
  derivs      4
  accuracy    8
$end

```

**Example 10.5** Compute HF/6-31G  $W(u, v)$  intracules for H<sub>2</sub>O using series summation up to  $n=25$  and 30 terms in the series evaluations of  $j_n(x)$  and  $i_n(x)$ .

```

$molecule
  0 1
  H1
  0 H1 r
  H2 0 r H1 theta

  r = 1.1
  theta = 106
$end

$rem
  EXCHANGE    hf
  BASIS       6-31G
  INTRACULE   true
  WIG_MEM     true
  N_WIG_SERIES 25
  N_I_SERIES  40
  N_J_SERIES  50
$end

$intracule
  int_type    2
  u_points    30  0.0  15.0
  v_points    20  0.0  10.0
$end

```

## 10.6 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 [26] and infrared activities. Vibrational frequencies are calculated by either using an analytic Hessian (if available, see 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.

When calculating analytic frequencies at the HF and DFT levels of theory, the coupled-perturbed SCF equations must be solved. This is the most time-consuming step in the calculation, and also consumes the most memory. The amount of memory required is  $\mathcal{O}(N^2M)$  where  $N$  is the number of basis functions, and  $M$  the number of atoms. This is an order more memory than is required for the SCF calculation, and is often the limiting consideration when treating larger systems analytically. Q-CHEM incorporates a new approach to this problem that avoids this memory bottleneck by solving the CPSCF equations in segments [27]. Instead of solving for all the perturbations at once, they are divided into several segments, and the CPSCF is applied for one segment at a time, resulting in a memory scaling of  $\mathcal{O}(N^2M/N_{\text{seg}})$ , where  $N_{\text{seg}}$  is the number of segments. This option is invoked automatically by the program.

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 [9, 28, 29].

### 10.6.1 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 7). 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).

Users should also be aware that the quality of the quadrature grid used in DFT calculations is more important when calculating second derivatives. The default grid for some atoms has changed in Q-CHEM 3.0 (see section 4.3.4) and for this reason vibrational frequencies may vary slightly from previous versions. It is recommended that a grid larger than the default grid is used when performing frequency calculations.

The standard output from a frequency analysis includes the following.

- 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

TYPE:  
LOGICAL

DEFAULT:  
FALSE

OPTIONS:  
FALSE Do not calculate Raman intensities.  
TRUE Do calculate Raman intensities.

RECOMMENDATION:  
None

**VIBMAN\_PRINT**

Controls level of extra print out for vibrational analysis.

TYPE:  
INTEGER

DEFAULT:  
1

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.  
7 Level 6 plus anharmonic cubic and quartic force field print out.

RECOMMENDATION:  
Use default.

**CPSCF\_NSEG**

Controls the number of segments used to calculate the CPSCF equations.

TYPE:  
INTEGER

DEFAULT:  
0

OPTIONS:  
0 Do not solve the CPSCF equations in segments.  
*n* User-defined. Use *n* segments when solving the CPSCF equations.

RECOMMENDATION:  
Use default.

**FDIFF\_STEPSIZE**

Displacement used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 0.001 Å.

OPTIONS:

$n$  Use a step size of  $n \times 10^{-5}$ .

RECOMMENDATION:

Use default, unless on a very flat potential, in which case a larger value should be used.

## 10.6.2 Example

**Example 10.6** 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.

```
$molecule
  0 1
  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
```

## 10.7 Anharmonic Vibrational Frequency

Computing vibrational spectra beyond the harmonic approximation has become an active area of research owing to the improved efficiency of computer techniques [31–34]. To calculate the exact vibrational spectrum within Born–Oppenheimer approximation, one has to solve the nuclear Schrödinger equation completely using numerical integration techniques, and consider the full configuration interaction of quanta in the vibrational states. This has only been carried out on di- or tri-atomic system [35,36]. The difficulty of this numerical integration arises because solving

exact the nuclear Schrödinger equation requires a complete electronic basis set, consideration of all the nuclear vibrational configuration states, and a complete potential energy surface (PES). Simplification of the Nuclear Vibration Theory (NVT) and PES are the doorways to accelerating the anharmonic correction calculations. There are five aspects to simplifying the problem:

- Expand the potential energy surface using a Taylor series and examine the contribution from higher derivatives. Small contributions can be eliminated, which allows for the efficient calculation of the Hamiltonian.
- Investigate the effect on the number of configurations employed in a variational calculation.
- Avoid using variational theory (due to its expensive computational cost) by using other approximations, for example, perturbation theory.
- Obtain the PES indirectly by applying a self-consistent field procedure.
- Apply an *anharmonic* wavefunction which is more appropriate for describing the distribution of nuclear probability on an anharmonic potential energy surface.

To incorporate these simplifications, new formulae combining information from the Hessian, gradient and energy are used as a default procedure to calculate the cubic and quartic force field of a given potential energy surface.

Here, we also briefly describe various NVT methods. In the early stage of solving the nuclear Schrödinger equation (around 1930s), second-order Vibrational Perturbation Theory (VPT2) [33, 37–40] was developed. However, problems occur when resonances exist in the spectrum. This becomes more problematic for larger molecules due to the greater chance of accidental degeneracies occurring. To avoid this problem, one can do a direct integration of the secular matrix using Vibrational Configuration Interaction (VCI) theory [41]. It is the most accurate method and also the least favored due to its computational expense. In Q-CHEM 3.0, we introduce a new approach to treating the wavefunction, transition-optimized shifted Hermite (TOSH) theory [42], which uses first-order perturbation theory, which avoids the degeneracy problems of VPT2, but which incorporates anharmonic effects into the wavefunction, thus increasing the accuracy of the predicted anharmonic energies.

### 10.7.1 Vibration Configuration Interaction Theory

To solve the nuclear vibrational Schrödinger equation, one can only use direct integration procedures for di-atomic molecules [35, 36]. For larger systems, a truncated version of full configuration interaction is considered to be the most accurate approach. When one applies the variational principle to the vibrational problem, a basis function for the nuclear wavefunction of the  $n$ th excited state of mode  $i$  is:

$$\psi_i^{(n)} = \phi_i^{(n)} \prod_{j \neq i}^m \phi_j^{(0)} \quad (10.37)$$

where the  $\phi_i^{(n)}$  represents the harmonic oscillator eigenfunctions for normal mode  $q_i$ , which can be expressed in terms of Hermite polynomials:

$$\phi_i^{(n)} = \left( \frac{\omega_i^{\frac{1}{2}}}{\pi^{\frac{1}{2}} 2^n n!} \right)^{\frac{1}{2}} e^{-\frac{\omega_i q_i^2}{2}} H_n(q_i \sqrt{\omega_i}) \quad (10.38)$$

With the basis function defined in eq. (10.37), the  $n$ th wavefunction can be described as a linear combination of the Hermite polynomials:

$$\Psi^{(n)} = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \sum_{k=0}^{n_3} \cdots \sum_{m=0}^{n_m} c_{ijk\dots m}^{(n)} \psi_{ijk\dots m}^{(n)} \quad (10.39)$$

where  $n_i$  is the number of quanta in the  $i$ th mode. We propose the notation VCI( $n$ ) where  $n$  is the total number of quanta *i.e.*:

$$n = n_1 + n_2 + n_3 + \cdots + n_m \quad (10.40)$$

To determine this expansion coefficient  $c^{(n)}$ , we integrate the  $\hat{H}$ , as in eq. (4.1), with  $\Psi^{(n)}$  to get the eigenvalues

$$c^{(n)} = E_{\text{VCI}(n)}^{(n)} = \langle \Psi^{(n)} | \hat{H} | \Psi^{(n)} \rangle \quad (10.41)$$

This gives us the anharmonically corrected frequencies to  $n$ -quanta accuracy for a  $m$ -mode molecule. The size of the secular matrix on the right hand of eq. (10.41) is  $((n+m)!/n!m!)^2$ , and the storage of this matrix can easily surpass the memory limit of a computer. Although this method is highly accurate, we need to seek for other approximations for computing large molecules.

### 10.7.2 Vibrational Perturbation Theory

Vibrational perturbation theory has been historically popular for calculating molecular spectroscopy. Nevertheless, it is notorious for the incapability of dealing with resonance cases. In addition, the non-standard formulas for various symmetries of molecules forces the users to modify inputs on a case-by-case basis [43–45], which narrows the accessibility of this method. VPT applies perturbation treatments on the same Hamiltonian as in eq. (4.1), but divides it into an unperturbed part,  $\hat{U}$ , and a perturbed part,  $\hat{V}$ .

$$\hat{U} = \sum_i^m \left( -\frac{1}{2} \frac{\partial^2}{\partial q_i^2} + \frac{\omega_i^2}{2} q_i^2 \right) \quad (10.42)$$

$$\hat{V} = \frac{1}{6} \sum_{ijk=1}^m \eta_{ijk} q_i q_j q_k + \frac{1}{24} \sum_{ijkl=1}^m \eta_{ijkl} q_i q_j q_k q_l \quad (10.43)$$

One can then apply second-order perturbation theory to get the  $i$ th excited state energy:

$$E^{(i)} = \hat{U}^{(i)} + \langle \Psi^{(i)} | \hat{V} | \Psi^{(i)} \rangle + \sum_{j \neq i} \frac{|\langle \Psi^{(i)} | \hat{V} | \Psi^{(j)} \rangle|^2}{\hat{U}^{(i)} - \hat{U}^{(j)}} \quad (10.44)$$

The denominator in eq. (10.44) can be zero either because of symmetry or accidental degeneracy. Various solutions, which depend on the type of degeneracy that occurs, have been developed which ignore the zero-denominator elements from the Hamiltonian [43–46]. An alternative solution has been proposed by Barone [33] which can be applied to all molecules by changing the masses of one or more nuclei in degenerate cases. The disadvantage of this method is that it will break the degeneracy which results in fundamental frequencies no longer retaining their correct symmetry. He proposed:

$$E_{\text{VPT2}}^{(i)} = \sum_i \omega_i \left(n_i + \frac{1}{2}\right) + \sum_{i \leq j} x_{ij} \left(n_i + \frac{1}{2}\right) \left(n_j + \frac{1}{2}\right) + \dots \quad (10.45)$$

where the anharmonic constants  $x_{ii}$  and  $x_{ij}$  are given by:

$$x_{ii} = \frac{1}{16\omega_i^2} \left( \eta_{iiii} - \frac{5\eta_{iii}^2}{3\omega_i^2} - \sum_{i \neq j}^m \frac{(8\omega_i^2 - 3\omega_j^2)\eta_{iiij}^2}{\omega_j^2(4\omega_i^2 - \omega_j^2)} \right) \quad (10.46)$$

$$x_{ij} = \frac{1}{4\omega_i\omega_j} \left( \eta_{iijj} - \frac{2\eta_{iii}^2}{4\omega_i^2 - \omega_j^2} - \frac{2\eta_{ijj}^2}{4\omega_j^2 - \omega_i^2} - \frac{\eta_{iii}\eta_{ijj}}{\omega_i^2} - \frac{\eta_{jjj}\eta_{iij}}{\omega_j^2} - \sum_{i \neq j \neq k}^m \frac{\eta_{iik}\eta_{jjk}}{\omega_k^2} \right) \quad (10.47)$$

Here, we ignore the vib-rotational terms (Coriolis coupling), and rotational constants since the contribution from these terms are relatively small (less than  $10\text{cm}^{-1}$ ).

### 10.7.3 Transition-Optimized Shifted Hermite Theory

So far, every aspect of solving the nuclear wave equation has been considered, except the wavefunction. Since Schrödinger proposed his equation, the nuclear wavefunction has traditionally been expressed in terms of Hermite functions, which are designed for the harmonic oscillator case. Recently [42] a modified representation has been presented. To demonstrate how this approximation works, we start with a simple example. For a di-atomic molecule, the Hamiltonian with up to quartic derivatives can be written as:

$$\hat{H} = -\frac{1}{2} \frac{\partial^2}{\partial q^2} + \frac{1}{2} \omega^2 q^2 + \eta_{iii} q^3 + \eta_{iiii} q^4 \quad (10.48)$$

and the wavefunction is expressed as in eq. (10.38). Now, if we shift the center of the wavefunction by  $\sigma$ , which is equivalent to a translation of the normal coordinate  $q$ , the shape will still remain the same, but the anharmonic correction can now be incorporated into the wavefunction. For a ground vibrational state, the wavefunction is written as:

$$\phi^{(0)} = \left(\frac{\omega}{\pi}\right)^{\frac{1}{4}} e^{-\frac{\omega}{2}(q-\sigma)^2} \quad (10.49)$$

Similarly, for the first excited vibrational state, we have:

$$\phi^{(1)} = \left(\frac{4\omega^3}{\pi}\right)^{\frac{1}{4}} (q - \sigma) e^{-\frac{\omega}{2}(q-\sigma)^2} \quad (10.50)$$

Therefore, the energy difference between the first vibrational excited state and the ground state is:

$$\Delta E_{\text{TOSH}} = \omega + \frac{\eta_{iiii}}{8\omega^2} + \frac{\eta_{iii}\sigma}{2\omega} + \frac{\eta_{iiii}\sigma^2}{4\omega} \quad (10.51)$$

which is the fundamental vibrational frequency from first-order perturbation theory. We know from the first-order perturbation theory with an ordinary wavefunction within a QFF PES, the energy is:



$$\Delta E_{\text{VPT1}} = \omega + \frac{\eta_{iii}}{8\omega^2} \quad (10.52)$$

The differences between these two wavefunctions are the two extra terms arising from the shift in eq. (10.51). To determine the shift, we compare the energy with that from second-order perturbation theory:

$$\Delta E_{\text{VPT2}} = \omega + \frac{\eta_{iii}}{8\omega^2} - \frac{5\eta_{iii}^2}{24\omega^4} \quad (10.53)$$

Since  $\sigma$  is a very small quantity compared with the other variables, we ignore the contribution of  $\sigma^2$  and compare  $\Delta E_{\text{TOSH}}$  with  $\Delta E_{\text{VPT2}}$ , which yields an initial guess for  $\sigma$ :

$$\sigma = -\frac{5}{12} \frac{\eta_{iii}}{\omega^3} \quad (10.54)$$

Because the only difference between this approach and the ordinary wavefunction is the shift in the normal coordinate, we call it “Transition-optimized shifted Hermite” (TOSH) functions [42]. This approximation gives second-order accuracy at only first-order cost.

For poly-atomic molecules, we consider eq. (10.51), and propose the energy of the  $i$ th mode expressed as:

$$\Delta E_{\text{TOSH}}^{(i)} = \nu_i = \omega_i + \frac{1}{8\omega_i} \sum_{j=1}^m \frac{\eta_{ijj}}{\omega_j} + \frac{1}{2\omega_i} \sum_{j=1}^m \eta_{ijj} \sigma_j + \frac{1}{4\omega_i} \sum_{j=1}^m \eta_{ijj} \sigma_j^2 \quad (10.55)$$

Following the same approach as for the di-atomic case, by comparing this with the energy from second-order perturbation theory, we obtain the shift as

$$\sigma_i = \delta_{ij} \frac{\eta_{iii}}{6\omega_i^3} - \frac{\eta_{ijj}}{2\omega_i \omega_j^2} - \frac{\eta_{ijj}}{4\omega_i(-2\omega_i - \omega_j)\omega_j} - \sum_{k \neq i}^N \frac{\eta_{jkk}}{4\omega_j^2 \omega_k} \quad (10.56)$$

## 10.8 Job Control

The following *\$rem* variables can be used to control the calculation of anharmonic frequencies.

### ANHAR

Performing various nuclear vibrational theory (TOSH, VPT2, VCI) calculations to obtain vibrational anharmonic frequencies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Carry out the anharmonic frequency calculation.

FALSE Do harmonic frequency calculation.

RECOMMENDATION:

Since this calculation involves the third and fourth derivatives at the minimum of the potential energy surface, it is recommended that the `GEOM_OPT_TOL_DISPLACEMENT`, `GEOM_OPT_TOL_GRADIENT` and `GEOM_OPT_TOL_ENERGY` tolerances are set tighter. Note that VPT2 calculations will fail if the system involves resonances (either accidental or due to symmetry). See the VCI *\$rem* variable for more details about increasing the accuracy of anharmonic calculations.

**VCI**

Specifies the number of quanta involved in the VCI calculation.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined

RECOMMENDATION:

The availability depends on the memory of the machine. For example, a machine with 1.5 GB memory and for molecules with fewer than 3 atoms, VCI(10) can be carried out, for molecule containing fewer than 5 atoms, VCI(7) can be carried out, for molecule containing fewer than 6 atoms, VCI(4) can be carried out. For molecules larger than hexa-atomics, VCI(2) is usually available. VCI(1) and VCI(3) usually overestimated the true energy while VCI(4) usually gives an answer close to the converged energy.

**ENG\_DER**

Controls what types of gradient information are used to compute higher derivatives. The default uses a combination of energy, gradient and Hessian information, which makes the force field calculation faster.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy, gradient, and Hessian information.

1 Use energy information only.

2 Use Hessian information only.

RECOMMENDATION:

When the molecule is larger than benzene, ENG\_DER=2 is recommended.

**10.8.1 Examples**

**Example 10.7** A four-quanta anharmonic frequency calculation on formaldehyde at the EDF2/6-31G\* optimized ground state geometry, which is obtained in the first part of the job. It is necessary to carry out the harmonic frequency first and this will print out an approximate time for the subsequent anharmonic frequency calculation. If a FREQ job has already been performed, the anharmonic calculation can be restarted using the saved scratch files from the harmonic calculation.

```

$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
  JOBTYP  OPT
  EXCHANG  EDF2

```

```

BASIS 6-31G*
GEOM_OPT_TOL_DISPLACEMENT 1
GEOM_OPT_TOL_GRADIENT 1
GEOM_OPT_TOL_ENERGY 1
$end
@@@
$molecule
  READ
$end
$rem
  JOBTYP E      FREQ
  EXCHANG E      EDF2
  BASIS 6-31G*
$end
@@@
$molecule
  READ
$end
$rem
  JOBTYP E      FREQ
  EXCHANG E      EDF2
  BASIS 6-31G*
  ANHAR TRUE
  VCI 4
$end

```

## 10.8.2 Isotopic Substitutions

By default Q-CHEM calculates vibrational frequencies using the atomic masses of the most abundant isotopes (taken from the Handbook of Chemistry and Physics, 63<sup>rd</sup> Edition). Masses of other isotopes can be specified using the *\$isotopes* section and by setting the ISOTOPES *\$rem* variable to TRUE. The format of the *\$isotopes* section is as follows:

```

$isotopes
  number_of_isotope_loops tp_flag
  number_of_atoms [temp pressure] (loop 1)
  atom_number1 mass1
  atom_number2 mass2
  ...
  number_of_atoms [temp pressure] (loop 2)
  atom_number1 mass1
  atom_number2 mass2
  ...
$end

```

**Note:** Only the atoms whose masses are to be changed from the default values need to be specified. After each loop all masses are reset to the default values. Atoms are numbered according to the order in the *\$molecule* section.

An initial loop using the default masses is always performed first. Subsequent loops use the user-specified atomic masses. Only those atoms whose masses are to be changed need to be included in the list, all other atoms will adopt the default masses. The output gives a full frequency analysis for each loop. Note that the calculation of vibrational frequencies in the additional loops only involves a rescaling of the computed Hessian, and therefore takes little additional computational time.

The first line of the *\$isotopes* section specifies the number of substitution loops and also whether the temperature and pressure should be modified. The *tp\_flag* setting should be set to 0 if the default temperature and pressure are to be used (298.18 K and 1 atm respectively), or 1 if they are to be altered. Note that the temperatures should be specified in Kelvin (K) and pressures in atmospheres (atm).

### ISOTOPES

Specifies if non-default masses are to be used in the frequency calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use default masses only.

TRUE Read isotope masses from *\$isotopes* section.

RECOMMENDATION:

None

### 10.8.3 Example

**Example 10.8** 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.

```

$molecule
  0 1
  C  1.08900  0.00000  0.00000
  C -1.08900  0.00000  0.00000
  H  2.08900  0.00000  0.00000
  H -2.08900  0.00000  0.00000
$end

$rem
  BASIS          3-21G
  JOBTYP        opt
  EXCHANGE       hf
  CORRELATION    none
$end

@@@

$molecule
  read
$end

$rem
  BASIS          3-21G
  JOBTYP        freq
  EXCHANGE       hf
  CORRELATION    none
  SCF_GUESS      read
  ISOTOPES       1
$end

$isotopes
  2  0          ! two loops, both at std temp and pressure
  4
  1  13.00336  ! All atoms are given non-default masses

```

```

      2  13.00336
      3  2.01410
      4  2.01410
    2
      3  2.01410  ! H's replaced with D's
      4  2.01410
$end

```

## 10.9 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.

If switched on for a geometry optimization, the NBO package will only be invoked at the end of the last optimization step.

### 10.9.1 Job Control

The following *\$rem* variable should be set:

#### NBO

Controls the use of the NBO package.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not invoke the NBO package.

TRUE Do invoke the NBO package.

RECOMMENDATION:

None

The general format for passing options from Q-CHEM to the NBO program is shown below:

```

$nbo
  {NBO program keywords, parameters and options}
$end

```

**Note:** (1) *\$rem* variable NBO must be set to TRUE before the *\$nbo* keyword is recognized.  
 (2) Q-CHEM does not support facets of the NBO package which require multiple job runs

## 10.10 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:

**Example 10.9** A job that evaluates the H<sub>2</sub> 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.

```

$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

```

General format for the *\$plots* section of the Q-CHEM input deck.

*\$plots*

One comment line

Specification of the 3–D mesh of points on 3 lines:

$$N_x \quad x_{\min} \quad x_{\max}$$

$$N_y \quad y_{\min} \quad y_{\max}$$

$$N_z \quad z_{\min} \quad z_{\max}$$

A line with 4 integers indicating how many things to plot:

$$N_{\text{MO}} \quad N_{\text{Rho}} \quad N_{\text{Trans}} \quad N_{\text{DA}}$$

An optional line with the integer list of MO's to evaluate (only if  $N_{\text{MO}} > 0$ )

$$\text{MO}(1) \quad \text{MO}(2) \quad \dots \quad \text{MO}(N_{\text{MO}})$$

An optional line with the integer list of densities to evaluate (only if  $N_{\text{Rho}} > 0$ )

$$\text{Rho}(1) \quad \text{Rho}(2) \quad \dots \quad \text{Rho}(N_{\text{Rho}})$$

An optional line with the integer list of transition densities (only if  $N_{\text{Trans}} > 0$ )

$$\text{Trans}(1) \quad \text{Trans}(2) \quad \dots \quad \text{Trans}(N_{\text{Trans}})$$

An optional line with states for detachment/attachment densities (if  $N_{\text{DA}} > 0$ )

$$\text{DA}(1) \quad \text{DA}(2) \quad \dots \quad \text{DA}(N_{\text{DA}})$$

*\$end*

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 ( $N_{\text{MO}}$ ), electron densities ( $N_{\text{Rho}}$ ), transition densities and attach/detach densities ( $N_{\text{DA}}$ ), 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  $1 \dots N_{\alpha}$ ,  $N_{\alpha} + 1 \dots N_{\alpha} + N_{\beta}$ ; electron densities numbered where 0 = ground state, 1 = first excited state, 2 = second excited state, *etc.*; and attach/detach specified from state  $1 \rightarrow N_{\text{DA}}$ .

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, as shown below:

```

x1  y1  z1      a1  a2  ...  aN
x2  y1  z1      b1  b2  ...  bN
...

```

The Q-CHEM *\$plots* utility allows the user to plot transition densities and detachment/attachment densities directly from amplitudes saved from a previous calculation, without having to solve the post-SCF (CIS, RPA, TDA, or TDDFT) equations again. To take advantage of this feature, the same Q-CHEM scratch directory must be used, and the SKIP\_CIS\_RPA *\$rem* variable must be set to TRUE. To further reduce computational time, the SCF\_GUESS *\$rem* can be set to READ.

#### SKIP\_CIS\_RPA

Skips the solution of the CIS, RPA, TDA or TDDFT equations for wavefunction analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE / FALSE

RECOMMENDATION:

Set to true to speed up the generation of plot data if the same calculation has been run previously with the scratch files saved.

## 10.11 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 ACSII file, plot.esp, in the format  $x, y, z, esp$  for each point.

TYPE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1 read grid input via the *\$plots* section of the input deck

0 Generate the ESP values at all nuclear positions.

+ $n$  read  $n$  grid points from the ACSII file ESPGrid.

RECOMMENDATION:

None

The following example illustrates the evaluation of electrostatic potentials on a grid, note that IANLTY must also be set to 200.

**Example 10.10** A job that evaluates the electrostatic potential for H<sub>2</sub> 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.

```

$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
  IGDESP -1
$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
  0 0 0 0
  0
$end

```

The electrostatic potential is a complicated object and it is not uncommon to model it using a simplified representation based on atomic charges. For this purpose it is well known that Mulliken charges perform very poorly. Several definitions of ESP-derived atomic charges have been given in the literature, however, most of them rely on a least-squares fitting of the ESP evaluated on a selection of grid points. Although these grid points are usually chosen so that the ESP is well modeled in the “chemically important” region, it still remains that the calculated charges will change if the molecule is rotated. Recently an efficient rotationally invariant algorithm was proposed [47] that sought to model the ESP not by direct fitting, but by fitting to the multipole moments. By doing so it was found that the fit to the ESP was superior to methods that relied on direct fitting of the ESP. The calculation requires the traceless form of the multipole moments and these are also printed out during the course of the calculations. To request these multipole-derived charges the following *\$rem* option should be set:



**MM\_CHARGES**

Requests the calculation of multipole-derived charges (MDCs).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculates the MDCs and also the traceless form of the multipole moments

RECOMMENDATION:

Set to TRUE if MDCs or the traceless form of the multipole moments are desired.

The calculation does not take long.

## 10.12 Spin and charge densities at the nuclei

Gaussian basis sets violate nuclear cusp conditions [48–50]. This may lead to large errors in wavefunction at nuclei, particularly for spin density calculations [51]. This problem can be alleviated by using an averaging operator that compute wavefunction density based on constraints that wavefunction must satisfy near Coulomb singularity [52, 53]. The derivation of operators is based on hyper virial theorem [54] and presented in Ref. [52]. Application to molecular spin densities for spin-polarized [53] and DFT [55] wavefunctions show considerable improvement over traditional delta function operator.

One of the simplest forms of such operators is based on the Gaussian weight function

$$\exp[-(Z/r_0)^2 (\mathbf{r} - \mathbf{R})^2] \quad (10.57)$$

sampling the vicinity of a nucleus of charge  $Z$  at  $\mathbf{R}$ . Parameter  $r_0$  has to be small enough to neglect two-electron contributions of the order  $O(r_0^4)$  but large enough for meaningful averaging. The range of values between 0.15 to 0.3 *a.u.* is shown to be adequate, with final answer being relatively insensitive to the exact choice of  $r_0$  [52, 53]. The value of  $r_0$  is chosen by RC\_R0 keyword in the units of 0.001 *a.u.* The averaging operators are implemented for single determinant Hartree-Fock and DFT, and correlated SSG wavefunctions. Spin and charge densities are printed for all nuclei in a molecule, including ghost atoms.

**RC\_R0**

Determines the parameter in the Gaussian weight function used to smooth the density at the nuclei.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Corresponds the traditional delta function spin and charge densities

$n$  corresponding to  $n \times 10^{-3}$  *a.u.*

RECOMMENDATION:

We recommend value of 250 for a typical spit valence basis. For basis sets with increased flexibility in the nuclear vicinity the smaller values of  $r_0$  also yield adequate spin density.

## 10.13 NMR Shielding Tensors

NMR spectroscopy is a powerful technique to yield important information on molecular systems in chemistry and biochemistry. Since there is no direct relationship between the measured NMR signals and structural properties, the necessity for a reliable method to predict NMR chemical shifts arises. Examples for such assignments are numerous, for example, assignments of solid-state spectra [56,57]. The implementation within Q-CHEM uses gauge-including atomic orbitals (GIAO's) [58–60] to calculate the NMR chemical shielding tensors. This scheme has been proven to be reliable and accurate for many applications [61].

The shielding tensor,  $\sigma$ , is a second-order property depending on the external magnetic field,  $B$ , and the nuclear magnetic spin momentum,  $m_k$ , of nucleus  $k$ .

$$\Delta E = -m_j(1 - \sigma)B \quad (10.58)$$

Using analytical derivative techniques to evaluate  $\sigma$ , the components of the 3x3 tensor are computed as

$$\sigma_{ij} = \sum_{\mu\nu} P_{\mu\nu} \frac{\partial^2 h_{\mu\nu}}{\partial B_i \partial m_{j,k}} + \sum_{\mu\nu} \frac{\partial P_{\mu\nu}}{\partial B_i} \frac{\partial h_{\mu\nu}}{\partial m_{j,k}} \quad (10.59)$$

where  $i$  and  $j$  are Cartesian coordinates. To solve for the necessary perturbed densities  $\partial P/\partial B_{x,y,z}$ , a new CPSCF method based on a density matrix based formulation [62,66] is used. This formulation is related to a density matrix based CPSCF (D-CPSCF) formulation employed for the computation of vibrational frequencies [63]. Alternatively, an MO-based CPSCF calculation of shielding tensors can be chosen by the variable MOPROP. Features of the NMR package include:

- Restricted HF-GIAO and KS-DFT-GIAO NMR chemical shifts calculations
- LinK/CFMM support to evaluate Coulomb- and Exchange-like matrices
- Density matrix-based coupled-perturbed SCF (D-CPSCF)
- DIIS acceleration
- Support of basis sets up to  $d$  functions
- Support of LSDA/GGA/Hybrid XC-functionals

### 10.13.1 Job Control

The JOBTYP must be set to NMR to request the NMR chemical shifts.

#### D\_CPSCF\_PERTNUM

Specifies whether to do the perturbations one at a time, or all together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perturbed densities to be calculated all together.

1 Perturbed densities to be calculated one at a time.

RECOMMENDATION:

None

**D\_SCF\_CONV\_1**

Sets the convergence criterion for the level-1 iterations. This preconditions the density for the level-2 calculation, and does not include any two-electron integrals.

TYPE:

INTEGER

DEFAULT:

4 corresponding to a threshold of  $10^{-4}$ .

OPTIONS:

$n < 10$  Sets convergence threshold to  $10^{-n}$ .

RECOMMENDATION:

The criterion for level-1 convergence must be less than or equal to the level-2 criterion, otherwise the D-CPSCF will not converge.

**D\_SCF\_CONV\_2**

Sets the convergence criterion for the level-2 iterations.

TYPE:

INTEGER

DEFAULT:

4 Corresponding to a threshold of  $10^{-4}$ .

OPTIONS:

$n < 10$  Sets convergence threshold to  $10^{-n}$ .

RECOMMENDATION:

None

**D\_SCF\_MAX\_1**

Sets the maximum number of level-1 iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

$n$  User defined.

RECOMMENDATION:

Use default.

**D\_SCF\_MAX\_2**

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

$n$  User defined.

RECOMMENDATION:

Use default.

**D\_SCF\_DIIS**

Specifies the number of matrices to use in the DIIS extrapolation in the D-CPSCF.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

$n$   $n = 0$  specifies no DIIS extrapolation is to be used.

RECOMMENDATION:

Use the default.

## 10.14 Linear-Scaling NMR chemical shifts: GIAO-HF and GIAO-DFT

The importance of nuclear magnetic resonance (NMR) spectroscopy for modern chemistry and biochemistry cannot be overestimated. Despite tremendous progress in experimental techniques, the understanding and reliable assignment of observed experimental spectra remains often a highly difficult task, so that quantum chemical methods can be extremely useful both in the solution and the solid state (*e.g.*, Refs. [56, 57, 64–66] and references therein).

The cost for the computation of NMR chemical shifts within even the simplest quantum chemical methods such as Hartree-Fock (HF) or density functional (DFT) approximations increases conventionally with the third power of the molecular size  $M$ ,  $\mathcal{O}(M^3)$ , where  $\mathcal{O}(\cdot)$  stands for the scaling order. Therefore, the computation of NMR chemical shieldings has so far been limited to molecular systems in the order of 100 atoms without molecular symmetry.

For larger systems it is crucial to reduce the increase of the computational effort to linear, which has been recently achieved by Kussmann and Ochsenfeld [62, 66]. In this way, the computation of NMR chemical shifts becomes possible at both HF or DFT level for molecular systems with 1000 atoms and more, while the accuracy and reliability of traditional methods is fully preserved. In our formulation we use gauge-including atomic orbitals (GIAO) [58, 59, 67], which have proven to be particularly successful [68]. For example, for many molecular systems the HF (GIAO-HF) approach provides typically an accuracy of 0.2–0.4 ppm for the computation of  $^1\text{H}$  NMR chemical shifts (*e.g.* Refs. [56, 57, 64–66]).

NMR chemical shifts are calculated as second derivatives of the energy with respect to the external magnetic field  $\mathbf{B}$  and the nuclear magnetic spin  $m_{Nj}$  of a nucleus  $N$ :

$$\sigma_{ij}^N = \frac{\partial^2 E}{\partial B_i \partial M_{Nj}} \quad (10.60)$$

where  $i, j$  are  $x, y, z$  coordinates.

For the computation of the NMR shielding tensor it is necessary to solve for the response of the one-particle density matrix with respect to the magnetic field, so that the solution of the coupled perturbed SCF (CPSCF) equations either within the HF or the DFT approach is required.

These equations can be solved within a density matrix-based formalism for the first time with only linear-scaling effort for molecular systems with a non-vanishing HOMO-LUMO gap [66]. The solution is even simpler in DFT approaches without explicit exchange, since present density functionals are not dependent on the magnetic field.

The present implementation of NMR shieldings in Q-CHEM employs the LinK (linear exchange K) method [69, 70] for the formation of exchange contributions [66]. Since the derivative of the density matrix with respect to the magnetic field is skew-symmetric, its Coulomb-type contractions vanish. For the remaining Coulomb-type matrices the CFMM method [14] is adapted [66]. In addition, a multitude of different approaches for the solution of the CPSCF equations can be selected within Q-CHEM.

The so far largest molecular system for which NMR shieldings have been computed, contained 1003 atoms and 8593 basis functions (GIAO-HF/6-31G\*) without molecular symmetry [66].

## 10.15 Linear-Scaling Computation of Electric Properties

The search for new optical devices is a major field of materials sciences. Here, polarizabilities and hyperpolarizabilities provide particularly important information on molecular systems. The response of the molecular systems in the presence of an external monochromatic oscillatory electric field is determined by the solution of the TDSCF equations, where the perturbation is represented as the interaction of the molecule with a single Fourier component within the dipole approximation

$$\hat{H}^{(S)} = \frac{1}{2} \hat{\mu} \mathcal{E} (e^{-i\omega t} + e^{+i\omega t}) \quad (10.61)$$

$$\hat{\mu} = -e \sum_{i=1}^N \hat{r}_i \quad (10.62)$$

where  $\mathcal{E}$  is the E-field vector,  $\omega$  the corresponding frequency,  $e$  the electronic charge and  $\mu$  the dipole moment operator. Starting from Frenkel's variational principle the TDSCF equations can be derived by standard techniques of perturbation theory [72]. As a solution we yield the first ( $\mathbf{P}^x(\pm\omega)$ ) and second order (*e.g.*  $\mathbf{P}^{xy}(\pm\omega, \pm\omega)$ ) perturbed density matrices with which the following properties are calculated:

- Static polarizability:  $\alpha_{xy}(0; 0) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^y(\omega = 0)]$
- Dynamic polarizability:  $\alpha_{xy}(\pm\omega; \mp\omega) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^y(\pm\omega)]$
- Static hyperpolarizability:  $\beta_{xyz}(0; 0, 0) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^{yz}(\omega = 0, \omega = 0)]$
- Second harmonic generation:  $\beta_{xyz}(\mp 2\omega; \pm\omega, \pm\omega) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^{yz}(\pm\omega, \pm\omega)]$
- Electro-optical Pockels effect:  $\beta_{xyz}(\mp\omega; 0, \pm\omega) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^{yz}(\omega = 0, \pm\omega)]$
- Optical rectification:  $\beta_{xyz}(0; \pm\omega, \mp\omega) = \text{Tr} [\mathbf{H}^{\mu_x} \mathbf{P}^{yz}(\pm\omega, \mp\omega)]$

where  $\mathbf{H}^{\mu_x}$  is the matrix representation of the  $x$  component of the dipole moments.

The TDSCF calculation is the most time consuming step and scales asymptotically as  $\mathcal{O}(N^3)$  because of the AO-MO transformations. The scaling behavior of the two-electron integral formations, which dominate over a wide range because of a larger prefactor, can be reduced by LinK/CFMM from quadratic to linear ( $\mathcal{O}(N^2) \rightarrow \mathcal{O}(N)$ ).

Third-order properties can be calculated with the equations above after a second-order TDSCF calculation (MOPROP: 101/102) or by use of Wigner's  $(2n + 1)$  rule [73] (MOPROP: 103/104). Since the second order TDSCF depends on the first-order results, the convergence of the algorithm may be problematically. So we recommend the use of 103/104 for the calculation of first hyperpolarizabilities.

These optical properties can be computed for the first time using linear-scaling methods (LinK/CFMM) for all integral contractions [62]. Although the present implementation available in Q-CHEM still uses MO-based time-dependent SCF (TDSCF) equations both at the HF and DFT level, the prefactor of this  $\mathcal{O}(M^3)$  scaling step is rather small, so that the reduction of the scaling achieved for the integral contractions is most important. Here, all derivatives are computed analytically.

Further specifications of the dynamic properties are done in the section `$fdpfreq` in the following format:

```
$fdpfreq
  property
  frequencies
  units
$end
```

The first line is only required for third order properties to specify the kind of first hyperpolarizability:

- `StaticHyper` Static Hyperpolarizability
- `SHG` Second harmonic generation
- `EOPockels` Electro-optical Pockels effect
- `OptRect` Optical rectification

Line number 2 contains the values (FLOAT) of the frequencies of the perturbations. Alternatively, for dynamic polarizabilities an equidistant sequence of frequencies can be specified by the keyword `WALK` (see example below). The last line specifies the units of the given frequencies:

- `au` Frequency (atomic units)
- `eV` Frequency (eV)
- `nm` Wavelength (nm) → Note that 0 nm will be treated as 0.0 *a.u.*
- `Hz` Frequency (Hertz)
- `cmInv` Wavenumber ( $\text{cm}^{-1}$ )

### 10.15.1 Examples for section `$fdpfreq`

**Example 10.11** Static and Dynamic polarizabilities, atomic units:

```
$fdpfreq
  0.0 0.03 0.05
  au
$end
```

**Example 10.12** Series of dynamic polarizabilities, starting with 0.00 incremented by 0.01 up to 0.10:

```
$fdpfreq
  walk 0.00 0.10 0.01
  au
$end
```

**Example 10.13** Static first hyperpolarizability, second harmonic generation and electro-optical Pockels effect, wavelength in nm:

```
$fdpfreq
  StaticHyper SHG EOPockels
  1064
  nm
$end
```

### 10.15.2 Features of mopropman

- Restricted/unrestricted HF and KS-DFT CPSCF/TDSCF
- LinK/CFMM support to evaluate Coulomb- and Exchange-like matrices
- DIIS acceleration
- Support of LSDA/GGA/Hybrid XC-functionals listed below
- Analytical derivatives

The following XC-functionals are supported:

Exchange:

- Dirac
- Becke 88

Correlation:

- Wigner
- VWN (both RPA and No. 5 parameterizations)
- Perdew-Zunger 81
- Perdew 86 (both PZ81 and VWN (No. 5) kernel)
- LYP

### 10.15.3 Job Control

The following options can be used:

**MOPROP**

Specifies the job for mopropman.

TYPE:

INTEGER

DEFAULT:

0 Do not run mopropman.

OPTIONS:

1 NMR chemical shielding tensors.

2 Static polarizability.

100 Dynamic polarizability.

101 First hyperpolarizability.

102 First hyperpolarizability, reading First order results from disk.

103 First hyperpolarizability using Wigner's  $(2n + 1)$  rule.

104 First hyperpolarizability using Wigner's  $(2n + 1)$  rule, reading first order results from disk.

RECOMMENDATION:

None.

**MOPROP\_PERTNUM**

Set the number of perturbed densities that will to be treated together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 All at once.

$n$  Treat the perturbed densities batchwise.

RECOMMENDATION:

Use default

**MOPROP\_CONV\_1ST**

Sets the convergence criteria for CPSCF and 1st order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

None

**MOPROP\_CONV\_2ND**

Sets the convergence criterium for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

None



**MOPROP\_criteria\_1ST**

The maximal number of iterations for CPSCF and first-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

$n$  Set maximum number of iterations to  $n$ .

RECOMMENDATION:

Use default.

**MOPROP\_MAXITER\_2ND**

The maximal number of iterations for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

$n$  Set maximum number of iterations to  $n$ .

RECOMMENDATION:

Use default.

**MOPROP\_DIIS**

Controls the use of Pulays DIIS.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

0 Turn off DIIS.

5 Turn on DIIS.

RECOMMENDATION:

None

**MOPROP\_DIIS\_DIM\_SS**

Specified the DIIS subspace dimension.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

0 No DIIS.

$n$  Use a subspace of dimension  $n$ .

RECOMMENDATION:

None

**SAVE\_LAST\_GPX**

Save last  $\mathbf{G}[\mathbf{P}^x]$  when calculating dynamic polarizabilities in order to call mopropman in a second run with MOPROP = 102.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 False

1 True

RECOMMENDATION:

None

## 10.16 Atoms in Molecules

Q-CHEM can output a file suitable for analysis with the Atoms in Molecules package (AIMPAC). The source for AIMPAC can be freely downloaded from the website:

<http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm>

Users should check this site for further information about installing and running AIMPAC. The AIMPAC input file is created by specifying a filename for the WRITE\_WFN *\$rem*.

**WRITE\_WFN**

Specifies whether or not a wfn file is created, which is suitable for use with AIMPAC. Note that the output to this file is currently limited to  $f$  orbitals, which is the highest angular momentum implemented in AIMPAC.

TYPE:

STRING

DEFAULT:

(NULL) No output file is created.

OPTIONS:

*filename* Specifies the output file name. The suffix *.wfn* will be appended to this name.

RECOMMENDATION:

None

# References and Further Reading

- [1] Ground State Methods (Chapters 4 and 5).
- [2] Excited State Methods (Chapter 6; particularly. attachment–detachment analysis).
- [3] Basis Sets (Chapter 7).
- [4] NBO manual.
- [5] L. Onsager, *J. Am. Chem. Soc.*, **58**, 1486, (1936).
- [6] R. C. Weast, *Handbook of chemistry and physics*; 70 ed.; R. C. Weast, Ed.; CRC Press: Boca Raton, FL, (1989).
- [7] M. W. Wong, M. J. Frisch and K. B. Wiberg, *J. Am. Chem. Soc.*, **113**, 4776, (1991).
- [8] J. Tomasi and M. Perisco, *Chem. Rev.*, **94**, 2027, (1994).
- [9] D. M. Chipman, *J. Chem. Phys.*, **112**, 5558, (2000).
- [10] D. M. Chipman, *Theor. Chem. Acc.*, **107**, 80, (2002).
- [11] D. M. Chipman and M. Dupuis, *Theor. Chem. Acc.*, **107**, 90, (2002).
- [12] V. I. Lebedev., *Zh. Vych. Mat. Mat. Fiz.*, **16**, 29, (1976).
- [13] V. I. Lebedev., *Sib. Mat. Zh.*, **18**, 132, (1977).
- [14] V. I. Lebedev and D. N. Laikov, *Dokl. Math.*, **366**, 741, (1999).
- [15] J. Florián, and A. Warshel, *J. Phys. Chem. B*, **101**, 5583, (1997).
- [16] J. Florián, and A. Warshel: CHEMSOL 1.0, University of Southern California, Los Angeles, (1997).
- [17] J. Florián, and A. Warshel, *J. Phys. Chem.*, **103**, 10282, (1999).
- [18] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction To Advanced Electronic Structure Theory*; (Dover, New York, 1998).
- [19] P. M. W. Gill, D. P. O’Neill and N. A. Besley, *Theor. Chem. Acc.*, **109**, (2003).
- [20] A. M. Lee and P. M. W. Gill, *Chem. Phys. Lett*, **313**, 271, (1999).
- [21] P. M. W. Gill., *Chem. Phys. Lett*, **270**, 193, (1997).
- [22] N. A. Besley, A. M. Lee and P. M. W. Gill., *Mol. Phys*, **100**, 1763, (2002).
- [23] N. A. Besley, D. P. O’Neill and P. M. W. Gill., *J. Chem. Phys*, **118**, 2033, (2003).
- [24] E. Wigner, *Phys. Rev*, **40**, 749, (1932).

- [25] J. Cioslowski and G. Liu, *J. Chem. Phys.*, **105**, 4151, (1996).
- [26] B. G. Johnson and J. Florián, *Chem. Phys. Lett.*, **247**, 120, (1995).
- [27] [20] P.P. Korambath, J. Kong, T.R. Furlani and M. Head-Gordon, *Mol. Phys.*, **100**, 1755, (2002).
- [28] C. W. Murray, G. J. Laming, N. C. Handy and R. D. Amos, *Chem. Phys. Lett.*, **199**, 551, (1992).
- [29] A. P. Scott and L. Radom, *J. Phys. Chem.*, **100**, 16502, (1996).
- [30] B. G. Johnson, P. M. W. Gill and J. A. Pople, *J. Chem. Phys.*, **98**, 5612, (1993).
- [31] K. Yagi, K. Hirao, T. Taketsugu, M. W. Schmidt, and M. S. Gordon, *J. Chem. Phys.*, **121**, 1383, (2004).
- [32] R. Burcl, N. C. Handy, and S. Carter, *Spectro. Acta A*, **59**, 1881, (2003).
- [33] V. Barone, *J. Chem. Phys.*, **122**, 014108, (2005).
- [34] A. Miani, E. Cané, P. Palmieri, A. Trombetti, and N. C. Handy, *J. Chem. Phys.*, **112**, 248, (2000).
- [35] S. D. Peyerimhoff, *Encyclopaedia of Computational Chemistry*, chapter Spectroscopy: Computational Methods, page 2646, John Wiley & Sons, Ltd, 1998.
- [36] J. Tucker Carrington, *Encyclopaedia of Computational Chemistry*, chapter Vibrational Energy Level Calculations, page 3157, John Wiley & Sons, Ltd, 1998.
- [37] A. Adel and D. M. Dennison, *Phys. Rev.*, **43**, 716, (1933).
- [38] E. B. Wilson and J. J. B. Howard, *J. Chem. Phys.*, **4**, 260, (1936).
- [39] H. H. Nielsen, *Phys. Rev.*, **60**, 794, (1941).
- [40] J. Neugebauer and B. A. Hess, *J. Chem. Phys.*, **118**, 7215, (2003).
- [41] R. J. Whitehead and N. C. Handy, *J. Mol. Spec.*, **55**, 356, (1975).
- [42] C. Y. Lin, A. T. B. Gilbert and P. M. W. Gill, *in preparation*.
- [43] W.D. Allen, Y. Yamaguchi, A.G. Császár, D.A. Clabo Jr., R.B. Remington and H.F. Schaefer III *Chem. Phys.*, **145**, 427, (1990).
- [44] I. M. Mills, *Molecular Spectroscopy: Modern Research*, chapter 3.2 Vibration-Rotation Structure in Asymmetric- and Symmetric-Top Molecules, pages 115, edited by K. N. Rao and C. W. Mathews, Academic Press, New York, 1972.
- [45] D. Clabo, W. Allen, R. Remington, Y. Yamaguchi, and H. Schaefer, *Chem. Phys.*, **123**, 187, (1988).
- [46] H. Nielsen, *Rev. Mod. Phys.*, **23**, 90, (1951).
- [47] A.C. Simmonett, A.T.B. Gilbert and P.M.W. Gill, *Mol. Phys.*, **103**, 2789, (2005).
- [48] T. Kato, *Commun. Pure Appl. Math.*, **10**, 151, (1957).
- [49] R. T. Pack and W. Byers Brown, *J. Chem. Phys.*, **45**, 556, (1966).
- [50] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **104**, 9908, (1996).
- [51] D. M. Chipman, *Theor. Chim. Acta*, **76**, 73, (1989).

- [52] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **105**, 1470, (1996).
- [53] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.*, **105**, 1479, (1996).
- [54] J. O. Hirschfelder, *J. Chem. Phys.*, **33**, 1462, (1960).
- [55] B. Wang, J. Baker, and P. Pulay, *Phys. Chem. Chem. Phys.*, **2**, 2131, (2000).
- [56] C. Ochsenfeld, *Phys. Chem. Chem. Phys.*, **2**, 2153, (2000).
- [57] C. Ochsenfeld, S. P. Brown, I. Schnell, J. Gauss and H. W. Spiess, *J. Am. Chem. Soc.*, **123**, 2597, (2001).
- [58] R. Ditchfield, *Mol. Phys.*, **27**, 789, (1974).
- [59] K. Wolinski, J. F. Hinton, P. Pulay, *J. Am. Chem. Soc.*, **112**, 8251, (1990).
- [60] M. Häser, R. Ahlrichs, H. P. Baron, P. Weiss and H. Horn, *Theo. Chim. Acta.*, **83**, 455, (1992).
- [61] T. Helgaker, M. Jaszuński and K. Ruud, *Chem. Rev.*, **99**, 293, (1990).
- [62] J. Kussmann and C. Ochsenfeld, to be published
- [63] C. Ochsenfeld and M. Head-Gordon, *Chem. Phys. Lett.*, **270**, 399, (1997).
- [64] S. P. Brown, T. Schaller, U. P. Seelbach, F. Koziol, C. Ochsenfeld, F.-G. Klärner, H. W. Spiess, *Angew. Chem. Int. Ed.*, **40**, 717, (2001).
- [65] C. Ochsenfeld, F. Koziol, S. P. Brown, T. Schaller, U. P. Seelbach, F.-G. Klärner, *Solid State Nucl. Magn. Reson.*, **22**, 128, (2002).
- [66] C. Ochsenfeld, J. Kussmann, F. Koziol, *Angew. Chem.*, **116**, 4585, (2004).
- [67] F. London, *J. Phys. Radium*, **8**, 397, (1937).
- [68] J. Gauss, *Ber. Bunsenges. Phys. Chem.*, **99**, 1001, (1995).
- [69] C. Ochsenfeld, C. A. White, M. Head-Gordon, *J. Chem. Phys.*, **109**, 1663, (1998).
- [70] *Chem. Phys. Lett.*, **327**, 216, (2000).
- [71] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.*, **230**, 8, (1994).
- [72] H. Sekino, R.J. Bartlett, *J. Chem. Phys.*, **85**, 976, (1986).
- [73] S.P. Karna, M. Dupuis, *J. Comp. Chem.*, **12**, 487, (1991).

# 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:

- User input file
- $\$HOME/.qchemrc$  file
- $\$QC/config/preferences$  file
- Program defaults

The order of preference of initialization is as above, 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 above.

**Note:** The *.qchemrc* and *preferences* files are not requisites for running Q-CHEM and currently only support  $\$rem$  keywords.

### 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. The format of both of these files is as follows:

```
$rem
  rem_variable  option  comment
  rem_variable  option  comment
  ...
$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.

```
$rem
  INCORE_INTS_BUFFER  4000000  More integrals in memory
  DIIS_SUBSPACE_SIZE  5         Modify max DIIS subspace size
  THRESH              10
$end
```

### 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 (AO2MO\_DISK, INCORE\_INTS\_BUFFER, MEM.STATIC, 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.

#### GUI

Controls the output of auxiliary information for third party packages.

#### TYPE:

INTEGER

#### DEFAULT:

0

#### OPTIONS:

0 No auxiliary output is printed.

2 Auxiliary information is printed to the file *Test.FChk*.

#### RECOMMENDATION:

Use default unless the additional information is required. Please note that any existing *Test.FChk* file will be overwritten.



# 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:

- Initial starting geometry.
- Algorithm used to determine the step  $\mathbf{h}$ .
- Quality of the Hessian (second derivative) matrix.
- 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

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

$$E(\mathbf{x}_0 + \mathbf{h}) = E(\mathbf{x}_0) + \mathbf{h}^t \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \frac{1}{2} \mathbf{h}^t \frac{d^2 E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \mathbf{h} + \dots \quad (\text{A.1})$$

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

$$\frac{dE}{d\mathbf{x}} \equiv \mathbf{g} \text{ (gradient vector)}, \quad \frac{d^2E}{d\mathbf{x}_1 d\mathbf{x}_2} \equiv \mathbf{H} \text{ (Hessian matrix)} \quad (\text{A.4})$$

(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

$$\mathbf{h} = \sum \frac{-F_i}{b_i} \mathbf{u}_i \quad (\text{A.5})$$

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.3 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

$$\mathbf{h} = \sum \frac{-F_i}{(b_i - \lambda)} \mathbf{u}_i \quad (\text{A.6})$$

in which  $\lambda$  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  $\lambda$ . References [2–4] all utilize the same basic approach (A.6) but differ in the means of determining the value of  $\lambda$ .

The EF algorithm [1] utilizes the rational function approach presented in [4], yielding an eigenvalue equation of the form

$$\begin{pmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} \quad (\text{A.7})$$

from which a suitable  $\lambda$  can be obtained. Expanding (A.7) gives

$$(\mathbf{H} - \lambda)\mathbf{h} + \mathbf{g} = 0 \quad (\text{A.8})$$

$$\mathbf{g}^t \mathbf{h} = \lambda \quad (\text{A.9})$$

In terms of a diagonal Hessian representation, (A.8) rearranges to (A.6), and substitution of (A.6) into the diagonal form of (A.9) gives

$$\sum \frac{-F_i^2}{(b_i - \lambda)} = \lambda \quad (\text{A.10})$$

which can be used to evaluate  $\lambda$  iteratively.

The eigenvalues,  $\lambda$ , of the RFO equation (A.7) have the following important properties [4]:

- The  $(n + 1)$  values of  $\lambda$  bracket the  $n$  eigenvalues of the Hessian matrix  $\lambda_i < b_i < \lambda_{i+1}$ .
- At a stationary point, one of the eigenvalues,  $\lambda$ , of (A.7) is zero and the other  $n$  eigenvalues are those of the Hessian at the stationary point.
- 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

$$\begin{pmatrix} b_1 & F_1 \\ F_1 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ 1 \end{pmatrix} \quad (\text{A.11})$$

$$\begin{pmatrix} b_2 & & & F_2 \\ & \ddots & \mathbf{0} & \vdots \\ & & b_n & F_n \\ F_2 & \cdots & F_n & 0 \end{pmatrix} \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} \quad (\text{A.12})$$

where it is assumed that we are maximizing along the lowest Hessian mode  $\mathbf{u}_1$ . Note that  $\lambda_p$  is the highest eigenvalue of (A.11) (it is always positive and approaches zero at convergence) and  $\lambda_n$  is the lowest eigenvalue of (A.12) (it is always negative and again approaches zero at convergence).

Choosing these values of  $\lambda$  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.6) to give the final step

$$\mathbf{h} = \frac{-F_1}{(b_1 - \lambda_p)} \mathbf{u}_1 - \sum_{i=2}^n \frac{-F_i}{(b_i - \lambda_n)} \mathbf{u}_i \quad (\text{A.13})$$

If this step is greater than the maximum allowed, it is scaled down. For minimization only one shift parameter,  $\lambda_n$ , would be used which would act on all modes.

In (A.11) and (A.12) 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  $k^{\text{th}}$  mode (instead of the lowest), (A.11) is replaced by

$$\begin{pmatrix} b_k & F_k \\ F_k & 0 \end{pmatrix} \begin{pmatrix} h_k \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_k \\ 1 \end{pmatrix} \quad (\text{A.14})$$

and (A.12) would now exclude the  $k^{\text{th}}$  mode but include the lowest. Since what was originally the  $k^{\text{th}}$  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.4 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 pseudo symmetry. 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:

- 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.
- 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).
- 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]

$$\Delta\mathbf{q} = \mathbf{B}\Delta\mathbf{X} \quad (\text{A.15})$$

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  $nm$  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

$$\mathbf{G}(\mathbf{U}\mathbf{R}) = (\mathbf{U}\mathbf{R}) \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \quad (\text{A.16})$$

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.16) 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:

- 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.
- 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.



- 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.5 Constrained Optimization

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

$$R_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (\text{A.17})$$

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

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \sum_{i=1}^m \lambda_i C_i(\mathbf{x}) \quad (\text{A.18})$$

which replaces the function  $F(\mathbf{x})$  in the unconstrained case. Here, the  $\lambda_i$  are the so-called Lagrange multipliers, one for each constraint  $C_i(\mathbf{x})$ . Differentiating (A.18) with respect to  $\mathbf{x}$  and  $\lambda$  gives

$$\begin{aligned}\frac{dL(\mathbf{x}, \lambda)}{dx_j} &= \frac{dF(\mathbf{x})}{dx_j} - \sum_{i=1}^m \lambda_i \frac{dC_i(\mathbf{x})}{dx_j} \\ \frac{dL(\mathbf{x}, \lambda)}{d\lambda_i} &= -C_i(\mathbf{x})\end{aligned}\quad (\text{A.19})$$

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  $(\mathbf{x}, \lambda)$  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

$$\nabla^2 \mathbf{L} = \begin{pmatrix} \frac{d^2 L(\mathbf{x}, \lambda)}{dx_j dx_k} & \frac{d^2 L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} \\ \frac{d^2 L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} & \frac{d^2 L(\mathbf{x}, \lambda)}{d\lambda_j d\lambda_i} \end{pmatrix} \quad (\text{A.20})$$

where

$$\frac{d^2 L(\mathbf{x}, \lambda)}{dx_j dx_k} = \frac{d^2 F(\mathbf{x})}{dx_j dx_k} - \sum \lambda_i \frac{d^2 F(\mathbf{x})}{dx_j dx_k} \quad (\text{A.21})$$

$$\frac{d^2 L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} = \frac{-dC_i(\mathbf{x})}{dx_j} \quad (\text{A.22})$$

$$\frac{d^2 L(\mathbf{x}, \lambda)}{d\lambda_j d\lambda_i} = 0 \quad (\text{A.23})$$

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.19), and Lagrangian second derivative matrix, given by (A.20), 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,  $\lambda_i$ , 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  $\lambda$ , one for each constraint. The total dimension of the constrained optimization problem,  $nm$ , 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  $m^{\text{th}}$  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.11) is replaced by

$$\begin{pmatrix} b_1 & & & F_1 \\ & \ddots & \mathbf{0} & \vdots \\ & & \mathbf{0} & b_m & F_m \\ F_1 & \cdots & F_m & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} \quad (\text{A.24})$$

and (A.12) by

$$\begin{pmatrix} b_{m+1} & & & F_{m+1} \\ & \ddots & \mathbf{0} & \vdots \\ & & \mathbf{0} & b_{m+n} & F_{m+n} \\ F_{m+1} & \cdots & F_{m+n} & 0 \end{pmatrix} \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} \quad (\text{A.25})$$

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.24) includes the  $m$  constraint modes along which a negative Lagrangian Hessian eigenvalue is required, and (A.25) includes all the other modes.

Equations (A.24) and (A.25) 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  $m + l$ .

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  $0^\circ$  and  $180^\circ$  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 Baker:1993[12], exactly analogous to splitting a  $180^\circ$  bond angle into two  $90^\circ$  angles in a  $Z$ -matrix.

**Note:** Exact  $0^\circ$  and  $180^\circ$  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.6 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

$$\mathbf{C}^{\text{proj}} = \sum \langle \mathbf{C} | \mathbf{U}_k \rangle \mathbf{U}_k \quad (\text{A.26})$$

where the summation is over all  $n$  active coordinates  $\mathbf{U}_k$ . The projected vector  $\mathbf{C}^{\text{proj}}$  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

$$\mathbf{V} = \{ \mathbf{C}^{\text{proj}}, \mathbf{U}_k \ k = 1..n \} \quad (\text{A.27})$$

This set of vectors is Schmidt orthogonalized according to the standard procedure

$$\tilde{\mathbf{V}}_k = \alpha_k \left( \mathbf{V}_k - \sum_{l=1}^{k-1} \langle \mathbf{V}_k | \tilde{\mathbf{V}}_l \rangle \tilde{\mathbf{V}}_l \right) \quad (\text{A.28})$$

where the first vector taken,  $\mathbf{V}_1$ , is  $\mathbf{C}^{\text{proj}}$ . The  $\alpha_k$  in (A.28) 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

$$C_i(R_i) = R_i - R_0 \quad (\text{A.29})$$

$$dC_i(R_i)/dR_i = 1; \quad dC_i(R_i)/dR_j = 0 \quad (\text{A.30})$$

$$d^2C_i(R_i)/dR_i dR_j = 0 \quad (\text{A.31})$$

where in (A.29),  $R_0$  is the desired value of the constrained coordinate, and  $R_i$  is its current value. From (A.30) we see that the constraint normals,  $dC_i(\mathbf{R})/dR_i$ , are simply unit vectors and the Lagrangian Hessian matrix, (A.20), 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  $0^\circ$  and  $180^\circ$  dihedral angle constraints. In Cartesian coordinates it is not possible to formally constrain bond angles and torsions to exactly  $0^\circ$  or  $180^\circ$  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  $0^\circ$  and  $180^\circ$  dihedral angle constraints can be imposed just as easily as any other value.  $180^\circ$  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  $180^\circ$  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.7 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

$$\mathbf{x}_n = \sum_{i=1}^m c_i \mathbf{x}_i \quad (\text{A.32})$$

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.*,  $\mathbf{x}_f = \mathbf{x}_i + \mathbf{e}_i$ , where  $\mathbf{e}_i$  is an error vector, then it is obvious that if the conditions

$$\mathbf{r} = \sum c_i \mathbf{e}_i \quad (\text{A.33})$$

and

$$\sum c_i = 1 \quad (\text{A.34})$$

are satisfied, then the relation

$$\sum c_i \mathbf{x}_i = \mathbf{x}_f \quad (\text{A.35})$$

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

$$\mathbf{e}_i = -\mathbf{H}^{-1} \mathbf{g}_i \quad (\text{A.36})$$

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.33), together with the constraint equation, (A.34), leads to a system of  $(m + 1)$  linear equations

$$\begin{pmatrix} B_{11} & \cdots & B_{1m} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ B_{m1} & \cdots & B_{mm} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \\ -\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (\text{A.37})$$

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  $\lambda$  is a Lagrange multiplier.

The coefficients  $c_i$  determined from (A.37) are used to calculate an intermediate interpolated geometry

$$\mathbf{x}'_{m+1} = \sum c_i \mathbf{x}_i \quad (\text{A.38})$$

and its corresponding interpolated gradient

$$\mathbf{g}'_{m+1} = \sum c_i \mathbf{g}_i \quad (\text{A.39})$$

A new, independent geometry is generated from the interpolated geometry and gradient according to

$$\mathbf{x}_{m+1} = \mathbf{x}'_{m+1} - \mathbf{H}^{-1} \mathbf{g}'_{m+1} \quad (\text{A.40})$$

**Note:** Convergence is theoretically guaranteed regardless of the quality of the Hessian matrix (as long as it is positive definite), and the original GDIIIS 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.32) and, subsequently, by neglecting earlier geometries, and eliminating any geometries more than a certain distance (default 0.3 *a.u.*) from the current geometry.

# References and Further Reading

- [1] J. Baker, *J. Comp. Chem.*, **7**, 385, (1986).
- [2] C. J. Cerjan and W. H. Miller, *J. Chem. Phys.*, **75**, 2800, (1981).
- [3] J. Simons, P. Jorgensen, H. Taylor and J. Ozment, *J. Phys. Chem.*, **87**, 2745, (1983).
- [4] A. Banerjee, N. Adams, J. Simons and R. Shepard, *J. Phys. Chem.*, **89**, 52, (1985).
- [5] P. Csaszar and P. Pulay, *J. Mol. Struct. Theochem.*, **114**, 31, (1984).
- [6] P. Pulay, *J. Comp. Chem.*, **3**, 556, (1982).
- [7] J. Baker, A. Kessi and B. Delley, *J. Chem. Phys.*, **105**, 192, (1996).
- [8] P. Pulay, G. Fogarasi, F. Pang and J. E. Boggs, *J. Am. Chem. Soc.*, **101**, 2550, (1979).
- [9] G. Fogarasi, X. Zhou, P. W. Taylor and P. Pulay, *J. Am. Chem. Soc.*, **114**, 8191, (1992).
- [10] P. Pulay and G. Fogarasi, *J. Chem. Phys.*, **96**, 2856, (1992).
- [11] J. Baker, *J. Comp. Chem.*, **13**, 240, (1992).
- [12] J. Baker and D. Bergeron, *J. Comp. Chem.*, **14**, 1339, (1993).
- [13] J. Baker, *J. Comp. Chem.*, **18**, 1079, (1997).
- [14] D. Poppinger, *Chem. Phys. Letts.*, **35**, 550, (1975).
- [15] J. Baker and W. J. Hehre, *J. Comp. Chem.*, **12**, 606, (1991).
- [16] J. Baker, *J. Comp. Chem.*, **14**, 1085, (1993).
- [17] H. B. Schlegel, *Theor. Chim. Acta.*, **66**, 333, (1984).
- [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 fueled 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 1950s, 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  $\mathcal{O}(N)$  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 summarized by considering a time line 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  $\theta$  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.4 Shell-Pair Data

It is common to characterize 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 centers. 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 center 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.5 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:

- The quartet is equivalent, by point group symmetry, to another quartet already treated.
- The quartet can be ignored on the basis of cheaply computed ERI bounds [7] on the largest quartet bra-ket.
- 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.6 Fundamental ERI

The fundamental ERI [2] and the basis of all ERI algorithms is usually represented

$$\begin{aligned} [\mathbf{0}]^{(0)} &= [ss|ss]^{(0)} \\ &= D_A D_B D_C D_D \iint e^{-\alpha|\mathbf{r}_1-\mathbf{A}|^2} e^{-\beta|\mathbf{r}_1-\mathbf{B}|^2} \left[ \frac{1}{r_{12}} \right] e^{-\gamma|\mathbf{r}_2-\mathbf{C}|^2} e^{-\delta|\mathbf{r}_2-\mathbf{D}|^2} d\mathbf{r}_1 d\mathbf{r}_2 \quad (\text{B.3}) \end{aligned}$$

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.7 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 above time line. Initially, angular momentum was calculated by taking derivatives of the fundamental ERI with respect to one of the Cartesian coordinates of the nuclear center. 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.8 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

$$\begin{aligned}
 (ss|ss) &= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \sum_{l=1}^3 D_{Ai} D_{Bj} D_{Ck} D_{Dl} \times \\
 &\quad \iint e^{-\alpha_i |\mathbf{r}_1 - \mathbf{A}|^2} e^{-\beta_j |\mathbf{r}_1 - \mathbf{B}|^2} \left[ \frac{1}{r_{12}} \right] e^{-\gamma_k |\mathbf{r}_2 - \mathbf{C}|^2} e^{-\delta_l |\mathbf{r}_2 - \mathbf{D}|^2} d\mathbf{r}_1 d\mathbf{r}_2 \\
 &= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \sum_{l=1}^3 [s_i s_j | s_k s_l]
 \end{aligned} \tag{B.5}$$

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 modeling [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.9 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  $\mathcal{O}(N^3)$  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.10 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 brackets prefer late contraction, highly contracted brackets 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.11 More Efficient Hartree–Fock Gradient and Hessian Evaluations

Q-CHEM combines the Head–Gordon–Pople (HGP) method [8] and the COLD prism method [3] for Hartree–Fock gradient and Hessian evaluations. All two–electron four–center integrals are classified according to their angular momentum types and degrees of contraction. For each type of integrals, the program chooses one with a lower cost. In practice, the HGP method is chosen for most integral classes in a gradient or Hessian calculation, and thus it dominates the total CPU time.

Recently the HGP codes within Q-CHEM were completely rewritten for the evaluation of the  $P II^x P$  term in the gradient evaluation, and the  $P II^{xy} P$  term in the Hessian evaluation. Our emphasis is to improve code efficiency by reducing cache misses rather than by reducing FLOP counts. Some timing results from a Hartree–Fock calculation on azt are shown below.

Basis Set	AIX			Linux		
	Old	New	New/Old	Old	New	New/Old
Gradient Evaluation: $P II^x P$ Term						
3-21G	34 s	20 s	0.58	25 s	14 s	0.56
6-31G**	259 s	147 s	0.57	212 s	120 s	0.57
DZ	128 s	118 s	0.92	72 s	62 s	0.86
cc-pVDZ	398 s	274 s	0.69	308 s	185 s	0.60
Hessian Evaluation: $P II^{xy} P$ term						
	Old	New	New/Old	Old	New	New/Old
3-21G	294 s	136 s	0.46	238 s	100 s	0.42
6-31G**	2520 s	976 s	0.39	2065 s	828 s	0.40
DZ	631 s	332 s	0.53	600 s	230 s	0.38
cc-pVDZ	3202 s	1192 s	0.37	2715 s	866 s	0.32

Table B.1: The AIX timings were obtained on an IBM RS/6000 workstation with AIX4 operating system, and the Linux timings on an Opteron cluster where the Q-CHEM executable was compiled with an intel 32-bit compiler.

## B.12 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.

# References and Further Reading

- [1] P. M. W. Gill, M. Head–Gordon and J. A. Pople, *J. Phys. Chem.*, **94**, 5564, (1990).
- [2] P. M. W. Gill, *Advan. Quantum Chem.*, **25**, 142, (1994).
- [3] T. R. Adams, R. D. Adamson and P. M. W. Gill, *J. Chem. Phys.*, **107**, 124, (1997).
- [4] M. J. Frisch, B. G. Johnson, P. M. W. Gill, D. J. Fox and R. H. Nobes, *Chem. Phys. Lett.*, **206**, 225, (1993).
- [5] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Int. J. Quantum Chem.*, **40**, 745, (1991).
- [6] P. M. W. Gill and J. A. Pople, *Int. J. Quantum Chem.*, **40**, 753, (1991).
- [7] P. M. W. Gill, B. G. Johnson and J. A. Pople, *Chem. Phys. Lett.*, **217**, 65, (1994).
- [8] M. Head–Gordon and J. A. Pople, *J. Chem. Phys.*, **89**, 5777, (1988).
- [9] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.*, **206**, 229, (1993).
- [10] B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.*, **206**, 239, (1993).
- [11] S. F. Boys, *Proc. Roy. Soc., A*, **A200**, 542, (1950).
- [12] S. F. Boys, G. B. Cook, C. M. Reeves and I. Shavitt, *Nature*, **178**, 1207, (1956).
- [13] J. A. Pople and W. J. Hehre, *J. Comput. Phys.*, **27**, 161, (1978).
- [14] C. A. White, B. G. Johnson, P. M. W. Gill and M. Head–Gordon, *Chem. Phys. Lett.*, **230**, 8, (1994).
- [15] R. D. Adamson, J. P. Dombroski and P. M. W. Gill, *Chem. Phys. Lett.*, **254**, 329, (1996).
- [16] J. P. Dombroski, S. W. Taylor and P. M. W. Gill, *J. Phys. Chem.*, **100**, 6272, (1996).
- [17] M. Challacombe and E. Schwegler, *J. Chem. Phys.*, **106**, 5526, (1997).
- [18] E. Schwegler and M. Challacombe, *J. Chem. Phys.*, **105**, 2726, (1996).
- [19] M. Dupuis, J. Rys and H. F. King, *J. Chem. Phys.*, **65**, 111, (1976).
- [20] L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.*, **26**, 218, (1978).
- [21] J. Almlöf, K. Faegri and K. Korsell, *J. Comput. Chem.*, **3**, 385, (1982).
- [22] S. Obara and A. Saika, *J. Chem. Phys.*, **84**, 3963, (1986).
- [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.*, **106**, 9708, (1996).



- 
- [25] S. Obara and A. Saika, *J. Chem. Phys.*, **89**, 1540, (1988).
- [26] R. D. Adamson, Honours Thesis. "Shell-pair economisation" Massey University: Palmerston North, 1995.
- [27] M. J. S. Dewar, *Org. Mass. Spec.*, **28**, 305, (1993).
- [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.*, **253**, 268, (1996).
- [30] M. Challacombe, E. Schwegler and J. Almlöf, *J. Chem. Phys.*, **104**, 4685, (1996).
- [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.*, **102**, 8448, (1995).
- [34] W. Yang, *Phys. Rev. Lett.*, **66**, 1438, (1991).
- [35] W. Yang, *Phys. Rev. A*, **44**, 7823, (1991).
- [36] W. Yang and T.-S. Lee, *J. Chem. Phys.*, **103**, 5674, (1995).
- [37] T.-S. Lee, D. M. York and W. Yang, *J. Chem. Phys.*, **105**, 2744, (1996).
- [38] C. Ochsenfeld and M. Head-Gordon, *Chem. Phys. Lett.*, **270**, 399, (1997).

# Appendix C

## Q-Chem Quick Reference

### C.1 Q-Chem Text Input Summary

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

Table C.1: Q-CHEM user input section keywords. See the *\$QC/samples* directory with your release for specific examples of Q-CHEM input using these keywords.

- Note:**
- (1) Users are able to enter keyword sections in any order.
  - (2) Each keyword section must be terminated with the *\$end* keyword.
  - (3) Not all keywords have to be entered, but *\$rem* and *\$molecule* are compulsory.
  - (4) Each keyword section will be described below.
  - (5) The entire Q-CHEM input is case-insensitive.
  - (6) Multiple jobs are separated by the string *@@@* on a single line.

#### C.1.1 Keyword: *\$molecule*

Four methods are available for inputting geometry information:

- *Z*-matrix (Angstroms and degrees):  

```
$molecule
  {Z-matrix}
  {blank line, if parameters are being used}
  {Z-matrix parameters, if used}
$end
```
- Cartesian Coordinates (Angstroms):  

```
$molecule
  {Cartesian coordinates}
  {blank line, if parameter are being used}
  {Coordinate parameters, if used}
$end
```
- Read from a previous calculation:  

```
$molecule
  read
$end
```
- Read from a file:  

```
$molecule
  read filename
$end
```

### C.1.2 Keyword: *\$rem*

See also the list of *\$rem* variables at the end of this Appendix. The general format is:

```
$rem
  REM_VARIABLE  VALUE  [optional comment]
$end
```

### C.1.3 Keyword: *\$basis*

The format for the user-defined basis section is as follows:

```
$basis
  X    0
  L    K    scale
   $\alpha_1$    $C_1^{L_{min}}$    $C_1^{L_{min}+1}$   ...   $C_1^{L_{max}}$ 
   $\alpha_2$    $C_2^{L_{min}}$    $C_2^{L_{min}+1}$   ...   $C_2^{L_{max}}$ 
   $\vdots$      $\vdots$      $\vdots$      $\ddots$      $\vdots$ 
   $\alpha_K$    $C_K^{L_{min}}$    $C_K^{L_{min}+1}$   ...   $C_K^{L_{max}}$ 
****
$end
```

where

$X$	Atomic symbol of the atom (atomic number not accepted)
$L$	Angular momentum symbol (S, P, SP, D, F, G)
$K$	Degree of contraction of the shell (integer)
$scale$	Scaling to be applied to exponents (default is 1.00)
$a_i$	Gaussian primitive exponent (positive real number)
$C_i^L$	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. Note that more than one contraction coefficient per line is one required for compound shells like SP. As with all Q-CHEM input deck information, all input is case-insensitive.

#### C.1.4 Keyword: *\$comment*

Note that the entire input deck is echoed to the output file, thus making the *\$comment* keyword largely redundant.

```
$comment
  User comments - copied to output file
$end
```

#### C.1.5 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,  $\hat{P}_0$ ,  $\hat{P}_1$ , ...,  $\hat{P}_{L-1}$ )
    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
****
$end
```

**Note:** (1) All of the information in the *\$ecp* block is case-insensitive.  
 (2) The  $L$  value may not exceed 4. That is, nothing beyond  $G$  projectors is allowed.  
 (3) The power of  $r$  (which includes the Jacobian  $r^2$  factor) must be 0, 1 or 2.

#### C.1.6 Keyword: *\$external\_charges*

All input should be given in atomic units.

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

**C.1.7 Keyword: *\$intracule***

```

$intracule
  int_type  0  Compute  $P(u)$  only
              1  Compute  $M(v)$  only
              2  Compute  $W(u, v)$  only
              3  Compute  $P(u)$ ,  $M(v)$  and  $W(u, v)$ 
              4  Compute  $P(u)$  and  $M(v)$ 
              5  Compute  $P(u)$  and  $W(u, v)$ 
              6  Compute  $M(v)$  and  $W(u, v)$ 
  u_points  Number of points, start, end.
  v_points  Number of points, start, end.
  moments  0-4  Order of moments to be computed ( $P(u)$  only).
  derivs   0-4  order of derivatives to be computed ( $P(u)$  only).
  accuracy  $n$   ( $10^{-n}$ ) specify accuracy of intracule interpolation table ( $P(u)$  only).
$end

```

**C.1.8 Keyword: *\$isotopes***

Note that masses should be given in atomic units.

```

$isotopes
  number_extra_loops  tp_flag
  number_of_atoms    [temp pressure]
  atom_number1 mass1
  atom_number2 mass2
  ...
$end

```

**C.1.9 Keyword: *\$multipole\_field***

Multipole fields are all in atomic units.

```

$multipole_field
  field_component1  value1
  field_component2  value2
  ...
$end

```

**C.1.10 Keyword: *\$nbo***

Refer to Chapter 10 and the NBO manual for further information. Note that the NBO *\$rem* variable must be set to ON to initiate the NBO package.

```

$nbo
  [ NBO options ]
$end

```

**C.1.11 Keyword: *\$occupied***

```

$occupied
  1  2  3  4 ... nalpha
  1  2  3  4 ... nbeta
$end

```

**C.1.12 Keyword: *\$opt***

Note that units are in Angstroms and degrees. Also see the summary in the next section of this Appendix.

```

$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

```

**C.1.13 Keyword: *\$svp***

```

$svp
<KEYWORD>=<VALUE>, <KEYWORD>=<VALUE>, ...
<KEYWORD>=<VALUE>
$end

```

For example, the section may look like this:

```
$svp
  RHOISO=0.001, DIELST=78.39, NPTLEB=110
$end
```

#### C.1.14 Keyword: *\$svpirf*

```
$svpirf
  <# point> <x point> <y point> <z point> <charge> <grid weight>
  <# point> <x normal> <y normal> <z normal>
$end
```

#### C.1.15 Keyword: *\$plots*

```
$plots
  One comment line
  Specification of the 3-D mesh of points on 3 lines:
     $N_x$   $x_{\min}$   $x_{\max}$ 
     $N_y$   $y_{\min}$   $y_{\max}$ 
     $N_z$   $z_{\min}$   $z_{\max}$ 
  A line with 4 integers indicating how many things to plot:
     $N_{\text{MO}}$   $N_{\text{Rho}}$   $N_{\text{Trans}}$   $N_{\text{DA}}$ 
  An optional line with the integer list of MO's to evaluate (only if  $N_{\text{MO}} > 0$ )
    MO(1) MO(2) ... MO( $N_{\text{MO}}$ )
  An optional line with the integer list of densities to evaluate (only if  $N_{\text{Rho}} > 0$ )
    Rho(1) Rho(2) ... Rho( $N_{\text{Rho}}$ )
  An optional line with the integer list of transition densities (only if  $N_{\text{Trans}} > 0$ )
    Trans(1) Trans(2) ... Trans( $N_{\text{Trans}}$ )
  An optional line with states for detachment/attachment densities (if  $N_{\text{DA}} > 0$ )
    DA(1) DA(2) ... DA( $N_{\text{DA}}$ )
$end
```

#### C.1.16 Keyword *\$van\_der\_waals*

Note: all radii are given in angstroms.

```
$van_der_waals
  1
  atomic_number    VdW_radius
$end
```

(alternative format)

```
$van_der_waals
  2
  sequential_atom_number    VdW_radius
$end
```

### C.1.17 Keyword: *\$xc\_functional*

```

$xc_functional
  X exchange_symbol coefficient
  X exchange_symbol coefficient
  ...
  C correlation_symbol coefficient
  C correlation_symbol coefficient
  ...
  K coefficient
$end

```

## C.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*

Values in angstroms; *value* > 0:

```
stre atom1 atom2 value
```

### *angles*

Values in degrees,  $0 \leq \text{value} \leq 180$ ; *atom2* is the middle atom of the bend:

```
bend atom1 atom2 atom3 value
```

### *out-of-plane-bends*

Values in degrees,  $-180 \leq \text{value} \leq 180$  *atom2*; angle between *atom4* and the *atom1-atom2-atom3* plane:

```
outp atom1 atom2 atom3 atom4 value
```

### *dihedral angles*

Values in degrees,  $-180 \leq \text{value} \leq 180$ ; angle the plane *atom1-atom2-atom3* makes with the plane *atom2-atom3-atom4*:

```
tors atom1 atom2 atom3 atom4 value
```

### *coplanar bends*

Values in degrees,  $-180 \leq \text{value} \leq 180$ ; bending of *atom1-atom2-atom3* in the plane *atom2-atom3-atom4*:

```
linc atom1 atom2 atom3 atom4 value
```

### *perpendicular bends*

Values in degrees,  $-180 \leq \text{value} \leq 180$ ; bending of *atom1-atom2-atom3* perpendicular to the plane *atom2-atom3-atom4*:

```
linp atom1 atom2 atom3 atom4 value
```

### C.2.1 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
```

### C.3 *\$rem* Variable List

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

A short description of what the variable controls

TYPE:

Defines the variable as either INTEGER, LOGICAL or STRING.

DEFAULT:

Describes Q-CHEM's internal default, if any exists.

OPTIONS:

Lists options available for the user

RECOMMENDATION:

Gives a quick recommendation.

#### C.3.1 General

BASIS	BASIS_LIN_DEP_THRESH
EXCHANGE	CORRELATION
ECP	JOBTYPE
PURECART	

#### C.3.2 SCF Control

BASIS2	BASISPROJTYPE
DIIS_PRINT	DIIS_SUBSPACE_SIZE
DIRECT_SCF	INCFOCK
MAX_DIIS_CYCLES	MAX_SCF_CYCLES
PSEUDO_CANONICAL	SCF_ALGORITHM
SCF_CONVERGENCE	SCF_FINAL_PRINT
SCF_GUESS	SCF_GUESS_MIX
SCF_GUESS_PRINT	SCF_PRINT
THRESH	THRESH_DIIS_SWITCH
UNRESTRICTED	VARTHRESH

### C.3.3 DFT Options

CORRELATION	EXCHANGE
FAST_XC	INC_DFT
INCDFT_DENDIFF_THRESH	INCDFT_GRIDDIFF_THRESH
INCDFT_DENDIFF_VARTHRESH	INCDFT_GRIDDIFF_VARTHRESH
XC_GRID	XC_SMART_GRID

### C.3.4 Large Molecules

CFMM_ORDER	DIRECT_SCF
EPAO_ITERATE	EPAO_WEIGHTS
GRAIN	INCFOCK
INTEGRAL_2E_OPR	INTEGRALS_BUFFER
LIN_K	MEM_STATIC
MEM_TOTAL	METECO
OMEGA	PAO_ALGORITHM
PAO_METHOD	THRESH
VARTHRESH	

### C.3.5 Correlated Methods

See also *\$rem* variables that start *CC\_* which control the coupled-cluster package within Q-CHEM.

AO2MO_DISK	CD_ALGORITHM
CORE_CHARACTER	CORRELATION
MEM_STATIC	MEM_TOTAL
N_FROZEN_CORE	N_FROZEN_VIRTUAL
PRINT_CORE_CHARACTER	

### C.3.6 Excited States

CC_NHIGHSPIN	CC_NLOWSPIN
CC_REFSYM	CC_SPIN_FLIP
CC_SYMMETRY	CC_TRANS_PROP
CIS_CONVERGENCE	CIS_GUESS_DISK
CIS_GUESS_DISK_TYPE	CIS_NROOTS
CIS_RELAXED_DENSITY	CIS_SINGLET
CIS_STATE_DERIV	CIS_TRIPLET
MAX_CIS_CYCLES	RPA
CC_STATE_DERIV	XCIS

### C.3.7 Geometry Optimizations

CIS_STATE_DERIV	FDIFF_STEPSIZE
GEOM_OPT_COORDS	GEOM_OPT_DMAX
GEOM_OPTHESSIAN	GEOM_OPT_LINEAR_ANGLE
GEOM_OPT_MAX_CYCLES	GEOM_OPT_MAX_DIIS
GEOM_OPT_MODE	GEOM_OPT_PRINT
GEOM_OPTSYMFLAG	GEOM_OPT_PRINT
GEOM_OPTTOL_ENERGY	GEOM_OPT_TOL_DISPLACEMENT
GEOM_OPT_TOL_ENERGY	GEOM_OPT_TOL_GRADIENT
GEOMP_OPT_UPDATE	IDERIV
JOBTYPE	SCF_GUESS_ALWAYS
CC_STATE_DERIV	

### C.3.8 Vibrational Analysis

DORAMAN	CPSCF_NSEG
FDIFF_STEPSIZE	IDERIV
ISOTOPE	JOBTYPE
VIBMAN_PRINT	

### C.3.9 Reaction Coordinate Following

JOBTYPE	RPATH_COORDS
RPATH_DIRECTION	RPATH_MAX_CYCLES
RPATH_MAX_STEPSIZE	RPATH_PRINT
RPATH_TOL_DISPLACEMENT	

### C.3.10 NMR Calculations

D_CPSCF_PERTNUM	D_SCF_CONV_1
D_SCF_CONV_2	D_SCF_DIIS
D_SCF_MAX_1	D_SCF_MAX_2
JOBTYPE	

### C.3.11 Wavefunction Analysis and Molecular Properties

CHEMSOL	CHEMSOL_EFIELD
CHEMSOL_NN	CHEM_SOL_PRINT
CIS_RELAXED_DENSITY	IGDESP
INTRACULE	MULTIPOLE_ORDER
NBO	POP_MULLIKEN
PRINT_DIST_MATRIX	PRINT_ORBITALS
READ_VDW	SOLUTE_RADIUS
SOLVENT_DIELECTRIC	STABILITY_ANALYSIS
WAVEFUNCTION_ANALYSIS	WRITE_WFN

### C.3.12 Symmetry

CC_SYMMETRY	CC_REF_SYM
SYM_IGNORE	SYMMETRY
SYMMETRY_DECOMPOSITION	SYM_TOL

### C.3.13 Printing Options

CC_PRINT	CHEMSOL_PRINT
DIIS_PRINT	GEOM_OPT_PRINT
MOM_PRINT	PRINT_CORE_CHARACTER
PRINT_DIST_MATRIX	PRINT_GENERAL_BASIS
PRINT_ORBITALS	RPATH_PRINT
SCF_FINAL_PRINT	SCF_GUESS_PRINT
SCF_PRINT	VIBMAN_PRINT
WRITE_WFN	

### C.3.14 Resource Control

AO2MO_DISK	CC_CLK_TNSR_BUFFSIZE
CC_TMPBUFFSIZE	DIRECT_SCF
INTEGRALS_BUFFER	MAX_SUB_FILE_NUM
MEM_TOTAL	MEM_STATIC

### C.3.15 Alphabetical Listing

#### AIMD\_FICT\_MASS

Specifies the value of the fictitious electronic mass  $\mu$ , in atomic units, where  $\mu$  has dimensions of  $(\text{energy}) \times (\text{time})^2$ .

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified

RECOMMENDATION:

Values in the range of 50–200 *a.u.* have been employed in test calculations; consult [12] for examples and discussion.

#### AIMD\_INIT\_VELOC

Specifies the method for selecting initial nuclear velocities.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

THERMAL Random sampling of nuclear velocities from a Maxwell–Boltzmann distribution. The user must specify the temperature in Kelvin via the *\$rem* variable AIMD\_TEMP.

ZPE Choose velocities in order to put zero–point vibrational energy into each normal mode, with random signs. This option requires that a frequency job to be run beforehand.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *\$velocity* section.

#### AIMD\_METHOD

Selects an *ab initio* molecular dynamics algorithm.

TYPE:

STRING

DEFAULT:

BOMD

OPTIONS:

BOMD Born–Oppenheimer molecular dynamics.

CURVY Curvy–steps Extended Lagrangian molecular dynamics.

RECOMMENDATION:

BOMD yields exact classical molecular dynamics, provided that the energy is tolerably conserved. ELMD is an approximation to exact classical dynamics whose validity should be tested for the properties of interest.

**AIMD\_MOMENTS**

Requests that multipole moments be output at each time step.

TYPE:

INTEGER

DEFAULT:

0 Do not output multipole moments.

OPTIONS:

$n$  Output the first  $n$  multipole moments.

RECOMMENDATION:

None

**AIMD\_STEPS**

Specifies the requested number of molecular dynamics steps.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

None.

**AIMD\_TEMP**

Specifies a temperature (in Kelvin) for Maxwell-Boltzmann velocity sampling.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified number of Kelvin.

RECOMMENDATION:

This variable is only useful in conjunction with `AIMD_INIT_VELOC = THERMAL`. Note that the simulations are run at constant energy, rather than constant temperature, so the mean nuclear kinetic energy will fluctuate in the course of the simulation.

**ANHAR**

Performing various nuclear vibrational theory (TOSH, VPT2, VCI) calculations to obtain vibrational anharmonic frequencies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Carry out the anharmonic frequency calculation.

FALSE Do harmonic frequency calculation.

RECOMMENDATION:

Since this calculation involves the third and fourth derivatives at the minimum of the potential energy surface, it is recommended that the GEOM\_OPT\_TOL\_DISPLACEMENT, GEOM\_OPT\_TOL\_GRADIENT and GEOM\_OPT\_TOL\_ENERGY tolerances are set tighter. Note that VPT2 calculations will fail if the system involves resonances (either accidental or due to symmetry). See the VCI *\$rem* variable for more details about increasing the accuracy of anharmonic calculations.

**AO2MO\_DISK**

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

TYPE:

INTEGER

DEFAULT:

2000 Corresponding to 2000 Mb.

OPTIONS:

*n* User-defined number of megabytes.

RECOMMENDATION:

Should be set as large as possible, discussed in Section 5.3.1.

**BASIS2**

Sets the small basis set to use in basis set projection.

TYPE:

STRING

DEFAULT:

No second basis set default.

OPTIONS:

Symbol Use standard basis sets as per Chapter 7.

RECOMMENDATION:

BASIS2 should be smaller than BASIS. There is little advantage to using a basis larger than a minimal basis.

**BASISPROJTYPE**

Determines which method to use when projecting the density matrix of BASIS2

TYPE:

STRING

DEFAULT:

FOPPROJECTION

OPTIONS:

FOPPROJECTION Construct the Fock matrix in the second basis

OVPROJECTION Projects MO's from BASIS2 to BASIS.

RECOMMENDATION:

None

**BASIS\_LIN\_DEP\_THRESH**

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of  $10^{-6}$

OPTIONS:

$n$  Sets the threshold to  $10^{-n}$

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in your basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

**BASIS**

Specifies the basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen User defined (*\$basis* keyword required).

Symbol Use standard basis sets as per Chapter 7.

Mixed Use a mixture of basis sets (see Chapter 7).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

**CC\_AMPL\_RESP**

If set to TRUE, adds amplitude response terms to one-particle and two-particle CCSD density matrices before calculation of properties. CC\_PROP must be set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no amplitude response terms will be added to density matrices)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost is always about the cost of an analytic gradient calculation, independent of whether or not the two-particle properties are requested. Besides, adding amplitude response terms without orbital response will unlikely improve the quality of the properties. However, it can be used for debugging purposes.

**CC\_BLK\_TNSR\_BUFFSIZE**

Specifies the maximum size, in Mb, of the buffers for in-core storage of block-tensors.

TYPE:

INTEGER

DEFAULT:

50% of MEM\_TOTAL

OPTIONS:

$n$  Integer number of 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\_FINAL**

Whether to semi-canonicalize orbitals at the end of the ground state calculation.

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

$n$  User-defined integer

RECOMMENDATION:

Smaller values can be tried in cases that do not converge.

**CC\_CANONIZE**

Whether to semi-canonicalize orbitals at the start of the calculation (*i.e.* Fock matrix is diagonalized in each orbital subspace)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Should not normally have to be altered.

**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) [ $10^{-n}$ ].

TYPE:

INTEGER

DEFAULT:

8 Energies.

8 Gradients.

OPTIONS:

$n$  Corresponding to  $10^{-n}$  convergence criterion.

RECOMMENDATION:

None

**CC\_DCONVERGENCE**

Convergence criterion for the RMS residuals of excited state vectors

TYPE:

INTEGER

DEFAULT:

5 Corresponding to  $10^{-5}$

OPTIONS:

$n$  Corresponding to  $10^{-n}$  convergence criterion

RECOMMENDATION:

Use default. Should normally be set to the same value as CC\_DTHRESHOLD.

**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.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

$n$  User-defined integer

RECOMMENDATION:

None

**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.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

$N$  User-defined integer

RECOMMENDATION:

None

**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.

TYPE:

DOUBLE

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

$abcde$  Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

None

**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.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

 $n$  User-defined integer

RECOMMENDATION:

None

**CC\_DIIS\_SIZE**

Specifies the maximum size of the DIIS space.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

 $n$  User-defined integer

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.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

 $n$  User-defined

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\_DIIS**

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

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Activates procedure 2 initially, and procedure 1 when gradients are smaller than DIIS12.SWITCH.

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\_DIP**

Initializes a EOM-DIP-CCSD calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform an EOM-DIP-CCSD calculation

1 Do perform an EOM-DIP-CCSD calculation

RECOMMENDATION:

None

**CC\_DMAXITER**

Maximum number of iteration allowed for Davidson diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

$n$  User-defined number of iterations

RECOMMENDATION:

Default is usually sufficient

**CC\_DMAXVECTORS**

Specifies maximum number of vectors in the subspace for the Davidson diagonalization.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

$n$  Up to  $n$  vectors per root before the subspace is reset

RECOMMENDATION:

Larger values increase storage but accelerate and stabilize convergence.

**CC\_DOV\_THRESH**

Specifies the 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.

TYPE:

DOUBLE

DEFAULT:

2502 Corresponding to 0.25

OPTIONS:

$abcde$  Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

Increase to 0.5 or 0.75 for non-convergent coupled-cluster calculations.

**CC\_DO\_CISDT**

Controls the calculation of CISDT

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute triples (do CISD only)

1 Do compute the triples correction (full CISDT)

RECOMMENDATION:

None

**CC\_DO\_DISCONNECTED**

Determines whether disconnected terms included in the EOM-OD equations

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Inclusion of disconnected terms has very small effects and is not necessary.

**CC\_DO\_SMALL\_TRIPLES**

Preconverges the EOM-CCSD block of the Hamiltonian by doing several EOM-CCSD iterations before switching to EOM(3)-CCSD using the EOM-CCSD vectors as a guess for EOM-CC(2,3)

TYPE:

INTEGER

DEFAULT:

0 Do not do any EOM-CCSD iterations

OPTIONS:

$n$  Do  $n$  iterations

RECOMMENDATION:

None

**CC\_DO\_TRIPLES**

This keyword initializes a EOM-CC(2,3) calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No EOM-CC(2,3) calculation

1 Perform a EOM-CC(2,3) calculation

RECOMMENDATION:

None

**CC\_DTHRESHOLD**

Specifies threshold for including a new expansion vector in the iterative Davidson diagonalization. Their norm must be above this threshold.

TYPE:

INTEGER

DEFAULT:

00105 Corresponding to 0.00001

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

Use default unless converge problems are encountered. Should normally be set to the same values as CC\_DCONVERGENCE, if convergence problems arise try setting to a value less than CC\_DCONVERGENCE.

**CC\_EA**

If TRUE, calculates EOM-EA-CCSD excitation energies and properties (upon request).

TYPE:

LOGICAL

DEFAULT:

FALSE (no EOM-EA-CCSD states will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

See comments above for CC\_IP

**CC\_EOM\_AMPL\_RESP**

If set to TRUE, adds amplitude response terms to one-particle and two-particle EOM-CCSD density matrices before calculation of properties. CC\_EXSTATES\_PROP must be set to TRUE.

TYPE:

INTEGER

DEFAULT:

FALSE (no amplitude response terms will be added to density matrices)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost is always about the cost of an analytic gradient calculation for each state, independent of whether or not the two-particle properties are requested. Besides, adding amplitude response terms without orbital response will unlikely improve the quality of the properties. However, it can be used for debugging purposes.

**CC\_EOM\_FULL\_RESP**

If set to TRUE, adds both amplitude and orbital response terms to one- and two-particle EOM-CCSD density matrices before calculation of the properties. CC\_EXSTATES\_PROP must be set to TRUE. If both CC\_EOM\_AMPL\_RESP=TRUE and CC\_EOM\_FULL\_RESP=TRUE, the CC\_EOM\_AMPL\_RESP=TRUE will be ignored.

TYPE:

LOGICAL

DEFAULT:

FALSE No orbital response terms will be added to density matrices, and no amplitude response will be calculated if CC\_EOM\_AMPL\_RESP = FALSE

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost for the full response properties calculation is about the same as the cost of the analytic gradient for each state. Adding full response terms improves quality of calculated properties, but usually it is a small but expensive correction. Use it only if you really need accurate properties.

**CC\_EOM\_TWOPART\_PROP**

Request for calculation of non-relaxed two-particle EOM-CCSD target state properties. The two-particle properties currently include  $\langle S^2 \rangle$ . The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of  $\langle S^2 \rangle$ . The variable CC\_EXSTATES\_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are extremely computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation for each state). Do not request the two-particle properties unless you really need them.

**CC\_EXSTATES\_PROP**

Whether or not the non-relaxed (expectation value) one-particle EOM-CCSD target state properties will be calculated. The properties currently include permanent dipole moment, the second moments  $\langle X^2 \rangle$ ,  $\langle Y^2 \rangle$ , and  $\langle Z^2 \rangle$  of electron density, and the total  $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$  (in atomic units). Incompatible with JOBTYP=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (EOM-CCSD equations for the left eigenvectors) need to be solved for properties, approximately doubling the cost of calculation for each irrep. Sometimes the equations for left and right eigenvectors converge to different sets of target states. In this case, the simultaneous iterations of left and right vectors will diverge, and the properties for several or all the target states may be incorrect! The problem can be solved by varying the number of requested states, specified with CC\_NLOWSPIN and CC\_NHIGHSPIN, or the number of guess vectors (CC\_NGUESS\_SINGLES). The cost of the one-particle properties calculation itself is low. The one-particle density of an EOM-CCSD target state can be analyzed with NBO package by specifying the state with CC\_REFSYM and CC\_STATE\_DERIV and requesting NBO=TRUE and CC\_EXSTATES\_PROP=TRUE.

**CC\_E\_CONV**

Convergence desired on the change in total energy, between iterations.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n$   $10^{-n}$  convergence criterion.

RECOMMENDATION:

None

**CC\_FULL\_RESP**

If set to TRUE, adds both amplitude and orbital response terms to one- and two-particle CCSD density matrices before calculation of the properties. CC\_PROP must be set to TRUE. If both CC\_AMPL\_RESP=TRUE and CC\_FULL\_RESP=TRUE, the CC\_AMPL\_RESP=TRUE will be ignored.

TYPE:

LOGICAL

DEFAULT:

FALSE No orbital response terms will be added to density matrices, and no amplitude response will be calculated if CC\_AMPL\_RESP = FALSE

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The cost for the full response properties calculation is about the same as the cost of the analytic gradient. Adding full response terms improves quality of calculated properties, but usually it is a small but expensive correction. Use it only if you really need accurate properties.



**CC\_HESS\_THRESH**

Minimum allowed value for the orbital Hessian. Smaller values are replaced by this constant.

TYPE:

DOUBLE

DEFAULT:

102 Corresponding to 0.01

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

RECOMMENDATION:

None

**CC\_INCL\_CORE\_CORR**

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

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

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).

**CC\_IP**

If TRUE, calculates EOM–IP–CCSD excitation energies and properties (upon request).

TYPE:

LOGICAL

DEFAULT:

FALSE (no EOM–IP–CCSD states will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

When CC\_IP is set to TRUE, it can change the convergence of Hartree–Fock iterations compared to the same job with CC\_IP=FALSE, because a very diffuse basis function is added to a center of symmetry *before* the Hartree–Fock iterations start. For the same reason, BASIS2 keyword is incompatible with CC\_IP=TRUE. In order to read Hartree–Fock guess from a previous job, you must specify CC\_IP=TRUE (even if you do not request for any correlation or excited states) in that previous job. Currently, the second moments of electron density are incorrect for the EOM–IP–CCSD target states. CC\_NLOWSPIN should also be set to specify the number of target EOM-IP states.

**CC\_ITERATE\_ON**

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 space iterations is given by this option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Up to  $n$  occupied-virtual iterations per overall cycle

RECOMMENDATION:

Can be useful for non-convergent active space calculations

**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.

TYPE:

INTEGER

DEFAULT:

0 No “mixed” iterations

OPTIONS:

$n$  Up to  $n$  occupied-virtual iterations per overall cycle

RECOMMENDATION:

Can be useful for non-convergent active space calculations.

**CC\_MAXITER**

Maximum number of iterations to optimize the coupled-cluster energy.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

$n$  up to  $n$  iterations to achieve convergence.

RECOMMENDATION:

None

**CC\_MP2NO\_GRAD**

If CC\_MP2NO\_GUESS is TRUE, what kind of one-particle density matrix is used to make the guess orbitals?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE 1 PDM from MP2 gradient theory.

FALSE 1 PDM expanded to 2<sup>nd</sup> order in perturbation theory.

RECOMMENDATION:

The two definitions give generally similar performance.

**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.

TYPE:

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.

RECOMMENDATION:

None

**CC\_NGUESS\_DOUBLES**

Specifies number of excited state guess vectors which are double excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

*n* Include *n* guess vectors that are double excitations

RECOMMENDATION:

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.

TYPE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

*n* Include *n* guess vectors that are single excitations

RECOMMENDATION:

Should be greater or equal than the number of excited states requested.

**CC\_NHIGHSPIN**

Sets the number of high-spin excited state roots to find. Works only for singlet reference states and triplet excited states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

[*i*, *j*, *k*...] Find *i* excited states in the first symmetry, *j* states in the second symmetry *etc.*

RECOMMENDATION:

None

**CC\_NLOWSPIN**

Sets the number of low-spin excited state roots to find. In the case of closed-shell reference states, excited singlet states will be found. For any other reference state, all states (*e.g.* both singlet and triplet) will be calculated.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

[*i*, *j*, *k*...] Find *i* excited states in the first symmetry, *j* states in the second symmetry *etc.*

RECOMMENDATION:

None

**CC\_ORBS\_PER\_BLOCK**

Specifies target (and maximum) size of blocks in orbital space.

TYPE:

INTEGER

DEFAULT:

16

OPTIONS:

*n* Orbital block size of *n* orbitals.

RECOMMENDATION:

None

**CC\_PRECONV\_DOUBLES**

When TRUE, doubly-excited vectors are converged prior to a full excited states calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Occasionally necessary to ensure a doubly excited state is found.

**CC\_PRECONV\_FZ**

In active space methods, whether to preconverge other wavefunction variables for fixed initial guess of active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No pre-iterations before active space optimization begins.

*n* Maximum number of pre-iterations via this procedure.

RECOMMENDATION:

None

**CC\_PRECONV\_SD**

Solves the EOM-CCSD equations, prints energies, then uses EOM-CCSD vectors as initial vectors in EOM-CC(2,3)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Turns the preconverging on

RECOMMENDATION:

Turning this option on is recommended

**CC\_PRECONV\_SINGLES**

When TRUE, singly-excited vectors are converged prior to a full excited states calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

**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.

TYPE:

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\_PRECONV\_T2Z**

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

TYPE:

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\_PROP**

Whether or not the non-relaxed (expectation value) one-particle CCSD properties will be calculated. The properties currently include permanent dipole moment, the second moments  $\langle X^2 \rangle$ ,  $\langle Y^2 \rangle$ , and  $\langle Z^2 \rangle$  of electron density, and the total  $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$  (in atomic units). Incompatible with JOBTYP=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (lambda CCSD equations) for properties with the cost approximately the same as CCSD equations. Use default if you do not need properties. The cost of the properties calculation itself is low. The CCSD one-particle density can be analyzed with NBO package by specifying NBO=TRUE, CC\_PROP=TRUE and JOBTYP=FORCE.

**CC\_QCCD\_THETA\_SWITCH**

QCCD calculations switch from OD to QCCD when the rotation gradient is below this threshold [ $10^{-n}$ ]

TYPE:

INTEGER

DEFAULT:

2  $10^{-2}$  switchover

OPTIONS:

$n$   $10^{-n}$  switchover

RECOMMENDATION:

None

**CC\_REFSYM**

Together with CC\_STATE\_DERIV, selects which EOM state is to be optimized.

Note that the symmetries are numbered from 0 up, as are the roots.

TYPE:

INTEGER

DEFAULT:

0 Corresponding to the first symmetry

OPTIONS:

$n$   $n < \text{order of the symmetry point group}$ .

RECOMMENDATION:

None

**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.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

$n$   $n$  iterations between resetting orbital rotations to zero.

RECOMMENDATION:

None

**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

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

**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.

TYPE:

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\_RESTR\_AMPL**

Controls the restriction on amplitudes if there are restricted orbitals

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 All amplitudes are in the full space

1 Amplitudes are restricted, if there are restricted orbitals

RECOMMENDATION:

None

**CC\_RESTR\_TRIPLES**

Controls which space the triples correction is computed in

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Triples are computed in the full space

1 Triples are restricted to the active space

RECOMMENDATION:

None

**CC\_REST\_OCC**

Sets the number of restricted occupied orbitals including frozen occupied orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Restrict  $n$  occupied orbitals.

RECOMMENDATION:

None

**CC\_REST\_VIR**

Sets the number of restricted virtual orbitals including frozen virtual orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Restrict  $n$  virtual orbitals.

RECOMMENDATION:

None



**CC\_SPIN\_FLIP\_MS**

Controls the spin projection of the excitation operator

TYPE:

INTEGER

DEFAULT:

0 For  $\alpha \rightarrow \alpha$  excitation

1 For  $\alpha \rightarrow \beta$  excitations

OPTIONS:

2 Allows double spin-flip calculations, *i.e.*, two electrons are flipped in the excitation operator:  $\alpha\alpha \rightarrow \beta\beta$ .

RECOMMENDATION:

This keyword should be used together with `CC_SPIN_FLIP = 1`.

**CC\_SPIN\_FLIP**

Selects whether do perform a standard excited state calculation, or a spin-flip calculation. Spin multiplicity should be set to 2 for systems with an odd number of electrons, and 3 for systems with an even number of electrons.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

**CC\_STATE\_DERIV**

Together with `CC_REFSYM`, selects which EOM state is to be considered for optimization or property calculations. If `CC_TRANS_PROP` is also selected, the transition properties will be calculated between this state and all the other states.

TYPE:

INTEGER

DEFAULT:

-1 Turns off optimization/property calculations

OPTIONS:

$n$  Optimize the  $n^{\text{th}}$  excited state.

RECOMMENDATION:

None

**CC\_SYMMETRY**

Controls the use of symmetry in coupled-cluster calculations

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Use the point group symmetry of the molecule

FALSE Do not use point group symmetry (all states will be of  $A$  symmetry).

RECOMMENDATION:

Should be turned off for any finite difference calculations, *e.g.* second derivatives.

**CC\_THETA\_CONV**

Convergence criterion on the RMS difference between successive sets of orbital rotation angles [ $10^{-n}$ ].

TYPE:

INTEGER

DEFAULT:

5 Energies

6 Gradients

OPTIONS:

$n$   $10^{-n}$  convergence criterion.

RECOMMENDATION:

Use default

**CC\_THETA\_GRAD\_CONV**

Convergence desired on the RMS gradient of the energy with respect to orbital rotation angles [ $10^{-n}$ ].

TYPE:

INTEGER

DEFAULT:

7 Energies

8 Gradients

OPTIONS:

$n$   $10^{-n}$  convergence criterion.

RECOMMENDATION:

Use default

**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.

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

*abcde* Integer code is mapped to  $abc \times 10^{-de}$

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 01001 translates to 0.1

**CC\_TMPBUFFSIZE**

Maximum size, in Mb, of additional buffers for temporary arrays used to work with individual blocks or matrices.

TYPE:

INTEGER

DEFAULT:

3% of MEM\_TOTAL

OPTIONS:

*n* Integer number of Mb

RECOMMENDATION:

Should not be smaller than the size of the largest possible block.

**CC\_TRANS\_PROP**

Whether or not the transition dipole moment (in atomic units) and oscillator strength for the EOM-CCSD target states will be calculated. By default, the transition dipole moment is calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment between a set of EOM-CCSD states and another EOM-CCSD state, the CC\_REFSYM and CC\_STATE\_DERIV must be specified for this state.

TYPE:

LOGICAL

DEFAULT:

FALSE (no transition dipole and oscillator strength will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case if transition properties between the CCSD reference and EOM-CCSD target states are requested) need to be solved for transition properties, approximately doubling the computational cost. The cost of the transition properties calculation itself is low.

**CC\_TWOPART\_PROP**

Request for calculation of non-relaxed two-particle CCSD properties. The two-particle properties currently include  $\langle S^2 \rangle$ . The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of  $\langle S^2 \rangle$ . The variable CC\_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are extremely computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

**CC\_T\_CONV**

Convergence criterion on the RMS difference between successive sets of coupled-cluster doubles amplitudes [ $10^{-n}$ ]

TYPE:

INTEGER

DEFAULT:

8 energies

10 gradients

OPTIONS:

$n$   $10^{-n}$  convergence criterion.

RECOMMENDATION:

Use default

**CC\_Z\_CONV**

Convergence criterion on the RMS difference between successive doubles  $Z$ -vector amplitudes [ $10^{-n}$ ].

TYPE:

INTEGER

DEFAULT:

8 Energies

10 Gradients

OPTIONS:

$n$   $10^{-n}$  convergence criterion.

RECOMMENDATION:

Use Default

**CC\_PRINT**

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

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

$0 \rightarrow 7$  higher values can lead to deforestation...

RECOMMENDATION:

Increase if you need more output and don't like trees

**CD\_ALGORITHM**

Determines the algorithm for MP2 integral transformations.

TYPE:

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:

Semi-direct is usually most efficient, and will normally be chosen by default.

**CFMM\_ORDER**

Controls the order of the multipole expansions in CFMM calculation.

TYPE:

INTEGER

DEFAULT:

15 For single point SCF accuracy

25 For tighter convergence (optimizations)

OPTIONS:

$n$  Use multipole expansions of order  $n$

RECOMMENDATION:

Use default.

**CHEMSOL\_EFIELD**

Determines how the solute charge distribution is approximated in evaluating the electrostatic field of the solute.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Exact solute charge distribution is used.

0 Solute charge distribution is approximated by Mulliken atomic charges.

This is a faster, but less rigorous procedure.

RECOMMENDATION:

None.

**CHEMSOL\_NN**

Sets the number of grids used to calculate the average hydration free energy.

TYPE:

INTEGER

DEFAULT:

5  $\Delta G_{\text{hydr}}$  will be averaged over 5 different grids.

OPTIONS:

$n$  Number of different grids (Max = 20).

RECOMMENDATION:

None.

**CHEMSOL\_PRINT**

Controls printing in the CHEMSOL part of the Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Limited printout.

1 Full printout.

RECOMMENDATION:

None

**CHEMSOL**

Controls the use of CHEMSOL in Q-CHEM.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use CHEMSOL.

1 Perform a CHEMSOL calculation.

RECOMMENDATION:

None

**CISTR\_PRINT**

Controls level of output

TYPE:

LOGICAL

DEFAULT:

FALSE Minimal output

OPTIONS:

TRUE Increase output level

RECOMMENDATION:

None

**CIS\_CONVERGENCE**

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

TYPE:

INTEGER

DEFAULT:

6 CIS convergence threshold  $10^{-6}$

OPTIONS:

$n$  Corresponding to  $10^{-n}$

RECOMMENDATION:

None

**CIS\_GUESS\_DISK\_TYPE**

Determines the type of guesses to be read from disk

TYPE:

INTEGER

DEFAULT:

Nil

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATION:

Must be specified if CIS\_GUESS\_DISK is TRUE.

**CIS\_GUESS\_DISK**

Read the CIS guess from disk (previous calculation)

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Create a new guess

True Read the guess from disk

RECOMMENDATION:

Requires a guess from previous calculation.

**CIS\_N\_ROOTS**

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

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

$n$   $n > 0$  Looks for  $n$  CIS excited states

RECOMMENDATION:

None

**CIS\_RELAXED\_DENSITY**

Use the relaxed CIS density for attachment/detachment density analysis

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the relaxed CIS density in analysis

True Use the relaxed CIS density in analysis.

RECOMMENDATION:

None

**CIS\_SINGLET**

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

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for singlet states

FALSE Do not solve for singlet states.

RECOMMENDATION:

None

**CIS\_STATE\_DERIV**

Sets CIS state for excited state optimizations and vibrational analysis

TYPE:

INTEGER

DEFAULT:

0 Does not select any of the excited states

OPTIONS:

$n$  Select the  $n^{\text{th}}$  state.

RECOMMENDATION:

Check to see that the states do no change order during an optimization

**CIS\_TRIPLETS**

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

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for triplet states

FALSE Do not solve for triplet states.

RECOMMENDATION:

None

**CORE\_CHARACTER**

Selects how the core orbitals are determined in the frozen-core approximation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy-based definition.

1-4 Use Muliken-based definition (see Table 5.3.2 for details).

RECOMMENDATION:

Use default, unless performing calculations on molecules with heavy elements.



**CORRELATION**

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

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

None	No Correlation.
VWN	Vosko–Wilk–Nusair parameterization #5
LYP	Lee–Yang–Parr
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	

RECOMMENDATION:

Consult the literature and reviews for guidance

**CPSCF\_NSEG**

Controls the number of segments used to calculate the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not solve the CPSCF equations in segments.

*n* User–defined. Use *n* segments when solving the CPSCF equations.

RECOMMENDATION:

Use default.

**CUTOCC**

Specifies occupied orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTOCC/100

DEFAULT:

50

OPTIONS:

0-200

RECOMMENDATION:

None

**CUTVIR**

Specifies virtual orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTVIR/100

DEFAULT:

0 No truncation

OPTIONS:

0-100

RECOMMENDATION:

None

**CVGLIN**

Convergence criterion for solving linear equations by the conjugate gradient iterative method (relevant if LINEQ=1 or 2).

TYPE:

FLOAT

DEFAULT:

1.0E-7

OPTIONS:

Real number specifying the actual criterion.

RECOMMENDATION:

The default value should be used unless convergence problems arise.

**DEUTERATE**

Requests that all hydrogen atoms be replaced with deuterium.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not replace hydrogens.

OPTIONS:

TRUE Replace hydrogens with deuterium.

RECOMMENDATION:

Replacing hydrogen atoms reduces the fastest vibrational frequencies by a factor of 1.4, which allow for a larger fictitious mass and time step in ELMD calculations. There is no reason to replace hydrogens in BOMD calculations.

**DIELST**

The static dielectric constant.

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

real number specifying the constant.

RECOMMENDATION:

The default value 78.39 is appropriate for water solvent.

**DIIS\_ERR\_RMS**

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Use default, the maximum error provides a more reliable criterion.

**DIIS\_PRINT**

Controls the output from DIIS SCF optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

- 0 Minimal print out.
- 1 Chosen method and DIIS coefficients and solutions.
- 2 Level 1 plus changes in multipole moments.
- 3 Level 2 plus Multipole moments.
- 4 Level 3 plus extrapolated Fock matrices.

RECOMMENDATION:

Use default

**DIIS\_SUBSPACE\_SIZE**

Controls the size of the DIIS subspace during the SCF.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

RECOMMENDATION:

None

**DIRECT\_SCF**

Controls direct SCF.

TYPE:

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.

**DORAMAN**

Controls calculation of Raman intensities. Requires JOBTYP to be set to FREQ

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate Raman intensities.

TRUE Do calculate Raman intensities.

RECOMMENDATION:

None

**DUAL\_BASIS\_ENERGY**

Activates dual-basis SCF (HF or DFT) energy correction.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Analytic first derivative available for HF (see JOBTYP)

Can be used in conjunction with MP2 or RI-MP2

See BASIS, BASIS2, BASISPROJTYPE

RECOMMENDATION:

Use Dual-Basis to capture large-basis effects at smaller basis cost. Particularly useful with RI-MP2, in which HF often dominates. Use only proper subsets for small-basis calculation.

**D\_CPSCF\_PERTNUM**

Specifies whether to do the perturbations one at a time, or all together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perturbed densities to be calculated all together.

1 Perturbed densities to be calculated one at a time.

RECOMMENDATION:

None

**D\_SCF\_CONV\_1**

Sets the convergence criterion for the level-1 iterations. This preconditions the density for the level-2 calculation, and does not include any two-electron integrals.

TYPE:

INTEGER

DEFAULT:

4 corresponding to a threshold of  $10^{-4}$ .

OPTIONS:

$n < 10$  Sets convergence threshold to  $10^{-n}$ .

RECOMMENDATION:

The criterion for level-1 convergence must be less than or equal to the level-2 criterion, otherwise the D-CPSCF will not converge.

**D\_SCF\_CONV\_2**

Sets the convergence criterion for the level-2 iterations.

TYPE:

INTEGER

DEFAULT:

4 Corresponding to a threshold of  $10^{-4}$ .

OPTIONS:

$n < 10$  Sets convergence threshold to  $10^{-n}$ .

RECOMMENDATION:

None

**D\_SCF\_DIIS**

Specifies the number of matrices to use in the DIIS extrapolation in the D-CPSCF.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

$n = 0$  specifies no DIIS extrapolation is to be used.

RECOMMENDATION:

Use the default.

**D\_SCF\_MAX\_1**

Sets the maximum number of level-1 iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

$n$  User defined.

RECOMMENDATION:

Use default.

**D\_SCF\_MAX\_2**

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

*n* User defined.

RECOMMENDATION:

Use default.

**ECP**

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No pseudopotential

OPTIONS:

General, Gen User defined. (*\$ecp* keyword required)

Symbol Use standard pseudopotentials discussed above.

RECOMMENDATION:

Pseudopotentials are recommended for first row transition metals and heavier elements. Consult the reviews for more details.

**ENG\_DER**

Controls what types of gradient information are used to compute higher derivatives. The default uses a combination of energy, gradient and Hessian information, which makes the force field calculation faster.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy, gradient, and Hessian information.

1 Use energy information only.

1 Use Hessian information only.

RECOMMENDATION:

When the molecule is larger than benzene, ENG\_DER=2 is recommended.

**EPAO\_ITERATE**

Controls iterations for EPAO calculations (see PAO\_METHOD).

TYPE:

INTEGER

DEFAULT:

0 Use uniterated EPAOs based on atomic blocks of SPS.

OPTIONS:

*n* Optimize the EPAOs 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 optimized EPAOs in which case a value of  $n = 500$  is reasonable.

**EPAO\_WEIGHTS**

Controls algorithm and weights for EPAO calculations (see *PAO\_METHOD*).

TYPE:

INTEGER

DEFAULT:

115 Standard weights, use 1<sup>st</sup> and 2<sup>nd</sup> order optimization

OPTIONS:

15 Standard weights, with 1<sup>st</sup> order optimization only.

RECOMMENDATION:

Use default, unless convergence failure is encountered.

**EXCHANGE**

Specifies the exchange level of theory.

TYPE:

STRING

DEFAULT:

HF for wavefunction correlation methods. Otherwise none.

OPTIONS:

HF	Hartree–Fock
Slater, S	Slater
Becke, B	Becke 1988
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	B3LYP based on correlation functional #5 of Vosko, Wilk, and Nusair rather than their functional #3
EDF1	EDF1
M05	M05
M05-2X	M05-2X
M06-L	M06-L
M06-HF	M06-HF
M06	M06
M06-2X	M06-2X
General, Gen	User defined combination of K, X and C (refer DFT section, Chapter 4).

RECOMMENDATION:

Consult the literature and reviews for guidance

**FAST\_XC**

Controls direct variable thresholds to accelerate exchange correlation (XC) in DFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Turn FAST\_XC on.

FALSE Do not use FAST\_XC.

RECOMMENDATION:

Caution: FAST\_XC improves the speed of a DFT calculation, but may occasionally cause the SCF calculation to diverge.

**FDIFF\_STEPSIZE**

Displacement used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 0.001 Å.

OPTIONS:

$n$  Use a step size of  $n \times 10^{-5}$ .

RECOMMENDATION:

Use default, unless on a very flat potential, in which case a larger value should be used.

**FOCK\_EXTRAP\_ORDER**

Specifies the polynomial order  $N$  for Fock matrix extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

$N$  Extrapolate using an  $N$ th-order polynomial ( $N > 0$ ).

RECOMMENDATION:

None

**FOCK\_EXTRAP\_POINTS**

Specifies the number  $M$  of old Fock matrices that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

$M$  Save  $M$  Fock matrices for use in extrapolation ( $M > N$ )

RECOMMENDATION:

Higher-order extrapolations with more saved Fock matrices are faster and conserve energy better than low-order extrapolations, up to a point. In many cases, the scheme ( $N = 6$ ,  $M = 12$ ), in conjunction with SCF\_CONVERGENCE = 6, is found to provide about a 50% savings in computational cost while still conserving energy.



**FTC\_CLASS\_THRESH\_MULT**

Together with FTC\_CLASS\_THRESH\_ORDER, determines the cutoff threshold for included a shell-pair in the *dd* class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Multiplicative part of the FTC classification threshold. Together with the default value of the FTC\_CLASS\_THRESH\_ORDER this leads to the  $5 \times 10^{-5}$  threshold value.

OPTIONS:

*n* User specified.

RECOMMENDATION:

Use the default. If diffuse basis sets are used and the molecule is relatively big then tighter FTC classification threshold has to be used. According to our experiments using Pople-type diffuse basis sets, the default  $5 \times 10^{-5}$  value provides accurate result for an alanine5 molecule while  $1 \times 10^{-5}$  threshold value for alanine10 and  $5 \times 10^{-6}$  value for alanine15 has to be used.

**FTC\_CLASS\_THRESH\_ORDER**

Together with FTC\_CLASS\_THRESH\_MULT, determines the cutoff threshold for included a shell-pair in the *dd* class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Logarithmic part of the FTC classification threshold. Corresponds to  $10^{-5}$

OPTIONS:

*n* User specified

RECOMMENDATION:

Use the default.

**FTC\_SMALLMOL**

Controls whether or not the operator is evaluated on a large grid and stored in memory to speed up the calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a big pre-calculated array to speed up the FTC calculations

0 Use this option to save some memory

RECOMMENDATION:

Use the default if possible and use 0 (or buy some more memory) when needed.

**FTC**

Controls the overall use of the FTC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use FTC in the Coulomb part

1 Use FTC in the Coulomb part

RECOMMENDATION:

Use FTC when bigger and/or diffuse basis sets are used.

**GEOM\_OPT\_COORDS**

Controls the type of optimization coordinates.

TYPE:

INTEGER

DEFAULT:

-1

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  $10^{-3}$ .

TYPE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

*n* User-defined cutoff.

RECOMMENDATION:

Use default.

**GEOM\_OPT\_HESSIAN**

Determines the initial Hessian status.

TYPE:

STRING

DEFAULT:

DIAGONAL

OPTIONS:

DIAGONAL Set up diagonal Hessian.

READ Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

RECOMMENDATION:

An accurate initial Hessian will improve the performance of the optimizer, but is expensive to compute.

**GEOM\_OPT\_LINEAR\_ANGLE**

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

*n* User-defined level.

RECOMMENDATION:

Use default.

**GEOM\_OPT\_MAX\_CYCLES**

Maximum number of optimization cycles.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

*n* User defined positive integer.

RECOMMENDATION:

The default should be sufficient for most cases. Increase if the initial guess geometry is poor, or for systems with shallow potential wells.

**GEOM\_OPT\_MAX\_DIIS**

Controls maximum size of subspace for GDIIS.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use GDIIS.

-1 Default size =  $\min(\text{NDEG}, \text{NATOMS}, 4)$  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 a transition state search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Mode following off.

*n* Maximize along mode *n*.

RECOMMENDATION:

Use default, for geometry optimizations.

**GEOM\_OPT\_PRINT**

Controls the amount of OPTIMIZE print output.

TYPE:

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.

**GEOM\_OPT\_SYMFLAG**

Controls the use of symmetry in OPTIMIZE.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

- 1 Make use of point group symmetry.
- 0 Do not make use of point group symmetry.

RECOMMENDATION:

Use default.

**GEOM\_OPT\_TOL\_DISPLACEMENT**

Convergence on maximum atomic displacement.

TYPE:

INTEGER

DEFAULT:

1200  $\equiv 1200 \times 10^{-6}$  tolerance on maximum atomic displacement.

OPTIONS:

$n$  Integer value (tolerance =  $n \times 10^{-6}$ ).

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 optimization cycles.

TYPE:

INTEGER

DEFAULT:

100  $\equiv 100 \times 10^{-8}$  tolerance on maximum gradient component.

OPTIONS:

$n$  Integer value (tolerance = value  $n \times 10^{-8}$ ).

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.

TYPE:

INTEGER

DEFAULT:

300  $\equiv 300 \times 10^{-6}$  tolerance on maximum gradient component.

OPTIONS:

$n$  Integer value (tolerance =  $n \times 10^{-6}$ ).

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\_UPDATE**

Controls the Hessian update algorithm.

TYPE:

INTEGER

DEFAULT:

-1

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\_PRINT**

Controls the amount of geometric information printed at each step.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Prints out all geometric information; bond distances, angles, torsions.

FALSE Normal printing of distance matrix.

RECOMMENDATION:

Use if you want to be able to quickly examine geometric parameters at the beginning and end of optimizations. Only prints in the beginning of single point energy calculations.

**GRAIN**

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

TYPE:

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

RECOMMENDATION:

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

**GUI**

Controls the output of auxiliary information for third party packages.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No auxiliary output is printed.

2 Auxiliary information is printed to the file *Test.FChk*.

RECOMMENDATION:

Use default unless the additional information is required. Please note that any existing *Test.FChk* file will be overwritten.

**GVB\_AMP\_SCALE**

Scales the default orbital amplitude iteration step size by  $n/1000$  for IP/RCC. PP amplitude equations are solved analytically, so this parameter does not affect PP.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

$n$  User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but in some highly-correlated systems it can help with convergence to use smaller values.

**GVB\_GUESS\_MIX**

Similar to SCF\_GUESS\_MIX, it breaks alpha–beta symmetry for UPP by mixing the alpha HOMO and LUMO orbitals according to the user–defined fraction of LUMO to add the HOMO. 100 corresponds to a 1:1 ratio of HOMO and LUMO in the mixed orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  User–defined,  $0 \leq n \leq 100$

RECOMMENDATION:

25 often works well to break symmetry without overly impeding convergence.

**GVB\_LOCAL**

Sets the localization scheme used in the initial guess wave function.

TYPE:

INTEGER

DEFAULT:

2 Pipek–Mezey orbitals

OPTIONS:

1 Boys localized orbitals

2 Pipek–Mezey orbitals

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. It can be helpful to try both to ensure the global minimum has been found.

**GVB\_N\_PAIRS**

Alternative to CC\_REST\_OCC and CC\_REST\_VIR for setting active space size in GVB and valence coupled cluster methods.

TYPE:

INTEGER

DEFAULT:

PP active space (1 occ and 1 virt for each valence electron pair)

OPTIONS:

$n$  user–defined

RECOMMENDATION:

Use default unless one wants to study a special active space. When using small active spaces, it is important to ensure that the proper orbitals are incorporated in the active space. If not, use the *\$reorder\_mo* feature to adjust the SCF orbitals appropriately.

**GVB\_ORB\_CONV**

The GVB–CC wave function is considered converged when the root–mean–square orbital gradient and orbital step sizes are less than  $10^{-\text{GVB\_ORB\_CONV}}$ . Adjust THRESH simultaneously.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

$n$  User–defined

RECOMMENDATION:

Use 6 for PP(2) jobs or geometry optimizations. Tighter convergence (*i.e.* 7 or higher) cannot always be reliably achieved.

**GVB\_ORB\_MAX\_ITER**

Controls the number of orbital iterations allowed in GVB-CC calculations. Some jobs, particularly unrestricted PP jobs can require 500-1000 iterations.

TYPE:

INTEGER

DEFAULT:

256

OPTIONS:

User-defined number of iterations.

RECOMMENDATION:

Default is typically adequate, but some jobs, particularly UPP jobs, can require 500-1000 iterations if converged tightly.

**GVB\_ORB\_SCALE**

Scales the default orbital step size by  $n/1000$ .

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

$n$  User-defined, 0-1000

RECOMMENDATION:

Default is usually fine, but for some stretched geometries it can help with convergence to use smaller values.

**GVB\_RESTART**

Restart a job from previously-converged GVB-CC orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Useful when trying to converge to the same GVB solution at slightly different geometries, for example.

**GVB\_UNRESTRICTED**

Controls restricted versus unrestricted PP jobs. Usually handled automatically.

TYPE:

LOGICAL

DEFAULT:

same value as UNRESTRICTED

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set this variable explicitly only to do a UPP job from an RHF or ROHF initial guess.



**GVB\_PRINT**

Controls the amount of information printed during a GVB-CC job.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

*n* User-defined

RECOMMENDATION:

Should never need to go above 0 or 1.

**IDERIV**

Controls the order of derivatives that are evaluated analytically. The user is not normally required to specify a value, unless numerical derivatives are desired. The derivatives will be evaluated numerically if IDERIV is set lower than JOBTYP requires.

TYPE:

INTEGER

DEFAULT:

Set to the order of derivative that JOBTYP requires

OPTIONS:

2 Analytic second derivatives of the energy (Hessian)

1 Analytic first derivatives of the energy.

0 Analytic energies only.

RECOMMENDATION:

Usually set to the maximum possible for efficiency. Note that IDERIV will be set lower if analytic derivatives of the requested order are not available.

**IGDEFIELD**

Triggers the calculation of the electrostatic potential and/or the electric field at the points given in the file ESPGrid.

TYPE:

INTEGER

DEFAULT:

UNDEFINED

OPTIONS:

0 Computes ESP.

1 Computes ESP and EFIELD.

2 Computes EFIELD.

RECOMMENDATION:

Must use this *\$rem* when IGDESP is specified.

**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.

TYPE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1 read grid input via the *\$plots* section of the input deck

0 Generate the ESP values at all nuclear positions.

+ $n$  read  $n$  grid points from the ACSII file ESPGrid.

RECOMMENDATION:

None

**INCDFT\_DENDIFF\_THRESH**

Sets the threshold for screening density matrix values in the IncDFT procedure.

TYPE:

INTEGER

DEFAULT:

SCF\_CONVERGENCE + 3

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

**INCDFT\_DENDIFF\_VARTHRESH**

Sets the lower bound for the variable threshold for screening density matrix values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT\_DENDIFF\_THRESH is reached. This means this value must be set lower than INCDFT\_DENDIFF\_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

**INCDFT\_GRIDDIFF\_THRESH**

Sets the threshold for screening functional values in the IncDFT procedure

TYPE:

INTEGER

DEFAULT:

SCF\_CONVERGENCE + 3

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

**INCDFT\_GRIDDIFF\_VARTHRESH**

Sets the lower bound for the variable threshold for screening the functional values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT\_GRIDDIFF\_THRESH is reached. This means that this value must be set lower than INCDFT\_GRIDDIFF\_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

$n$  Corresponding to a threshold of  $10^{-n}$ .

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

**INCDFT**

Toggles the use of the IncDFT procedure for DFT energy calculations.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not use IncDFT

TRUE Use IncDFT

RECOMMENDATION:

Turning this option on can lead to faster SCF calculations, particularly towards the end of the SCF. Please note that for some systems use of this option may lead to convergence problems.

**INCFOCK**

Iteration number after which the incremental Fock matrix algorithm is initiated

TYPE:

INTEGER

DEFAULT:

1 Start INCFOCK after iteration number 1

OPTIONS:

User-defined (0 switches INCFOCK off)

RECOMMENDATION:

May be necessary to allow several iterations before switching on INCFOCK.

**INTCAV**

A flag to select the surface integration method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Single center Lebedev integration.

1 Single center spherical polar integration.

RECOMMENDATION:

The Lebedev integration is by far the more efficient.

**INTEGRALS\_BUFFER**

Controls the size of in-core integral storage buffer.

TYPE:

INTEGER

DEFAULT:

15 15 Megabytes.

OPTIONS:

User defined size.

RECOMMENDATION:

Use the default, or consult your systems administrator for hardware limits.

**INTEGRAL\_2E\_OPR**

Determines the two-electron operator.

TYPE:

INTEGER

DEFAULT:

-2 Coulomb Operator.

OPTIONS:

-1 Apply the CASE approximation.

-2 Coulomb Operator.

RECOMMENDATION:

Use default unless the CASE operator is desired.

**INTRACULE**

Controls whether intracule properties are calculated (see also the *\$intracule* section).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE No intracule properties.

TRUE Evaluate intracule properties.

RECOMMENDATION:

None

**IOPPRD**

Specifies the choice of system operator form.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Symmetric form.

1 Non-symmetric form.

RECOMMENDATION:

The default uses more memory but is generally more efficient, we recommend its use unless there is shortage of memory available.

**IROTGR**

Rotation of the cavity surface integration grid.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

- 0 No rotation.
- 1 Rotate initial *xyz* axes of the integration grid to coincide with principal moments of nuclear inertia (relevant if ITRNGR=1)
- 2 Rotate initial *xyz* axes of integration grid to coincide with principal moments of nuclear charge (relevant if ITRNGR=2)
- 3 Rotate initial *xyz* axes of the integration grid through user-specified Euler angles as defined by Wilson, Decius, and Cross.

RECOMMENDATION:

The default is recommended unless the knowledgeable user has good reason otherwise.

**ISHAPE**

A flag to set the shape of the cavity surface.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

- 0 use the electronic isodensity surface.
- 1 use a spherical cavity surface.

RECOMMENDATION:

Use the default surface.

**ISOTOPES**

Specifies if non-default masses are to be used in the frequency calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

- FALSE Use default masses only.
- TRUE Read isotope masses from *\$isotopes* section.

RECOMMENDATION:

None

**ITRNGR**

Translation of the cavity surface integration grid.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

- 0 No translation (*i.e.*, center of the cavity at the origin of the atomic coordinate system)
- 1 Translate to the center of nuclear mass.
- 2 Translate to the center of nuclear charge.
- 3 Translate to the midpoint of the outermost atoms.
- 4 Translate to midpoint of the outermost non-hydrogen atoms.
- 5 Translate to user-specified coordinates in Bohr.
- 6 Translate to user-specified coordinates in Angstroms.

RECOMMENDATION:

The default value is recommended unless the single-center integrations procedure fails.

**JOBTYPE**

Specifies the type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

- SP Single point energy.
- OPT Geometry Minimization.
- TS Transition Structure Search.
- FREQ Frequency Calculation.
- FORCE Analytical Force calculation.
- RPATH Intrinsic Reaction Coordinate calculation.
- NMR NMR chemical shift calculation.

RECOMMENDATION:

Job dependent

**LINEQ**

Flag to select the method for solving the linear equations that determine the apparent point charges on the cavity surface.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

- 0 use LU decomposition in memory if space permits, else switch to LINEQ=2
- 1 use conjugate gradient iterations in memory if space permits, else use LINEQ=2
- 2 use conjugate gradient iterations with the system matrix stored externally on disk.

RECOMMENDATION:

The default should be sufficient in most cases.

**LIN\_K**

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

TYPE:

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.

**MAX\_CIS\_CYCLES**

Maximum number of CIS iterative cycles allowed

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

*n* User-defined number of cycles

RECOMMENDATION:

Default is usually sufficient.

**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.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM

*n* *n* DIIS iterations before switching to (G)DM.

RECOMMENDATION:

None

**MAX\_SCF\_CYCLES**

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

**MAX\_SUB\_FILE\_NUM**

Sets the maximum number of sub files allowed.

TYPE:

INTEGER

DEFAULT:

16 Corresponding to a total of 32Gb for a given file.

OPTIONS:

*n* User-defined number of gigabytes.

RECOMMENDATION:

Leave as default, or adjust according to your system limits.

**MEM\_STATIC**

Sets the memory for individual program modules.

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb.

OPTIONS:

*n* User-defined number of megabytes.

RECOMMENDATION:

For direct and semi-direct MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (32–160 Mb), as discussed above.

**MEM\_TOTAL**

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 (2 Gb)

OPTIONS:

*n* User-defined number of megabytes.

RECOMMENDATION:

Use default, or set to the physical memory of your machine. Note that if more than 1GB is specified for a Coupled Cluster job, the memory is allocated as follows

12% MEM\_STATIC

3% CC\_TMPBUFFSIZE

50% CC\_CC\_BLK\_TNSR\_BUFFSIZE

35% Other memory requirements:

**METECO**

Sets the threshold criteria for discarding shell-pairs.

TYPE:

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.



**MM\_CHARGES**

Requests the calculation of multipole-derived charges (MDCs).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculates the MDCs and also the traceless form of the multipole moments

RECOMMENDATION:

Set to TRUE if MDCs or the traceless form of the multipole moments are desired.

The calculation does not take long.

**MOLDEN\_FORMAT**

Requests a MOLDEN-formatted input file containing information from a Q-CHEM job.

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

True Append MOLDEN input file at the end of the Q-CHEM output file.

RECOMMENDATION:

None.

**MOM\_PRINT**

Switches printing on within the MOM procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Printing is turned off

TRUE Printing is turned on.

RECOMMENDATION:

None

**MOM\_START**

Determines when MOM is switched on to stabilize DIIS iterations.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE) MOM is not used

*n* MOM begins on cycle *n*.

RECOMMENDATION:

Set to 1 if preservation of initial orbitals is desired. If MOM is to be used to aid convergence, an SCF without MOM should be run to determine when the SCF starts oscillating. MOM should be set to start just before the oscillations.

**MOPROP\_CONV\_1ST**

Sets the convergence criterium for CPSCF and 1st order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

None

**MOPROP\_CONV\_2ND**

Sets the convergence criterium for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$  Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

None

**MOPROP\_DIIS\_DIM\_SS**

Specified the DIIS subspace dimension.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

0 No DIIS.

$n$  Use a subspace of dimension  $n$ .

RECOMMENDATION:

None

**MOPROP\_DIIS**

Controls the use of Pulays DIIS.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

0 Turn off DIIS.

5 Turn on DIIS.

RECOMMENDATION:

None

**MOPROP\_MAXITER\_1ST**

The maximal number of iterations for CPSCF and first-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

*n* Set maximum number of iterations to *n*.

RECOMMENDATION:

Use default.

**MOPROP\_MAXITER\_2ND**

The maximal number of iterations for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

*n* Set maximum number of iterations to *n*.

RECOMMENDATION:

Use default.

**MOPROP\_PERTNUM**

Set the number of perturbed densities that will to be treated together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 All at once.

*n* Treat the perturbed densities batchwise.

RECOMMENDATION:

Use default

**MOPROP**

Specifies the job for mopropman.

TYPE:

INTEGER

DEFAULT:

0 Do not run mopropman.

OPTIONS:

1 NMR chemical shielding tensors.

2 Static polarizability.

100 Dynamic polarizability.

101 First hyperpolarizability.

102 First hyperpolarizability, reading First order results from disk.

103 First hyperpolarizability using Wigner's  $(2n + 1)$  rule.

104 First hyperpolarizability using Wigner's  $(2n + 1)$  rule, reading first order results from disk.

RECOMMENDATION:

None.

**MULTIPOLE\_ORDER**

Determines highest order of multipole moments to print if wavefunction analysis requested.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

$n$  Calculate moments to  $n^{\text{th}}$  order.

RECOMMENDATION:

Use default unless higher multipoles are required.

**NBO**

Controls the use of the NBO package.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not invoke the NBO package.

TRUE Do invoke the NBO package.

RECOMMENDATION:

None

**NPTLEB**

The number of points used in the Lebedev grid for the single-center surface integration. (Only relevant if INTCAV=0).

TYPE:

INTEGER

DEFAULT:

1202

OPTIONS:

Valid choices are: 6, 18, 26, 38, 50, 86, 110, 146, 170, 194, 302, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, or 5294.

RECOMMENDATION:

The default value has been found adequate to obtain the energy to within 0.1 kcal/mol for solutes the size of monosubstituted benzenes.

**NPTTHE, NPTPHI**

The number of  $(\theta, \phi)$  points used for single-centered surface integration (relevant only if INTCAV=1).

TYPE:

INTEGER

DEFAULT:

8,16

OPTIONS:

$\theta, \phi$  specifying the number of points.

RECOMMENDATION:

These should be multiples of 2 and 4 respectively, to provide symmetry sufficient for all Abelian point groups. Defaults are too small for all but the tiniest and simplest solutes.

**N\_FROZEN\_CORE**

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

TYPE:

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.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n$  Freeze  $n$  virtual orbitals.

RECOMMENDATION:

None

**N\_I\_SERIES**

Sets summation limit for series expansion evaluation of  $i_n(x)$ .

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

**N\_J\_SERIES**

Sets summation limit for series expansion evaluation of  $j_n(x)$ .

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

**N\_SOL**

Specifies number of atoms or orbitals in *\$solute*

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

**N\_WIG\_SERIES**

Sets summation limit for Wigner integrals.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n < 100$

RECOMMENDATION:

Increase  $n$  for greater accuracy.

**OMEGA**

Controls the degree of attenuation of the Coulomb operator.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

$n$  Corresponding to  $\omega = n/1000$

RECOMMENDATION:

None

**PAO\_ALGORITHM**

Algorithm used to optimize polarized atomic orbitals (see PAO\_METHOD)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use efficient (and riskier) strategy to converge PAOs.

1 Use conservative (and slower) strategy to converge PAOs.

RECOMMENDATION:

None

**PAO\_METHOD**

Controls evaluation of polarized atomic orbitals (PAOs).

TYPE:

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.

RECOMMENDATION:

None

**POP\_MULLIKEN**

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE (or 0) Do not calculate Mulliken Population.

TRUE (1) Calculate Mulliken population

2 Also calculate shell populations for each occupied orbital.

RECOMMENDATION:

Leave as TRUE. Trivial additional calculation

**PRINT\_CORE\_CHARACTER**

Determines the print level for the CORE\_CHARACTER option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No additional output is printed.

1 Prints core characters of occupied MOs.

2 Print level 1, plus prints the core character of AOs.

RECOMMENDATION:

Use default, unless you are uncertain about what the core character is.

**PRINT\_DIST\_MATRIX**

Controls the printing of the inter-atomic distance matrix

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

0 Turns off the printing of the distance matrix

$n$  Prints the distance matrix if the number of atoms in the molecule is less than or equal to  $n$ .

RECOMMENDATION:

Use default unless distances are required for large systems

**PRINT\_GENERAL\_BASIS**

Controls print out of built in basis sets in input format

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATION:

Useful for modification of standard basis sets.

**PRINT\_ORBITALS**

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

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print any orbitals.

TRUE Prints occupied orbitals plus 5 virtuals.

NVIRT Number of virtuals to print.

RECOMMENDATION:

Use TRUE unless more virtuals are desired.

**PSEUDO\_CANONICAL**

When SCF\_ALGORITHM = DM, this controls the way the initial step, and steps after subspace resets are taken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use Roothaan steps when (re)initializing

TRUE Use a steepest descent step when (re)initializing

RECOMMENDATION:

The default is usually more efficient, but choosing TRUE sometimes avoids problems with orbital reordering.

**PURECART**

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

2111 Cartesian  $h$ -functions and pure  $g, f, d$  functions

OPTIONS:

*hgf d* Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets



**QMMM\_CHARGES**

Controls the printing of QM charges to file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Writes a charges.dat file with the Mulliken charges from the QM region.

FALSE No file written.

RECOMMENDATION:

Use default unless running calculations with CHARMM where charges on the QM region need to be saved.

**QMMM\_PRINT**

Controls the amount of output printed from a QM/MM job.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Limit molecule, point charge, and analysis printing.

FALSE Normal printing.

RECOMMENDATION:

Use default unless running calculations with CHARMM.

**QM\_MM**

Turns on the Q-CHEM/CHARMM interface.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do QM/MM calculation through the Q-CHEM/CHARMM interface.

FALSE Turn this feature off.

RECOMMENDATION:

Use default unless running calculations with CHARMM.

**RADSPH**

Sphere radius used to specify the cavity surface (Only relevant for ISHAPE=1).

TYPE:

FLOAT

DEFAULT:

Half the distance between the outermost atoms plus 1.4 Angstroms.

OPTIONS:

Real number specifying the radius in bohr (if positive) or in Angstroms (if negative).

RECOMMENDATION:

Make sure that the cavity radius is larger than the length of the molecule.

**RC\_R0**

Determines the parameter in the Gaussian weight function used to smooth the density at the nuclei.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Corresponds the traditional delta function spin and charge densities  
 $n$  corresponding to  $n \times 10^{-3}$  a.u.

RECOMMENDATION:

We recommend value of 250 for a typical split valence basis. For basis sets with increased flexibility in the nuclear vicinity the smaller values of  $r_0$  also yield adequate spin density.

**READ\_VDW**

Controls the input of user-defined atomic radii for CHEMSOL calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use default CHEMSOL parameters.  
TRUE Read from the *\$van\_der\_waals* section of the input file.

RECOMMENDATION:

None.

**RHOISO**

Value of the electronic isodensity contour used to specify the cavity surface. (Only relevant for ISHAPE = 0).

TYPE:

FLOAT

DEFAULT:

0.001

OPTIONS:

Real number specifying the density in electrons/bohr<sup>3</sup>.

RECOMMENDATION:

The default value is optimal for most situations. Increasing the value produces a smaller cavity which ordinarily increases the magnitude of the solvation energy.

**ROTTHE ROTPHI ROTCHI**

Euler angles ( $\theta$ ,  $\phi$ ,  $\chi$ ) in degrees for user-specified rotation of the cavity surface. (relevant if IROTGR=3)

TYPE:

FLOAT

DEFAULT:

0,0,0

OPTIONS:

$\theta$ ,  $\phi$ ,  $\chi$  in degrees

RECOMMENDATION:

None.

**RPATH\_COORDS**

Determines which coordinate system to use in the IRC search.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use mass-weighted coordinates.

2 Use *Z*-matrix coordinates.

RECOMMENDATION:

Use default.

**RPATH\_DIRECTION**

Determines the direction of the eigenmode to follow. This will not usually be known prior to the Hessian diagonalization.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Descend in the positive direction of the eigenmode.

-1 Descend in the negative direction of the eigenmode.

RECOMMENDATION:

It is usually not possible to determine in which direction to go *a priori*, and therefore both directions will need to be considered.

**RPATH\_MAX\_CYCLES**

Specifies the maximum number of points to find on the reaction path.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

*n* User-defined number of cycles.

RECOMMENDATION:

Use more points if the minimum is desired, but not reached using the default.

**RPATH\_MAX\_STEPSIZE**

Specifies the maximum step size to be taken (in thousandths of *a.u.*).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 *a.u.*

OPTIONS:

*n* Step size = *n*/1000.

RECOMMENDATION:

None.

**RPATH\_PRINT**

Specifies the print output level.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

*n*

RECOMMENDATION:

Use default, little additional information is printed at higher levels. Most of the output arises from the multiple single point calculations that are performed along the reaction pathway.

**RPATH\_TOL\_DISPLACEMENT**

Specifies the convergence threshold for the step. If a step size is chosen by the algorithm that is smaller than this, the path is deemed to have reached the minimum.

TYPE:

INTEGER

DEFAULT:

5000 Corresponding to 0.005 *a.u.*

OPTIONS:

*n* User-defined. Tolerance =  $n/1000000$ .

RECOMMENDATION:

None.

**RPA**

Do an RPA calculation in addition to a CIS calculation

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not do an RPA calculation

True Do an RPA calculation.

RECOMMENDATION:

None

**SAVE\_LAST\_GPX**

Save last  $\mathbf{G}[\mathbf{P}^x]$  when calculating dynamic polarizabilities in order to call mopropman in a second run with MOPROP = 102.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 False

1 True

RECOMMENDATION:

None

**SCF\_ALGORITHM**

Algorithm used for converging the SCF.

TYPE:

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 geometric direct minimization is recommended. 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. Note that in Q-CHEM 3.0 the DIIS error is measured by the maximum error rather than the RMS error as in previous versions.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations.

7 For geometry optimizations and vibrational analysis.

8 For SSG calculations, see Chapter 5.

OPTIONS:

User-defined

RECOMMENDATION:

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

**SCF\_FINAL\_PRINT**

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

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

0 No extra print out.

1 Orbital energies and break-down of SCF energy.

2 Level 1 plus MOs and density matrices.

3 Level 2 plus Fock and density matrices.

RECOMMENDATION:

The break-down of energies is often useful (level 1).

**SCF\_GUESS\_ALWAYS**

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

TYPE:

LOGICAL

DEFAULT:

False

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.

RECOMMENDATION:

Use default unless SCF convergence issues arise

**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.

TYPE:

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*.

**SCF\_GUESS\_PRINT**

Controls printing of guess MOs, Fock and density matrices.

TYPE:

INTEGER

DEFAULT:

0

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.

RECOMMENDATION:

None

**SCF\_GUESS**

Specifies the initial guess procedure to use for the SCF.

TYPE:

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. Note that because the density is made spherical, this may favor an undesired state for atomic systems, especially transition metals.

**SCF\_PRINT**

Controls level of output from SCF procedure to Q-CHEM output file.

TYPE:

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. These levels are primarily for program debugging.

**SKIP\_CIS\_RPA**

Skips the solution of the CIS, RPA, TDA or TDDFT equations for wavefunction analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE / FALSE

RECOMMENDATION:

Set to true to speed up the generation of plot data if the same calculation has been run previously with the scratch files saved.

**SOLUTE\_RADIUS**

Sets the Onsager solvent model cavity radius.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

$n \quad a_0 = n \times 10^{-5}$ .

RECOMMENDATION:

Use equation (10.1).

**SOLVENT\_DIELECTRIC**

Sets the dielectric constant of the Onsager solvent continuum.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

$n \quad \varepsilon = n \times 10^{-5}$ .

RECOMMENDATION:

As per required solvent.

**SSG**

Controls the calculation of the SSG wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute the SSG wavefunction

1 Do compute the SSG wavefunction

RECOMMENDATION:

See also the UNRESTRICTED and DIIS\_SUBSPACE\_SIZE *\$rem* variables.

**STABILITY\_ANALYSIS**

Performs stability analysis for a HF or DFT solution.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform stability analysis.

FALSE Do not perform stability analysis.

RECOMMENDATION:

Set to TRUE when a HF or DFT solution is suspected to be unstable.



**STATE\_TO\_OPT**

An alternative to the above two *\$rem* variables which allows the state and symmetry to be specified using a single variable

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[*i,j*] optimize the *j*th state of the *i*th irrep.

RECOMMENDATION:

Do not use in conjunction with CC\_REFSYM and CC\_STATE\_DERIV

**SVP\_CAVITY\_CONV**

Determines the convergence value of the iterative isodensity cavity procedure.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

*n* Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

The default value unless convergence problems arise.

**SVP\_CHARGE\_CONV**

Determines the convergence value for the charges on the cavity. When the change in charges fall below this value, if the electron density is converged, then the calculation is considered converged.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

*n* Convergence threshold set to  $10^{-n}$ .

RECOMMENDATION:

The default value unless convergence problems arise.

**SVP\_GUESS**

Specifies how and if the solvation module will use a given guess for the charges and cavity points.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No guessing.

1 Read a guess from a previous Q-CHEM solvation computation.

2 Use a guess specified by the \$svpirf section from the input

RECOMMENDATION:

It is helpful to also set SCF\_GUESS to READ when using a guess from a previous Q-CHEM run.

**SVP\_MEMORY**

Specifies the amount of memory for use by the solvation module.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

*n* corresponds to the amount of memory in MB.

RECOMMENDATION:

The default should be fine for medium size molecules with the default Lebedev grid, only increase if needed.

**SVP\_PATH**

Specifies whether to run a gas phase computation prior to performing the solvation procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 runs a gas-phase calculation and after convergence runs the SS(V)PE computation.

1 does not run a gas-phase calculation.

RECOMMENDATION:

Running the gas-phase calculation provides a good guess to start the solvation stage and provides a more complete set of solvated properties.

**SVP**

Sets whether to perform the isodensity solvation procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform the SS(V)PE solvation procedure.

TRUE Perform the SS(V)PE solvation procedure.

RECOMMENDATION:

NONE

**SYMMETRY\_DECOMPOSITION**

Determines symmetry decompositions to calculate.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No symmetry decomposition.

1 Calculate MO eigenvalues and symmetry (if available).

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices.

RECOMMENDATION:

None

**SYMMETRY**

Controls the efficiency through the use of point group symmetry for calculating integrals.

TYPE:

LOGICAL

DEFAULT:

TRUE Use symmetry for computing integrals.

OPTIONS:

TRUE Use symmetry when available.

FALSE Do not use symmetry.

RECOMMENDATION:

Use default unless benchmarking.

**SYM\_IGNORE**

Controls whether or not Q-CHEM determines the point group of the molecule.

TYPE:

LOGICAL

DEFAULT:

FALSE Do determine the point group.

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use default unless you do not want the molecule to be reoriented.

**SYM\_TOL**

Controls the tolerance for determining point group symmetry. Differences in atom locations less than  $10^{-\text{SYM\_TOL}}$  are treated as zero.

TYPE:

INTEGER

DEFAULT:

5 corresponding to  $10^{-5}$ .

OPTIONS:

User defined.

RECOMMENDATION:

Use the default unless the molecule has high symmetry which is not being correctly identified. Note that relaxing this tolerance too much may introduce errors into the calculation.

**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

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

**THRESH**

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

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

OPTIONS:

$n$  for a threshold of  $10^{-n}$ .

RECOMMENDATION:

Should be at least three greater than SCF\_CONVERGENCE. Increase for more significant figures, at greater computational cost.

**TIME\_STEP**

Specifies the molecular dynamics time step, in atomic units ( $1 \text{ a.u.} = 0.0242 \text{ fs}$ ).

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

Smaller time steps lead to better energy conservation; too large a time step may cause the job to fail entirely. Make the time step as large as possible, consistent with tolerable energy conservation.

**TRANX, TRANY, TRANZ**

$x$ ,  $y$ , and  $z$  value of user-specified translation (only relevant if ITRNGR is set to 5 or 6)

TYPE:

FLOAT

DEFAULT:

0, 0, 0

OPTIONS:

$x$ ,  $y$ , and  $z$  relative to the origin in the appropriate units.

RECOMMENDATION:

None.

**TRNSS**

Controls whether reduced single excitation space is used

TYPE:

LOGICAL

DEFAULT:

FALSE Use full excitation space

OPTIONS:

TRUE Use reduced excitation space

RECOMMENDATION:

None

**TRTYPE**

Controls how reduced subspace is specified

TYPE:

INTEGER

DEFAULT:

1 Select orbitals localized on a set of atoms

OPTIONS:

2 Specify a set of orbitals

RECOMMENDATION:

None

**UNRESTRICTED**

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE (Restricted) Closed-shell systems.

TRUE (Unrestricted) Open-shell systems.

OPTIONS:

TRUE (Unrestricted) Open-shell systems.

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 Section 4.5 on initial guesses).

**VARTHRESH**

Controls the temporary integral cut-off threshold.  $tmp\_thresh = 10^{-VARTHRESH} \times DIIS\_error$

TYPE:

INTEGER

DEFAULT:

0 Turns VARTHRESH off

OPTIONS:

*n* User-defined threshold

RECOMMENDATION:

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

**VCI**

Specifies the number of quanta involved in the VCI calculation.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined

RECOMMENDATION:

The availability depends on the memory of the machine. For example, a machine with 1.5 GB memory and for molecules with fewer than 3 atoms, VCI(10) can be carried out, for molecule containing fewer than 5 atoms, VCI(7) can be carried out, for molecule containing fewer than 6 atoms, VCI(4) can be carried out. For molecules larger than hexa-atoms, VCI(2) is usually available. VCI(1) and VCI(3) usually overestimated the true energy while VCI(4) usually gives an answer close to the converged energy.

**VIBMAN\_PRINT**

Controls level of extra print out for vibrational analysis.

TYPE:

INTEGER

DEFAULT:

1

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.
- 7 Level 6 plus anharmonic cubic and quartic force field print out.

RECOMMENDATION:

Use default.

**WAVEFUNCTION\_ANALYSIS**

Controls the running of the default wavefunction analysis tasks.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

- TRUE Perform default wavefunction analysis.
- FALSE Do not perform default wavefunction analysis.

RECOMMENDATION:

None

**WIG\_GRID**

Specify angular Lebedev grid for Wigner intracule calculations.

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Lebedev grids up to 5810 points.

RECOMMENDATION:

Larger grids if high accuracy required.

**WIG\_LEB**

Use Lebedev quadrature to evaluate Wigner integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Evaluate Wigner integrals through series summation.

TRUE Use quadrature for Wigner integrals.

RECOMMENDATION:

None

**WIG\_MEM**

Reduce memory required in the evaluation of  $W(u, v)$ .

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use low memory option.

TRUE Use low memory option.

RECOMMENDATION:

The low memory option is slower, use default unless memory is limited.

**WRITE\_WFN**

Specifies whether or not a wfn file is created, which is suitable for use with AIM-PAC. Note that the output to this file is currently limited to  $f$  orbitals, which is the highest angular momentum implemented in AIMPAC.

TYPE:

STRING

DEFAULT:

(NULL) No output file is created.

OPTIONS:

*filename* Specifies the output file name. The suffix *.wfn* will be appended to this name.

RECOMMENDATION:

None

**XCIS**

Do an XCIS calculation in addition to a CIS calculation

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not do an XCIS calculation

True Do an XCIS calculation (requires ROHF ground state).

RECOMMENDATION:

None

**XC\_GRID**

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

0 SG-0/SG-1 hybrid

OPTIONS:

0 Use SG-0 for H, C, N, and O, SG-1 for all other atoms.

1 Use SG-1 for all atoms.

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 4.3.4.

$-mn$  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  $n = 2N^2$ .

RECOMMENDATION:

Use default unless numerical integration problems arise. Larger grids may be required for optimization and frequency calculations.

**XC\_SMART\_GRID**

Uses SG-0 (where available) for early SCF cycles, and switches to the (larger) grid specified by XC\_GRID (which defaults to SG-1, if not otherwise specified) for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.