TURBOMOLE

Program Package for \textit{ab initio}

Electronic Structure Calculations

USER’S MANUAL

\textsc{Turbomole} Version 7.0
June 16, 2015
Contents

1 Preface and General Information 11
   1.1 Contributions and Acknowledgements .......................... 11
   1.2 Features of TURBOMOLE ......................................... 13
   1.3 How to Quote Usage of TURBOMOLE .............................. 13
   1.4 Modules and Their Functionality ................................ 23
   1.5 Tools .............................................................. 26

2 Installation of TURBOMOLE 30
   2.1 Install TURBOMOLE command line version ..................... 30
      2.1.1 Settings for each user: ..................................... 30
      2.1.2 Setting system type and $PATH by hand .................... 31
      2.1.3 Testing the installation ..................................... 31
   2.2 Installation problems: How to solve ......................... 32

3 How to Run TURBOMOLE 34
   3.1 A ‘Quick and Dirty’ Tutorial .................................... 34
      3.1.1 Single Point Calculations: Running TURBOMOLE Modules .. 36
      3.1.2 Energy and Gradient Calculations .......................... 36
      3.1.3 Calculation of Molecular Properties ....................... 38
      3.1.4 Modules and Data Flow .................................... 38
   3.2 Parallel Runs .................................................... 38
      3.2.1 Running Parallel Jobs — MPI case ......................... 40
      3.2.2 Running Parallel Jobs — SMP case ........................ 45

4 Preparing your input file with DEFINE 49
## CONTENTS

4.0.3 Universally Available Display Commands in `DEFINE` . . . . . 50
4.0.4 Specifying Atomic Sets . . . . . . . . . . . . . . . . . . . . 50
4.0.5 `control` as Input and Output File . . . . . . . . . . . . . 50
4.0.6 Be Prepared . . . . . . . . . . . . . . . . . . . . . . . . 51

4.1 The Geometry Main Menu . . . . . . . . . . . . . . . . . . . . 52
  4.1.1 Description of commands . . . . . . . . . . . . . . . . . 54
  4.1.2 Internal Coordinate Menu . . . . . . . . . . . . . . . . 57
  4.1.3 Manipulating the Geometry . . . . . . . . . . . . . . . 62

4.2 The Atomic Attributes Menu . . . . . . . . . . . . . . . . . 62
  4.2.1 Description of the commands . . . . . . . . . . . . . . 65

4.3 Generating MO Start Vectors . . . . . . . . . . . . . . . . . 67
  4.3.1 The MO Start Vectors Menu . . . . . . . . . . . . . . 67
  4.3.2 Assignment of Occupation Numbers . . . . . . . . . . 70
  4.3.3 Orbital Specification Menu . . . . . . . . . . . . . . 72
  4.3.4 Roothaan Parameters . . . . . . . . . . . . . . . . . . 73
  4.3.5 Start-MOs for broken symmetry treatments ("flip") . . 73

4.4 The General Options Menu . . . . . . . . . . . . . . . . . . 76
  4.4.1 Important commands . . . . . . . . . . . . . . . . . . . 77
  4.4.2 Special adjustments . . . . . . . . . . . . . . . . . . . . 83
  4.4.3 Relax Options . . . . . . . . . . . . . . . . . . . . . 85
  4.4.4 Definition of External Electrostatic Fields . . . . . . . 89
  4.4.5 Properties . . . . . . . . . . . . . . . . . . . . . . . . 90

5 Calculation of Molecular Structure and *Ab Initio* Molecular Dynamics 99

5.1 Structure Optimizations using the `JOBEX` Script . . . . . . . . 99
  5.1.1 Options . . . . . . . . . . . . . . . . . . . . . . . . . . 99
  5.1.2 Output . . . . . . . . . . . . . . . . . . . . . . . . . . 100

5.2 Program `STATPT` . . . . . . . . . . . . . . . . . . . . . . . 101
  5.2.1 General Information . . . . . . . . . . . . . . . . . . . 101
  5.2.2 Hessian matrix . . . . . . . . . . . . . . . . . . . . . . 102
  5.2.3 Finding Minima . . . . . . . . . . . . . . . . . . . . . 103
  5.2.4 Finding transition states . . . . . . . . . . . . . . . . 103
CONTENTS

5.3 Program Relax ........................................ 104
  5.3.1 Purpose ........................................ 104
  5.3.2 Optimization of General Coordinates ........ 105
  5.3.3 Force Constant Update Algorithms ............ 106
  5.3.4 Definition of Internal Coordinates ............ 108
  5.3.5 Structure Optimizations Using Internal Coordinates .... 108
  5.3.6 Structure Optimization in Cartesian Coordinates .... 109
  5.3.7 Optimization of Basis Sets (SCF only) ........ 110
  5.3.8 Simultaneous Optimization of Basis Set and Structure .... 110
  5.3.9 Optimization of Structure and a Global Scaling Factor .... 111
  5.3.10 Conversion from Internal to Cartesian Coordinates .... 111
  5.3.11 Conversion of Cartesian Coordinates, Gradients and Force Constants to Internals111
  5.3.12 The m-Matrix ................................ 112
  5.3.13 Initialization of Force Constant Matrices ........ 112
  5.3.14 Look at Results ................................ 113

5.4 Force Field Calculations ............................... 113
  5.4.1 Purpose ........................................ 113
  5.4.2 How to Perform a Uff Calculation ............... 114
  5.4.3 The Uff implementation ......................... 114

5.5 Molecular Dynamics Calculations ...................... 116

5.6 Counterpoise-Corrections using the JOBBSSE Script .... 118
  5.6.1 Options ........................................ 119
  5.6.2 Output .......................................... 120

5.7 Reaction Path Optimization .......................... 121
  5.7.1 Background and Program structure ............... 121
  5.7.2 Input Structure ................................ 121
  5.7.3 How it works ................................... 122

6 Hartree–Fock and DFT Calculations for Molecular Systems 125
  6.1 Background Theory .................................. 127
  6.2 Exchange-Correlation Functionals Available ........ 128
    6.2.1 Exchange-Correlation Functionals from XCFun library .... 131
6.3 Restricted Open-Shell Hartree–Fock .................................. 135
  6.3.1 Brief Description ................................................. 135
  6.3.2 One Open Shell .................................................... 135
  6.3.3 More Than One Open Shell ..................................... 138
  6.3.4 Miscellaneous ..................................................... 140

6.4 Relativistic effects ..................................................... 142
  6.4.1 One- and two-component relativistic methods ............... 142
  6.4.2 How to use .......................................................... 144

6.5 Periodic Electrostatic Embedded Cluster Method ................. 146
  6.5.1 General Information ............................................... 146
  6.5.2 Theoretical Background ......................................... 146
  6.5.3 Calculation Setup ................................................ 147

6.6 Dispersion Correction for DFT Calculations ....................... 154

6.7 Energy Decomposition Analysis (EDA) ............................ 156
  6.7.1 How to perform .................................................. 156

7 DFT Calculations for Molecular and Periodic Systems ........... 158
  7.1 Functionalities of RIPER ............................................ 158
  7.2 Theoretical Background ............................................. 158
  7.3 How to Perform a Calculation ..................................... 162
    7.3.1 Prerequisites ..................................................... 162
    7.3.2 Creating the Input File ....................................... 162
    7.3.3 Example .......................................................... 162

8 Hartree–Fock and DFT Response Calculations: Stability, Dynamic Response Properties, and Excited
  States ............................................................................ 164
  8.1 Functionalities of Esecf and Egrad ................................. 164
  8.2 Theoretical Background .............................................. 165
  8.3 Implementation ........................................................ 168
  8.4 How to Perform ........................................................ 169
    8.4.1 Preliminaries ....................................................... 169
    8.4.2 Polarizabilities and Optical Rotations ....................... 170
    8.4.3 Stability Analysis ................................................. 171
    8.4.4 Vertical Excitation and CD Spectra ......................... 171
CONTENTS

8.4.5 Excited State Geometry Optimizations .......... 174
8.4.6 Excited State Force Constant Calculations .......... 174
8.4.7 Polarizability Derivatives and Raman Spectra .......... 175

9 Second-order Möller–Plesset Perturbation Theory .......... 176
9.1 Functionalities of mpgrad, rcc2, and pnoccsd .......... 176
9.1.1 How to quote ........................................ 177
9.2 Some Theory .......................................... 178
9.3 How to Prepare and Perform MP2 Calculations .......... 179
9.4 General Comments on MP2 Calculations, Practical Hints .......... 183
9.5 RI-MP2-F12 Calculations .................................. 184
9.6 LT-SOS-RI-MP2 with $O(N^4)$ scaling costs .......... 189
9.7 OSV-PNO-MP2 and OSV-PNO-MP2-F12 calculations .......... 191

10 Second-Order Approximate Coupled-Cluster (CC2) Calculations .......... 193
10.1 CC2 Ground-State Energy Calculations ................. 198
10.2 Calculation of Excitation Energies ................. 200
10.3 First-Order Properties and Gradients ................. 204
10.3.1 Ground State Properties, Gradients and Geometries .......... 204
10.3.2 Excited State Properties, Gradients and Geometries .......... 206
10.3.3 Visualization of densities and Density analysis .......... 209
10.3.4 Fast geometry optimizations with RI-SCF based gradients .......... 211
10.4 Transition Moments .................................. 211
10.4.1 Ground to excited state transition moments .......... 212
10.4.2 Transition moments between excited states .......... 213
10.5 Ground State Second-order Properties with MP2 and CC2 .......... 214
10.6 Parallel RI-MP2 and RI-CC2 Calculations ................. 214
10.7 Spin-component scaling approaches (SCS/SOS) .......... 215

11 CCSD, CCSD(F12*) and CCSD(T) calculations .......... 217
11.1 Characteristics of the Implementation and Computational Demands .......... 219

12 Random Phase Approximation Calculations: Energy and First-Order Properties .......... 227
12.1 Ground State Energy Theory .................................. 227
## CONTENTS

12.2 Gradients Theory ......................................................... 229
12.3 Further Recommendations ............................................. 231
12.4 Comments on the Output ................................................ 232

13 Many body perturbation theory in the $GW$ approximation 233
  13.1 Theoretical background ............................................... 233
  13.2 $GW$ features .......................................................... 234
  13.3 General recipe for $G_0W_0$ calculations .......................... 234

14 Calculation of Vibrational Frequencies and Vibrational Spectra 236
  14.1 Analysis of Normal Modes in Terms of Internal Coordinates .... 238
  14.2 Calculation of Raman Spectra ........................................ 239
  14.3 Vibrational frequencies with fixed atoms using NumForce ........ 239
  14.4 Interface to hotFCHT .................................................. 240

15 First order electron-vibration coupling 241
  15.1 Theoretical background ............................................... 241
  15.2 $evib$ features ........................................................ 242
  15.3 General usage of $evib$ .............................................. 242

16 Calculation of NMR Shieldings 244
  16.1 Prerequisites .......................................................... 244
  16.2 How to Perform a SCF of DFT Calculation ......................... 244
  16.3 How to Perform a MP2 calculation .................................. 245
  16.4 Chemical Shifts ....................................................... 245
  16.5 Other Features and Known Limitations ............................. 246

17 Molecular Properties, Wavefunction Analysis, and Interfaces to Visualization Tools247
  17.1 Wavefunction analysis and Molecular Properties .................. 247
  17.2 Interfaces to Visualization Tools ................................... 251

18 Frozen Density Embedding calculations 257
  18.1 Background Theory ................................................... 257
  18.2 Frozen Density Embedding calculations using the FDE script ...... 259
18.2.1 Options ......................................................... 262
18.2.2 FDE with hybrid and orbital-dependent functionals .... 267

19 Orbital Dependent Kohn-Sham Density Functional Theory 269
19.1 Theoretical Background .......................................................... 269
19.2 Implementation ................................................................. 271
19.2.1 OEP-EXX ................................................................. 271
19.2.2 LHF ................................................................. 271
19.3 How to Perform ................................................................. 272
19.4 How to plot the exchange potential ........................................ 276
19.5 How to quote ................................................................. 277

20 Treatment of Solvation Effects with Cosmo 278

21 Keywords in the control file 285
21.1 Introduction ................................................................. 285
21.2 Format of Keywords and Comments ...................................... 285
21.2.1 General Keywords .......................................................... 285
21.2.2 Keywords for System Specification .................................... 287
21.2.3 Keywords for redundant internal coordinates in $redund_inp$ 289
21.2.4 Keywords for Module uff .................................................. 290
21.2.5 Keywords for Module woelfling ........................................ 295
21.2.6 Keywords for Modules dscf and ridft ................................ 296
21.2.7 Keywords for Module riper ................................................ 317
21.2.8 Keywords for Periodic Electrostatic Embedded Cluster Method 320
21.2.9 Keywords for COSMO .................................................. 322
21.2.10 Keywords for Modules grad and rdgrad .......................... 327
21.2.11 Keywords for Module aoforce .......................................... 328
21.2.12 Keywords for Module evib .............................................. 331
21.2.13 Keywords for Module escf ............................................... 331
21.2.14 Keywords for Module rirpa .............................................. 334
21.2.15 Keywords for Module egrad ............................................. 335
21.2.16 Keywords for Module mpgrad .......................................... 335
CONTENTS

21.2.17 Keywords for Module ricc2 .................................. 336
21.2.18 Keywords for Module pnoccsd ................................. 348
21.2.19 Keywords for Module relax ................................... 351
21.2.20 Keywords for Module statpt ................................. 359
21.2.21 Keywords for Module moloch ................................. 360
21.2.22 Keywords for wave function analysis and generation of plotting data 364
21.2.23 Keywords for Module frog ................................. 373
21.2.24 Keywords for Module mpshift ................................. 379
21.2.25 Keywords for Parallel Runs ................................. 380

22 Sample control files 383

22.1 Introduction ......................................................... 383
22.2 NH$_3$ Input for a RHF Calculation .............................. 384
22.3 NO$_2$ input for an unrestricted DFT calculation ............... 388
22.4 TaCl$_5$ Input for an RI-DFT Calculation with ECPs .......... 392
22.5 Basisset optimization for Nitrogen ............................. 399
22.6 ROHF of Two Open Shells .................................... 402

23 The Perl-based Test Suite Structure 405

23.1 General .......................................................... 405
23.2 Running the tests .................................................. 405
23.3 Taking the timings and benchmarking ............................ 407
23.4 Modes and options of the TTEST script ....................... 407

Bibliography 411

Index 425
Chapter 1

Preface and General Information

1.1 Contributions and Acknowledgements

TURBOMOLE [1] is a development of University of Karlsruhe and Forschungszentrum Karlsruhe GmbH 1989-2007, TURBOMOLE GmbH, since 2007. The following people have made contributions:

Reinhart Ahlrichs, Markus Klaus Armbruster, Rafał A. Bachorz, Michael Bär, Hans-Peter Baron, Rüdiger Bauernschmitt, Florian A. Bischoff, Stephan Böcker, Asbjörn M. Burow, Nathan Crawford, Peter Deglmann, Fabio Della Sala, Michael Diedenhofen, Michael Ehrig, Karin Eichkorn, Simon Elliott, Daniel Friese, Filipp Furche, Andreas Glöß, Nora Graf, Lukáš Grajciar, Frank Haase, Marco Häser, Christof Hättig, Arnim Hellweg, Benjamin Helmich, Sebastian Höfener, Hans Horn, Christian Huber, Uwe Huniar, Marco Kattannek, Wim Klopper, Andreas Köhn, Christoph Kömler, Markus Kollwitz, Michael Kühn, Klaus May, Paola Nava, Christian Ochsenfeld, Holger Öhm, Mathias Pabst, Holger Patzelt, Dmitrij Rappoport, Oliver Rubner, Ansgar Schäfer, Gunnar Schmitz, Uwe Schneider, Marek Sierka, David P. Tew, Oliver Treutler, Barbara Unterreiner, Malte von Arnim, Florian Weigend, Patrick Weis, Horst Weiss, Nina Winter
We acknowledge help from

- Michael Dolg, University of Stuttgart, now: University of Cologne
- Jürgen Gauss, University of Mainz
- Christoph van Wüllen, University of Bochum, now: TU Kaiserslautern
- Stefan Brode, BASF AG, Ludwigshafen
- Heinz Schiffer, HOECHST AG, Frankfurt
- Ove Christiansen and Tobias Schwabe, Aarhus University, T. S. now: Hamburg University

and financial support by the University of Karlsruhe, BASF AG, BAYER AG, HOECHST AG, the DFG, and the "Fonds der Chemischen Industrie".

Contact address:

Abteilung für Theoretische Chemie
Institut für Physikalische Chemie
Karlsruher Institut für Technologie
Kaiserstr. 12
D-76131 Karlsruhe
E-mail: info@turbomole.com
Web: http://www.turbomole.com

Support is provided by COSMOlogic GmbH&Co.KG, see http://www.cosmologic.de
Email: turbomole@cosmologic.de
1.2 Features of TURBOMOLE

TURBOMOLE has been specially designed for UNIX workstations and PCs and efficiently exploits the capabilities of this type of hardware. TURBOMOLE consists of a series of modules; their use is facilitated by various tools.

Outstanding features of TURBOMOLE are

- semi-direct algorithms with adjustable main memory and disk space requirements
- full use of all point groups
- efficient integral evaluation
- stable and accurate grids for numerical integration
- low memory and disk space requirements

1.3 How to Quote Usage of TURBOMOLE

Please quote the usage of the program package under consideration of the version number:


A LaTeX template could look like this:
@misc{TURBOMOLE,
title = {{TURBOMOLE V7.0 2015}, a development of {University of Karlsruhe} and {Forschungszentrum Karlsruhe GmbH}, 1989-2007, {TURBOMOLE GmbH}, since 2007; available from \ 
{\tt http://www.turbomole.com}.}}

Scientific publications require proper citation of methods and procedures employed. The output headers of TURBOMOLE modules include the relevant papers. One may also use the following connections between: method [module] number in the subsequent list (For module ricc2 see also Section 10).
• Programs and methods
  - general program structure and features: I
  - HF-SCF [dscf, ridft]: II
  - DFT (quadrature) [dscf, ridft, escf, aoforce]: IV, d (m grids)
  - RI-DFT [ridft, aoforce, escf, riper]: c, d, XXIII (marij), VII (escf), XXIV (aocharacter), XL (lmidf)
  - periodic DFT [riper]: XXXIX
  - MP2 [mpgrad]: III
  - RI-MP2 [ricc2]: energies and gradients VIII, XXIX, f, and (static) polarizabilities XXXVI
  - PNO-MP2 [pno0ccsd]: energies XXXVII
  - stability analysis [escf]: V
  - electronic excitations with CIS, RPA, TD-DFT [escf]: VI, VII, XVIII, XXVII
  - excited state structures and properties with CIS, RPA, TD-DFT [e0grad]: XIX, XXVI, XXVII
  - RI-CC2 [ricc2]:
    * singlet XII and triplet excitation energies XIII
    * transition moments and first-order properties of excited states XV and first-order properties for triplet states XIV
    * ground state geometry optimizations XXI
    * excited state geometry optimizations and relaxed properties XXII
    * parallelization XXIX
    * spin-component scaled (SCS) variants XXXII
    * frequency-dependent and static polarizabilities XXXVI
  - RI-ADC(2), RI-CIS(D) and RI-CIS(D∞) [ricc2]: XXVIII
  - SOS variants of MP2, CIS(D), CIS(D∞), ADC(2) and CC2 with $O(N^4)$-scaling XXXIII
  - analytical second derivatives (force fields) [aoforce]: XVI, XVII
  - RI-JK [ridft]: XX
  - NMR chemical shifts [mpshift]: IX (MP2)
  - parallel DFT [ridft]: X
  - geometry optimization in redundant internal coordinates [relax]: XI
  - RI integral evaluation: XXV
  - explicitly correlated F12 methods for ground state energies [ccsdf12and pno0ccsd]: MP2-F12 XXXIV, PNO-MP2-F12 XXXVIII, MP3-F12 XXXV, MP4(F12*) XXXX, CCSD(F12) XXXI, CCSD(F12*) XXXX, CCSD(F12)(T) XXXI, CCSD(F12*)(T) XXXV
• Orbital and auxiliary basis sets

  – basis sets:
    * SV, SV(P), SVP, DZ (a), TZV, TZVP, TZVPP (b), TZVPP(Rb-Hg) (f), QZV, QZVP, QZVPP (i)
    * new balanced basis sets (with smaller ECPs, i.e. the def2 basis sets): j
    * all-electron basis sets for Rb to Xe (SVPall, SVPPall, TZVPall, TZVP-Pall): g
    * references for the correlation consistent basis sets (cc-pVXZ, etc.) can be found e.g. at
      http://tyr0.chem.wsu.edu/~kipeters/Pages/cc_append.html or
    Note, that most of the correlation consistent basis sets in the basis set library of TURBOMOLE have been downloaded from the latter EMSL web site and therefore users are requested to include in addition to the original scientific reference an appropriate citation (see web site) in any publications resulting from the use of these basis sets.
    * property–optimized augmentations: def2-SVPD, def2-TZVPD, def2-TZVPPD, def2-QZVPD, def2-QZVPPD (n).
    * basis sets for Dirac–Fock ECPs, i.e. the dhf basis sets: o.
  – auxiliary basis sets for RI-DFT: c, d, e
  – auxiliary basis sets for RI-MP2: f, k, h (for Dunning basis sets)

Further references of papers not from the TURBOMOLE group are given in the bibliography. The following publications describe details of the methodology implemented in TURBOMOLE:

**Methods**


IV. Efficient Molecular Numerical Integration Schemes.


1.3. HOW TO QUOTE USAGE OF TURBOMOLE


XXIX. Distributed memory parallel implementation of energies and gradients for second-order Møller-Plesset perturbation theory with the resolution-of-the-identity ap-

XXX. Distributed memory parallel implementation of energies and gradients for second-order Møller-Plesset perturbation theory with the resolution-of-the-identity ap-


1.3. HOW TO QUOTE USAGE OF TURBOMOLE


Basis sets

The following tables can be used to find the proper citations of the standard orbital and auxiliary basis sets in the TURBOMOLE basis set library.

**Orbital basis sets, elements H–Kr**

<table>
<thead>
<tr>
<th></th>
<th>H,He</th>
<th>Li</th>
<th>Be</th>
<th>B–Ne</th>
<th>Na,Mg</th>
<th>Al–Ar</th>
<th>K</th>
<th>Ca</th>
<th>Sc–Zn</th>
<th>Ga–Kr</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVP,SV(P)</td>
<td>r</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>TZVP</td>
<td>r</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
</tr>
<tr>
<td>TZVPP</td>
<td>r</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>QZVP,QZVPP</td>
<td>r</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>def2-SV(P)</td>
<td>r</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>def2-SVP</td>
<td>r</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>j</td>
<td>a</td>
<td>a</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>def2-TZVP</td>
<td>r</td>
<td>f</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def2-TZVPP</td>
<td>r</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def2-XVPD/XVPPD, X=S,T,Q</td>
<td>r</td>
<td>j</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>f</td>
<td>j</td>
<td></td>
</tr>
</tbody>
</table>

Note: For H–Kr def-SV(P), def-SVP, ... are identical with the basis sets without def prefix. def2-QZVPP and def2-QZVP are identical with QZVPP and QZVP. def2-XVPD/XVPPD denotes the property–optimized augmentations def2-SVPD, def2-TZVPD, def2-TZVPPD, def2-QZVPD, def2-QZVPPD.

**Orbital basis sets, elements Rb–Rn**

<table>
<thead>
<tr>
<th></th>
<th>Rb</th>
<th>Sr</th>
<th>Y–Cd</th>
<th>In–Cs</th>
<th>Ba</th>
<th>La–Hg</th>
<th>Tl</th>
<th>At</th>
<th>Rn</th>
</tr>
</thead>
<tbody>
<tr>
<td>def-SVP,def-SV(P),def-TZVP</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>d</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def-TZVPP</td>
<td>f</td>
<td>d</td>
<td>f</td>
<td>d</td>
<td>f</td>
<td>d</td>
<td>j</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def2-SV(P)</td>
<td>j</td>
<td>d</td>
<td>d</td>
<td>j</td>
<td>d</td>
<td>d</td>
<td>j</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def2-SVP</td>
<td>j</td>
<td>d</td>
<td>j</td>
<td>j</td>
<td>d</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>def2-TZVP,def2-TZVPP</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-QZVP,def2-QZVP</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td>j</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-XVPD/XVPPD, X=S,T,Q</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Auxiliary basis sets for RI-DFT (Coulomb fitting)**

<table>
<thead>
<tr>
<th></th>
<th>H–Kr</th>
<th>Rb–At</th>
<th>Rn</th>
</tr>
</thead>
<tbody>
<tr>
<td>(def-)SVP,(def-)SV(P)</td>
<td>c</td>
<td>d</td>
<td>l</td>
</tr>
<tr>
<td>(def-)TZVP</td>
<td>d</td>
<td>d</td>
<td>l</td>
</tr>
<tr>
<td>def2 universal</td>
<td>l</td>
<td>l</td>
<td>l</td>
</tr>
</tbody>
</table>
### Auxiliary basis sets for RI-MP2 and RI-CC, elements H–Ar

<table>
<thead>
<tr>
<th>Basis Set</th>
<th>H</th>
<th>He</th>
<th>Li</th>
<th>Be</th>
<th>B–F</th>
<th>Ne</th>
<th>Na,Mg</th>
<th>Al–Cl</th>
<th>Ar</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVP,SV(P)</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>TZVP,TZVPP</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>QZVP,QZVPP</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-SV(P)</td>
<td>f</td>
<td>k</td>
<td>m</td>
<td>f</td>
<td>f</td>
<td>k</td>
<td>m</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-SVP</td>
<td>f</td>
<td>k</td>
<td>m</td>
<td>f</td>
<td>f</td>
<td>k</td>
<td>m</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-TZVP,def2-TZVPP</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>m</td>
<td>f</td>
<td>k</td>
<td>m</td>
<td>m</td>
<td>k</td>
</tr>
<tr>
<td>def2-XVPD/XVPPD, X=S,T,Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(aug-)cc-pVXZ, X=D–Q</td>
<td>h</td>
<td>h</td>
<td>k</td>
<td>k</td>
<td>h</td>
<td>h</td>
<td>k</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>(aug-)cc-pV5Z</td>
<td>k</td>
<td>k</td>
<td>-</td>
<td>-</td>
<td>k</td>
<td>k</td>
<td>-</td>
<td>k</td>
<td>k</td>
</tr>
<tr>
<td>cc-pCVXZ, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>k</td>
<td>k</td>
<td>-</td>
<td>k</td>
<td>k</td>
</tr>
</tbody>
</table>

Note: the auxiliary basis sets for the (aug-)cc-pV(X+d)Z basis sets for Al–Ar are identical with the (aug-)cc-pVXZ auxiliary basis sets.

### Auxiliary basis sets for RI-MP2 and RI-CC, elements K–Kr

<table>
<thead>
<tr>
<th>Basis Set</th>
<th>K</th>
<th>Ca</th>
<th>Sc–Zn</th>
<th>Ga–Br</th>
<th>Kr</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVP,SV(P)</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>TZVP,TZVPP</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>QZVP,QZVPP</td>
<td>f</td>
<td>k</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-SV(P)</td>
<td>m</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-SVP</td>
<td>m</td>
<td>f</td>
<td>m</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-TZVP,def2-TZVPP</td>
<td>m</td>
<td>f</td>
<td>m</td>
<td>f</td>
<td>k</td>
</tr>
<tr>
<td>def2-XVPD/XVPPD, X=S,T,Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>q</td>
</tr>
<tr>
<td>(aug-)cc-pVXZ, X=D–Q</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>(aug-)cc-pV5Z</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>cc-pCWVXZ, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>(aug-)cc-pVXZ-PP, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>cc-pwCVXZ-PP, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>

### Auxiliary basis sets for RI-MP2 and RI-CC, elements Rb–Rn

<table>
<thead>
<tr>
<th>Basis Set</th>
<th>Rb</th>
<th>Sr</th>
<th>Y–Cd</th>
<th>In–Xe</th>
<th>Cs</th>
<th>Ba</th>
<th>La–Hg</th>
<th>Tl–At</th>
<th>Rn</th>
</tr>
</thead>
<tbody>
<tr>
<td>def-SV,def-SV(P)</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-SV,def2-SV(P)</td>
<td>m</td>
<td>f</td>
<td>f</td>
<td>m</td>
<td>m</td>
<td>f</td>
<td>f</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>def-TZVP,def-TZVPP</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-TZVP,def2-TZVPP</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-QZVP,def2-QZVPP</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>def2-XVPD/XVPPD, X=S,T,Q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>q</td>
</tr>
<tr>
<td>(aug-)cc-pVXZ-PP, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>cc-pwCVXZ-PP, X=D–5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>p</td>
<td>p</td>
</tr>
</tbody>
</table>


g. Contracted all-electron Gaussian basis sets for Rb to Xe. R. Ahlrichs and K. May; Phys. Chem. Chem. Phys., 2, 943 (2000).


k. Optimization of auxiliary basis sets for RI-MP2 and RI-CC2 calculation: Core-valence and quintuple-ζ basis sets for H to Ar and QZVPP basis sets for Li to Kr. C. Hättig; Phys. Chem. Chem. Phys. 7, 59 (2005).


r. unpublished.
1.4 Modules and Their Functionality

For references see Bibliography.

**define** interactive input generator which creates the input file `control`. `define` supports most basis sets in use, especially the only fully atom optimized consistent basis sets of SVP and TZV quality [2–6] available for the atoms H–Rn, excluding lanthanides. `define` determines the molecular symmetry and internal coordinates allowing efficient geometry optimization. `define` allows to perform a geometry optimization at a force field level to preoptimize the geometry and to calculate a Cartesian Hessian matrix. `define` sets the keywords necessary for single point calculations and geometry optimizations within a variety of methods. There are also many features to manipulate geometries of molecules: just try and see how it works.

**uff** performs a geometry optimization at a force field level. The Universal Force Field (UFF) [7] is implemented. Beyond this it calculates an analytical Hessian (Cartesian) which will be used as a start Hessian for an *ab initio* geometry optimization.

**dscf** for (semi–)direct SCF–HF and DFT calculations (see keywords for functionals supported). `dscf` supports restricted closed-shell (RHF), spin-restricted ROHF as well as UHF runs. `dscf` includes an in-core version for small molecules.

**grad** requires a successful `dscf` run and calculates the gradient of the energy with respect to nuclear coordinates for all cases treated by `dscf`.

**ridft** and **rdgrad** perform (direct) SCF–HF and DFT calculations—as `dscf` and `grad`—within the very efficient RI–J approximation for the interelectronic Coulomb term. These programs also permit to approximate HF exchange within the RI–K approximation. The exchange correlation functionals supported are specified in `define`.

**riper** performs DFT calculations for molecules and periodic systems using the RI technique combined with the CFMM scheme. In addition, the low-memory RI implementation based on preconditioned conjugate gradient algorithm is available for molecular systems. Both RHF and UHF runs are supported.

**mpgrad** requires a well converged SCF run—by `dscf`, see keywords—and performs closed-shell RHF or UHF calculations yielding single point MP2 energies and, if desired, the corresponding gradient.

**ricc2** calculates electronic ground and excitation energies, transition moments and properties of ground and excited states at the MP2, CIS, CIS(D), ADC(2) and CC2 level using either a closed-shell RHF or a UHF SCF reference function. Calculates R12 basis set limit correction for MP2
energies. Employs the RI technique to approximate two-electron integrals. [8–15].

ccsd12 calculations of electronic ground state energies beyond MP2/CC2: RI-MP2-F12, MP3, MP3-F12, MP4, MP4(F12*), CCSD, CCSD(F12), CCSD(F12*), CCSD(F12)(T), CCSD(F12*)(T) and electronic excitation energies at the CCSD level. [16–19]

pnoccsd calculations of electronic ground state energies with PNO-based methods (currently restricted to MP2 and MP2-F12). [20, 21]

relax requires a gradient run—by grad, rdgrad, ricc2, egrad, or mpgrad—and proposes a new structure based on the gradient and the approximated force constants. The approximated force constants will be updated.

statpt performs structure optimization using the "Trust Radius Image Minimization" algorithm. It can be used to find minima or transition structures (first order saddle points). Transition structure searches usually require initial Hessian matrix calculated analytically or the transition vector from the lowest eigenvalue search.

frog executes one molecular dynamics (MD) step. Like relax, it follows a gradient run: these gradients are used as classical Newtonian forces to alter the velocities and coordinates of the nuclei.

aoforce requires a well converged SCF or DFT run—by dscf or ridft, see keywords—and performs an analytic calculation of force constants, vibrational frequencies and IR intensities. aoforce is also able to calculate only the lowest Hessian eigenvalues with the corresponding eigenvectors which reduces computational cost. The numerical calculation of force constants is also possible (see tool Numforce in Section 1.5).

escf requires a well converged SCF or DFT run and calculates time-dependent and dielectric properties (spin-restricted closed-shell or spin-unrestricted open-shell reference):

- static and frequency-dependent polarizabilities within the SCF approximation
- static and frequency-dependent polarizabilities within the time-dependent Kohn–Sham formalism, including hybrid functionals such as B3-LYP
- electronic excitations within the RHF and UHF CI(S) restricted CI method
- electronic excitations within the so-called SCF-RPA approximation (poles of the frequency dependent polarizability)
– electronic excitations within the time dependent Kohn–Sham formalism (adiabatic approximation). It can be very efficient to use the RI approximation here, provided that the functional is of non-hybrid type: we recommend B-P86 (but slightly better results are obtained for the hybrid functional B3-LYP) [22].

– stability analysis of single-determinant closed-shell wave functions (second derivative of energy with respect to orbital rotations) [23].

**egrad** computes gradients and first-order properties of excited states. Well converged orbitals are required. The following methods are available for spin-restricted closed shell or spin-unrestricted open-shell reference states:

– CI-Singles approximation (TDA)
– Time-dependent Hartree–Fock method (RPA)
– Time-dependent density functional methods

**egrad** can be employed in geometry optimization of excited states (using **jobex**, see Section 5.1), and in finite difference force constant calculations (using **Numforce**). Details see [24].

**rirpa** calculates ground state energies and analytic first-order properties within the random phase approximation (RPA), see Section 12.

**mpshift** requires a converged SCF or DFT run for closed shells. **mpshift** computes NMR chemical shieldings for all atoms of the molecule at the SCF, DFT or MP2 level within the GIAO ansatz and the (CPHF) SCF approximation. From this one gets the NMR chemical shifts by comparison with the shieldings for the standard compound usually employed for this purpose, e.g., TMS for carbon shifts. Note that NMR shielding typically requires more flexible basis sets than necessary for geometries or energies. ECPs are not supported in **mpshift** [25].

**freeh** calculates thermodynamic functions from molecular data in a control file; an **aoforce** or a **NumForce** run is a necessary prerequisite.

**intense** calculates Raman scattering cross sections from molecular data in a control file; an **aoforce** and an **egrad** run are a necessary prerequisite. Please use the **Raman** script to run these three steps in an automated way.

**woelfling** computes a finite number of structures along reaction paths within different interpolation algorithms. It provides an initial path using a modified Linear Synchronous Transit. See Section 5.7 for details. Please use the **woelfling-job** script to run optimizations with it.
1.5 Tools

Note: these tools are very helpful and meaningful for many features of TURBOMOLE.

This is a brief description of additional TURBOMOLE tools. Further information will be available by running the programs with the argument -help.

**actual**  please use: actual -help

**aoforce2g98**  usage: aoforce2g98 aoforce.out > g98.out
converts output from the aoforce program to Gaussian 98 style, which can be interpreted by some molecular viewer (e.g. jmol) to animate the normal coordinates.

**bend**  example: bend 1 2 3
displays the bending angle of three atoms specified by their number from the control file. Note that unlike in the TURBOMOLE definition of internal coordinates the apex atom is the second!

**cbasopt**  optimize auxiliary basis sets for RI-MP2 and RI-CC2 calculations. Uses ricc2 to calculate the error functional and its gradient and relax as optimization module. For further details call cbasopt -h.

**cc2cosmo**  manages macro iterations for RI-MP2, RI-CC2 or RI-ADC(2) calculations in an equilibrated solvent environment described by cosmo (see Chapter 20).

**cgnce**  plots energies as a function of SCF iteration number (gnuplot required).

**cosmoprep**  sets up control file for a cosmo run (see Chapter 20).

**dist**  example: dist 1 2
calculates atomic distances from TURBOMOLE input files; dist -l 4 gives all interatomic distances to 4 a.u. (5 a.u. is the default).

**DRC**  automates dynamic reaction coordinate calculations forward and backward along the imaginary vibrational mode of a transition state structure. A transition state optimization with a subsequent frequency calculation is prerequisite. For further details call DRC -h.

**eiger**  displays orbital eigenvalues obtained from data group $scfmo$. Shows HOMO-LUMO gap, occupation, checks if there are holes in the occupation, and much more.

**evalgrad**  reads the gradient file and prints the energies of each cycle versus bond lengths or angles. Five operational modes are possible:
evalgrad  prints the energy.
evalgrad 1  prints the coordinate of atom 1.
1.5. **TOOLS**

**evalgrad 1 2** prints the distance between atoms 1 and 2.
**evalgrad 1 2 3** prints the bending angle as defined in *Bend*.
**evalgrad 1 2 3 4** prints the torsional angle as defined in *Tors*.

**FDE**

*drive* the Frozen Density Embedding calculations.

**hcore**

prepares the control file for a Hamilton core guess.

**jobex**

usage: see Section 5.1

is the TURBOMOLE driver for all kinds of optimizations.

**kdg**

example: *kdg scfdiis*

kills a data group (here $scfdiis$) in the control file.

**lhfprep**

prepares for Localized Hartree-Fock calculations by adjusting parameters of the control file.

**log2x**

converts the file logging an MD trajectory into coordinates in frames appropriate for *jmol* animation program.

**log2egy**

extracts the energy data (KE, total energy, PE) from an MD log file.

**log2rog**

computes the radius of gyration, geometric radius and diameter from an MD log file.

**mdprep**

interactive program to prepare for an MD run, checking in particular the *mdmaster* file (**mdprep** is actually a FORTRAN program).

**MECPprep**

prepares the input for minimum-energy crossing point calculations. The subdirectories *state1* and *state2* will be created. Multiplicity and charge for the two states can be set. For further details call **MECPprep** -h.

**MECPopt**

*driver* for geometry optimizations of minimum-energy crossing points. The electronic structure calculations are carried out in the subdirectories *state1* and *state2* and the optimizer step is performed in the starting directory. For further details call **MECPopt** -h.

**mp2prep**

prepares MP2 calculations interactively by adjusting parameters of the control file according to your system resources.

**Numforce**

calculates numerically force constants, vibrational frequencies, and IR intensities. (Note that the name of the shell script is *NumForce* with capital F.)

**outp**

example: *outp 1 2 3 4*

displays the out-of-plan angle between atom1 and the plane that is defined by the last three atoms. atom1 is fixed at atom4.

**past**

translates and rotates coordinates in the principal axis system and prints out the rotational constants.
CHAPTER 1. PREFACE AND GENERAL INFORMATION

raman calculates vibrational frequencies and Raman intensities. See Section 14.2 for explanation.

screwer distorts a molecule along a vibrational mode.

scanprep prepares a series of control files with frozen internal coordinates. The data group $constraints (e.g. provided by TmoleX) is evaluated. For further details call scanprep -h.

vibration distorts a molecule along a vibrational mode or generates a plot of an IR spectrum (gnuplot required)

sdg shows data group from control file: for example sdg energy shows the list of calculated energies.

sysname returns the name of your system, used in almost all TURBOMOLE scripts.

stati prepares the control file for a statistics run.

t2x converts TURBOMOLE coordinates to xyz format.

tm2aomix creates an input file for the AOMix program. AOMix a software the analysis of molecular orbitals. For more information see: (http://www.sg-chem.net/aomix).

Uses tm2molden as described below by automatically adding the $aomix keyword to the control file.

tm2molden is a versatile tool to create

- molden format input file for the Molden program,
- AOMix input files or
- detailed information about the largest AO contributions to the MOs.

Molden is a graphical interface for displaying the molecular density, MOs, normal modes, and reaction paths. For more information about molden see: http://www.cmbi.ru.nl/molden/molden.html.

This format is also often used as input for other program packages or property tools.

If tm2molden finds the keyword $aomix in the control file, it will write out an AOMix input file, see: http://www.sg-chem.net/aomix.

Finally, tm2molden can be used to print out the largest contributions of the AO basis functions to the molecular orbitals.

Usage:

```
tm2molden mostat [molist] [above <threshold>]
e.g.: tm2molden mostat 230-240,251,255
      tm2molden mostat 434-440 above 0.001
      tm2molden mostat above 0.02
```
1.5. **TOOLS**

Only contributions which are larger than a certain percentage (default is 1%) are printed, this value can be changed with the **above** option (as absolute value, so 1% is 0.01). Without a list of orbitals (the numbering follows the output of **eiger**) all MOs are printed.

**tors** is a script to query a dihedral angle in a molecular structure:
e.g. **tors 1 2 3 4** gives the torsional angle of atom 4 out of the plane of atoms 1, 2 and 3.

**tbtim** is used to convert timings output files from **TURBOBENCH** calculations to **LATEX** tables (for options please type **TBTIM -help**).

**tblist** is used to produce summaries of timings from **TURBOBENCH** calculations to **LATEX** format. (for options please type **TBLIST -help**).

**uhfuse** transforms the UHF MOs from a given symmetry to another symmetry, which is **C$_1$** by default (just enter **uhfuse**). but can be specified (e.g. as **C$_{2v}$**) by entering **uhfuse -s c2v**. Now this functionality is included in the MO definition menu of **define** program, see Section 4.3.1.

**woelfling-job** optimizes a reaction path with **woelfling**.
For further information please type **woelfling-job -h**.

**x2t** converts standard xyz files into **TURBOMOLE** coordinates.
Chapter 2

Installation of TURBOMOLE

2.1 Install TURBOMOLE command line version

Installation requires familiarity with some simple UNIX commands. The TURBOMOLE package is generally shipped as one `tar` file. This has to be uncompressed

```
gunzip turbomole_70.tar.gz
```

and unpacked

```
tar -xvf turbomole_70.tar
```

to produce the whole directory structure.

**Note:** Do not install or run TURBOMOLE as root or with root permissions!

2.1.1 Settings for each user:

The environmental variable `$TURBODIR` must be set to the directory where TURBOMOLE has been unpacked, for example:

```
TURBODIR=/my_disk/my_name/TURBOMOLE
```

Then, the most convenient way to extend your path to the TURBOMOLE scripts and binaries is to source the file `Config_turbo_env`:

```
source $TURBODIR/Config_turbo_env
```

If you have a csh or tcsh as default login shell use

```
source $TURBODIR/Config_turbo_env.tcsh
```

instead.

It is recommended to add the two lines given above to your `.bashrc` (or `.profile` or wherever you prefer to add your local settings).
2.1.2 Setting system type and $PATH by hand

Check that the **Sysname** tool works on your computer:

```
$TURBODIR/scripts/sysname
```

should return the name of your system and this should match a bin/[arch] subdirectory.

If **Sysname** does not print out a single string matching a directory name in $TURBODIR/bin/, and if one of the existing binary versions does work, you can force **sysname** to print out whatever is set in the environment variable $TURBOMOLE_SYSNAME:

```
TURBOMOLE_SYSNAME=em64t-unknown-linux-gnu
```

Please make sure not to append _mpi or _smp to the string when setting $TURBOMOLE_SYSNAME, even if you intend to run parallel calculations. **sysname** will append this string automatically to the system name if $PARA_ARCH is set to MPI or SMP (see chapter 3.2.1 how to set up parallel environment).

You can call **TURBOMOLE** executables and tools easily from anywhere if you add the corresponding directories to your path (kornshell or bash syntax):

```
PATH=$PATH:$TURBODIR/scripts
PATH=$PATH:$TURBODIR/bin/'sysname'
```

Note that **sysname** is set in back quotes which tells the shell to substitute the entry by the output of **sysname**.

Now the **TURBOMOLE** executables can be called from a directory with the required input files. For example to call dscf and save the output:

```
$TURBODIR/bin/'sysname'/dscf > dscf.out
```

or if the path is OK, simply

```
dscf > dscf.out
```

Executable modules are in the bin/[arch] directory (for example, Linux modules are in bin/em64t-unknown-linux-gnu). Tools (including jobex) are in scripts and (auxiliary) basis sets are kept in the directories basen, jbasen, jkbasen, cbasen, xbasen and cabasen. Coordinates for some common chemical fragments are supplied in structures. The documentation and a tutorial can be found in the folder DOC.

2.1.3 Testing the installation

In addition, some sample calculations are supplied in **Turbotest** so that the modules can be tested. Just run **TTEST** from this directory to run all tests or **TTEST -help** to get help on how this works:
cd $TURBODIR/TURBOTEST

test

2.2 Installation problems: How to solve

Please check your user limits!
If one or several tests of the test suite fail, it is very likely that your user limits for stack size and/or memory are too small.

sh/bash/ksh users: please do a

    ulimit -a

to get your actual limits. The output should look like:

<table>
<thead>
<tr>
<th>Limit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>core file size (blocks)</td>
<td>0</td>
</tr>
<tr>
<td>data seg size (kbytes)</td>
<td>unlimited</td>
</tr>
<tr>
<td>file size (blocks)</td>
<td>unlimited</td>
</tr>
<tr>
<td>max locked memory (kbytes)</td>
<td>unlimited</td>
</tr>
<tr>
<td>max memory size (kbytes)</td>
<td>unlimited</td>
</tr>
<tr>
<td>open files</td>
<td>1024</td>
</tr>
<tr>
<td>pipe size (512 bytes)</td>
<td>8</td>
</tr>
<tr>
<td>stack size (kbytes)</td>
<td>unlimited</td>
</tr>
<tr>
<td>cpu time (seconds)</td>
<td>unlimited</td>
</tr>
<tr>
<td>max user processes</td>
<td>8191</td>
</tr>
<tr>
<td>virtual memory (kbytes)</td>
<td>unlimited</td>
</tr>
</tbody>
</table>

The most important entries are data size, stack size, max memory size, and virtual memory. Those should be either unlimited or as big as your total RAM.

To set, e.g. the stack size to the maximum allowed size on your system (the so called hard limit), do:

    ulimit -s hard

csh/tcsh users: please do limit instead of ulimit and check the output.

Again, like given above, the limits should be at least as high as your memory available. The syntax for changing the limits to unlimited using csh/tcsh is:

    limit stacksize hard

And please note that on 32bit machines, unlimited can be the same as 4GB (4194304 kbytes).

If you are using a queuing system:
Note that if you are submitting jobs to a queue, the user limits might be different from what you get when you log in on the machines! To check your limits, you have to add `ulimit` or `limit` in the script that is sent to the queue:

```
....
ulimit -a > mylimits.out
jobex -ri -c 200 -statpt > jobex.out
...
```

send it to the queue and check the file `mylimits.out` to find out which limits are set.

**Parallel version:**

The parallel binaries are being started by the `mpirun` command which often uses `ssh` to start a process on a remote node. The limits for the stack size can not be set by the user in such a case, so everything in `$HOME/.profile`, `$HOME/.bashrc`, etc. will not help to get rid of the problem.

To check the limits on a remote node, try (sh/bash/ksh syntax):

```
ssh <hostname> ulimit -a
```

If the `ssh` command gives a lower stack size than `unlimited` or a large number, you have to change the file

```
/etc/security/limits.conf
```

on all nodes where the parallel binaries might run, and add there the line (example for 4GB limit)

```
* soft stack 4194303
```

Redo `ssh <hostname>ulimit -a` and you should get 4GB stack size limit, as it is set in `limits.conf` now.
Chapter 3

How to Run TURBOMOLE

3.1 A ‘Quick and Dirty’ Tutorial

A detailed tutorial for the usage of TURBOMOLE on the command line can be found in the DOC directory of your TURBOMOLE installation or on the web site of COSMOlogic, see http://www.cosmologic.de/

All TURBOMOLE modules need the control file as input file. The control file provides directly or by cross references the information necessary for all kinds of runs and tasks (see Section 21). define provides step by step the control file: Coordinates, atomic attributes (e.g. basis sets), MO start vectors and keywords specific for the desired method of calculation. We recommend generating a set of Cartesian coordinates for the desired molecule using special molecular design software and converting this set into TURBOMOLE format (see Section 22.2) as input for define. Alternatively the graphical user interface TmoleX can be used to import and/or build molecules.

A straightforward way to perform a TURBOMOLE calculation from scratch is as follows:

- generate your atomic coordinates by any tool or program you are familiar with,
- save it as an .xyz file which is a standard output format of all programs, or use a conversion tool like babel,
- use the TURBOMOLE script x2t to convert your .xyz file to the TURBOMOLE coord file:
  \[ x2t \text{ xyzinputfile } > \text{ coord} \]
- since input files for TURBOMOLE are always called control, each input has to be placed in a different directory. Create a new directory and copy the coord file there,
- call define; after specifying the title, you get the coord menu—just enter a coord to read in the coordinates. Use desy to let define determine the point group automatically.
If you want to do geometry optimizations, we recommend to use generalized internal coordinates; iRed generates them automatically.

- you may then go through the menus without doing anything: just press <Enter>, * or q—whatever ends the menu, or by confirming the proposed decision of define again by just pressing <Enter>. This way you get the necessary specifications for a (SCF-based) run with SV(P) as the default basis set which is roughly 6-31G*.

- for more accurate SCF or DFT calculations choose larger basis sets, e.g. TZVP by entering b all def-TZVP or b all def2-TZVP in the basis set menu.

- ECPs which include (scalar) relativistic corrections are automatically used beyond Kr.

- an initial guess for MOs and occupation numbers is provided by eht

- for DFT you have to enter dft in the last menu and then enter on

- for efficient DFT calculations you best choose the RI approximation by entering ri and providing roughly 3/4 of the memory (with m number; number in MB) your computer has available. (Auxiliary basis sets are provided automatically) In the printout of an ridft run you can check how much is really needed; a top statement will tell you if you overplayed your cards.

- B-P86 is the default functional. It has a good and stable performance throughout the periodic system.

- for an HF or DFT run without RI, you simply enter:
  
  [nohup] dscf > dscf.out &

  or, for a RI-DFT run:
  
  [nohup] ridft > ridft.out &

- for a gradient run, you simply enter:
  
  [nohup] grad > grad.out &

  or
  
  [nohup] rdgrad > rdgrad.out &

- for a geometry optimization simply call jobex:
  for a standard SCF input:
  
  [nohup] jobex &

  for a standard RI-DFT input:
  
  [nohup] jobex -ri &

- many features, such as NMR chemical shifts on SCF and DFT level, do not require further modifications of the input, just call e.g. mpsift after the appropriate energy calculation (mpshift runs with SCF or DFT using a hybrid-functional need a file size of the semi-direct file twoint that is non-zero).
3.1.1 Single Point Calculations: Running TURBOMOLE Modules

All calculations are carried out in a similar way. First you have to run define to obtain the control file or to add/change the keywords you need for your purpose. This can also be done manually with an editor. Given a bash and a path to $TURBODIR/bin/[arch] (see installation, Chapter 2) you call the appropriate module in the following way (e.g. module dscf):

```
nohup dscf > dscf.out &
```

nohup means that the command is immune to hangups, logouts, and quits. & runs a background command. The output will be written to the file dscf.out. Several modules write some additional output to the control file. For the required keywords see Section 21. The features of TURBOMOLE will be described in the following section.

3.1.2 Energy and Gradient Calculations

Energy calculations may be carried out at different levels of theory.

Hartree–Fock–SCF

use modules dscf and grad or ridft and rdgrad to obtain the energy and gradient. The energy can be calculated after a define run without any previous runs. dscf and grad need no further keywords ridft and rdgrad only need the keyword $rij. The gradient calculation however requires a converged dscf or ridft run.

Density functional theory

DFT calculations are carried out in exactly the same way as Hartree–Fock calculations except for the additional keyword $dft. For DFT calculations with the fast Coulomb approximation you have to use the modules ridft and rdgrad instead of dscf and grad. Be careful: dscf and grad ignore RI–K flags and will try to do a normal calculation, but they will not ignore RI–J flags ($rij) and stop with an error message. To obtain correct derivatives of the DFT energy expression in grad or rdgrad the program also has to consider derivatives of the quadrature weights—this option can be enabled by adding the keyword weight derivatives to the data group $dft.

For a semi-direct dscf calculation (Hartree–Fock or DFT) you first have to perform a statistics run. If you type

```
stat dscf
nohup dscf > dscf.stat &
```
3.1. A ‘QUICK AND DIRTY’ TUTORIAL

the disk space requirement (MB) of your current $thime$ and $thize$ combination will be computed and written to the data group $scfintunit size=integer$ (see Section 21.2.6). The requirement of other combinations will be computed as well and be written to the output file $dscf.stat$. The size of the integral file can be set by the user to an arbitrary (but reasonable) number. The file will be written until it reaches the given size and $dscf$ will continue in direct mode for the remaining integrals. Note that TURBOMOLE has no 2GB file size limit.

MP2 and MP2-F12

MP2 calculations need well converged SCF runs (the SCF run has to be done with at least the density convergence $\text{denconv 1.d-7}$, and $\text{scfconv 7}$ as described in Section 21). This applies also to the spin-component scaled (SCS and SOS) and explicitly-correlated (F12) variants of MP2. For MP2 and MP2-F12 calculations in the RI approximation use the $\text{ricc2}$ or $\text{pnoocc}$ modules. The module $\text{mpgrad}$ calculates the conventional (non-RI and non-F12) MP2 energy its gradient (only recommended for test calculations). The input can be prepared with the $\text{mp2}$, $\text{cc}$, or $\text{pnoocc}$ menu in $\text{define}$.

Excited states with CIS, TDHF and TDDFT (escf)

Single point excited state energies for CIS, TDHF, and TDDFT methods can be calculated using $\text{escf}$. Excited state energies, gradients, and other first order properties are provided by $\text{egrad}$. Both modules require well converged ground state orbitals.

Excited states with second-order wavefunction methods (ricc2)

The module $\text{ricc2}$ calculates beside MP2 and CC2 ground state energies also CIS (identical to CCS), CIS(D), CIS(D$_{\infty}$), ADC(2) or CC2 excitation energies using the resolution-of-the-identity (RI) approximation. Also available are spin-component scaled (SCS and SOS) variants of the second-order methods CIS(D), CIS(D$_{\infty}$), ADC(2) or CC2. Excited state gradients are available at the CCS, CIS(D$_{\infty}$), ADC(2), and CC2 levels and the spin-component scaled variants of the latter three methods. In addition, transition moments and first-order properties are available for some of the methods. For more details see Section 10. The input can be prepared using the $\text{cc}$ menu of $\text{define}$.

Coupled-Cluster methods beyond CC2: CCSD(F12*)(T) (ccsdf12)

Coupled-Cluster methods beyond CC2 as CCSD and CCSD(T) and Møller-Plesset perturbation theory beyond MP2 and explicitly-correlated F12 variants thereof are since Release V7.0 implemented in the $\text{ccsdf12}$ program. The F12 variants of these methods have a much faster basis set convergence and are therefore more efficient. We recommend in particular CCSD(F12*) and CCSD(F12*)(T). Excitation energies are only available for (conventional) CCSD.
3.1.3 Calculation of Molecular Properties

See Section 1.4 for the functionality and Section 21 for the required keywords of the modules dscf, ridft, mpshift, escf, and ricc2.

3.1.4 Modules and Data Flow

See Figure 3.1.

3.2 Parallel Runs

Some of the TURBOMOLE modules are parallelized using the message passing interface (MPI) for distributed and shared memory machines or with OpenMP or multi-threaded techniques for shared memory and multi-core machines.

Generally there are two hardware scenarios which determine the kind of parallelization that is possible to use:

- On a single node with several CPUs and/or cores using the same memory (shared memory), the user can run all parallelized modules of TURBOMOLE. For some modules, both shared-memory and MPI versions are available, but it is recommended not to use the latter ones for performance reasons.
  
  How to run the parallel TURBOMOLE SMP version on multi-core and/or multi-CPU systems: Please see chapter 3.2.2.
  
- On a cluster a parallel calculation can be performed using several distinct nodes, each one with local memory and disks. This can be done with the MPI version.
  
  How to run the parallel TURBOMOLE MPI version on clusters: Please see chapter 3.2.1.

The list of programs parallelized includes presently:

- ridft — parallel ground state Hartree-Fock and DFT energies including RI-J and the multipole accelerated RI (MA-RI-J)
  
- rdgrad — parallel ground state gradients from ridft calculations
  
- dscf — Hartree-Fock and DFT ground state calculations for all available DFT functionals, without the usage of RI-J approximation
  
- riper — parallel ground state DFT energies for molecular and periodic systems.
  
- grad — parallel ground state gradients from dscf calculations
3.2. PARALLEL RUNS

USER INPUT:
coordinates

input generator
define

From TURBOMOLE library:
basis sets

ground state energy
dscf
ridft

HF/DFT/TDDFT
gradient
grad
rdgrad
egrad

MP2/CC2
energy+gradient
ricc2
mpgrad

geometry changes
minimum or transition state
relax
statpt

molecular dynamics
frog

vib. frequencies
aoforce
NumForce

TDDFT excited states,
response properties
escf, egrad

NMR shieldings
mpshift

Raman spectrum
Raman

CC2 excited states and
response properties
ricc2

(PNO-)CC and
(PNO-)CC-F12
calculations
ccsdf12
pnoccsd

thermodynamics
Freeh

Figure 3.1: The modules of TURBOMOLE and the main data flow between them.
• **ricc2** — parallel ground and excited state calculations of energies and gradients at MP2 and CC2 level using RI, as well as energy calculations of other wave function models, see chapter 10.6.

• **ccsdf12** — parallel ground state energies beyond MP2/CC2 and excitation energies beyond CC2.

• **pnoccsd** — parallel ground state energies at the OSV-PNO-MP2 and OSV-PNO-MP2-F12 level.

• **mpgrad** — parallel conventional (i.e. non-RI) MP2 energy and gradient calculations. Please note that RI-MP2 is one to two orders of magnitude faster than conventional MP2, so even serial RI-MP2 will be faster than parallel MP2 calculations.

• **aoforce** — parallel Hartree-Fock and DFT analytic 2nd derivatives for vibrational frequencies, IR spectra, generation of Hessian for transition state searches and check for minimum structures. SMP only.

• **escf** — parallel TDDFT, RPA, CIS excited state calculations (UV-Vis and CD spectra, polarizabilities). SMP only.

• **egrad** — parallel TDDFT, RPA, CIS excited state analytic gradients, including polarizability derivatives for RAMAN spectra. SMP only.

• **NumForce** — this script can used for a trivial parallelization of the numerical displaced coordinates.

Additional keywords necessary for parallel runs with the MPI binaries are described in Chapter 21. However, those keywords do not have to be set by the users. When using the parallel version of TURBOMOLE, scripts are replacing the binaries. Those scripts prepare a usual input, run the necessary steps and automatically start the parallel programs. The users just have to set environment variables, see Sec. 3.2.1 below.

To use the OpenMP parallelization only an environment variable needs to be set. But to use this parallelization efficiently one should consider a few additional points, e.g. memory usage, which are described in Sec. 3.2.2.

### 3.2.1 Running Parallel Jobs — MPI case

The parallel version of TURBOMOLE runs on all supported systems:

• workstation cluster with Ethernet, Infiniband, Myrinet (or other) connection

• SMP systems

• or combinations of SMP and cluster
Setting up the parallel MPI environment

In addition to the installation steps described in Section 2 (see page 30) you just have to set the variable PARA_ARCH to MPI, i.e. in sh/bash/ksh syntax:

```bash
export PARA_ARCH=MPI
```

This will cause sysname to append the string _mpi to the system name and the scripts like jobex will take the parallel binaries by default. To call the parallel versions of the programs ridft, rdgrad, dscf, grad, ricc2, or mpgrad from your command line without explicit path, expand your $PATH environment variable to:

```bash
export PATH=$TURBODIR/bin/'sysname':$PATH
```

The usual binaries are replaced now by scripts that prepare the input for a parallel run and start mpirun (or poe on IBM) automatically. The number of CPUs that shall be used can be chosen by setting the environment variable PARNODES:

```bash
export PARNODES=8
```

The default for PARNODES is 2.

Finally the user can set a default scratch directory that must be available on all nodes. Writing scratch files to local directories is highly recommended, otherwise the scratch files will be written over the network to the same directory where the input is located. The path to the local disk can be set with

```bash
export TURBOTMPDIR=/scratch/username
```

This setting is automatically recognized by the parallel ridft and ricc2 programs. Note:

- This does not set the path for the integral scratch files for dscf (see section below about twoint of keyword $scfintunit).
- In MPI parallel runs the programs attach to the name given in $TURBOTMPDIR node-specific extension (e.g. /scratch/username-001) to avoid clashes between processes that access the same file system. The jobs must have the permissions to create these directories. Therefore one must not set $TURBOTMPDIR to something like /scratch which would result in directory names like /scratch-001 which can usually not created by jobs running under a standard user id.

On all systems TURBOMOLE is using the MPI library that has been shipped with your operating system.

On Linux for PCs and Windows systems IBM Platform MPI (formerly known as HP-MPI) is used — see

IBM Platform MPI
COSMOlogic ships TURBOMOLE with a IBM Platform MPI Community Edition. TURBOMOLE users do not have to install or license IBM Platform MPI themselves. Parallel binaries will run out of the box on the fastest interconnect that is found - Infiniband, Myrinet, TCP/IP, etc.

The binaries that initialize MPI and start the parallel binaries (mpirun) are located in the $TURBODIR/mpirun_scripts/em64t-unknown-linux-gnu_mpi/PMPI/ directory.

**Note:** the parallel TURBOMOLE modules (except ricc2) need an extra server running in addition to the clients. This server is included in the parallel binaries and it will be started automatically—but this results in one additional task that usually does not need any CPU time. So if you are setting $PARNODES to N, N+1 tasks will be started.

If you are using a queuing system or if you give a list of hosts where TURBOMOLE jobs shall run on (see below), make sure that the number of supplied nodes match $SPPARNOES — e.g. if you are using 4 CPUs via a queuing system, make sure that $SPPARNOES is set to 4.

In some older versions of the LoadLeveler on IBM systems the total number of tasks must be set to $SPPARNOES + 1 (except for ricc2).

**Starting parallel jobs**

After setting up the parallel environment as described in the previous section, parallel jobs can be started just like the serial ones. If the input is a serial one, it will be prepared automatically for the parallel run.

The parallel versions of the programs dscf and grad need an integral statistics file as input which is generated by a parallel statistics run. This preparation step is done automatically by the scripts dscf and grad that are called in the parallel version. In this preparing step the size of the file that holds the 2e-integrals for semi-direct calculations twoint is recalculated and reset. It is highly recommended to set the path of this twoint file to a local scratch directory of each node by changing the line.

\[
\text{unit}=30 \text{ size}=?????? \text{ file}=\text{twoint}
\]

to

\[
\text{unit}=30 \text{ size}=?????? \text{ file}=/\text{local_scratchdir}/\text{twoint}
\]

For the additional mandatory or optional input for parallel runs with the ricc2 program see Section 10.6.

**Running calculations on different nodes**

If TURBOMOLE is supposed to run on a cluster, we highly recommend the usage of a queuing system like PBS. The parallel version of TURBOMOLE will automatically recognise that it is started from within the PBS environment and the binaries will run on the machines PBS provides for each job.
3.2. PARALLEL RUNS

Important: Make sure that the input files are located on a network directory like an NFS disk which can be accessed on all nodes that participate at the calculation.

A file that contains a list of machines has to be created, each line containing one machine name:

node1
node1
node2
node3
node4
node4

And the environment variable $HOSTS_FILE has to be set to that file:

```
export HOSTS_FILE=/nfshome/username/hostsfile
```

Note: Do not forget to set $PARNODES to the number of lines in $HOSTS_FILE.

Note: In general the stack size limit has to be raised to a reasonable amount of the memory (or to unlimited). In the serial version the user can set this by `ulimit -s unlimited` on bash/sh/ksh shells or `limit stacksize unlimited` on csh/tcsh shells. However, for the parallel version that is not sufficient if several nodes are used, and the `/etc/security/limits.conf` files on all nodes might have to be changed. Please see the following web site for details: [Turbomole User Forum](https://turbomole.org)

Testing the parallel binaries

The binaries `ridft`, `rdgrad`, `dscf`, `grad`, and `ricc2` can be tested by the usual test suite: go to `$TURBODIR/TURBOTEST` and call `TTEST`

Note: Some of the tests are very small and will only pass properly if 2 CPUs are used at maximum. Therefore `TTEST` will not run any test if $PARNODES is set to a higher value than 2.

If you want to run some of the larger tests with more CPUs, you have to edit the `DEFCRIT` file in `TURBOMOLE/TURBOTEST` and change the `$defmaxnodes` option.

Linear Algebra Settings

The number of CPUs and the algorithm of the linear algebra part of Turbomole depends on the settings of `$parallel_platform`:

- **cluster** – for clusters with TCP/IP interconnect. Communication is avoided by using an algorithm that includes only one or few CPUs.
- **MPP** – for clusters with fast interconnect like Infiniband or Myrinet. Number of CPUs that take part at the calculation of the linear algebra routines depends on the size of the input and the number of nodes that are used.
SMP – all CPUs are used and SCALapack (see http://www.netlib.org/scalapack/) routines are involved.

The scripts in $TURBODIR/mpirun_scripts automatically set this keyword depending on the output of sysname. All options can be used on all systems, but especially the SMP setting can slow down the calculation if used on a cluster with high latency or small bandwidth.

Sample simple PBS start script

```
#!/bin/sh
# Name of your run :
#PBS -N turbomole
#
# Number of nodes to run on:
#PBS -l nodes=4
#
# Export environment:
#PBS -V

# Set your TURBOMOLE pathes:

######## ENTER YOUR TURBOMOLE INSTALLATION PATH HERE ##########
export TURBODIR=/whereis/TURBOMOLE
############################################################
export PATH=$TURBODIR/scripts:$PATH

## set locale to C
unset LANG
unset LC_CTYPE

# set stack size limit to unlimited:
ulimit -s unlimited

# Count the number of nodes
PBS_L_NODENUMBER='wc -l < $PBS_NODEFILE'

# Check if this is a parallel job
if [ $PBS_L_NODENUMBER -gt 1 ]; then
    ###### Parallel job
    # Set environment variables for a MPI job
    export PARA_ARCH=MPI
    export PATH="$TURBODIR\bin/\sysname":"PATH"
    export PARNODES="expr $PBS_L_NODENUMBER"
else
3.2. PARALLEL RUNS

3.2.1 Sequentiel job

```bash
##### set the PATH for Turbomole calculations
export PATH="${TURBODIR}/bin/‘sysname’:${PATH}"
fi

#VERY important is to tell PBS to change directory to where
# the input files are:

cd $PBS_O_WORKDIR

######## ENTER YOUR JOB HERE ##################################
jobex -ri > jobex.out
###############################################################
```

3.2.2 Running Parallel Jobs — SMP case

The SMP version of TURBOMOLE currently combines three different parallelization schemes which all use shared memory:

- `dscf`, `ricc2` and `riper` are also partially parallelized with OpenMP for applications on shared-memory, in particular multi-CPU and multi-core, machines.
- `aoforce`, `escf` and `egrad` are currently parallelized as described in [26]
- `ridft` and `rdgrad` are parallelized with MPI using the Global Arrays toolkit, but use shared memory on SMP systems.

Setting up the parallel SMP environment

In addition to the installation steps described in Section 2 (see page 30) you just have to set the variable PARA_ARCH to SMP, i.e. in sh/bash/ksh syntax:

```bash
export PARA_ARCH=SMP
```

This will cause `sysname` to append the string `_smp` to the system name and the scripts like `jobex` will take the parallel binaries by default. To call the parallel versions of the programs `ridft`, `rdgrad`, `dscf`, `riper`, `ricc2`, `aoforce`, `escf` or `egrad` from your command line without explicit path, expand your $PATH environment variable to:

```bash
export PATH=$TURBODIR/bin/‘sysname’:PATH
```

The usual binaries are replaced now by scripts that prepare the input for a parallel run and start the job automatically. The number of CPUs that shall be used can be chosen by setting the environment variable `PARNODES`:

```bash
export PARNODES=8
```
CHAPTER 3. HOW TO RUN TURBOMOLE

The default for PARNODES is 2.

**NOTE:** Depending on what you are going to run, some care has to be taken that the system settings like memory limits, etc. will not prevent the parallel versions to run. See the following sections.

**OpenMP parallelization of dscf, ricc2, ccsdf12, pnoccsd, and riper**

The OpenMP parallelization does not need any special program startup. The binaries can be invoked in exactly the same manner as for sequential (non-parallel) calculations. The only difference is, that before the program is started the environment variable PARNODES has to be set to the number or threads that should be used by the program, the scripts will set OMP_NUM_THREADS to the same value and start the OpenMP binaries. The number of threads is essentially the max. number of CPU cores the program will try to utilize. To exploit e.g. all eight cores of a machine with two quad-core CPUs set

```bash
export PARNODES=8
```

(for csh and tcsh use `setenv PARNODES=8`).

Presently the OpenMP parallelization of ricc2 comprises all functionalities apart from the $O(N^4)$-scaling LT-SOS-RI functionalities (which are only parallelized with MPI) and expectation values for $\hat{S}^2$ (not parallelized). Note that the memory specified with $\text{maxcor}$ is for OpenMP-parallel calculation the maximum amount of memory that will be dynamically allocated by all threads together. To use your computational resources efficiently, it is recommended to set this value to about 75% of the physical memory available for your calculations, but to at most 16000 (megabytes).

(Due to the use of integer*4 arithmetics the ricc2 program is presently limited to 16 Gbytes.)

In the dscf program the OpenMP parallelization covers presently only the Hartree-Fock coulomb and exchange contributions to the Fock matrix in fully integral-direct mode and is mainly intended to be used in combination with OpenMP parallel runs of ricc2. Nevertheless, the OpenMP parallelization can also be used in DFT calculations, but the numerical integration for the DFT contribution to the Fock matrix will only use a single thread (CPU core) and thus the overall speed up will be less good.

Localized Hartree-Fock calculations (dscf program) are parallelized using OpenMP. In this case an almost ideal speedup is obtained because the most expensive part of the calculation is the evaluation of the Fock matrix and of the Slater-potential, and both of them are well parallelized. The calculation of the correction-term of the grid will use a single thread.

The OpenMP parallelization of riper covers all contributions to the Fock matrix. Hence an almost ideal speedup is obtained.

**Restrictions:**
3.2. PARALLEL RUNS

- In the ricc2 program the parts related to RI-MP2-F12, LT-SOS-RI-MP2 or calculation of expectation values for $\hat{S}^2$ do not (yet) use OpenMP parallelization. If the OpenMP parallelization is switched on (by setting OMP_NUM_THREADS) these parts will still be executed sequentially.

- In the dscf program the DFT part will only be executed sequentially by a single thread and the $\textit{incore}$ option will be ignored if more than one thread is used. Semi-direct dscf calculations (i.e. if a size larger than 0 is given two-electron integral scratch file in $\textit{scfintunit}$) can not be combined with the OpenMP parallel runs. (The program will than stop with error message in the first Fock matrix construction.)

Multi-thread parallelization of aoforce, escf and egrad

The parallelization of those modules is described in [26] and is based on fork() and Unix sockets. Except setting PARANODES which triggers the environment variable SMP_Cpus, nothing has to be set in addition. Alternatively, the binaries can be called with -smcpus <N> command line option or with the keyword $\textit{smp_cpus}$ in the control file.

Multi-thread parallelization of dscf, grad, ridft and rdgrad

Instead of the default binaries used in the SMP version, setting

$\textit{TM\_PAR\_FORK=on}$

will cause the TURBOMOLE scripts to use the multi-threaded versions of dscf, grad, ridft and rdgrad. The efficiency of the parallelization is usually similar to the default version, but for ridft and rdgrad RI-K is not parallelized. If density convergence criteria ($\textit{denconv}$) is switched on using ridft and if no RI-K is being used, the multi-threaded version should be used.

Global Arrays parallelization of ridft and rdgrad

ridft and rdgrad are parallelized with MPI using the Global Arrays toolkit.

Those versions are automatically used when setting PARA_ARCH=SMP. Due to the explicit usage of shared memory on an SMP system, user has to be allowed to use sufficient shared memory:

- In addition to the usual stack size limit problem, make sure that your maximum shared memory you are allowed to use on your system is large enough (a few GB or 70-90 percent of the total memory should be sufficient):
  - cat /proc/sys/kernel/shmmax shows the amount of allowed shared memory
  - use sysctl to set new values, but you have to be root to do that!
• the default shared memory that is used is per process for the matrices is 300 MB for heap and 10 MB for stack. For large cases this can be too small and an error will be given in the output. In order to increase the default values, just set:

$paroptions
   ga_memperproc <stacksize> <heapsize>

stacksize and heapsize have to be given in words, i.e. units of 8 Byte (1 MB is equivalent to 131072 word).
Chapter 4

Preparing your input file with DEFINE

define is the general interactive input generator of TURBOMOLE. During a session with define, you will create the control file which controls the actions of all other TURBOMOLE programs. During your define session you will be guided through four main menus:

1. The geometry main menu: This first menu allows you to build your molecule, define internal coordinates for geometry optimizations, determine the point group symmetry of the molecule, adjust internal coordinates to the desired values and related operations. Beyond this one can perform a geometry optimization at a force field level to preoptimize the geometry and calculate a Cartesian analytical Hessian. After leaving this menu, your molecule to be calculated should be fully specified.

2. The atomic attributes menu: Here you will have to assign basis sets and/or effective core potentials to all atoms. The SV(P) basis is assigned automatically as default, as well as ECPs (small core) beyond Kr.

3. The occupation numbers and start vectors menu: In this menu you should choose eht to start from Extended Hückel MO vectors. Then you have to define the number of occupied orbitals in each irreducible representation.

4. The general menu: The last menu manages a lot of control parameters for all TURBOMOLE programs.

Most of the menu commands are self-explanatory and will only be discussed briefly. Typing * (or q) terminates the current menu, writes data to control and leads to the next while typing & goes back to the previous menu.
4.0.3 Universally Available Display Commands in Define

There are some commands which may be used at (almost) every stage of your define session. If you build up a complicated molecular geometry, you will find the dis command useful. It will bring you to the following little submenu:

**ANY COMMAND WHICH STARTS WITH THE 3 LETTERS dis IS A DISPLAY COMMAND. AVAILABLE DISPLAY COMMANDS ARE:**

- **disc** <range>: DISPLAY CARTESIAN COORDINATES
- **dist** <real>: DISPLAY DISTANCE LIST
- **disb** <range>: DISPLAY BONDING INFORMATION
- **disa** <range>: DISPLAY BOND ANGLE INFORMATION
- **Disi** <range>: DISPLAY VALUES OF INTERNAL COORDINATES
- **disg** <range>: GRAPHICAL DISPLAY OF MOL. GEOMETRY

<range> IS A SET OF ATOMS REFERENCED
<real> IS AN OPTIONAL DISTANCE THRESHOLD (DEFAULT=5.0)

As an example consider disc 1,3-6,10,11 which displays the Cartesian coordinates of atoms 1,3,4,5,6,10, and 11.

Hit >return< to continue or enter any display command.

Of course, you may enter each of these display commands directly without entering the general command dis before. The option disg needs special adaption to the computational environment, however, and will normally not be available.

4.0.4 Specifying Atomic Sets

For many commands in define you will have to specify a set of atoms on which that command shall act. There are three ways to do that:

- You may enter all or none, the meaning of which should be clear (entering none makes not much sense in most cases, however).
- You may specify a list of atomic indices like 1, 3, 5, 6 or 2, 4-6, 7, 8-10 or similar.
- You may also enter atomic identifiers which means strings of at most eight characters: the first two contain the element symbol and the remaining six could be used to distinguish different atoms of the same type. For example, if you have several carbon atoms in your molecule, you could label some c ring and others c chain to distinguish them. Whenever you want to enter an atomic identifier, you have to put it in double quotation marks: "c ring".

You should take into account that define also creates, from the atoms you entered, all others according to symmetry. If necessary, you will therefore have to lower the (formal) symmetry before executing a command.

4.0.5 control as Input and Output File

define may be used to update an existing control file, which is helpful if only the basis set has been changed. In this case just keep all data, i.e. reply with <enter> on
all questions, and only specify new start MOs. The more general usage is described now.

At the beginning of each define session, you will be asked to enter the name of the file to be created. As mentioned earlier, all TURBOMOLE programs require their input to be on a file named control, but it may be useful at this moment to choose another name for this file (e.g. if you have an old input file control and you do not want to overwrite it). Next you will be asked to enter the name of an old file which you want to use as input for this session. This prevents you from creating the new input from scratch if you want to make only minor changes to an old control file. It is possible to use the same file as input and output file during a define session (which means that it will only be modified). This may lead to difficulties, however, because define reads from the input file when entering each main menu and writes the corresponding data when leaving this menu. Therefore the input file may be in an ill-defined status for the next main menu (this will be the case, for example, if you add or change atoms in the first menu so that the basis set information is wrong in the second menu). define takes care of most—but not all—of these problems.

For these reasons, it is recommended to use a different filename for the input and the output file of the define session if you change the molecule to be investigated. In most cases involving only changes in the last three of the four main menus no problem should arise when using the same file as input and output.

4.0.6 Be Prepared

Atomic Coordinates

Molecules and their structures are specified by coordinates of its atoms, within the program invariably by Cartesian coordinates in atomic units (Ångström would also do). In TURBOMOLE these coordinates are contained in the file coord (see Section 22 “Sample control files” for an example).

Recommendation

We strongly recommend to create the coord file before calling define, only for small molecules one should use the interactive input feature of define. Set up the molecule by any program you like and write out coordinates in the xyz-format (XMo1 format), which is supported by most programs. Then use the TURBOMOLE tool x2t to convert it into a TURBOMOLE coord file (see Section 1.5).

Internal Coordinates

Structure optimizations, see jobex, are most efficient if carried out in internal coordinates and TURBOMOLE offers the following choices.

internals based on bond distances and angles, see Section 4.1.2.
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

redundant internals
defined as linearly independent combinations of internals (see ref. [27]), provided automatically by the command ired in the ‘geometry main menu’ in Section 4.1 below. This works in almost all cases and is efficient. The disadvantage is, that this is a black box procedure, the coordinates employed have no direct meaning and cannot be modified easily by the user.
cartesians
should always work but are inefficient (more cycles needed for convergence). Cartesians are the last resort if other options fail, they are assigned as default if one leaves the main geometry menu and no other internals have been defined.

4.1 The Geometry Main Menu

After some preliminaries providing the title etc. you reach the geometry main menu:

SPECIFICATION OF MOLECULAR GEOMETRY ( #ATOMS=0 SYMMETRY=c1 )
YOU MAY USE ONE OF THE FOLLOWING COMMANDS :
sy <group> <eps> : DEFINE MOLECULAR SYMMETRY (default for eps=3d-1)
desy <eps> : DETERMINE MOLECULAR SYMMETRY AND ADJUST
              COORDINATES (default for eps=1d-6)
susy : ADJUST COORDINATES FOR SUBGROUPS
ai : ADD ATOMIC COORDINATES INTERACTIVELY
a <file> : ADD ATOMIC COORDINATES FROM FILE <file>
aa <file> : ADD ATOMIC COORDINATES IN ANGSTROEM UNITS FROM FILE <file>
sub : SUBSTITUTE AN ATOM BY A GROUP OF ATOMS
i : INTERNAL COORDINATE MENU
ired : REDUNDANT INTERNAL COORDINATES
red_info : DISPLAY REDUNDANT INTERNAL COORDINATES
ff : UFF-FORCEFIELD CALCULATION
m : MANIPULATE GEOMETRY
frag : DEFINE FRAGMENTS FOR BSSE CALCULATION
w <file> : WRITE MOLECULAR COORDINATES TO FILE <file>
r <file> : RELOAD ATOMIC AND INTERNAL COORDINATES FROM FILE <file>
name : CHANGE ATOMIC IDENTIFIERS
del : DELETE ATOMS
dis : DISPLAY MOLECULAR GEOMETRY
banal : CARRY OUT BOND ANALYSIS
* : TERMINATE MOLECULAR GEOMETRY SPECIFICATION
    AND WRITE GEOMETRY DATA TO CONTROL FILE

IF YOU APPEND A QUESTION MARK TO ANY COMMAND AN EXPLANATION
OF THAT COMMAND MAY BE GIVEN

This menu allows you to build your molecule by defining the Cartesian coordinates interactively (ai) or by reading the coordinates from an external file (a, aa). The
structure can be manipulated by the commands sub, m, name and del. The command sy allows you to define the molecular symmetry while desy tries to determine automatically the symmetry group of a given molecule.

There exists a structure library which contains the Cartesian coordinates of selected molecules, e.g. CH$_4$. These data can be obtained by typing for example a ! ch4 or a ! methane. The data files are to be found in the directory $TURBODIR/structures$. The library can be extended.

You can perform a geometry optimization at a force field level to preoptimize the geometry. For this purpose the Universal Force Field (UFF) developed from Rappé et al. in 1992 [7] has been implemented in the uff module (see also Section 5.4). This can also be used to calculate an analytical approximate cartesian Hessian. If one does so, the start Hessian for the ab initio geometry optimization is this Hessian instead of the diagonal one ($forceinit on carthess$ for relax module).

**Recommendation**

Here is an easy way to get internal coordinates, which should work.

Have coord ready before calling define. In the main geometry menu proceed as follows to define redundant internals:

```
a coord   read coord

desy      determine symmetry, if you expect a higher symmetry, repeat with increased tolerance desy 0.1 , you may go up to desy 1..

ired      get redundant internals

*         quit main geometry menu
```

To define internals:

```
a coord   read coord

desy      determine symmetry

i         go to internal coordinate menu

iaut      automatic assignment of bends etc.

q         to quit bond analysis

imet      to get the metric, unnecessary internals are marked d now. If #ideg = #k in the head line you are done. Otherwise this did not work.

<enter>   go back to main geometry menu

*         quit main geometry menu
```

To define cartesians:
a coord  read coord

desy  determine symmetry

*  quit main geometry menu

4.1.1 Description of commands

Main Geometry Menu

In the headline of this menu you can see the current number of atoms and molecular symmetry (we use an input for PH\textsubscript{3} as example). The commands in this menu will now be described briefly:

\texttt{sy}  Definition of the Schönflies symbol of the molecular point group symmetry. If you enter only \texttt{sy}, \texttt{define} will ask you to enter the symbol, but you may also directly enter \texttt{sy c3v}. \texttt{define} will symmetrize the geometry according to the new Schönflies symbol and \textbf{will create new nuclei if necessary}. You therefore have to \textbf{take care} that you enter the correct symbol and \textbf{that your molecule is properly oriented}. All \texttt{TURBOMOLE} programs require the molecule to be in a standard orientation depending on its point group. For the groups \textit{C}_n, \textit{C}_{nv}, \textit{C}_{nh}, \textit{D}_n, \textit{D}_{nh} and \textit{D}_{nd} the \textit{z}-axis has to be the main rotational axis, secondary (twofold) rotational axis is always the \textit{x}-axis, \textit{\sigma}_v is always the \textit{zx}-plane and \textit{\sigma}_h the \textit{xy}-plane. \textit{O}_h is oriented as \textit{D}_{4h}. For \textit{T}_d, the threefold rotational axis points in direction (1,1,1) and the \textit{z}-axis is one of the twofold axes bisecting one vertex of the tetrahedron.

\texttt{desy}  \texttt{desy} allows you to determine the molecular symmetry automatically. The geometry does not need to be perfectly symmetric for this command to work. If there are small deviations from some point group symmetry (as they occur in experimentally determined structures), \texttt{desy} will recognize the higher symmetry and symmetrize the molecule properly. If symmetry is lower than expected, use a larger threshold: \texttt{<eps>} up to 1.0 is possible.

\texttt{susy}  \texttt{susy} leads you through the complete subgroup structure if you want to lower symmetry, e.g. to investigate Jahn–Teller distortions. The molecule is automatically reoriented if necessary. Example: \textit{T}_d \rightarrow \textit{D}_{2d} \rightarrow \textit{C}_{2v} \rightarrow \textit{C}_s.

\texttt{ai}  You may enter Cartesian atomic coordinates and atomic symbols interactively. After entering an atomic symbol, you will be asked for Cartesian coordinates for this type of atom until you enter *. If you enter & the atom counter will be decremented and you may re-define the last atom (but you surely won’t make mistakes, will you?). After entering *, \texttt{define} asks for the next atom type. Entering & here will allow you to re-define the last atom type and * to leave this mode and return to
the geometry main menu. Enter q as atom symbol if you want to use a
dummy center without nuclear charge. Symmetry equivalent atoms are
created immediately after you entered a set of coordinates.

This is a convenient tool to provide e.g. rings: exploit symmetry group
$D_{nh}$ to create an $n$-membered planar ring by putting an atom on the
x-axis.

**a file**

You may also read atomic coordinates (and possibly internal coordinates)
from file, where file must have the same format as the data group $\text{coord}$
in file control.

The Cartesian coordinates and the definitions of the internal coordinates
are read in free format; you only have to care for the keywords $\text{coord}$
and (optionally) $\text{intdef}$ and (important!) for the $\text{end}$ at the end of
the file. The atomic symbol follows the Cartesian coordinates separated
by (at least) one blank. For a description of the internal coordinate
definitions refer to 4.1.2.

Entering ‘!’ as first character of file will tell define to take file from the
structure library. (The name following the ‘!’ actually does not need to
be a filename in this case but rather a search string referenced in the
structure library contents file, see Section 4.1).

**aa file**

same as a, but assumes the atomic coordinates to be in Å rather than
a.u.

**sub**

This command allows you to replace one atom in your molecule by an-
other molecule. For example, if you have methane and you want to create
ethane, you could just substitute one hydrogen atom by another methane
molecule. The only requirement to be met by the substituted atom is
that it must have exactly one bond partner. The substituting molecule
must have an atom at the substituting site; in the example above it would
not be appropriate to use CH$_3$ instead of CH$_4$ for substitution. Upon
substitution, two atoms will be deleted and the two ones forming the
new bond will be put to a standard distance. define will then ask you
to specify a dihedral angle between the old and the new unit. It is also
possible to use a part of your molecule as substituting unit, e.g. if you
have some methyl groups in your molecule, you can create further ones
by substitution. Some attention is required for the specification of this
substituting unit, because you have to specify the atom which will be
deleted upon bond formation, too. If you enter the filename from which
the structure is to be read starting with ‘!’, the file will be taken from
the structure library (see Section 4.1). Definitions of internal coordinates
will be adjusted after substitution, but no new internal coordinates are
created.

**i**

This command offers a submenu which contains everything related to
internal coordinates. It is further described in Section 4.1.2.
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

m

This command offers a submenu which allows you to manipulate the molecular geometry, i.e. to move and rotate the molecule or parts of it. It is further described in Section 4.1.3.

frag

Here, the fragments will be defined as being used by the jobsse script in order to do a calculation using the counter-poise correction scheme. In this menu, up to three monomers can be defined, together with their charges and their symmetry. When assigning atom numbers to fragments, if x is entered instead of a number, the program will request the first and last atoms of a range. This will be useful for very large fragments.

w file

The command w writes your molecular geometry and your internal coordinates to file. Afterwards you will be back in the geometry main menu. If the filename entered starts with ‘!’, the structure will be written to the structure library.

name

name allows you to change atomic identifiers turning, e.g. oxygen atoms into sulfur atoms. After entering the identifier to be changed (remember the double quotation marks : "c ring"), you will be asked to enter the new one. You can use question marks for characters not to be changed, e.g. you enter "??ring" to change c chain to c ring. If you do not enter eight characters, your input will be filled up with trailing blanks.

del

The command del allows you to delete one or more atoms. After you entered the atomic list, define will show you a list of all atoms concerned and will ask you to confirm deleting these atoms. If any internal coordinate definitions exist, which rely on some of the deleted atoms, these definitions will be deleted, too.

banal

The command banal allows you to perform a bonding analysis, that is, define will try to decide which atoms are bonded and which are not (according to a table of standard bond lengths which is included in the code of define). You must have performed this command before you can use the display commands disb (display bonding information) or disa (display bond angle information). The standard bond lengths (and the bonding analysis available from these) are also needed for the commands sub and iaut (see internal coordinate menu, Section 4.1.2). If you want to change the standard bond lengths (or define more bond lengths, because not for all possible combinations of elements a standard length is available) you can do that by creating your own file with the non-default values and by specifying its full pathname in file .sys.data. The file has the following simple format:

```
c - h 2.2
h - h 2.0
...```

The format of the entries is almost arbitrary: the two element symbols have to be separated by a bar, the new bond distance follows in free format (in atomic units). If the file cannot be read properly, a warning message is displayed.

* This command leaves this first main menu and writes all data generated so far to file. The default output file is the file you choose in the first question during your define session (usually control). Now the data groups $coord$ and $intdef$ will be written to file. After leaving this menu, you will enter the atomic attributes menu, which is described in Section 4.2.

### 4.1.2 Internal Coordinate Menu

**INTERNAL COORDINATE MENU ( #ideg=6 #k=2 #f=0 #d=0 #i=0 )**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>imet &lt;a&gt;</td>
<td>PROVIDE B-MATRIX FOR ACTIVE INTERNAL COORDINATES (CHECK COMPLETENESS AND NUMERICAL QUALITY AND CHANGE REDUNDANT INTERNALS TO display)</td>
</tr>
<tr>
<td>idef</td>
<td>SUB-MENU FOR INTERACTIVE DEFINITION OF INTERNAL COORDINATES</td>
</tr>
<tr>
<td>ideg &lt;a&gt;</td>
<td>OUTPUT NUMBER OF TOT. SYMMETRIC INTERNAL DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>iaut</td>
<td>TRY AUTOMATIC DEFINITION OF INTERNAL COORDINATES</td>
</tr>
<tr>
<td>iman &lt;a&gt;</td>
<td>MANIPULATE GEOMETRY BY CHANGING INTERNAL COORDINATE VALUES</td>
</tr>
<tr>
<td>ic &lt;i&gt;</td>
<td>CHANGE STATUS OF INTERNAL COORDINATE &lt;i&gt; TO &lt;x&gt;</td>
</tr>
<tr>
<td>irem &lt;i&gt;</td>
<td>REMOVE INTERNAL COORDINATE &lt;i&gt;, e.g. irem d TO REMOVE ALL display COORDS</td>
</tr>
<tr>
<td>dis</td>
<td>ANY DISPLAY COMMAND e.g. disi OR disc</td>
</tr>
<tr>
<td>disiat &lt;i&gt;</td>
<td>AS disi BUT STARTING AT INTERNAL COORD. NUMBER i</td>
</tr>
</tbody>
</table>

WHERE  
<a>= OPTIONAL ATOMIC SET (DEFAULT=all)  
<i>= INDEX(LIST) OF INTERNAL COORDINATE(S) LIKE 3-6,8 OR <i>=<x>  
<x>= STATUS OF INTERNAL COORDINATE = k, f, d OR i

ADDING A QUESTION MARK TO ANY COMMAND MAY PROVIDE EXPLANATIONS

ENTER COMMAND OR HIT >return< TO GET BACK TO GEOMETRY MAIN MENU

The parameters in the headline of this menu have the following meanings:

- **#ideg** is the total number of symmetry restricted degrees of freedom.
- **#k** is the number of *active* internal coordinates specified up to now. Only these coordinates are optimized during a geometry optimization.
- **#f** is the number of *fixed* internal coordinates specified. These coordinates will be included in the B-matrix (see command imet), but their values will not be changed during geometry optimization.
#d is the number of internal coordinates whose values will only be displayed (e.g. by command disi), but no gradients will be calculated for these coordinates nor will they be included in the geometry optimization.

#i means the number of coordinates which are defined, but will be completely ignored, i.e. they are not even displayed on the screen and will not be used by any program (this is the waste-paper-basket of define).

Note that the #k plus #f must equal the number of degrees of freedom (#ideg) of your molecule, if you want to perform a geometry optimization. If you have less coordinates than degrees of freedom, you will have to specify further ones (commands idef or iaut, see below); if you have more coordinates than degrees of freedom, you will have to throw away some of them (commands irem or imet, see below).

The commands in this menu allow you to define internal coordinates for your molecule, adjust your geometry to special values of these internal coordinates and to control the numeric reliability of the chosen set of internal coordinates. In detail, the commands act as follows.

**Description of commands**

**imet a**  This command computes the so-called B-matrix, which is the matrix of the derivatives of the (active and fixed) internal coordinates with respect to Cartesian coordinates. This matrix is used in program relax for the conversion from Cartesian coordinates and gradients to internal ones and vice versa. If this matrix is singular (or even nearly singular) this indicates a linear dependency of your internal coordinate set. As a consequence the geometry update (more exactly the transformation of the updated internal coordinates into Cartesian ones) will fail. This may also happen in the course of a geometry optimization if the coordinates run into linear dependency during their optimization. imet checks the B-matrix and removes linear dependent internal coordinates from your list (their status is changed from #k or #f to #d). If B is only near singular, a warning is issued and the lowest eigenvalue(s) as well as the corresponding eigenvector(s) are displayed. In this case, you should try to find better internal coordinates (although this may not always be possible). After the command imet, there may be too few (active plus fixed) internal coordinates, but certainly not too many (because linear dependencies have been eliminated). Perhaps you will have to add new ones or — better! — try command iaut or irem in the preceding menu.

Command imet should be used always after creating internal coordinates with iaut or idef (especially after iaut, because this command creates usually an overcomplete set of internal coordinates).

**idef**  idef unfolds a little submenu where you can define internal coordinates manually. The exact procedure of the definition will be described below in a separate section.
4.1. THE GEOMETRY MAIN MENU

ideg a

This command gives you the number of symmetry restricted degrees of freedom (for the atomic set specified by a). Without symmetry, this is just $3N - 6$, where $N$ is the number of atoms, but if there is symmetry, some of these degrees of freedom will violate symmetry and therefore are not valid. For geometry optimizations, only the symmetry allowed degrees of freedom are needed, because the symmetry requirements are imposed anyway. In connection with the optional atomic set $a$ this command can help you to find out, in which part of a complicated molecule internal coordinates are missing, if you fail to get the full number of #ideg (which equals the result of ideg all) for the molecule as a whole.

iaut

iaut tries an automatic definition of internal coordinates. This command relies on a recursive procedure which tries to simplify the molecule as far as possible and then starts the definition of internal coordinates. At present not all molecular topologies are supported, therefore it may happen that no internal coordinates can be assigned to your molecule or at least a part of it. However, for all cases in which an automatic assignment of coordinates is possible, iaut has up to now proved to provide very good internal coordinates. If iaut works for your molecule (and in most non-pathological cases it will) we recommend strongly to use these coordinates, as they may help you to save several cycles in the geometry optimization procedure. After creating internal coordinates with iaut you should always use imet (see above), because iaut may provide an overcomplete set of coordinates. All coordinates which conflict with the molecular symmetry are set to ignore by iaut.

iman a

iman allows you to modify the values of internal coordinates. If you specify a list of atoms $a$ only those internal coordinates which refer to only these atoms will be handled. You will get a list of all (active and fixed) internal coordinates and their current values and you will be able to enter a new value for each of them if you like. Default (<enter>) keeps the value shown. Be aware that all distances are given in atomic units ($1 \text{ a.u.} = 52.9 \text{ pm}$).

ic ix

This option allows you to change the status of a coordinate, e.g. from active to display or every other combination. The syntax is ic 5 d, if coordinate no. 5 is to be set to display, or ic k d, if all active coordinates are to be set to display.

irem i

This option allows you to delete definitions of internal coordinates from your list. The indices of the internal coordinates always refer to the full list of coordinates including display and ignore coordinates. To make sure you delete the right ones, use disi. Also the indices will immediately change if you delete coordinates. If you want to delete several coordinates, this is therefore done most easily if you delete them in order of descending indices (because deletion of a coordinate has only an effect on the coordinates with higher indices). After choosing the coordinates
to be deleted, a list of all coordinates involved will be displayed and you will be asked to confirm deletion.

The syntax is simply irem 5 to delete internal coordinate no. 5 or irem d to remove all ‘display’ coordinates.

Hitting <return> will bring you back to the geometry main menu.

Interactive Definition of Internal Coordinates

If you choose idef in the internal coordinate menu, you will get the following information:

```
ENTER INTERNAL COORDINATE DEFINITION COMMAND
  <x> <type> <indices>
WHERE  <x>   = k  f  d  i
     <type> = stre invr bend outp tors linc linp
                 comp ring pyrm bipy pris cube octa
```

These commands will be explained in detail if you enter <x> <type>? for some choice of <x> and <type>, e.g. k stre ?

```
DEFAULT=GO BACK TO INTERNAL MAIN MENU DISPLAY=dis
```

The <x> means the status (see page 57) of the internal coordinate entered (k, f, d, i). The syntax is:

```
k stre 1 2
d tors 3 6 2 7
f bend 3 4 5
i outp 3 4 7 9
```

Note that in the third example atom 5 is the central atom of the angle!

Specification of available internal coordinates

The following types of coordinates are available:

- **stre**: The stre (for stretch) describes a distance between two atoms. It needs only two atomic indices to be given, the order of which is arbitrary.

- **invr**: The invr coordinate (for inverse r) describes an inverse distance. The declaration is the same as for stre, but in some cases (if you are far away from the minimum) the use of invr may result in better convergence.

- **bend**: bend describes a bond angle. It requires three atoms to be specified, of which the third one is the atom at the apex.

- **outp**: Out-of-plane angle: outp abcd is the angle between bond a − d and plane b − c − d.
Dihedral angle: *tors abcd* is the angle between the planes $a - b - c$ and $b - c - d$.

This is a special coordinate type to describe the bending of a near-linear system. *linc abcd* describes the collinear bending of $a - b - c$ (where the angle is defined as for *bend*: the apex atom appears last) in the plane of $b - c - d$ (see also below, command *linp*). The system $b - c - d$ has to be non-linear, of course.

This coordinate is similar to *linc*, but describes the bending of $a - b - c$ perpendicular to the plane $b - c - d$. These two types of coordinates are in most cases sufficient to describe the bending of near-linear systems. An example may help you to understand these two coordinate types. Consider ketene, $H_2CCO$, which contains a linear system of three atoms. Without symmetry, this molecule has 9 degrees of freedom. You could choose the four bond lengths, two CCH angles and the out-of-plane angle of the C–C bond out of the CHH–plane. But then two degrees of freedom still remain, which cannot be specified using these *normal* coordinate types. You can fix these by using *linc* and *linp*. The two coordinates *linc 1 3 2 4* and *linp 1 3 2 4* (where 1=oxygen, 2=carbon, 3=carbon, 4=hydrogen) would solve the problem.

The type *comp* describes a *compound* coordinate, i.e. a linear combination of (primitive) internal coordinates. This is often used to prevent strong coupling between (primitive) internal coordinates and to achieve better convergence of the geometry optimization. The use of linear combinations rather than primitive coordinates is especially recommended for rings and cages (see ref. [28]). Command *iaut* uses linear combinations in most cases.

After you entered *k comp n* where *n* is the number of primitive internal coordinates to be combined, you will be asked to enter the type of the coordinate (*stre, bend, . . .*). Then you will have to enter the weight (the coefficient of this primitive coordinate in the linear combination) and the atomic indices which define each coordinate. The definition of the primitive coordinates is the same as described above for the corresponding coordinate types. It is not possible to combine internal coordinates of different types.

This type helps you to define special ring coordinates. You only have to enter *k ring n* where *n* is the ring size. Then you will be asked for the atomic indices of all atoms which constitute the ring and which must be entered in the same order as they appear in the ring. The maximum number of atoms in the ring is 69 (but in most cases the ring size will be limited by the maximum number of atoms which is allowed for *define*).

Hitting <return> will bring you back to the internal coordinate menu where you can see the new number of internal coordinates in the headline.
4.1.3 Manipulating the Geometry

Note that the molecular geometry can be modified with the iman command of the internal coordinate menu described earlier, if internal coordinates has been defined. Another option is to select m in the geometry main menu which provides the following submenu:

CARTESIAN COORDINATE MANIPULATION MENU :
mov e : TRANSLATE AND/OR ROTATE PART OF THE MOLECULE
inert : MOVE MOLECULE SO THAT COORDINATE AXES BECOME PRINCIPAL AXES OF INERTIA
mback : RESTORE PREVIOUS MOLECULAR GEOMETRY
dis : DISPLAY MOLECULAR GEOMETRY
YOU MAY APPEND A QUESTION MARK TO ANY OF THESE COMMANDS FOR FURTHER EXPLANATIONS.
HIT >return< OR USE ANY GEOMETRY COMMAND NOT IN THIS LIST TO TERMINATE THIS MENU.
UPON TERMINATION THE MOLECULAR SYMMETRY WILL BE ENFORCED ACCORDING TO SYMMETRY GROUP c3v .

The meaning of the commands inert and mback should be clear; command move allows you to manipulate the geometry of your molecule. After entering move, you will be asked to specify a set of atoms on which the command shall act. You can use this to manipulate only a part of your molecule, e.g. if you are building a structure from subunits and you want to adjust their relative arrangement. As long as you stay in this menu, the molecular symmetry needs not be correct (so that you can try different movements and/or rotations), but as soon as you leave it, the geometry will be symmetrized according to the present Schönflies symbol. After you specified the atomic set to be considered, you get the following information:

INPUT DIRECTION OF MOVEMENT OR LOCATION OF ROTATION AXIS EITHER AS A COORDINATE TRIPLE SEPARATED BY BLANKS, OR AS TWO ATOMIC INDICES SEPARATED BY KOMMA, OR x OR y OR z OR ENTER ANY DISPLAY COMMAND FIRST OR & TO GO BACK

You can thus specify the direction of movement (or the rotational axis) in the form 0. 0. 1. or simply z (which both describes the z-axis) or 1.3256 -3.333 0.2218 for an arbitrary axis. If you want to specify an axis which is related to your molecule, you may also enter two atomic indices which define it. After having specified the axis, you have to enter the distance of movement and the angle of rotation. If you want to perform a simple rotation, enter 0 for the distance of movement and if you want to simply move your structure, enter 0 for the rotational angle.

You can leave this menu and return to the geometry main menu by hitting <return> or by entering any command of the geometry main menu.

4.2 The Atomic Attributes Menu

After you specified the molecular geometry and symmetry and wrote this data to file, you will encounter the atomic attributes menu, which is the second of the four
main menus. You will enter this menu, if all necessary data cannot be read from your input file or if you do not use an input file. This menu deals with the specification of basis sets and other data related to the atom type:

```
ATOMIC ATTRIBUTE DEFINITION MENU ( #atoms=5   #bas=5   #ecp=0   )

b : ASSIGN ATOMIC BASIS SETS
bb : b RESTRICTED TO BASIS SET LIBRARY
bl : LIST ATOMIC BASIS SETS ASSIGNED
bm : MODIFY DEFINITION OF ATOMIC BASIS SET
bp : SWITCH BETWEEN 5d/7f AND 6d/10f
lib : SELECT BASIS SET LIBRARY
ecp : ASSIGN EFFECTIVE CORE POTENTIALS
ecpb : ecp RESTRICTED TO BASIS SET LIBRARY
ecpi : GENERAL INFORMATION ABOUT EFFECTIVE CORE POTENTIALS
ecpl : LIST EFFECTIVE CORE POTENTIALS ASSIGNED
ecprm: REMOVE EFFECTIVE CORE POTENTIAL(S)
ecm : ASSIGN NUCLEAR CHARGES (IF DIFFERENT FROM DEFAULTS)
cem : ASSIGN NUCLEAR CHARGES FOR EMBEDDING
m : ASSIGN ATOMIC MASSES (IF DIFFERENT FROM DEFAULTS)
dis : DISPLAY MOLECULAR GEOMETRY
dat : DISPLAY ATOMIC ATTRIBUTES YET ESTABLISHED
h : EXPLANATION OF ATTRIBUTE DEFINITION SYNTAX
* : TERMINATE THIS SECTION AND WRITE DATA OR DATA REFERENCES TO control
GOBACK=& (TO GEOMETRY MENU !)
```

The headline gives you the number of atoms, the number of atoms to which basis sets have already been assigned and the number of atoms to which effective core potentials have already been assigned. Most of the commands in this menu deal with the specification of basis sets and pseudopotentials.

**Basis sets available**

The following basis sets are available on \$TURBODIR/basen/, which you may inspect to see which other basis sets are supported automatically. The corresponding publications can be found here 1.3.

- **SV(P) or def-SV(P)** for routine SCF or DFT. Quality is about 6–31G*.
- **TZVP or def-TZVP** for accurate SCF or DFT. Quality is slightly better than 6–311G**.
- **TZVPP or def-TZVPP** for MP2 or close to basis set limit SCF or DFT. Comparable to 6–311G(2df).
- **QZVP and QZVPP** for highly correlated treatments; quadruple zeta + 3d2f1g or 4d2f1g (beyond Ne), 3p2d1f for H.

These basis sets are available for atoms H–Kr, and the split-valence (SV) and valence-triple-ζ (TZV) basis sets types with ECPs also for Rb–Rn, except lanthanides.
For calculations with the programs ricc2, ccddf12, and pnoccsd optimized auxiliary basis sets are available for the basis sets SV(P), SVP, TZVP, TZVPP, and QZVPP.

**NEW**: New sets of basis functions, partly identical with those mention above, denoted def2-XYZ are available for atoms H–Rn \[6\]. The def2 basis sets for 5p and 6p block elements are designed for small core ECPs (ECP-28, ECP-46 and ECP-60). For each family, SV, TZV, and QZV, we offer two sets of polarization functions leading to:

- def2-SV(P) and def2-SVP
- def2-TZVP and def2-TZVPP
- def2-QZVP and def2-QZVPP

We strongly recommended the new def2-basis, since they have been shown to provide consistent accuracy across the periodic table.

**Recommendation**

Use the same basis set type for all atoms; use ECPs beyond Kr since this accounts for scalar relativistic effects.

**New basis sets** (def2-XYZ): MP2 implies RI-MP2 and RICC2

- exploratory MP2: SVP
- almost quantitative DFT: SV(P), HF: SVP, MP2: TZVPP; properties (HF and DFT): TZVPP
- quantitative DFT: TZVP, HF: TZVPP, MP2: QZVPP
- basis set limit DFT: QZVP, HF: QZVP

If you want a better basis than SV(P), assigned automatically, use **all def2-TZVP** (or another basis). The assignment can be checked by **b1**.

Diffuse functions should only be added if really necessary. E.g. for small anions or treatment of excited states use: **TZVP** instead of **SVP + diffuse**. This is more accurate and usually faster. Only for excited states of small molecules or excited states with (a partial) Rydberg character add additional diffuse functions (e.g. by using the aug-cc-pVTZ basis) as well as the keyword **diffuse**, for more information, see page 298 in the keyword section.

**Old basis sets** (def-XYZ): For standard correlated calculations (MP2, RI-MP2, RI-CC2) use the doubly-polarized TZVPP (or def-TZVPP) basis.

**Correlation-Consistent (Dunning) Basis Sets**

Dunning basis sets like cc-pVDZ, cc-pVTZ, cc-pVQZ are also supported, e.g. by **all cc-pVTZ**. But these basis sets employ generalized contractions for which TURBOMOLE
is not optimized. This has in particular strong effects on the performance of all programs which use 4-index electron repulsion integrals, for RI-MP2 and RI-CC2 this is partially compensated by the RI-approximation.

The following correlation consistent basis sets are available in the TURBOMOLE basis set library:

- **cc-pVXZ** standard valence X-tuple zeta basis sets \((X = D, T, Q, 5, 6)\); available for H, He, Li–Ne, Na–Ar, K, Ca, Ga–Kr.
  - (cc-pV6Z only for H, He, B–Ne, Al–Ar; for Al–Ar also the recommended newer cc-pV(X+d)Z sets are available)

- **cc-pwCVXZ-PP** weighted core-valence x-tuple zeta basis sets \((X = D, T, Q, 5)\) are available for post-d main group elements Ga–Kr, In–Xe, and Tl–Rn.
  - (also pure valence basis sets cc-pVXZ-PP are available for these elements, but it is not recommended to use them)

- **cc-pwCVXZ** weighted core-valence X-tuple zeta basis sets \((X = D, T, Q, 5)\); available for H, He, B–Ne, Al–Ar, and Ga–Kr
  - (for Al–Ar also the recommended combination of the cc-pV(X+d)Z sets with the core valence functions \((wC)\), i.e. the cc-pwCV(X+d)Z basis set are available)

- **aug-** diffuse functions for combination with the basis sets cc-pVXZ, cc-pV(X+d)Z, cc-pwCVXZ, cc-pV(X+d)Z, cc-pVXZ-PP or cc-pwCVXZ-PP; available for H, He, B–Ne, Al–Ar with \(X = D–6\) and Ga–Kr, In–Xe, and Tl–Rn with \(X = D–5\).

- **cc-pVXZ-F12** with \(X = D, T, Q\) for use with the explicitly-correlated F12 variants of wavefunction methods (MP2-F12, CCSD(F12*), etc.)

For calculations with the programs that employ the RI approximation with a correlated wavefunction optimized auxiliary basis sets are available for most of the correlation consistent basis set series.

### 4.2.1 Description of the commands

- **b** With **b** you can specify basis sets for all atoms in your molecule. After entering **b** you will be asked to specify the atoms to which you want to assign basis sets. You can do this in the usual ways (refer to Section 4.0.4), including **all** and **none**. Then you will be asked to enter the **nickname** of the basis set to be assigned. There are two principal ways to do this:

  1) If you are in the **append** mode, the nickname you entered will be appended to the atomic symbol of the element under consideration.
     - This is especially useful if you want to assign basis sets to different atoms with one command. For example, if you want to assign basis
sets to hydrogen and oxygen atoms and you enter only DZ, the basis sets h DZ and o DZ will be read from the basis set library.

2) If you are in the non-append mode, no atomic symbol will be inserted in front of the nickname entered. Therefore you have to enter the full basis set nickname, e.g. h DZ. This mode is advantageous if you want to assign basis sets to dummy centers (i.e. points without nuclear charge but with basis functions, e.g. for counterpoise calculations) or if you want to use the basis set nickname none (which means no basis functions at this atom).

You can switch between the two modes with ‘+’ (switches to append mode) and ‘-’ (switches to non-append mode).

Once you have specified your basis set nickname, define will look in the standard input file (normally control) for this basis set. If it can not be found there, you can switch to the standard basis set library (if you did not use a standard input file, the standard library will be searched immediately). If the basis set cannot be found there, you are asked either to enter a new standard library (which will be standard only until you leave this menu) or another input file, where the basis set can be found. If you do not know the exact nickname of your basis set, you may abbreviate it by ‘?’, so you could enter h DZ? to obtain basis sets like h DZ, h DZP, h DZ special, etc. define will give you a list of all basis sets whose nicknames match your search string and allows you to choose among them. You may also use the command list to obtain a list of all basis sets cataloged.

bb bb does essentially the same as b but does not search your default input file for basis sets. Instead it will look in the basis set library immediately.

bl bl gives you a list of all basis sets assigned so far.

bm This command is used to modify basis sets which are already assigned. The corresponding submenu is self-explanatory, but we recommend to change directly the file basis.

bp The TURBOMOLE programs normally work with basis sets of 5d-functions (which means they delete the s-component of the full 6d-set). bp allows to switch between the proper 5d/7f-set and the Cartesian 6d/10f-set.

ecp This command allows you to specify effective core potentials for some atoms. The assignment works exactly like the specification of basis sets (see above).

ecpb This one does the same as command ecp, but restricted to the basis set library (the input file will not be used).

ecpi ecpi gives you some general information about what type of pseudopotentials is supported. For more information we refer to [29] and references therein.
4.3 Generating MO Start Vectors

4.3.1 The MO Start Vectors Menu

This menu serves to define the occupation numbers, and to generate the start vectors for HF-SCF and DFT calculations. They may be constructed from earlier SCF calculations (perhaps employing another basis set, type use), by Hamilton core guess (hcore), or by an extended Hückel calculation which can be performed automatically (eht). An occupation of the start orbitals will be proposed and can be modified if desired.

OCCUPATION NUMBER & MOLECULAR ORBITAL DEFINITION MENU

CHOOSE COMMAND

infsao : OUTPUT SAO INFORMATION
atb : Switch for writing MOs in ASCII or binary format
eht : PROVIDE MOS & OCCUPATION NUMBERS FROM EXTENDED HUECKEL GUESS
use <file> : SUPPLY MO INFORMATION USING DATA FROM <file>
man : MANUAL SPECIFICATION OF OCCUPATION NUMBERS
hcore : HAMILTON CORE GUESS FOR MOs
flip : FLIP SPIN OF A SELECTED ATOM
& : MOVE BACK TO THE ATOMIC ATTRIBUTES MENU
Recommendation

You will normally only need to enter `eht`. For the EHT-guess, `define` will ask for some specifications, and you should always choose the default, i.e. just `<enter>`. Of importance is only the molecular charge, specified as 0 (neutral, default), 1 or -1 etc.

Based on the EHT orbital energies `define` proposes an occupation. If you accept you are done, if not you get the “occupation number assignment menu” explained in 4.3.2.

Description of Commands

`infsao` Command `infsao` provides information about the symmetry adapted basis which is used for the SCF-calculation. To exploit the molecular symmetry as efficiently as possible, TURBOMOLE programs do not use the simple basis which you specified during your `define` session. Instead it builds linear combinations of equal basis functions on different—but symmetry equivalent—atoms. This basis is then called the SAO-basis (Symmetry Adapted Orbital). It has the useful property that each basis function transformed to this basis transforms belongs to one irreducible representation of the molecular point group (that is, the basis reflects the full molecular symmetry as specified by the Schönflies symbol). `infsao` gives you a listing of all symmetry adapted basis functions and their constituents either on file or on the screen. This may help you if you want to have a closer look at the SCF vectors, because the vector which is output by program `dscf` is written in terms of these SAOs.

`atb` Molecular orbitals can be written either in ASCII or in binary format. This command switches from one option to the other, and it is highly recommended to read which setting is currently active. ASCII format is portable and allows the usage of TURBOMOLE input files on different systems with incompatible binary format. Binary format is faster and smaller files will be written. The external program `atbandbta` can be used to transform existing `mos`, `alpha`, and `beta` files from ASCII to binary format and vice versa.

`eht` `eht` performs an extended Hückel calculation for your molecule. The orbital energies available from this calculation are then used to provide occupation numbers for your calculation and the Hückel MOs will be projected onto the space that is spanned by your basis set. This start-vectors are not as good as the ones which may be obtained by projection of an old SCF vector, but they are still better than the core Hamiltonian guess that is used if no start vectors are available. When using this
4.3. GENERATING MO START VECTORS

command, you will be asked if you want to accept the standard Hückel parameters and to enter the molecular charge. Afterwards you will normally get a list of the few highest occupied and lowest unoccupied MOs, their energies and their default occupation. If you don’t want to accept the default occupation you will enter the occupation number assignment menu, which is described in Section 4.3.2. Note that the occupation based on the Hückel calculation may be unreliable if the difference of the energies of the HOMO and the LUMO is less than 0.05 a.u. (you will get a warning). You will also have to enter this menu for all open-shell cases other than doublets.

**use file**

With command **use** you are able to use information about occupied MOs and start vectors from a former calculation on the same molecule. *file* should be the path and name of the *control* file of this former calculation, of which all data groups related to occupation numbers and vectors will be read. As the new generated data will overwrite the existing data if both resist in the same directory, it is best and in some cases necessary to have the data of the former calculation in another directory than the one you started the *define* session in. Then just type **use <path>/control** to construct a new SCF vector from the data of the old calculation, without changing the old data. The data groups `$closed shells` and `$open shells` will be taken for your new calculation and the SCF vector from the old calculation will be projected onto the space which is spanned by your present basis set. These start vectors are usually better than the ones you could obtain by an extended Hückel calculation.

**man**

**man** allows you to declare occupation numbers or change a previous declaration manually. After selecting this command, you will get a short information about the current occupation numbers:

```
---------------------------------------------
actual closed shell orbital selection    range
---------------------------------------------
  a1      #    1-  18
  a2      #    1-   1
  e       #    1-  13
---------------------------------------------

any further closed-shell orbitals to declare ? DEFAULT(y)
```

If you answer this question with *y*, you enter the orbital specification menu which will be described in Section 4.3.3.

The same procedure applies to the open-shell occupation numbers after you finished the closed-shell occupations.

**hcore**

**hcore** tells programs *dscf* and *ridft* to run without a start vector (it writes the data group `$scfmo none` to file *control*). *dscf* or *ridft* will then start from the core Hamiltonian start vector, which is the vector
obtained by diagonalizing the one-electron Hamiltonian. Note that you still have to specify the occupation numbers. This concerns only the first SCF run, however, as for the following calculations the converged vector of the previous iteration will be taken. A SCF calculation with a core Hamiltonian start vector typically will take $2 - 3$ iterations more than a calculation with an extended Hückel start vector (a calculation with the converged SCF vector of a previous calculation will need even less iterations, depending on how large the difference in the geometry between the two calculations is).

**flip**

flipping of spins at a selected atom. Requirements: converged UHF molecular orbitals and no symmetry ($C_1$). **define** will localize the orbitals, assign them to the atoms and give the user the possibility to choose atoms at which alpha-orbitals are moved to beta orbitals, or vice versa. This is useful for spin-broken start orbitals, but not for spatial symmetry breaking.

* This command (as well as **use** and **eht**) terminates this menu, but without providing a start vector. If the keyword **$scfmo** exists in your input file, it will be kept unchanged (i.e. the old vector will be taken), otherwise **$scfmo none** will be inserted into your output file, which forces a calculation without start vector to be performed. When you leave this menu, the data groups **$closed shells**, **$open shells** (optionally) and **$scfmo** will be written to file. You will then reach the last of the four main menus (the General Menu) which is described in Section 4.4.

### 4.3.2 Assignment of Occupation Numbers

If an automatic assignment of occupation numbers is not possible or you do not except the occupation numbers generated by the EHT, you enter the following menu:

**OCCUPATION NUMBER ASSIGNMENT MENU ( #e=60 #c=0 #o=0)**

- **s**: CHOOSE UHF SINGLET OCCUPATION
- **t**: CHOOSE UHF TRIPLET OCCUPATION
- **u <int>**: CHOOSE UHF WITH <int> UNPAIRED ELECTRONS
- **l <list>**: PRINT MO'S FROM EHT IN <list>, (DEFAULT=ALL)
- **p <index>**: PRINT MO-COEFFICIENTS OF SHELL <index>
- **c <list>**: CHOOSE SHELLS IN <list> TO BECOME CLOSED SHELLS
- **o <index>**: CHOOSE SHELL <index> TO BECOME AN RHF OPEN SHELL
- **a <list>**: CHOOSE SHELLS IN <list> TO BECOME UHF ALPHA SHELLS
- **b <list>**: CHOOSE SHELLS IN <list> TO BECOME UHF BETA SHELLS
- **v <list>**: CHOOSE SHELLS IN <list> TO BECOME EMPTY SHELLS
- **&**: REPEAT THE EXTENDED HUECKEL CALCULATION
- *****: SAVE OCCUPATION NUMBERS & GO TO NEXT ITEM
- **dis**: GEOMETRY DISPLAY COMMANDS
- **e**: CALCULATE EHT-ENERGY
- **f**: FURTHER ADVICE
4.3. GENERATING MO START VECTORS

<int> = INTEGER
<index> = INDEX OF MO-SHELL ACCORDING TO COMMAND s
<list> = LIST OF MO-SHELL INDICES (LIKE 1-5,7-8,11)

Recommendation

Enter 1 to get a list of all MO energies. Then make up your mind on what to do: closed shell, RHF open shell (not allowed for DFT) or UHF. Look at the examples below.

RHF c 1-41,43,45 to define these levels to be doubly occupied.

UHF a 1-5 alpha levels to be occupied, b 1-3,5 beta levels to be occupied. Or simply, s, t, or u 1 to get singlet, triplet or doublet occupation pattern.

ROHF c 1-41,43,45 levels to be doubly occupied; o 42 level 42 should be partially occupied. You will then be asked to specify the occupation. If there are more open shells you have to repeat, since only a single open shell can be specified at a time. Watch the headline of the menu, which tells you the number of electrons assigned to MOs.

Description of Commands

s list This command gives you a listing of all MOs and their energies as obtained from the extended Hückel calculation. For NH$_3$ in $C_3v$ and TZVP you get, e.g.:

<table>
<thead>
<tr>
<th>ORBITAL SYMMETRY (SHELL)</th>
<th>TYPE</th>
<th>ENERGY</th>
<th>SHELL DEGENERACY</th>
<th>SHELL DEG.</th>
<th>CUMULATED CL.SHL OCC.</th>
<th>OP.SHL OCC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1a1</td>
<td>-15.63244</td>
<td>2</td>
<td>2</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>2a1</td>
<td>-0.99808</td>
<td>2</td>
<td>4</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>1e</td>
<td>-0.64406</td>
<td>4</td>
<td>8</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>4</td>
<td>3a1</td>
<td>-0.57085</td>
<td>2</td>
<td>10</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>2e</td>
<td>0.30375</td>
<td>4</td>
<td>14</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>6</td>
<td>4a1</td>
<td>0.87046</td>
<td>2</td>
<td>16</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

TO CONTINUE, ENTER <return>

p index This allows you to get the linear combination of basis functions which form the MO-index. Note that this refers not to the basis set you specified, but to the extended Hückel basis. index must be a single index, not an index list.

c list This command allows you to specify closed shells. Their occupation will be 2 per MO, the total occupation the shell degeneracy which you can obtain by using command s. list is a list of shell indices like 1-13 or 1,3-5,7.
o index    This command allows you to specify open shells. index must be a single shell index, not an index list. You will then be asked for the number of electrons per MO which shall be contained in this shell. For example, for a fluorine atom you should choose o n (where n is the index of the p-shell) and an occupation of 5/3 per MO. You may enter the occupation numbers as simple integers or as integer fractions, e.g. 1 for the s-occupation in sodium, 5/3 for the p-occupation in fluorine.

v list    With this command you can remove an orbital occupation, if you specified a wrong one. list is again a list of shell indices in usual syntax.

&    This command has a different meaning in this menu than in the rest of define. Here it will repeat the extended Hückel calculation (perhaps you want to change some Hückel parameters for the next one).

*    * will not bring you back to the occupation numbers menu, but will terminate the whole occupation number and start vector section and will bring you to the last main menu, which is described in Section 4.4. If you want to leave this menu without assigning all electrons in your molecule to shells, define will issue a warning and suggest to continue defining occupation numbers. You can ignore this warning, if you do not want to assign all electrons.

e    Calculates and displays the extended Hückel total energy of your molecule.

f    f will give you some information about the commands in this menu.

You may overwrite occupation numbers once given by just redefining the corresponding shell. For example, if you choose shells 1–10 as closed shells and afterwards shell no. 9 as open shell (with any occupation number), the open shell will be correctly assigned.

4.3.3 Orbital Specification Menu

define provides the possibility to assign the occupation numbers of the MOs manually, if you like. To do that, use the command man in the occupation number main menu and you will arrive at the following submenu:

------------- ORBITAL SPECIFICATION MENU -------------

<label> <list> : select orbitals within <list>
-<label> <list> : skip orbitals within <list>
& : ignore input for last label
clear : clear all assignments
p(print) : print actual orbital selection
for help, type ? or help // for quit, type * or q(uit)

Depending on whether you are in the closed- or in the open-shell section, the commands of this menu refer only to the corresponding type of orbitals. The commands
4.3. GENERATING MO START VECTORS

of this menu do not need much explanation. <label> is the irrep label of one irreducible representation of the molecular point group (e.g. a1, b2, t1g, ...). <list> is a list of orbital indices within this irrep (e.g. 1, 2, 4 or 1-8, 10, 11). p or print will give you the same listing of the orbital occupations as you saw before entering this menu. After you leave this submenu, you will be back in the occupation numbers main menu.

4.3.4 Roothaan Parameters

In open-shell calculations within the restricted Hartree–Fock ansatz (ROHF), the coupling between the closed and the open shells must be specified using two parameters a and b, which depend on the type of the open shell, the number of electrons in it (the electron configuration), but also on the state to be calculated. For example, there are three states arising from the \( s^2p^2 \) configuration of an atom (\(^3\)P, \(^1\)D, \(^1\)S) which have different values of a and b. For the definition of these parameters and their use refer to Roothaan's original paper [30]. For simple cases, define sets these parameters automatically. If not, you have to enter them yourself. In this case, you will get the following message:

ROOTHAAN PARAMETERS a AND b COULD NOT BE PROVIDED ...
TYPE IN ROOTHAAN a AND b AS INTEGER FRACTIONS
OR ENTER val FOR AN AVERAGE OF STATES CALCULATION
OR ENTER & TO REPEAT OCCUPATION NUMBER ASSIGNMENT

Note that not all open shell systems can be handled in this way. It is possible to specify a and b for atomic calculations with \( s^n, p^n, d^1 \), and \( d^9 \) configurations and for calculations on linear molecules with \( \pi^n \) and \( \delta^n \) configurations. Furthermore, it is possible to do calculations on systems with half-filled shells (where \( a=1, b=2 \)). In the literature you may find tabulated values for individual states arising from \( d^n \) configurations, but these are not correct. Instead, these are parameters for an average of all states arising from these configurations. You can obtain these values if you enter val on the above question. For a detailed description see Section 6.3.

4.3.5 Start-MOs for broken symmetry treatments ("flip")

Broken-symmetry treatments suggested by e.g. Noodleman or Ruiz are a popular tool for the calculation of spin coupling parameters in the framework of DFT. As an example one might consider two coupled Cu\(^{II} \) centers, e.g. for a (hypothetical) arrangement like this:

```
$coord
  0.0  2.7  0.0  cu
  0.0 -2.7  0.0  cu
  0.0 -6.1  0.0  f
  0.0  6.1  0.0  f
  2.4  0.0  0.0  f
-2.4  0.0  0.0  f
$end
```
Chapter 4. Preparing Your Input File with Define

The high-spin case, a doublet with an excess alpha electron at each Cu atom, "aa" in an obvious notation, preserves $D_{2h}$ symmetry, while the low spin state "ba" does not. For a broken-symmetry treatment it is advisable to calculate the high-spin state first, and then broken-symmetry state(s); from the energy difference(s) one may calculate approximate values for the spin-spin coupling parameters as described by e.g. the above authors. Access to broken-symmetry states usually is possible by the choice of appropriate start-MOs, followed by an SCF-procedure. Start MOs may be obtained by first applying a localization procedure to the MOs of the high-spin state and then by "moving" localized alpha orbitals to the beta subset.

The preparation of broken-symmetry start-MOs can be done with define (semi-)automatically. Prerequisite is a converged wave function for the high-spin state in $C_1$-symmetry, that fulfills the aufbau principle.

If in this case one enters flip in the orbital definition menu, define selects the occupied valence orbitals of the system (by an orbital energy criterion, which one can usually accept, unless the system is highly charged and the orbital energies are shifted). Next a Boys orbital localization procedure is carried out, which - depending on the size of the problem - may take some time. Then the user is asked:

**Enter Indices of Atoms or Element to Be Manipulated (example: 1,2-3 or "mn")**

In case of our above example one may enter "cu", which immediately leads to the following output (a def-SV(P) basis and the B-P functional were used for the high-spin state):

<table>
<thead>
<tr>
<th>Relevant LMos for Atom</th>
<th>1 Cu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alpha:</strong></td>
<td></td>
</tr>
<tr>
<td>index</td>
<td>occupation</td>
</tr>
<tr>
<td>15</td>
<td>1.000</td>
</tr>
<tr>
<td>18</td>
<td>1.000</td>
</tr>
<tr>
<td>20</td>
<td>1.000</td>
</tr>
<tr>
<td>22</td>
<td>1.000</td>
</tr>
<tr>
<td>23</td>
<td>1.000</td>
</tr>
<tr>
<td><strong>Beta:</strong></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>1.000</td>
</tr>
<tr>
<td>41</td>
<td>1.000</td>
</tr>
<tr>
<td>43</td>
<td>1.000</td>
</tr>
<tr>
<td>46</td>
<td>1.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relevant LMos for Atom</th>
<th>2 Cu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alpha:</strong></td>
<td></td>
</tr>
<tr>
<td>index</td>
<td>occupation</td>
</tr>
<tr>
<td>16</td>
<td>1.000</td>
</tr>
<tr>
<td>17</td>
<td>1.000</td>
</tr>
<tr>
<td>19</td>
<td>1.000</td>
</tr>
<tr>
<td>21</td>
<td>1.000</td>
</tr>
<tr>
<td>24</td>
<td>1.000</td>
</tr>
</tbody>
</table>
4.3. GENERATING MO START VECTORS

BETA:

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>1.000</td>
<td>-0.326</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.33</td>
<td>0.08</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td>42</td>
<td>1.000</td>
<td>-0.326</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.33</td>
<td>0.08</td>
<td>0.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td>44</td>
<td>1.000</td>
<td>-0.321</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>45</td>
<td>1.001</td>
<td>-0.318</td>
<td>0.05</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
<td>0.43</td>
<td>0.51</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

\(a_{2b} : \) FLIPPING ALPHA TO BETA (default)

\(b_{2a} : \) FLIPPING BETA TO ALPHA

\(r : \) repeat atom choice

As evident from the second column, for each Cu atom five localized alpha and four localized beta orbitals were generated which are of d-type (the sixth column labelled "d" shows values close to 1, the other columns such close to 0). The six columns at the right show the individual contributions of the six cartesian d-functions.

What has to be done to generate start MOs for the "ba"-case? Obviously one of the five localized alpha spin orbitals from the first Cu atom (atom label 1 cu) has to become a beta spin orbital. These five orbitals have the indices 15, 18, 20, 22, 23. In order to avoid linear dependencies, it is advisable to take the orbital that has no beta-analogue. This can be found by comparing the contributions of the six d-functions. In the present example this is the case for the localized alpha orbitals 15 and 18: in contrast to all localized beta orbitals they show significant contributions from \(d_{xy} \). One thus enters \(a_{2b} 15\)

and after confirming the replacement of original MOs with the generated start-MOs one is finally asked

It is advisable to modify damping and orbital shift in the following way:

\$scfdamp start=5.000 step=0.050 min=0.500

\$scforbitalshift automatic=1.0

\$scfiterlimit 999

Do you want to replace the corresponding entries in the control-file? (y)

which should be confirmed, as otherwise the prepared spin state might be destroyed during the SCF iterations.

From this input one may start the SCF(HF/DFT)-procedure. For recommended choices of DFT functionals and formulae to calculate the coupling parameters from these energy differences please consult the papers of the above-mentioned authors. For reasons of economy, a pre-optimization by a pure (non-hybrid) DFT-functional is reasonable.

Important: For the converged wave function one should check, whether the resulting state is really the desired one. This can quite reliably be done by a Mulliken population analysis. For this purpose, add \$pop to the control file, type \texttt{ridft -proper}
or `dscf -proper`, respectively, and check the signs of the calculated numbers of unpaired electrons in the output.

### 4.4 The General Options Menu

After you specified all data concerning the molecule you want to examine, you are on your way to the last of the four main menus. Before reaching it, you will perhaps get a message like the following:

**DO YOU WANT TO DELETE DATA GROUPS LIKE**

- $energy
- $grad
- $hessian
- $hessian (projected)
- $last energy change
- $maximum norm of internal gradient
- $dipgrad
- $vibrational normal modes
- $vibrational spectrum
- $cartesianforce interspace

**LEFT OVER FROM PREVIOUS CALCULATIONS ? DEFAULT(n)**

`define` has scanned your input file for this session and found some data groups which might have become obsolete. If they are still acceptable depends on the changes you made during your present `define` session. They are obviously incorrect if you changed the molecule under consideration; but any change in the basis sets or the occupation numbers will make them dangerous, too, because you might not know some day if they really refer to the basis set which is defined in this control file. As a rough guide, delete them whenever you have made changes in one of the first three main menus during your `define` session.

After that you will reach the last main menu of `define` which helps you to control the actions of all TURBOMOLE programs. The meanings of the various options are explained in more detail in the description of the individual programs, therefore only a short explanation will be given here.

Now have a look at the menu:

**GENERAL MENU : SELECT YOUR TOPIC**

- scf : SELECT NON-DEFAULT SCF PARAMETER
- mp2 : OPTIONS AND DATA GROUPS FOR rimp2 and mpgrad
- pnocc : OPTIONS AND DATA GROUPS FOR pnoccsd
- cc : OPTIONS AND DATA GROUPS FOR ricc2
- ex : EXCITED STATE AND RESPONSE OPTIONS
- prop : SELECT TOOLS FOR SCF-ORBITAL ANALYSIS
- drv : SELECT NON-DEFAULT INPUT PARAMETER FOR EVALUATION OF ANALYTICAL ENERGY DERIVATIVES (GRADIENTS, FORCE CONSTANTS)
- rex : SELECT OPTIONS FOR GEOMETRY UPDATES USING RELAX
- stp : SELECT NON-DEFAULT STRUCTURE OPTIMIZATION PARAMETER
4.4. THE GENERAL OPTIONS MENU

This menu serves very different purposes. The next subsection deals with commands required to activate and/or specify specific methods of calculation. The subsequent subsection describes commands used to select non-default options. Standard SCF calculations do not require special action, just leave the menu. The final subsection describes the settings for property calculations.

4.4.1 Important commands

DFT calculations

Command \texttt{dft} leads you to the menu:

\begin{verbatim}
STATUS OF DFT_OPTIONS:
DFT is NOT used
  functional b-p
  gridsize m3

ENTER DFT-OPTION TO BE MODIFIED

func: TO CHANGE TYPE OF FUNCTIONAL
grid: TO CHANGE GRIDSIZE
on: TO SWITCH ON DFT

Just <ENTER>, q or '*' terminate this menu.
\end{verbatim}

To activate DFT input \texttt{on} and then specify the grid for the quadrature of exchange-correlation terms. \textsc{Turbomole} offers grids 1 (coarse) to 7 (finest), and the multiple grids \texttt{m3} to \texttt{m5} \cite{4}. The latter employ a coarser grid during SCF iterations, and grid 3 to grid 5 in the final SCF iteration and the gradient evaluation. Default is grid \texttt{m3}, for clusters with more than 50 atoms use \texttt{m4}.

The functionals supported are obtained with the command \texttt{func}:
### Survey of Available Exchange-Correlation Energy Functionals

<table>
<thead>
<tr>
<th>Functional</th>
<th>Type</th>
<th>Exchange</th>
<th>Correlation</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slater-Dirac exchange</td>
<td>LDA</td>
<td>S</td>
<td></td>
<td>1,2</td>
</tr>
<tr>
<td>s-vwn</td>
<td>LDA</td>
<td>S</td>
<td>VWN(V)</td>
<td>1-3</td>
</tr>
<tr>
<td>vwn</td>
<td>LDA</td>
<td></td>
<td>VWN(V)</td>
<td>3</td>
</tr>
<tr>
<td>s-vwn_Gaussian</td>
<td>LDA</td>
<td>S</td>
<td>VWN(III)</td>
<td>1-3</td>
</tr>
<tr>
<td>pwlda</td>
<td>LDA</td>
<td>S</td>
<td>PW</td>
<td>1,2,4</td>
</tr>
<tr>
<td>becke-exchange</td>
<td>GGA</td>
<td>S+B88</td>
<td></td>
<td>1,2,5</td>
</tr>
<tr>
<td>b-lyp</td>
<td>GGA</td>
<td>S+B88</td>
<td>LYP</td>
<td>1,2,6</td>
</tr>
<tr>
<td>b-vwn</td>
<td>GGA</td>
<td>S+B88</td>
<td>VWN(V)</td>
<td>1-3,5</td>
</tr>
<tr>
<td>lyp</td>
<td>GGA</td>
<td></td>
<td>LYP</td>
<td>6</td>
</tr>
<tr>
<td>b-p</td>
<td>GGA</td>
<td>S+B88</td>
<td>VWN(V)+P86</td>
<td>1-3,5,7</td>
</tr>
<tr>
<td>pbe</td>
<td>GGA</td>
<td>S+PBE(X)</td>
<td>PW+PBE(C)</td>
<td>1,2,4,8</td>
</tr>
<tr>
<td>tpss</td>
<td>MGGA</td>
<td>S+TPSS(X)</td>
<td>PW+TPSS(C)</td>
<td>1,2,4,14</td>
</tr>
<tr>
<td>bh-lyp</td>
<td>HYB</td>
<td>0.5(S+B88)</td>
<td>LYP</td>
<td>1,2,5,6,9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.5HF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b3-lyp</td>
<td>HYB</td>
<td>0.8S+0.72B88</td>
<td>0.19VWN(V)</td>
<td>1-3,5,6,10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.2HF</td>
<td>+0.81LYP</td>
<td></td>
</tr>
<tr>
<td>b3-lyp_Gaussian</td>
<td>HYB</td>
<td>0.8S+0.72B88</td>
<td>0.19VWN(III)</td>
<td>1-3,5,6,10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.2HF</td>
<td>+0.81LYP</td>
<td></td>
</tr>
<tr>
<td>pbe0</td>
<td>HYB</td>
<td>0.75(S+PBE(X))</td>
<td>PW+PBE(C)</td>
<td>1,2,4,8,11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.25HF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tpssh</td>
<td>HYB</td>
<td>0.9(S+TPSS(X))</td>
<td>PW+TPSS(C)</td>
<td>1,2,4,14,15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.1HF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lhf</td>
<td>EXX</td>
<td>EXX</td>
<td></td>
<td>12,13</td>
</tr>
<tr>
<td>b97-d</td>
<td>GGA</td>
<td>B97 refit</td>
<td>B97 refit</td>
<td>16</td>
</tr>
<tr>
<td>b2-plyp</td>
<td>DHYB</td>
<td>0.47(SB88)+0.53HF</td>
<td>0.73LYP+0.27PT2</td>
<td>17</td>
</tr>
</tbody>
</table>

Default is b-p, i.e. B-P86, which is probably best for the whole of Chemistry [31]. For main group compounds we recommend b3-lyp; note that GAUSSIAN uses partly different implementations [31].

The programs dscf and grad are used to carry out conventional DFT treatments, i.e. \( J \) and \( K \) are evaluated without approximations.

### RI-J Calculations

For non-hybrid functionals we strongly recommend the RI-J procedure, which speeds up calculations by a factor 10 at least (as compared to conventional treatments) without sacrificing accuracy. Command ri gives:

#### STATUS OF RI-OPTIONS:

- **RI is not used**
- Memory for RI: 200 MB
- Filename for auxbasis: auxbasis

**Enter RI-option to be modified**

m: Change memory for RI
4.4. THE GENERAL OPTIONS MENU

f: CHANGE FILENAME
jbas: ASSIGN AUXILIARY RI-J BASIS SETS
on: TO SWITCH ON RI

Use <ENTER>, q, end, or * to leave this menu.

Activate RI-J with on, and choose with m the memory you can dedicate to store three-center integrals (Keyword: $ricore), default is 200 MB. The more memory, the faster the calculation.

A rough guide: put $ricore to about 2/3 of the memory of the computer. Use OS specific commands (top on most UNIX systems), during an ridft run to find the actual memory usage and then adjust $ricore, the keyword in control specifying memory.

If the option jbas is selected, define enters a submenu which allows the assignment of auxiliary basis sets (for an explanation of the menu items see Section 4.2). Where available, the program will select by default the auxiliary basis sets optimized for the orbital basis used. Please note that treatment of systems with diffuse wavefunctions may also require an extension of the auxiliary basis. For this cases enlarge the sets of s- and p-functions with diffuse functions.

The RI-J option is only supported by programs ridft and rdgrad, if you use jobex to optimize molecular geometry, put: nohup jobex -ri ...

MARI-J option

RI-J calculations can be done even more efficiently with the Multipole Accelerated RI-J (MARI-J) option, especially for larger molecules where almost linear scaling is achieved [32].

Parameters:
1) precision parameter: 1.00E-06
2) maximum multipole l-moment: 10
3) maximum number of bins: 8
4) minimum separation of bins: 0.00
5) maximum allowed extension: 20.00
6) threshold for multipole neglect: 1.00E-18

Enter the number to change a value or <return> to accept all.

Just rely on the defaults.

Multiple auxiliary basis sets

With the command trunc you can switch on this option. Effect: a reduced auxiliary (or fitting) basis to represent the electron density is employed during SCF iterations, the final SCF iteration and the gradient are computed with the full auxiliary basis.

truncated RI ALREADY SWITCHED ON
DO YOU WANT TO SWITCH OFF truncation? (default=no)
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

Note: trunc is presently not compatible with marij!

RI in SCF calculations

Considerable savings in CPU times are achieved with the RI technique for both Coulomb \( J \) and exchange \( K \) terms in SCF calculations, the RI-JK method [33], provided large basis sets are employed, e.g. TZVPP, cc-pVTZ, or cc-pVQZ. With rijk you get:

STATUS OF RI-OPTIONS:
RI IS NOT USED
Memory for RI: 200 MB
Filename for auxbasis: auxbasis

ENTER RI-OPTION TO BE MODIFIED
m: CHANGE MEMORY FOR RI
f: CHANGE FILENAME
jkbas: ASSIGN AUXILIARY RI-JK BASIS SETS
on: TO SWITCH ON RI
Use <ENTER>, q, end, or * to leave this menu

For an explanation of the menu items see Section 4.4.1. RI-JK calculations can be carried out with the program ridft.

Optimization to minima and transition structures using Statpt

Structure optimizations can be carried out by the program statpt. For minimizations no additional keywords are required. The default values are assumed, which work in most of the cases. Structure optimization is performed in internal coordinates if they have been set. Otherwise, Cartesian coordinates are used. One can switch the optimization in internal coordinates on or off, respectively in internal redundant or cartesian coordinates. For transition structure optimizations the index of transition vector has to be set to an integer value \( > 0 \) (0 means structure minimization). The value of the index specifies transition vector to follow during the saddle point search. Note, that Hessian eigenpairs are stored in ascending order of the eigenvalues, i.e. the eigenpair with the smallest eigenvector has the index 1.

The command stp gives:

------------------------------------------------------------------------
CONVERGENCE CRITERIA:
------------------------------------------------------------------------

thre 1.000000E-06  thre : threshold for ENERGY CHANGE
thrd 1.000000E-03  thrd : threshold for MAX. DISPL. ELEMENT
thrg 1.000000E-03  thrg : threshold for MAX. GRAD. ELEMENT
rmsd 5.000000E-04  rmsd : threshold for RMS OF DISPL.
rmsg 5.000000E-04  rmsg : threshold for RMS OF GRAD.
4.4. THE GENERAL OPTIONS MENU

defl : set default values.

------------------------------------------------------------------------

OPTIMIZATION refers to

<table>
<thead>
<tr>
<th>int</th>
<th>off</th>
<th>int : INTERNAL coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdn</td>
<td>off</td>
<td>rdn : REDUNDANT INTERNAL coordinates</td>
</tr>
<tr>
<td>crt</td>
<td>on</td>
<td>crt : CARTESIAN coordinates</td>
</tr>
</tbody>
</table>

NOTE : options int and crt exclude each other

ENTER STATPT-OPTIONS TO BE MODIFIED

| itvc | 0   | itvc : change INDEX OF TRANSITION VECTOR |
| updte | bfgs | updte : change method of HESSIAN UPDATE |
| hsfreq | 0   | hsfreq : frequency of HESSIAN CALCULATION |
| kptm | 0   | kptm : FREEZING transition vector INDEX |
| hdiag | 5.000000E-01 | hdiag : change DIAGONAL HESSIAN ELEMENTS |
| rmax | 3.000000E-01 | rmax : change MAX. TRUST RADIUS |
| rmin | 1.000000E-04 | rmin : change MIN. TRUST RADIUS |
| trad | 3.000000E-01 | trad : change TRUST RADIUS |

Just <ENTER>, q or '*' terminate this menu.

Excited states, frequency-dependent properties, and stability analysis

Excited state calculations with RPA or CIS (based on HF-SCF) and TDDFT procedures as well as stability analyses (SCF or DFT) are carried out by the program escf.

You will need a well converged HF-SCF or DFT calculation that were converged to at least $scfconv=7$, see Section 4.4.2.

Details of calculations are specified with the command ex:

MAIN MENU FOR RESPONSE CALCULATIONS

<table>
<thead>
<tr>
<th>OPTION</th>
<th>STATUS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>rpas</td>
<td>off</td>
<td>RPA SINGLET EXCITATIONS (TDHF OR TDDFT)</td>
</tr>
<tr>
<td>ciss</td>
<td>off</td>
<td>TDA SINGLET EXCITATIONS (CI SINGLES)</td>
</tr>
<tr>
<td>rpat</td>
<td>off</td>
<td>RPA TRIPLET EXCITATIONS (TDHF OR TDDFT)</td>
</tr>
<tr>
<td>cist</td>
<td>off</td>
<td>TDA TRIPLET EXCITATIONS (CI SINGLES)</td>
</tr>
<tr>
<td>polly</td>
<td>off</td>
<td>STATIC POLARIZABILITY</td>
</tr>
<tr>
<td>dynpol</td>
<td>off</td>
<td>DYNAMIC POLARIZABILITY</td>
</tr>
<tr>
<td>single</td>
<td>off</td>
<td>SINGLET STABILITY ANALYSIS</td>
</tr>
<tr>
<td>triple</td>
<td>off</td>
<td>TRIPLET STABILITY ANALYSIS</td>
</tr>
<tr>
<td>nonrel</td>
<td>off</td>
<td>NON-REAL STABILITY ANALYSIS</td>
</tr>
</tbody>
</table>

ENTER <OPTION> TO SWITCH ON/OFF OPTION, * OR q TO QUIT
If you have selected an option, e.g. \texttt{rpas}, and quit this menu, you will get another menu:

\begin{verbatim}
SELECT IRREP AND NUMBER OF STATES
ENTER ? FOR HELP, * OR Q TO QUIT, & TO GO BACK
\end{verbatim}

This should be self-evident.

**MP2, MP2-F12, RI-MP2, and PNO-MP2**

We recommend to use MP2 for standard applications together with the RI technique using the \texttt{ricc2} program. The \texttt{mpgrad} program is supplied for special benchmark application where the RI approximation needs to be avoided. The \texttt{pnoccsd} program is meant only for very large (> 100 atoms and > 3000 basis functions) MP2 and MP2-F12 single point energy calculations. This is more efficient and supports the frozen core option in the gradient calculation.

The entry \texttt{mp2} leads to a submenu which allows to set some keywords for MP2 and RI-MP2 calculations, e.g. defining frozen orbitals, maximum memory usage, or assign auxiliary basis sets for RI-MP2 calculations, etc. If you want to use \texttt{ricc2}, you have to use the entry \texttt{cc} and the submenu \texttt{ricc2} in order to assign MP2 as wavefunction model. For the \texttt{pnoccsd} program you have to use the entry \texttt{pnocc} and the submenu \texttt{pnoccsd} to assign the wavefunction model.

Conventional MP2 calculations with \texttt{mpgrad} require a number of additional settings for which it is recommended to invoke the interactive tool \texttt{mp2prep}. For geometry optimizations with \texttt{jobex} use \texttt{nohup jobex -level mp2 -ri ...}

**CC2 and CCSD calculations**

The entry \texttt{cc} leads to a submenu which allows to set a number of keywords essential for calculations with the programs \texttt{ricc2} and \texttt{ccsdf12}. In particular it allows the assignment of the wavefunction method(s), selection of auxiliary basis sets, the specification of frozen orbitals, and the definition of a scratch directory and of the maximum core memory usage.

The \texttt{ricc2} program must be used for excitation energies and response properties with second-order methods (MP2, CIS(D), ADC(2), CC2, etc. and their spin-scaled variants), while the \texttt{ccsdf12} program must be used for third- and higher-order methods (MP3, CCSD, CCSD(T), etc.).

**2nd analytical derivatives for SCF and DFT**

The program \texttt{aoforce} computes force constants and IR and Raman Spectra on SCF and DFT level. Analytical second derivative calculations can directly be started from converged SCF or DFT calculations. Note, that the basis is restricted to \textit{d}-functions, and ROHF as well as broken occupation numbers are not allowed. For
better efficiency, in case of larger systems, use the keyword $\texttt{maxcor}$ as described in Chapter 14 to reduce computational cost. RI will be used if the RI option for DFT has been specified.

### 4.4.2 Special adjustments

Adjustments described by the following menus are often better done directly in the control file; have a look at the keywords in Chapter 21. For common calculations just start with the defaults, and change keywords directly in control if you encounter problems with your calculation.

**SCF options**

**ENTER SCF-OPTION TO BE MODIFIED**

<table>
<thead>
<tr>
<th>Option</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv</td>
<td></td>
<td>ACCURACY OF SCF-ENERGY $\texttt{scfconv}$</td>
</tr>
<tr>
<td>thi</td>
<td></td>
<td>INTEGRAL STORAGE CRITERIA $\texttt{thize} \ $\texttt{thime}$</td>
</tr>
<tr>
<td>ints</td>
<td></td>
<td>INTEGRAL STORAGE ALLOCATION $\texttt{scfintunit}$</td>
</tr>
<tr>
<td>iter</td>
<td></td>
<td>MAXIMUM NUMBER OF ITERATIONS $\texttt{scfiterlimit}$</td>
</tr>
<tr>
<td>diis</td>
<td></td>
<td>DIIS CONVERGENCE ACCELERATION $\texttt{scfdiis}$</td>
</tr>
<tr>
<td>damp</td>
<td></td>
<td>OPTIONS FOR DAMPING $\texttt{scfdamp}$</td>
</tr>
<tr>
<td>shift</td>
<td></td>
<td>SHIFTING OF ORBITALS $\texttt{scforbitalshift}$</td>
</tr>
<tr>
<td>order</td>
<td></td>
<td>ORDERING OF ORBITALS $\texttt{scforbitalorder}$</td>
</tr>
<tr>
<td>fermi</td>
<td></td>
<td>THERMAL SMEARING OF OCC. NUMBERS $\texttt{fermi}$</td>
</tr>
</tbody>
</table>

By the command $\texttt{fermi}$ you can switch on smearing of occupation numbers, and thus automatically optimize occupations and spin.

**Menu drv**

The most important of the derivative menus is the first one which tells the programs which derivatives to calculate. This is only necessary for special purposes and you should better not change default options.

---

**derivative data groups '$\texttt{drvopt}$, $\texttt{drvtol}$'**

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crt</td>
<td>T</td>
<td>CARTESIAN 1st derivatives</td>
</tr>
<tr>
<td>sec</td>
<td>T</td>
<td>CARTESIAN 2nd derivatives</td>
</tr>
<tr>
<td>bas</td>
<td>F</td>
<td>energy derivatives with respect to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BASIS SET exponents/scaling factors/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>contraction coefficients</td>
</tr>
<tr>
<td>glb</td>
<td>F</td>
<td>energy derivative with respect to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a GLOBAL scaling factor</td>
</tr>
<tr>
<td>dip</td>
<td>T</td>
<td>cartesian 1st derivatives of DIPOLE MOMENT</td>
</tr>
<tr>
<td>pol</td>
<td>T</td>
<td>nuclear contribution to POLARIZABILITY</td>
</tr>
</tbody>
</table>
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

fa | F | SPECTROSCOPIC ANALYSIS only
tol 0.100D-06 derivative integral cutoff

use <opt> for enabling, -<opt> for disabling of logical switches
<&> will bring you back to GENERAL MENU without more changes
<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

The handling of these options is very simple. With the exception of tol, all are logical switches which are either true (or on, active) or false (or off, inactive). You can switch between the two states if you enter, for example, crt (to switch calculation of Cartesian first derivatives on) or -crt (to switch it off). The options crt, sec and bas should provide no problems. glb refers to a global scaling factor for all basis set exponents. Imagine that you would like to replace your basis set, which contains basis functions

\[ \chi_\mu = (x-x_0)^l(y-y_0)^m(z-z_0)^n \exp \left[-\eta_\mu (r-r_0)^2 \right] \]

by another basis set which contains basis functions

\[ \chi_\mu = (x-x_0)^l(y-y_0)^m(z-z_0)^n \exp \left[-\alpha \eta_\mu (r-r_0)^2 \right] \]

where \( \alpha \) is the same for all primitive basis functions \( \chi_\mu \). With command glb you are able to calculate analytical derivatives of the total energy with respect to \( \alpha \) and can thus easily determine the optimum \( \alpha \).

dip enables you to calculate the first derivatives of the electric dipole moment with respect to nuclear displacements which gives you infrared intensities. pol allows you to calculate the contribution of the nuclear rearrangement on the electric polarizability. fa finally performs only a frequency analysis which means that aoforce will read the force constant matrix (\$hessian or \$hessian (projected)), diagonalize it and give you the frequencies and normal modes. tol is not a logical switch as the other options in this menu, but a cutoff threshold for the derivative integrals, i.e. integrals below this threshold will be neglected in the derivative calculations.

Entering * will bring you to the second derivative submenu.

Debug Options for the Derivative Programs

The following menu deals only with some debug options for grad. Use them with caution, each of them can produce lots of useless output:
4.4. THE GENERAL OPTIONS MENU

derivative debug options '$drvdebug'

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disp1e</td>
<td>F</td>
<td>display 1e contributions to desired derivatives</td>
</tr>
<tr>
<td>only1e</td>
<td>F</td>
<td>calculate 1e contributions to desired derivatives only</td>
</tr>
<tr>
<td>debug1e</td>
<td>F</td>
<td>display 1e shell contributions to desired derivatives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(WARNING : this produces large outputs!)</td>
</tr>
<tr>
<td>debug2e</td>
<td>F</td>
<td>display 2e shell contributions to desired derivatives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(WARNING : this produces VERY large outputs!)</td>
</tr>
<tr>
<td>debugvib</td>
<td>F</td>
<td>debug switch for vibrational analysis (force only)</td>
</tr>
<tr>
<td>notrans</td>
<td>F</td>
<td>disable transfer relations (gradient only!)</td>
</tr>
<tr>
<td>novirial</td>
<td>F</td>
<td>disable virial scaling invariance in basis set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>optimizations (gradient only)</td>
</tr>
</tbody>
</table>

use <opt> for enabling, -<opt> for disabling option <opt>
<&> will bring you back to GENERAL MENU without more changes
<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

As there is no need to use these options normally and the menu text is self-explaining, no further description will be given. Note that all options are logical switches and may be enabled and disabled the same way as shown for the last menu. Entering * will bring you to the last derivative submenu.

4.4.3 Relax Options

Program relax has a huge variety of options to control its actions which in program define are grouped together in eight consecutive menus. These are only briefly described in the following sections; for a more detailed discussion of the underlying algorithms refer to the documentation of program relax (see Section 5.3). Only experts should try to change default settings.

Optimization Methods

The first of the relax subgenus deals with the type of optimization to be performed:

optimization options for RELAX

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>F</td>
<td>INTERNAL coordinates</td>
</tr>
<tr>
<td>crt</td>
<td>F</td>
<td>CARTESIAN coordinates</td>
</tr>
<tr>
<td>bas</td>
<td>F</td>
<td>BASIS SET exponents/scale factors</td>
</tr>
<tr>
<td>glb</td>
<td>F</td>
<td>GLOBAL scaling factor</td>
</tr>
</tbody>
</table>

use <opt> for enabling, -<opt> for disabling option <opt>
<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU
You can choose between a geometry optimization in the space of internal coordinates (in this case you will need definitions of internal coordinates, of course) or in the space of Cartesian coordinates (these possibilities are mutually exclusive, of course). Furthermore optimizations of basis set parameters (exponents, contraction coefficients and scaling factors) or of a global scaling factor is possible (these options are also exclusive, but can be performed simultaneous to a geometry optimization). For the geometry optimization you should normally use internal coordinates as they provide better convergence characteristics in most cases.

**Coordinate Updates**

The next submenu deals with the way relax updates the old coordinates. You may choose a maximum change for the coordinates or you can allow coordinate updates by means of extrapolation:

```
coordinate update options for RELAX

dqmax <real> : coordinates are allowed to change by at most
<real> (DEFAULT : 0.3000 ) a.u.
polish : perform an interpolation or extrapolation of
coordinates (DEFAULT :y)
-polish : disable inter/extrapolation

<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU
```

These options result in better convergence of your optimization in most cases.

**Interconversion Between Internal and Cartesian Coordinates**

The interconversion between internal and Cartesian coordinates is not possible directly (in this direction). Instead it is performed iteratively. The following options control this conversion:
4.4. THE GENERAL OPTIONS MENU

---

**interconversion options for RELAX**

<table>
<thead>
<tr>
<th>option</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>switch on interconversion (DEFAULT: off)</td>
</tr>
<tr>
<td>qconv &lt;r&gt;</td>
<td>set convergence threshold for interconversion of coordinates to &lt;r&gt;. DEFAULT : &lt;r&gt; = .1000E-09</td>
</tr>
<tr>
<td>iter &lt;i&gt;</td>
<td>allow at most &lt;i&gt; iterations for interconversion of coordinates. DEFAULT : &lt;i&gt; = 25</td>
</tr>
<tr>
<td>crtint</td>
<td>transform cartesian into internal coordinates (DEFAULT=n)</td>
</tr>
<tr>
<td>intcrt</td>
<td>transform internal into cartesian coordinates (DEFAULT=n)</td>
</tr>
<tr>
<td>grdint</td>
<td>transform cartesian into internal gradients (DEFAULT=n)</td>
</tr>
<tr>
<td>hssint</td>
<td>transform cartesian into internal hessian (DEFAULT=n)</td>
</tr>
</tbody>
</table>

---

use -<opt> for disabling any interconversion option

<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

The options qconv and iter are used in each normal relax run to determine the characteristics of the back-transformation of coordinates into the internal space. With the other options and interconversion switched on, you can force relax to perform only the specified coordinate transformation and write the transformed coordinates to file control. To achieve this, enter on to switch to the transformation-only mode, and one of the last four options, e.g. crtint, to specify the desired transformation.

---

**Updating the Hessian**

relax provides a variety of methods to generate an updated Hessian every cycle. This includes the well known methods such as BFGS, DFP, or MS update methods as well as some less common procedures:

---

**OPTIONS FOR UPDATING THE HESSIAN**

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>F</td>
<td>NO UPDATE (STEEPEST DESCENT)</td>
</tr>
<tr>
<td>bfgs</td>
<td>F</td>
<td>BROYDEN-FLETCHER-GOLDFARB-SHANNO UPDATE</td>
</tr>
<tr>
<td>dfp</td>
<td>F</td>
<td>DAVIDON-FLETCHER-POWELL UPDATE</td>
</tr>
<tr>
<td>bfgs-dfp</td>
<td>F</td>
<td>COMBINED (BFGS+DFP) UPDATE</td>
</tr>
<tr>
<td>ms</td>
<td>F</td>
<td>MURTAGH-SARGENT UPDATE</td>
</tr>
<tr>
<td>shlegel</td>
<td>F</td>
<td>SCHLEGEL UPDATE</td>
</tr>
<tr>
<td>diagup</td>
<td>F</td>
<td>DIAGONAL UPDATE (AHLRICHS/EHRIG)</td>
</tr>
<tr>
<td>multidim</td>
<td>F</td>
<td>RANK &gt; 2 BFGS-TYPE UPDATE</td>
</tr>
<tr>
<td>ahlrichs</td>
<td>T</td>
<td>MACRO : AHLRICHS UPDATE (DEFAULT)</td>
</tr>
</tbody>
</table>

USE <opt> FOR ENABLING OPTION <opt> AND THUS DISABLING ALL OTHER OPTIONS.

<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

We recommend to use the default method ahlrichs which provides excellent con-
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

General Boundary Conditions for Update

The force constant matrix will only be updated if at least \texttt{mingeo} cycles exist. The maximum number of cycles used for the update is specified by the parameter \texttt{maxgeo}. Normally the default values provided by \texttt{define} need not be changed.

\textbf{DEFINE BOUNDARY CONDITIONS FOR UPDATE}

\begin{verbatim}
mingeo <i> | START UPDATE IF THERE ARE AT LEAST <i> CYCLES
     | DEFAULT : min 3
maxgeo <i> | USE LAST <i> CYCLES FOR UPDATE, DEFAULT : max 4
\end{verbatim}

<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

Special Boundary Conditions for Ahlrichs and Pulay Updates

For the default update method \texttt{ahlrichs} some additional control parameters are available which can be defined in this menu:

\textbf{DEFINE BOUNDARY CONDITIONS FOR AHLRICHS OR PULAY UPDATE}

\begin{verbatim}
option     | description
modus <i>  | DEFINE MODUS FOR GDIIS PROCEDURE : MINIMIZE
     | <dq|dq> IF <i> = 0
     | <g|dq> IF <i> = 1
     | <g|g> IF <i> = 2
     | <dE> IF <i> = 3
     | DEFAULT : <i> = 1
fail <r>   | IGNORE GDIIS IF <g|dq> /| <g|dq> | IS
     | LARGER THAN -<r>. DEFAULT : <r> = 0.1
\end{verbatim}

<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

For detailed description consult Section 5.3.

\textbf{OPTIONS FOR MANIPULATING THE HESSIAN}

\begin{verbatim}
option     | description
diagonal   | RESTRICT UPDATE TO DIAGONAL-ELEMENTS IF
     | METHOD IS BFGS,DFP OR MS. DEFAULT=n
offreset   | DISCARD OFF-DIAGONAL ELEMENTS. DEFAULT=n
offdamp <r> | DAMP OFF-DIAGONAL ELEMENTS BY 1/(1+<r>) DEFAULT= 1.000
damp <real> | DAMP UPDATE BY 1/(1+<real>), DEFAULT= .0000E+00
scale <real> | SCALE INPUT HESSIAN BY <real>, DEFAULT= 1.000
allow <real> | SCALE INPUT HESSIAN BY <real>/|DE| IF |DE|,
4.4. THE GENERAL OPTIONS MENU

THE OBSERVED ABSOLUTE CHANGE IN ENERGY, IS
OBEYING THE CONDITION |DE| > <real> > 0.
DEFAULT : NO SCALING

min <real> | DO NOT ALLOW EIGENVALUES OF HESSIAN TO DROP
BELOW <real>. DEFAULT = .1000E-02
reset <real> | USE <real> AS A RESET VALUE FOR TOO SMALL
EIGENVALUES (CP. min). DEFAULT = .1000E-02
max <real> | DO NOT ALLOW EIGENVALUES OF HESSIAN TO BECOME
LARGER THAN <real>. DEFAULT = 1000.

WITH THE EXCEPTION OF min, reset AND max, ALL OPTIONS MAY BE
DISABLED BY ENTERING -<opt>
<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

Initialization of the Hessian

Finally there are some options to control the choice of the initial Hessian during your
geometry optimization:

FORCE CONSTANTS INITIALIZATION OPTIONS FOR RELAX

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>switch off initialization (DEFAULT: on)</td>
</tr>
<tr>
<td>cart</td>
<td>use analytical cartesian hessian provided by a</td>
</tr>
<tr>
<td></td>
<td>2nd derivatives calculation. DEFAULT(n)</td>
</tr>
<tr>
<td>diag</td>
<td>use diagonal matrix with diagonal elements set</td>
</tr>
<tr>
<td></td>
<td>individually within data groups $intdef or</td>
</tr>
<tr>
<td></td>
<td>$basis or $global. DEFAULT(n)</td>
</tr>
<tr>
<td>unit &lt;r&gt;</td>
<td>use multiple of the unit matrix ( H = &lt;r&gt;*E ).</td>
</tr>
<tr>
<td></td>
<td>DEFAULT(n) - DEFAULT &lt;r&gt; = 1.000</td>
</tr>
</tbody>
</table>

NOTE THAT THESE OPTIONS ARE MUTUALLY EXCLUSIVE
<RETURN> OR * OR q(uit) WILL TERMINATE THIS MENU

Option off will be used if you have already a good Hessian from a previous calcu-
lalion which may be used. cart describes an even better state where you have a
Hessian from a calculation of the second derivatives available (aoforce). The other
two options describe real procedures for initialization of the Hessian. Default values:
stretches (0.5), angles (0.2).

4.4.4 Definition of External Electrostatic Fields

This submenu allows you to calculate first and second numerical derivatives of the
energy with respect to an external electric field. The first three options should be
clear; 1st and 2nd are logical switches which are turned on and off the usual way
(1st or -1st) and delta is the increment for the numerical differentiation, that is,
the finite value of the external field, which replaces the (ideally) differential field:
### electrostatic field definition menu

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>F</td>
<td>numerical 1st derivative ( \frac{dE}{d\text{Field}} )</td>
</tr>
<tr>
<td>2nd</td>
<td>F</td>
<td>numerical 2nd derivative ( \frac{d^2E}{d\text{Field}^2} )</td>
</tr>
<tr>
<td>delta &lt;real&gt;</td>
<td></td>
<td>increment for numerical differentiation</td>
</tr>
<tr>
<td>geofield</td>
<td>F</td>
<td>geometry optimization with external field</td>
</tr>
<tr>
<td>man</td>
<td>F</td>
<td>explicit definition of electrostatic field(s)</td>
</tr>
</tbody>
</table>

**geofield** gives the possibility to perform a whole geometry optimization under the influence of a finite external field and thus to obtain the (distorted) minimum geometry in this field. To do this, an external electrostatic field must be defined explicitly which can be done using command **man**. Note that **geofield** must also be switched on if any properties are to be evaluated in the presence of an electric field. The most prominent example is the calculation of hyperpolarizabilities.

Take Care, due to some inconsistencies in **define** it is always necessary to switch on the field calculations manually. Therefore edit the **control** file after having finished your **define** session and enter on after the entries of **fields** and **geofield**.

#### 4.4.5 Properties

The program **moloch** used for this purpose is currently being revamped, and will then be much simpler to use. The subsequent description for an older version may not work in all cases—sorry for that.

If you enter **prop** in the general menu, **define** first will check whether the data group \( $\text{properties} \)\) does already exist in your **control** file or in a file referenced therein. If this is not the case you will be asked to specify the file on which \( $\text{properties} \)\) shall be written:

```
data group $\text{properties} \text{ has not yet been specified}
FOR INITIALIZING <moloch> KEYWORDS ENTER
   [return] : WRITE TO CONTROL FILE control (DEFAULT), or
   filename : WRITE TO ANOTHER FILE
```

Afterwards you will get the following submenu which allows you to control all possible actions of program **moloch**:
4.4. THE GENERAL OPTIONS MENU

switch on one or more of the following options <i>
<i> = 1,..., 9
for switching off option <i>, specify -<i>

( 1) trace off
( 2) moments off
( 3) potential off
( 4) cowan-griffin off
( 5) localization off
( 6) population analyses off
( 7) plot off
( 8) firstorder off

selecting an already active option indicates that suboptions shall be modified
* or q(uit) = quit | for help, type help <integer>

All options in this menu are selected by entering their number as indicated in the first column. For example, to switch on option trace enter 1. The flag off will then change to active. To switch off an option enter its negative number, e.g. -1 for trace. Most of the options require additional input and will therefore lead you to further submenus. These are briefly described below.

Option trace

trace will calculate the trace of density times overlap matrix:

\[ N = \text{tr}(DS) \]

If the orbitals are orthonormal, \( N \) should yield the total number of electrons in your molecule. If this is not true, your MO-vector will most probably be erroneous. For example, the vector might belong to another geometry or basis set. As this is a very sensitive test for errors like these and the calculation requires almost no time, you should always switch on this option.

Option moments

This option leads you to the following submenu:

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>point &lt;x&gt; &lt;y&gt; &lt;z&gt;</td>
<td>T</td>
<td>reference point = (x,y,z)</td>
</tr>
<tr>
<td>atom &lt;i&gt;</td>
<td>F</td>
<td>reference point = atom no. &lt;i&gt;</td>
</tr>
<tr>
<td>0th</td>
<td>T</td>
<td>compute 0th moment</td>
</tr>
<tr>
<td>1st</td>
<td>F</td>
<td>compute 1st moment</td>
</tr>
<tr>
<td>2nd</td>
<td>F</td>
<td>compute 2nd moment</td>
</tr>
<tr>
<td>3rd</td>
<td>F</td>
<td>compute 3rd moment</td>
</tr>
</tbody>
</table>

-<moment> : skip computation of <moment>
* or q(uit) : terminate input
CHAPTER 4. PREPARING YOUR INPUT FILE WITH DEFINE

This menu serves to specify the electrostatic moments to be calculated (0th=charge, 1st=dipole moment, 2nd=quadrupole moment, 3rd=octuple moment). The reference point is the origin of the coordinate system used in the calculation. The value of any calculated moment will be independent of this reference point, if all lower moments are zero. The default for the reference point is the origin, i.e. the coordinate system used for the calculation of the moments will be the same as the one in which the atomic coordinates are specified. The reference point may be changed by typing point with the three new coordinates appended. Alternatively you may choose the coordinates of one of the atoms as reference point by entering atom and the atom index.

**Option potential**

This option collects all possible quantities related to the electrostatic field created by the molecular charge distribution. This includes the following suboptions:

- pot - electrostatic potential
- fld - electrostatic field
- fldgrd - electrostatic field gradient
- shld - diamagnetic shielding
- file - file reference
- * - quit

The meaning of the four suboptions pot, fld, fldgrd and shld will probably present no problems to you. For each of them, however, you will have to specify at which point(s) this property should be calculated. This is accomplished by one or more data groups $points in file control. After you chose one or more of the above options, you will therefore reach the next submenu which deals with the specification of these data groups:

- there are 1 data groups $points
- manipulate data group(s) $points
  a - add another data group
  m <integer> - modify <integer>th data group
  m all - modify all data groups
  d <integer> - delete <integer>th data group
  d all - delete all data groups
  off <integer> - switch off <integer>th data group
  off all - switch off all data groups
  on <integer> - switch on <integer>th data group
  on all - switch on all data groups
  s - scan through data groups
  * - quit

The first line informs you how many of these data groups already exist in your control file. Each of these data groups may consist of several points at which the properties will be calculated. You may now create new data groups, delete old
ones or simply switch on or off individual data groups (without deleting them from control). The number of different data groups $points$ as well as the number of points in each of them are not limited. However, if you use many points, you should consider specifying them in a separate file. This is most easily done using option file in the potential menu. This option will create a file for your data groups $points$ and will write a reference of this file to file control.

**Option cowan-griffin**

This option activates the computation of the first order relativistic correction to the energy as given by the expectation value of the Cowan–Griffin operator.

**Option localization**

Specifying option localization will switch on a Boys localization of molecular orbitals. define by default chooses a set of MOs to be localized according to a certain threshold for the orbital energy. Information about these are displayed like this:

```
BOYS localization will be performed with respect to x y z
number of sweeps = 10000
subset of molecular orbitals to be localized :
  ---> all occupied molecular orbitals
    with orbital energy above -2.00000 Hartree
```

The subset of molecular orbitals to be localized is displayed:

```
shells to be localized
----------------------------------------
al  4-5 # 1- 5
  e  2 # 1- 2
----------------------------------------
you are employing default options for localization
do you want to modify them ? DEFAULT(n)
```

If you want to change the MO selection or other options for the localization enter y at this point (By default or when typing n you will reach the moloch options menu again). You will then be asked whether to change the MO selection method. If you want this, you will enter a little submenu where you can choose one of three possible selection procedures:

- **all** selects all occupied orbitals
- **thr** selects all occupied orbitals with orbital energy larger than a certain threshold
- **man** enables you to select the MOs manually later in this section

If the selection method thr is specified you then will be asked for the threshold to be applied for the selection. Afterwards you have the possibility to change some other topics concerning the localization:
• specify other localization directions

• switch on utilization of localized orbitals for population analysis and/or preparation of plot data within the same moloch run

• set the maximum number of sweeps in the localization procedure

• specify a file where localized orbitals shall be written to

Option population analyses

When activating this option you first have to specify whether the population analysis (PA) should be performed in the CAO (default) or AO basis. Afterwards define will ask you whether you want to perform a Mulliken population analysis. In this case, the following submenu will be displayed:

add or delete one or more special options for a Mulliken population analysis

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>spdf</td>
<td>F</td>
<td>compute MO contributions to atomic brutto populations</td>
</tr>
<tr>
<td>molap</td>
<td>F</td>
<td>compute MO contributions to atomic overlap populations</td>
</tr>
<tr>
<td>netto</td>
<td>F</td>
<td>compute atomic netto populations</td>
</tr>
<tr>
<td>irpspd</td>
<td>F</td>
<td>compute IRREP contributions to atomic brutto populations</td>
</tr>
<tr>
<td>irpmol</td>
<td>F</td>
<td>compute IRREP contributions to atomic overlap populations</td>
</tr>
<tr>
<td>mommul</td>
<td>F</td>
<td>print electrostatic moments resulting from atomic charges</td>
</tr>
</tbody>
</table>

Here you can activate several optional quantities to be computed along with the Mulliken PA. To switch on one or more of these options you must enter the corresponding option keywords, e.g. spdf netto for computation of atomic netto populations and MO contributions to atomic brutto populations. The status flags for these tasks will then change from F (false) to T (true). To switch off any option you simply have to enter the corresponding keyword preceded by a ‘-’, e.g. -netto for disabling calculation of atomic netto populations.

After having left the Mulliken PA section you will be asked whether a population analysis based on occupation numbers (a modified Roby–Davidson PA) should be performed by moloch. When typing y you will see the following submenu, where you can switch on several special options for the PA in the same manner as described above.
4.4. THE GENERAL OPTIONS MENU

add or delete one or more special options for a population analysis based on occupation numbers

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>momao</td>
<td>F</td>
<td>compute MO contributions to modified atomic orbital (MAO) occupation numbers</td>
</tr>
<tr>
<td>maodump</td>
<td>F</td>
<td>dump all MAOs onto standard output</td>
</tr>
<tr>
<td>maofile</td>
<td>F</td>
<td>write MAOs onto a separate file</td>
</tr>
<tr>
<td>select</td>
<td>F</td>
<td>write only those MAOs which have been employed in the population analysis</td>
</tr>
<tr>
<td>all</td>
<td>F</td>
<td>write all MAOs</td>
</tr>
</tbody>
</table>

note that the options select and all are complementary

-<option> : switch off <option>
* or q(uit) : leave this menu

Afterwards you have the possibility to change the criterion to be applied for the selection of modified atomic orbitals (MAOs) within the following little submenu:

global criterion for selection of Modified Atomic Orbitals (MAOs) :

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eig &lt;r&gt;</td>
<td>T</td>
<td>select by eigenvalues of the 'atomic' density matrices</td>
</tr>
<tr>
<td>occ &lt;r&gt;</td>
<td>F</td>
<td>select by occupation numbers</td>
</tr>
</tbody>
</table>

<r> is the selection threshold (DEFAULT= .1000)

* or q(uit) : leave this menu

The criterion applied by default is the so-called atomic density eigenvalue with a threshold of 0.1. You can switch the criterion to occupation numbers by entering occ. If you also want to change the threshold, you just have to append its new value to the selection keyword, e.g. occ .2. Finally you can select or disable various options in connection with the computation of shared electron numbers (SEN) within the following menu:
actual settings for data group $shared\electron\ numbers$
2-center shared electron numbers will be computed;
values are printed if absolute value exceeds 0.0100
3-center shared electron numbers will be computed;
values are printed if absolute value exceeds 0.0100
4-center shared electron numbers will be computed;
values are printed if absolute value exceeds 0.0100
add or delete one or more options for the
computation of Shared Electron Numbers (SEN)

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2c &lt;r&gt;</td>
<td>T</td>
<td>compute 2-center SEN and print if [</td>
</tr>
<tr>
<td>3c &lt;r&gt;</td>
<td>T</td>
<td>compute 3-center SEN and print if [</td>
</tr>
<tr>
<td>4c &lt;r&gt;</td>
<td>T</td>
<td>compute 4-center SEN and print if [</td>
</tr>
<tr>
<td>nosym</td>
<td>F</td>
<td>switch off use of symmetry</td>
</tr>
<tr>
<td>orbs</td>
<td>F</td>
<td>compute orbital contributions to SEN</td>
</tr>
<tr>
<td>irreps</td>
<td>F</td>
<td>compute irrep contributions to SEN</td>
</tr>
</tbody>
</table>

-<option> : switch off <option>
* or q(uit) : leave this menu

The procedure for changing the options is the same as described above. By default
calculation of 2-, 3- and 4-center SENs will be enabled with thresholds of 0.01 each.

**Option plot**

This option allows you to prepare the data needed for contour plots of orbital amplitudes or total electron densities. We do not recommend to prepare plotting data this way; an easier method—with an easier syntax—is to generate these data directly by the programs, where densities (also MP2 or excited ones) and Molecular orbitals are calculated. This is described in Chapter 17. If you nevertheless want to prepare the input for plotting data as needed by moloch using define, on activating plot you get the following menu:
4.4. THE GENERAL OPTIONS MENU

there are 1 data groups $grid

manipulate data group(s) $grid
  a - add another data group
  m <integer> - modify <integer>th data group
  m all - modify all data groups
  d <integer> - delete <integer>th data group
  d all - delete all data groups
  off <integer> - switch off <integer>th data group
  off all - switch off all data groups
  on <integer> - switch on <integer>th data group
  on all - switch on all data groups
  s - scan through data groups
  * - quit

The commands in this menu serve for the manipulation of data groups $grid in an analogous way as described for $points in the potential section above. $grid data groups contain the input information necessary to create the plot data by moloch (one data group for each plot). If you want to add a new data group you will enter this submenu:

specify the input orbital / input density :
  mo <label> - use occupied molecular orbital <label>
  mo density - use one electron density built from the occupied molecular orbitals
  lmo <i> - use localized molecular orbital no. <lmo>
  mao <i> <k> - use modified atomic orbital no. <i> centered on atom no. <k>
  help - explanation of the syntax for <label>
  * - quit

Here you may specify the orbital to be plotted. To plot the amplitude of the fifth orbital in irrep a1, e.g., you would enter mo 5a1. Equivalently you can use localized orbitals from a Boys localization procedure or modified atomic orbitals as obtained in a Roby–Davidson–Ahlrichs–Heinzmann population analysis. In the latter cases you will not have to enter an irrep label, as these orbitals are necessarily in C1 symmetry. Instead you will have to enter the index of the orbital to be plotted (and for option mao the index of the atom at which it is situated). In all cases you will additionally have to specify the plane in which the amplitudes or densities will be monitored. To do this, you have to declare two vectors which span that plane and the origin of this new coordinate system relative to the one in which the atomic coordinates are given. Furthermore, you will have to create a grid of points on this plane. The orbital amplitude or electron density will then be calculated for every point in this grid. The grid is created by telling define the range to be included along both vectors spanning the plane (where the unit in each direction is the length of the corresponding basis vector) and the number of points to be calculated in this range. It is advantageous to use a wide grid while you test the ranges or planes which give the best results and then to switch to a finer grid for the final calculation. Finally input (MO vector) and output (plot data) files can be specified.

In case you do not want to add a new data group as described above but to change
an existing one, you will be asked which one of the specifications you want to modify.
Chapter 5

Calculation of Molecular Structure and *Ab Initio* Molecular Dynamics

5.1 Structure Optimizations using the JOBEX Script

In its normal mode of operation, the shell script *jobex* controls and executes automatic optimizations of molecular geometry parameters. It will cycle through the direct SCF, gradient and force relaxation programs and stop if either the maximum number of cycles is reached or the convergence criteria (change in the total energy, maximum norm of the gradient) are fulfilled. By default, the executable programs are taken from the load modules library within the *TURBOMOLE* directory.

5.1.1 Options

Given a shell the usage is:

```
nohup jobex &
```

This command invokes structure optimization using the default program *statpt*. Structure optimizations using program *relax* can be performed using `-relax` flag:

```
nohup jobex -relax &
```

*nohup* means that the command is immune to hangups, logouts, and quits. & runs a background command. *jobex* accepts the following arguments controlling the level of calculation, convergence criteria and many more (for example *nohup jobex -gcart 4 &*):

```
-energy integer converge total energy up to 10^{(-<integer>)} Hartree (default: 6)
```
CHAPTER 5. STRUCTURE OPTIMIZATIONS

-gcart integer  converge maximum norm of cartesian gradient up to $10^{-<\text{integer}>}$ atomic units (default: 3)
-c integer  perform up to integer cycles (default: 100)
-dscf begin with a direct SCF step
-grad begin with a gradient step
-statpt begin with a force relaxation step
-relax use the relax program for force relaxation
-trans perform transition state search
-level level  define the optimization level, level=scf, mp2, cc2, uff, or rirpa (default is scf).
-ri use RI modules ridft and rdgrad (fast Coulomb approximation) instead of dscf and grad as well as rimp2 instead of mpgrad; obligatory option if -level rirpa
-rijk in connection with '-level cc2', the RI-JK versions of HF and CPHF are switched on
-ex perform excited state geometry optimization using egrad
-l <path> employ programs from directory <path>
-ls <path> load scripts from directory <path>
-md  a molecular dynamics (MD) run (using frog instead of relax)
-mdfile file commands for MD run are contained in this file (default: mddmaste)
-mdscript file option to execute a shell script before the frog step
-keep keep program output from all optimization steps
-help shows a short description of the commands above

5.1.2 Output

There will be an output written to file job.start which informs you about the current options. The convergence is signalled by the file converged; otherwise, you should find the file not.converged within your working directory. If jobex finds a file named stop or STOP in the working directory, jobex will stop after the present step has terminated. You can create stop by the command touch stop.

The output of the last complete cycle is written to file job.last, while the output of the running cycle is collected within the file job.<cycle>, where <cycle> is the index of the cycle. The convergence criteria and their current values are written out at the bottom of the job.last file.
5.2. Program **STATPT**

5.2.1 General Information

Stationary points are places on the potential energy surface (PES) with a zero gradient, i.e. zero first derivatives of the energy with respect to atomic coordinates. Two types of stationary points are of special importance to chemists. These are minima (reactants, products, intermediates) and first-order saddle points (transition states).

The two types of stationary points can be characterized by the curvature of the PES at these points. At a minimum the Hessian matrix (second derivatives of energy with respect to atomic coordinates) is positive definite, that is the curvature is positive in all directions. If there is one, and only one, negative curvature, the stationary point is a transition state (TS). Because vibrational frequencies are basically the square roots of the curvatures, a minimum has all real frequencies, and a saddle point has one imaginary vibrational "frequency".

Structure optimizations are most effectively done by so-called quasi-Newton–Raphson methods. They require the exact gradient vector and an approximation to the Hessian matrix. The rate of convergence of the structure optimization depends on anharmonicity of the PES and of the quality of the approximation to the Hessian matrix.

The optimization procedure implemented in **STATPT** belongs to the family of quasi-Newton–Raphson methods [34]. It is based on the restricted second-order method, which employs Hessian shift parameter in order to control the step length and direction. This shift parameter is determined by the requirement that the step size should be equal to the actual value of the trust radius, **tradius**, and ensures that the shifted Hessian has the correct eigenvalue structure, all positive for a minimum search, and one negative eigenvalue for a TS search. For TS optimization there is another way of describing the same algorithm, namely as a minimization on the "image" potential. The latter is known as TRIM (Trust Radius Image Minimization) [35].

For TS optimizations the TRIM method implemented in **STATPT** tries to maximize the energy along one of the Hessian eigenvectors, while minimizing it in all other directions. Thus, one “follows” one particular eigenvector, hereafter called the “transition” vector. After computing the Hessian for your guess structure you have to identify which vector to follow. For a good TS guess this is the eigenvector with negative eigenvalue, or imaginary frequency. A good comparison of different TS optimization methods is given in [36].

Structure optimizations using **STATPT** are controlled by the keyword `$statpt` to be present in the control file. It can be set either manually or by using the `stp` menu of `define`. The type of stationary point optimization depends on the value of `itrvec` specified as an option within `$statpt`. By default `itrvec` is set to 0, which implies a structure minimization. A value `itrvec > 0` implies a transition state optimization using the eigenvalue-following TRIM algorithm, where the index of the transition vector is specified by `itrvec`. Note, that **STATPT** orders eigenvalues (and
(eigenvectors) of the Hessian in ascending order, shifting six (or five in the case of linear molecules) zero translation and rotation eigenvalues to the end.

Note: this order differs from that used for vibrational frequencies in the control file, where rotational and translational eigenvalues are not shifted.

By default a structure optimization is converged when all of the following criteria are met:

- the energy change between two optimization cycles drops below the value given by \texttt{threchange} (default: \(10^{-6}\) a.u.),

- the maximum displacement element drops below the value given by \texttt{thrmax-displ} (default: \(10^{-3}\) a.u.),

- the maximum gradient element drops below the value given by \texttt{thrmaxgrad} (default: \(10^{-3}\) a.u.),

- the root mean square of the displacement elements drops below the value given by \texttt{thrrmsdispl} (default: \(5 \cdot 10^{-4}\) a.u.),

- the root mean square of the gradient elements drops below the value given by \texttt{thrrmsgrad} (default: \(5 \cdot 10^{-4}\) a.u.).

The default values for the convergence criteria can be changed using the \texttt{stp} menu of \texttt{define}. The necessary keywords are described in Section 21.2.20 below.

For structure optimization of minima with \texttt{statpt} as relaxation program just use:

\begin{verbatim}
jobex &
\end{verbatim}

TS optimizations are performed by the \texttt{jobex} invocation:

\begin{verbatim}
jobex -trans &
\end{verbatim}

\subsection{5.2.2 Hessian matrix}

The choice of the initial Hessian matrix has a great effect on the convergence of the structure optimization. At present, there are three choices for the Hessian matrix in \texttt{statpt}. For minimization, a diagonal matrix or approximate Hessian matrix from a forcefield calculation using \texttt{uff} (see Section 5.4) can be used. For transition state optimizations you have to provide either the “exact” Hessian or results from the lowest eigenvalue search (LES, see Section 14). Note also that you can calculate the Hessian with a smaller basis set and/or at a lower wavefunction level, and use it for higher level structure optimization. Usually, a Hessian matrix calculated in a minimal basis using RI-DFT is good enough for all methods implemented in TURBOMOLE.
\texttt{statpt} automatically takes the best choice of the Hessian from the \texttt{control} file. For minimizations it first looks for the exact Hessian and then for the UFF Hessian. If none of them is found it takes the scaled unit matrix. For transition state optimization the exact Hessian has a higher priority than the results of LES.

The results of LES can be used to obtain an initial Hessian matrix for transition state optimizations involving large molecules, where calculation of the full Hessian is too expensive. Note, that LES calculations for \texttt{statpt}, in addition to the $\$les$ keyword require the following keywords to be added \textit{manually} in the \texttt{control} file:

\begin{verbatim}
$h0hessian$
$nomw$
\end{verbatim}

The default Hessian update for minimization is \texttt{bfgs}, which is likely to remain positive definite. The \texttt{powell} update is the default for transition state optimizations, since the Hessian can develop a negative curvature as the search progresses.

\section*{5.2.3 Finding Minima}

Simply specify the $\$statpt$ keyword in the \texttt{control} file and run \texttt{jobex} as explained above. You can very often speedup the optimization by calculating the initial Hessian matrix using \texttt{uff}.

\section*{5.2.4 Finding transition states}

Locating minima on a PES is straightforward. In contrast, transition state optimization requires much more input. The diagonal guess Hessian will almost never work, so you must provide a computed one. The Hessian should be computed at your best guess as to what the TS should be.

The real trick here is to find a good guess for the transition state structure. The closer you are, the better. It is often difficult to guess these structures. One way to obtain a good guess is to built an approximate TS and to perform a constrained minimization by freezing internal coordinates that change most during the reaction. Alternatively, you can generate several structures intermediate to reactants and products, and compute the energy at each point. The maximum energy structure is usually a good guess for the true TS.

After obtaining a reasonable initial guess for the TS structure you have to perform a vibrational analysis (or LES calculation for a large molecule) and to identify the index of the transition vector to follow during the optimization. Ideally, this is a vector with a negative eigenvalue, or "imaginary" frequency. The best way to find the right vector is to use some graphical interface to visualize vibrations. For a reasonable guess structure there should be one vibration that resembles the reaction under study. Remember that \texttt{statpt} uses a different ordering of eigenvalues as compared to the \texttt{aoforce} output—six (five) zero eigenvalues are shifted to the end.
CHAPTER 5. STRUCTURE OPTIMIZATIONS

There is an important thing to remember at this point. Even such sophisticated optimization methods like TRIM will not replace your own chemical intuition about where transition states may be located. If you need to restart your run, do so with the coordinates which have the smallest RMS gradient. Note that the energy does not have necessarily to decrease in a transition state search (as opposed to minimizations). It is sometimes necessary to do restart several times (including a recomputation of the Hessian) before the saddle point can be located.

Assuming you do find the TS, it is always a good idea to recompute the Hessian at this structure. It is fairly common, especially when using symmetry, that at your “TS” there is a second imaginary frequency. This means that you have not found the correct TS. The proper procedure is to distort the structure along the “extra” imaginary normal mode using the tool screwer (see Section 1.5). Very often such a distortion requires also lowering the point group symmetry. The distortion must be large enough, otherwise the next run will come back to the invalid structure.

5.3 Program Relax

5.3.1 Purpose

relax drives and controls a non-linear optimization procedure to locate the minimum (or a stationary point) of a function \( f(x) \). In TURBOMOLE \( f \) is always the electronic energy, and the coordinates \( x \) will be referred to as general coordinates. They include

- cartesian atomic coordinates
- internal atomic coordinates
- exponents, contraction coefficients and scaling factors of basis functions
- a global scaling factor (a common scaling factor for all basis set exponents)

The optimization employs an iterative procedure based on gradients \( \nabla f \) of the current and, if available, previous iterations. Various procedures can be applied: steepest descent, Pulay’s DIIS, quasi–Newton, conjugate gradients, as well as combinations of them. relax carries out:

- update of general coordinates
- update of approximate hessians if needed
- conversion of coordinates (internal \( \leftrightarrow \) cartesian)

The mode of operation is chosen by the keywords \$optimize and \$interconversion and the corresponding options, which will be described in the following sections.
5.3. PROGRAM RELAX

5.3.2 Optimization of General Coordinates

After gradients $G^k$ have been calculated for coordinates $q^k$ in optimization cycle $k$, new coordinates (or basis set exponents) $q^{k+1}$ can be obtained from the quasi-Newton update:

$$q^{k+1} = q^k - F^k G^k$$

where $F^k$ is the inverse of an approximate force constant matrix $H^k$. This method would immediately converge to the equilibrium geometry if $F^k$ would be the inverse of the exact force constant matrix and the force field would be quadratic. In real applications usually none of these requirements is fulfilled. Often only a crude approximation to the force constant matrix $H^k$ is known. Sometimes a unit matrix is employed (which means coordinate update along the negative gradient with all coordinates treated on an equal footing).

The optimization of nuclear coordinates in the space of internal coordinates is the default task performed by relax and does not need to be enabled. Any other optimization task requires explicit specifications in data group $optimize$, which takes several possible options:

$optimize$ options

- internal on/off Structure optimization in internal coordinates.
- redundant on/off Structure optimization in redundant coordinates.
- cartesian on/off Structure optimization in cartesian coordinates.
- basis on/off Optimization of basis set exponents, contraction coefficients, scaling factors.
- global on/off Optimization of global scaling factor for all basis set exponents.

Note: All options except internal are switched off by default, unless they have been activated explicitly by specifying on.

Some of the options may be used simultaneously, e.g.

- internal, basis
- internal, global
- cartesian, basis

Other options have to be used exclusively, e.g.

- internal, cartesian
- basis, global
CHAPTER 5. STRUCTURE OPTIMIZATIONS

The update of the coordinates may be controlled by special options provided in data group \$coordinateupdate which takes as options:

\[ dqmax = \text{real} \] Maximum total coordinate change (default: 0.3).

\[ \text{interpolate on/off} \] Calculate coordinate update by inter/extrapolation using coordinates and gradients of the last two optimization cycles (default: \text{interpolate on}) if possible.

\[ \text{statistics integer/off} \] Display optimization statistics for the integer previous optimization cycles. Without integer all available information will be displayed. off suppresses optimization statistics.

The following data blocks are used by program \text{relax}:

1. Input data from gradient programs \text{grad, rdgrad, egrad, rimp2, mpgrad}, etc.:
   \[ \$\text{grad} \] cartesian atomic coordinates and their gradients.
   \[ \$\text{egrad} \] exponents and scale factors and their gradients.
   \[ \$\text{globgrad} \] global scale factor and its gradient.

2. Input data from force constant program \text{aoforce}:
   \[ \$\text{grad} \] cartesian atomic coordinates and their gradients.
   \[ \$\text{globgrad} \] global scale factor and its gradient.
   \[ \$\text{hessian} \] the force constant matrix in the space of cartesian coordinates.

3. Output data from program \text{relax}:
   \[ \$\text{coord} \] cartesian atomic coordinates.
   \[ \$\text{basis} \] exponents and scale factors.
   \[ \$\text{global} \] global scale factor.

For structure optimizations the use of (redundant) internal coordinates is recommended, see Section 4.0.6. Normally internal coordinates are not used for input or output by the electronic structure programs (\text{dscf, mpgrad}, etc.). Instead the coordinates, gradients, etc. are automatically converted to internal coordinates by \text{relax} on input and the updated positions of the nuclei are written in cartesi ans coordinates to the data group \$\text{coord}. Details are explained in the following sections.

5.3.3 Force Constant Update Algorithms

In a Newton-type geometry update procedure often only a crude approximation to the force constant matrix \( H^k \) is available. What can be done then is to update \( F^k = (H^k)^{-1} \) in each iteration using information about previous coordinates and gradients. This constitutes the quasi–Newton or variable metric methods of which there are a few variants:
1. Murtagh/Sargent (MS):

\[ F^k = F^{k-1} + Z^{k-1}(Z^{k-1})^\dagger \frac{dG^{k-1}}{(Z^{k-1})^\dagger dG^{k-1}} \]

2. Broyden/Fletcher/Goldfarb/Shanno (BFGS):

\[ F^k = F^{k-1} + \frac{S(dq^{k-1})^\dagger dq^{k-1} - dq^{k-1}(dG^{k-1})^\dagger F^{k-1} - F^{k-1}dG^{k-1}(dq^{k-1})^\dagger}{S1} \]

3. Davidon/Fletcher/Powell (DFP):

\[ F^k = F^{k-1} + \frac{(dq^{k-1})^\dagger dq^{k-1} - F^{k-1}dG^{k-1}(dq^{k-1})^\dagger F^{k-1}}{(S-1)S1} \]

4. combined method (BFGS/DFP): If \( S1 < (S-1)S1 \) and \( S1 > 0 \) perform DFP update, otherwise BFGS.

The meaning of the symbols above is as follows:

- \( F^k \): approximate inverse force constant matrix in the k-th iteration.
- \( q^k \): general coordinates in the k-th iteration.
- \( G^k \): gradients in the k-th iteration.
- \( dq^{k-1} = q^k - q^{k-1} \)
- \( dg^{k-1} = g^k - g^{k-1} \)
- \( Z^{k-1} = dq^{k-1} - F^{k-1}dG^{k-1} \)
- \( S1 = (dq^{k-1})^\dagger dq^{k-1} \)
- \( S = 1 + ((dg^{k-1})^\dagger F^{k-1}dG^{k-1})/(S1) \)

An alternative is to use update algorithms for the hessian \( H^k \) itself:

Ehrig, Ahlrichs: *Diagonal* update for the hessian by means of a least squares fit

\[ H^{k,i}_i = \sqrt{H_i^{k-1}(h_i + d_i)} \]

with the new estimate \( h \) for the diagonal elements obtained by

\[ h_i = \frac{\sum_k G_k^i dq_k^i}{\sum_k (dq_k^i)^2} \]

and the error \( d \) obtained by the regression

\[ d_i = \frac{\sqrt{\sum_k (dq_k^i)^2} - h_i^2}{k - 2} . \]
Another alternative is to use DIIS-like methods: structure optimization by direct inversion in the iterative subspace. (See ref. [37] for the description of the algorithm). The DIIS procedure can often be applied with good success, using static or updated force constant matrices.

Any of the algorithms mentioned above may be chosen. Recommended is the macro option **ahlrichs**, which leads to the following actions ($n$ is the maximum number of structures to be included for the update, default is $n = 4$):

- $\text{ncycles} < n$: geometry update by inter/extrapolation using the last 2 geometries.
- $\text{ncycles} \geq n$: diagonal update for the hessian as described above; DIIS–like update for the geometry.
- $||G|| < \text{thr}$: BFGS-type update of the hessian and quasi–Newton update of (generalized) coordinates.

References for the algorithms mentioned above: [34, 37–41]

### 5.3.4 Definition of Internal Coordinates

If structure optimizations are to be performed in the space of internal coordinates ($\texttt{optimize internal}$, is the default setting), appropriate internal coordinate definitions have to be provided on data block $\texttt{intdef}$. The types available and their definitions are described in Section 4.1.2. For recommendations about the choice of internal coordinates consult ref. [28]. Nevertheless the structure of $\texttt{intdef}$ will shortly be described. The syntax is (in free format):

```
1  k 1.00000000  bend 1 2 3  val=1.9500  fdiag=.6666
```

The first items have been explained in Chapter 4.

Two additional items $\texttt{val=}\text{real}$, $\texttt{fdiag=}\text{real}$ may be supplied for special purposes:

- $\text{val=}$ serves for the input of values for internal coordinates for the interconversion internal → cartesian coordinates; it will be read in by **relax** if the flag for interconversion of coordinates has been activated ($\texttt{interconversion on}$), or by the interactive input program **define** within the geometry specification menu.

- $\text{fdiag=}$. Serves for the input of (diagonal) force constants for the individual internal coordinates to initialize $\texttt{forceapprox}$.

### 5.3.5 Structure Optimizations Using Internal Coordinates

This is the default task of **relax** ($\texttt{optimize internal on}$ does not need to be specified!) You need as input the data groups:

- **intdef**
$grad$ cartesian coordinates and gradients as provided and accumulated in subsequent optimization cycles by the programs grad, or rdgrad etc.

$intdef$ definitions of internal coordinates.

$redundant$ definitions of redundant coordinates.

Output will be the updated coordinates on $coord$ and the updated force constant matrix on $forceapprox$. If any non-default force constant update option has been chosen, relax increments its counting variables $<\text{numgeo}>$, $<\text{numpul}>$ within command keyword $forceupdate$. If the approximate force constant has been initialized ($forceinit$ on ) relax switches the initialization flag to $forceinit$ off. Refer also to the general documentation of TURBOMOLE. It is recommended to check correctness of your definition of internal coordinates:

1. Calculate their values for your cartesian start coordinates using the relax program (see Section 5.3.11) or within a define session.

2. Have a look at the eigenvectors of the $B\text{m}B^\dagger$-matrix. Set some ‘?’ behind keyword $intdef$, if there are any eigenvalues close to zero ($< 10^{-2}$ is to be considered bad for small molecules, but there is no general rule) check those internal coordinates for consistency which contribute to the corresponding eigenvector(s)!

5.3.6 Structure Optimization in Cartesian Coordinates

For this task you have to specify:

$optimize$

```plaintext
cartesian on
internal off
```

These lines switch on the non-default optimization in cartesian coordinates and switch off the optimization in internal coordinates (this has to be done explicitly!). As input data groups you need only $grad$ as provided by one of the gradient programs. For the first coordinate update an approximate force constant matrix is needed in data group $forceapprox$. Output will be the updated coordinates on $coord$, and the updated force constant matrix on $forceapprox$.

The coordinates for any single atom can be fixed by placing an 'f' in the third to eighth column of the chemical symbol/flag group. As an example, the following coordinates specify acetone with a fixed carbonyl group:

$coord$

```plaintext
2.02693271108611 2.03672551266230 0.00000000000000 c
1.08247228252865 -0.68857387733323 0.00000000000000 c f
2.53154870318830 -2.48171472134488 0.00000000000000 o f
```
5.3.7 Optimization of Basis Sets (SCF only)

For this task you have to specify:

```
$optimize
  basis on
  internal off
```

This example would perform only a basis set optimization without accompanying geometry optimization. It is possible, of course, to optimize both simultaneously: Just leave out the last line of the example (internal off). Input data groups are:

- `$egrad` Basis set exponents, contraction coefficients, scaling factors and their respective gradients as provided and accumulated in subsequent optimization cycles by one of the programs grad or mpgrad, if $drvopt basis on has been set.

- `$basis` Description of basis sets used, see Section 4.2.

Output will be the updated basis on `$basis`, and the updated force constant matrix on `$forceapprox`.

For an example, see Section 22.5.

5.3.8 Simultaneous Optimization of Basis Set and Structure

The optimization of geometry and basis set may be performed simultaneously and requires the specification of:

```
$optimize
  internal on (or: cartesian on)
  basis on
```

and needs as input data groups `$grad` and `$egrad`. Output will be on `$coord`, `$basis`, also on `$forceapprox` (updated).
5.3.9 Optimization of Structure and a Global Scaling Factor

Optimization of a global scaling factor is usually not performed in geometry optimizations. It is a special feature for special applications by even more special users. As reference see [42].

To optimize the structure and a global scaling factor specify:

```plaintext
$optimize
  internal on (or: cartesian on)
  global on
```

You need as input data groups `$grad` and `$globgrad`, the latter contains the global scaling factors and their gradients accumulated in all optimization cycles. Output will be on `$coord`, `$global`, also on `$forceapprox` (updated). Note that for optimization of a global scaling factor a larger initial force constant element is recommended (about 10.0).

5.3.10 Conversion from Internal to Cartesian Coordinates

Due to translational and rotational degrees of freedom and the non-linear dependence of internal coordinates upon cartesian coordinates, there is no unique set of cartesian coordinates for a given set of internal coordinates. Therefore an iterative procedure is employed to calculate the next local solution for a given cartesian start coordinates. This task may be performed using the relax program, but it is much easier done within a define session.

5.3.11 Conversion of Cartesian Coordinates, Gradients and Force Constants to Internals

To perform this tasks, you have to activate the interconversion mode by

```plaintext
$interconversion on
  cartesian --> internal    coordinate gradient hessian
```

Note that any combination of the three options showed is allowed! The default value is `coordinate`, the two other have to be switched on explicitly if desired.

You need as input data groups:

- `intdef` Definitions of (redundant) internal coordinates
- `coord` Cartesian coordinates (for option ‘coordinate’)
- `grad` Cartesian coordinates and gradients as provided and accumulated in subsequent optimization cycles by the various gradient programs (for coordinate and gradient)
hessian Analytical force constant matrix (as provided by the force constant program aoforce) (only if option hessian is specified). The data group $hessian (projected) may be used alternatively for this purpose.

All output will be written to the screen except for option hessian (output to data group $forceapprox)

5.3.12 The m-Matrix

The m-matrix serves to fix position and orientation of your molecule during geometry optimizations. It cannot be used to fix internal coordinates! The m-matrix is a diagonal matrix of dimension $3n^2$ (where $n$ is the number of atoms). Normally $m$ will be initialized as a unit matrix by relax. As an example consider you want to restrict an atom to the xy-plane. You then set the $m(z)$–matrix element for this atom to zero. You can use at most six zero m-matrix diagonals (for linear molecules only five)—corresponding to translational and rotational degrees of freedom. Note that the condition of the $B_mB^\dagger$-matrix can get worse if positional restrictions are applied to the m-matrix. m-matrix elements violating the molecular point group symmetry will be reset to one. Non-default settings for m-matrix diagonals of selected atoms have to be specified within data group $m$-matrix as:

```
$m$-matrix
  1  0.0  0.0  0.0
 10 1.0  0.0  0.0
 11 1.0  1.0  0.0
```

5.3.13 Initialization of Force Constant Matrices

The most simple initial hessian is a unit matrix. However, better choices are preferable. For structure optimizations using internal coordinates you may use structural information to set up a diagonal force constant matrix with elements chosen in accord to the softness or stiffness of the individual modes. For detailed information refer to ref. [40]. For optimization of basis set parameters less information is available. When neither data block $forceapprox$ is available nor $forceinit on$ is set, the force constant matrix will be initialized as a unit matrix. Specifying the force constant initialization key $forceinit on diag=... will lead to:

```
diag=real Initialization with real as diagonal elements.
diag=default Initial force constant diagonals will be assigned the following default values:
```
5.4. FORCE FIELD CALCULATIONS

internal coordinates : stretches 0.50
angles 0.20
scaling factors : s,p 1.50
d 3.00
exponents : uncontracted 0.15
contracted 10.00
contraction coefficients : 100.00
global scaling factor : 15.00
cartesian force constants : 0.50

diag=individual Initial force constant diagonals will be taken from
$intdef fdiag=... or
$global fdiag=...
Similar initialization modes are NOT supported for geometry
optimization in cartesian space and for the optimization of basis
set parameters!
carthess Data group $hessian (projected) is used.

5.3.14 Look at Results

The energy file includes the total energy of all cycles of a structure optimization
completed so far. To get a display of energies and gradients use the UNIX command
grep cycle gradient which yields, e.g. H\textsubscript{2}O.

cycle = 1 SCF energy = -76.3432480651 |dE/dxyz| = 0.124274
cycle = 2 SCF energy = -76.3575482860 |dE/dxyz| = 0.082663
cycle = 3 SCF energy = -76.3626983371 |dE/dxyz| = 0.033998
cycle = 4 SCF energy = -76.3633251080 |dE/dxyz| = 0.016404
cycle = 5 SCF energy = -76.3634291559 |dE/dxyz| = 0.010640
cycle = 6 SCF energy = -76.3634910117 |dE/dxyz| = 0.000730

This should be self-evident. To see the current—or, if the optimization is con-
verged, the final—atomic distances use the tool dist. Bond angles, torsional an-
gles etc. are obtained with the tools bend, tors, outp, etc. In the file gradient
are the collected cartesian coordinates and corresponding gradients of all cycles.
The values of the general coordinates and corresponding gradients are an output of
relax written to job.<cycle> of job.last within jobex. To look at this search for
'Optimization statistics' in job.last or job.<cycle>.

5.4 Force Field Calculations

5.4.1 Purpose

uff preoptimizes a structure and calculates an analytical Hessian which can be used
as a start Hessian in a geometry optimization. This will accelerate the convergence
of an optimizations. For optimizations in cartesian space this will be faster by a factor of two for any molecule.

### 5.4.2 How to Perform a Uff Calculation

You have to generate cartesian coordinates (file `coord`), nothing else. You can start an single-point calculation calculation by typing

```
uff
```

To start an **uff** geometry optimization, one has to change the number of cycles (parameter `maxcycle`) in the block `$uff` in the file `control`. The output is the optimized structure (file `coord`), the analytical gradient (file `uffgradient`) and the analytical cartesian hessian (file `uffhessian0-0`). Furthermore the `control` file will be modified:

```
$forceinit on
carthess
$uffhessian file=uffhesian0-0
```

These commands have the effect to initialize the force constant matrix for a geometry optimization with the hessian one.

In some cases **uff** cannot recognize the connectivity, then one can specify the connectivity in the file `ufftopology`. The program will calculate the bond, angle, torsion, inversion and non-bonded terms (force field terms) based on the connectivity specified in the topology file.

### 5.4.3 The Uff implementation

The **uff** implementation follows the paper by Rappé [7]. The energy expression in **uff** is as follows:
5.4. FORCE FIELD CALCULATIONS

\[ E_{UFF} = \sum_{I<J}^{N_B} \frac{1}{2} K_{IJ} \cdot (r - r_{IJ})^2 \]  
(5.1)

\[ + \sum_{I<J}^{N_A} \left\{ \begin{array}{l} K_{IJ}^4 (1 - \cos(2\theta)) : \text{linear case} \\ K_{IJ}^3 (1 - \cos(3\theta)) : \text{trigonal planar case} \\ K_{IJ}^2 (1 - \cos(4\theta)) : \text{quadratic planar case} \\ K_{IJ} (1 - \cos(4\theta)) : \text{octahedral case} \end{array} \right. \\
K_{IJK} \cdot (C_A^A + C_1^A \cos \theta + C_2^A \cos(2\theta)) : \text{general case} \]

\[ + \sum_{I}^{N_T} \frac{1}{2} V_{\phi} \cdot (1 - \cos (n\phi_0) \cos(n\phi)) \]

\[ + \sum_{I}^{N_I} V_{\omega} \cdot (C_0^I + C_1^I \cos \omega + C_2^I \cos 2\omega) \]

\[ + \sum_{I<J}^{N_{nb}} D_{IJ} \cdot \left( -\frac{1}{6} \left( \frac{x_{IJ}}{x} \right)^6 + \left( \frac{x_{IJ}}{x} \right)^{12} \right) \]

\[ + \sum_{I<J}^{N_{nb}} q_I \cdot q_J \left( \frac{1}{\epsilon} \cdot \frac{1}{x} \right) \]

The Fourier coefficients \( C_0^A, C_1^A, C_2^A \) of the general angle terms are evaluated as a function of the natural angle \( \theta_0 \):

\[ C_2^A = \frac{1}{4 \sin^2 \theta_0} \]  
(5.2)

\[ C_1^A = -4 \cdot C_2^A \cos \theta_0 \]  
(5.3)

\[ C_0^A = C_2^A \left( 2 \cos^2 \theta_0 + 1 \right) \]  
(5.4)

The expressions in the energy term are:

\( N_B, N_A, N_T, N_I, N_{nb} \) the numbers of the bond-, angle-, torsion-, inversion- and the non bonded-terms.

\( K_{IJ}, K_{IJK} \) force constants of the bond- and angle-terms.

\( r, r_{IJ} \) bond distance and natural bond distance of the two atoms \( I \) and \( J \).

\( \theta, \theta_0 \) angle and natural angle for three atoms \( I - J - K \).

\( C_0^A, C_1^A, C_2^A \) Fourier coefficients of the general angle terms.

\( \phi, \phi_0 \) torsion angle and natural torison angle of the atoms \( I - J - K - L \).

\( V_{\phi} \) height of the torsion barrier.

\( n \) periodicity of the torsion potential.

\( \omega \) inversion- or out-of-plane-angle at atom \( I \).
V_\omega \quad \text{height of the inversion barrier.}

C_i^0, C_i^1, C_i^2 \quad \text{Fourier coefficients of the inversions terms.}

x, x_{IJ} \quad \text{distance and natural distance of two non bonded atoms I and J.}

D_{IJ} \quad \text{depth of the Lennard–Jones potential.}

q_I, \epsilon \quad \text{partial charge of atoms I and dielectric constant.}

One major difference in this implementation concerns the atom types. The atom types in Rappé’s paper have an underscore "_". In the present implementation an sp^3 C atom has the name "C 3" instead of "C_3". Particularly the bond terms are described with the harmonic potential and the non-bonded van der Waals terms with the Lennard–Jones potential. The partial charges needed for electrostatic nonbond terms are calculated with the Charge Equilibration Modell (QEq) from Rappé [43]. There is no cutoff for the non-bonded terms.

The relaxation procedure distinguishes between molecules with more than 90 atoms and molecules with less atoms. For small molecules it consists of a Newton step followed by a linesearch step. For big molecules a quasi-Newton relaxation is done. The BFGS update of the force-constant matrix is done [38, 44–46]. Pulay’s DIIS procedure is implemented for big molecule to accelerate the optimization [37, 47].

The coordinates for any single atom can be fixed by placing an ‘f’ in the third to eighth column of the chemical symbol/flag group. As an example, the following coordinates specify acetone with a fixed carbonyl group:

```
$coord
  2.02693271108611  2.03672551266230  0.00000000000000  c
  1.08247228252865  -0.68857387733323  0.00000000000000  c f
  2.53154870318830  -2.48171472134488  0.00000000000000  o f
 -1.78063790034738  -1.04586399389434  0.00000000000000  c
 -2.64348282517094  -0.13141435997713  1.68855816889786  h
 -2.23779643042546  -3.09026673535431  0.00000000000000  h
 -2.64348282517094  -0.13141435997713  -1.68855816889786 h
  1.3100893646566  3.07002878668872  1.68840815751978  h
  1.3100893646566  3.07002878668872  -1.68840815751978 h
  4.12184425921830  2.06288409251899  0.00000000000000  h
$end
```

5.5 Molecular Dynamics Calculations

*Ab initio* molecular dynamics (MD) can be carried out on the ground and excited state Born–Oppenheimer potential hypersurface. In addition non-adiabatic Tully-type *Surface Hopping* MD can be performed using TDDFT. At the start of an MD run the user must specify the initial atomic positions and velocities and give some
general instructions for the run. This is managed by running the interactive pro-
gram \texttt{Mdprep} and generating the command file \texttt{mdmaster}. If this is successful, the
MD run itself may be started: \texttt{jobex -md}. Time is then advanced in steps. The
electronic potential energy and its gradients are calculated quantum mechanically at
the required coordinates each timestep (as detailed above, e.g. \texttt{dscf} and \texttt{grad}). The
MD program \texttt{frog} uses the Leapfrog Verlet algorithm \cite{48} to turn the gradients into
new atomic positions and velocities. The atoms thus undergo classical Newtonian
dynamics on the \textit{ab initio} potential hypersurface. Trajectory information is recorded
in a log file (\texttt{mdlog}). It is possible to instruct \texttt{frog} to heat or cool the system, use
a thermostat for canonical dynamics, conserve total energy or read in new positions
or velocities: the appropriate keywords are described in Section 21.2.23 below.
5.6 Counterpoise-Corrections using the JOBBSSE Script

The shell script `jobbsse` controls and executes the automatic calculation of the counterpoise correction as it has been formulated by Boys and Bernadi (S. F. Boys and F. Bernadi, Mol. Phys., 19, 553 (1970)) to estimate the Basis Set Superposition Error (BSSE). For a dimer, the cp-correction takes the form for the monomers A and B:

\[ E_{AB}^{CP} = E_{AB} - (E_{A(B)} - E_A) - (E_{B(A)} - E_B) \]

Where parentheses denote ghost basis sets without electrons or nuclear charges. For a trimer `jobbsse` used by default the conventional so-called site-site functional counterpoise corrections:

\[ E_{ABC}^{CP} = E_{ABC} - (E_{A(BC)} - E_A) - (E_{B(AC)} - E_B) - (E_{C(AB)} - E_C) \]

`jobbsse` works similar as the `jobex` script: it cycles through the SCF/DFT and, if needed, gradient and force relaxation programs and stops if either the maximum number of cycles is reached or the convergence criteria (change in the total energy, maximum norm of the gradient) are fulfilled. It does either only energy calculations or a full geometry optimization including up to three fragments. By default, the executable programs are taken from the load modules library within the TURBOMOLE directory.

Note that you need to set up the fragments (and possibly their symmetries using `define` in the geometry menu beforehand. The general structure of a `jobbsse` calculation is as follows:

1. `bsseenergy` is invoked to generate input files for `define`, which is then used to prepare the control files (including occupation numbers, initial guess MOs, etc.) for the different “ghost“ and monomer calculations and shell scripts with commands for calculations on these fragments.

2. `jobbsse` cycles over the supermolecular complex and the fragments and computes the energies and, if requested, gradients for them. Then the counterpoise-corrected results are evaluated and written to the standard data groups ($energy and $grad).

3. For geometry optimizations one of the structure relaxation codes (`statpt` or `relax`) is invoked to update the coordinates and check for convergence. If the structure optimization is not converged `jobbsse` continues with the previous step.

Note, that counterpoise-corrected calculations with `jobbsse` are NOT as black-box as ordinary geometry optimizations with `jobex`. The input generated for the fragments are based on the default occupation numbers obtained from the EHT guess, default assignments for the frozen orbitals, memory, etc. Since this might be different from what is needed (or even fail), it is recommended to let `jobbsse` stop after the initial setup step using the flag `-setup` and to check carefully the assigned basis sets,
occupations number and subsystem symmetries. In particular, for MP2 or CC2 calculations with molecules containing not only the atoms H–Ar also the number of frozen orbitals should be checked, and if necessary corrected.

5.6.1 Options

Given a shell the usage is:

```
nohup jobbsse &
```

This command invokes cp-correction, and, if needed structure optimization using the default program `statpt`. Note, that the program needs to know which calculation is being done. Structure optimizations using program `relax` can be performed using `-relax` flag:

```
nohup jobbsse -opt -relax &
```

`nohup` means that the command is immune to hangups, logouts, and quits. `&` runs a background command. `jobbsse` accepts the following arguments controlling the level of calculation, convergence criteria and many more (for example `nohup jobbsse -gcart 4 &`):

- **-energy** integer converge total energy up to $10^{(-<\text{integer}>)}$ Hartree (default: 6)
- **-gcart** integer converge maximum norm of cartesian gradient up to $10^{(-<\text{integer}>)}$ atomic units (default: 3)
- **-c** integer perform up to integer cycles (default: 100)
- **-gradient** calculate the gradient as well
- **-opt** optimise the structure
- **-relax** use the `relax` program for force relaxation
- **-level** level define the optimization level, `level=scf, dft, mp2, or cc2` (default is `scf`). Note that the program needs this input! If the level is DFT, the grid will be automatically set to m4.
- **-ri** use RI modules `ridft` and `rdgrad` (fast Coulomb approximation) instead of `dscf` and `grad` as well as `rimp2` instead of `mpgrad`
- **-l** <path> employ programs from directory <path>
- **-mem** integer Is able to control the memory from outside `define` Note that if you did not define any memory, it is automatically set to 1 GB
-trimer calculates, in case we have a trimer:
\[ \text{Energy} = ABC - AB(C) + AB - AC(B) + AC - BC(A) + BC \]
rather than
\[ \text{Energy} = ABC - A(BC) + A - B(AC) + B - C(AB) + C \]
(note that the first term neglects the BSSE in the dimer)

-setup Interrupt calculation after the initial setup step to check and possibly correct the control files for the fragments and the supermolecule. To continue, start jobbsse without the -setup option.

-help shows a short description of the commands above

5.6.2 Output

There will be an output written to file bsse_out. In this file, you will find all individual energies computed which were used to calculate the last cp-corrected energy. The same holds true for the last gradients, which are written to grad_out.

The convergence criteria and their current values are written out at the not.converged file. For the possible options to control convergence check the subsection for the optimization program used (statpt, which is used by default, or relax). Since for weak complexes the force constants for intra- and intermolecular bonds very strongly in magnitude, it is recommended to use whenever possible redundant internal coordinates.
5.7 Reaction Path Optimization

5.7.1 Background and Program structure

The goal of self-consistent optimization of the reaction path (RP) is usually to obtain an initial guess for Transition State Search or an approximation to the barrier. Methods that use reactant and product structure to compute the RP are often referred to as 'double-ended' methods or, if the RP is discretized, 'chain-of-states' methods. [49–51]

The RP connects reactant and product, its highest point being the transition state. It is a steepest descent path, which means that its tangent \( t \) is always parallel to the gradient \( g \). The RP is in actual calculations discretized by a finite number of structures \( n \). The tangents are parallel to the gradients for all structures \( i = 1, \ldots, n \).

Assuming normalized tangents \( t_i^T t_i = 1 \), this can be written as:

\[
0 = (1 - t_i^T t_i) g_i
\]  

(5.5)

Several approximations for the tangents \( t_i \) exist [51, 52], usually using finite difference schemes. The most common methods, the Nudged Elastic Band (NEB) [51, 52] and String Method (SM) [53] prevent the structures from ‘sliding’ down the reaction path towards products and reactants with additional springs or interpolation/redistribution algorithms. The method used here achieves equal spacing via constrained optimization assuming a quadratic potential. [54] An initial path is provided using a slight variation of the Linear Synchronous Transit [49].

The structure of the optimization is the same as in other TURBOMOLE structure optimizations: As the jobex script drives optimizations by calling statpt/relax as well the SCF and gradient modules. The woelfling-job script drives optimizations by calling woelfling as well as the SCF and gradient modules. The woelfling-job scripts reads the current path from file path.xyz which is the output of the woelfling program. woelfling-job then creates folders to run the calculations of each structure in, gathers coordinates and gradients, and then calls woelfling again.

The aim of RP optimization is usually not to optimize the RP to some accuracy, but to obtain an initial guess for a TS optimization. It is in general not possible to find a convergence criterion (and a corresponding threshold) that guarantees a good initial guess. The maximum number of iterations and the convergence threshold are therefore relatively high and tight. One can extract a TS guess also during the course of the optimization. If the TS search is successful (or not) you can stop (or restart) the RP optimization. Apart from simply killing the program you can add a ’stop’ file in the (scratch) directory, in which the script runs. It will then terminate at the end of the current cycle and can easily be restarted.

5.7.2 Input Structure

Options can be modified using keywords in the $woelfling data-group. The most important options are:
The values above are the default values. If $\texttt{woelfling}$ is missing, it will be added during the first \texttt{woelfling} run and default values will be set. Most importantly, \texttt{ncoord} is the number of input structures provided, \texttt{ninter} is the number of structures to be used for discretization of the path and \texttt{maxit} is the number of cycles to run. If \texttt{maxit} 0, structures will be rotated/translated to minimize the cartesian distance, for \texttt{maxit} 1 structures will be used as provided. Using \texttt{method qg} instead of \texttt{method q}, a reaction path will be grown as in the growing string method. To start a RP optimization you need to provide at least a reactant and a product structure \texttt{ncoord} $\geq 2$. You may provide more structures if you have a guess for the reaction path. The input structures will be used to compute an initial guess with \texttt{ncoord} structures that is then optimized. Reactant and product structure will stay fixed throughout the optimization. All structures \textit{have to have} the same ordering of atoms!

The input structures are provided in a file \texttt{coords}, which contains merged coord files. All \texttt{ncoord} structures are given in the right order in the typical TURBOMOLE coord format.

### 5.7.3 How it works

#### Minimum Input/Quick and Dirty

1. Make a usual TURBOMOLE input using the coord file of either reactant of product structure.

2. Join the coord-files of reactant and product in a file \texttt{coords}.

3. Run \texttt{woelfling-job}

4. Check the output and the path (path.xyz) to extract a TS guess.

It is usually a good idea to check the initial path before starting the calculation. Once you have prepared the input, simply run \texttt{woelfling} directly and check \texttt{path.xyz}. If it looks reasonable, just run the \texttt{woelfling-job} script.

#### Unsuccessful Optimization

If there is no reasonable TS guess or a frequency calculation does not give the correct number of imaginary frequencies, you can:
5.7. REACTION PATH OPTIMIZATION

1. Check if have used the best structure as TS guess, maybe the structure with the highest energy is not the best.

2. Check if the RP has a reasonable amount of structures (if they are far apart, it is unlikely that a structure is close to the TS)

3. Check if the RP is reasonably converged (mean of $\text{rms}(g^i)$ in output $< 1.0d-3$; path is continuous in terms of energy and structure)
   
   (a) If it is not yet converged, converge it.
   
   (b) If it is not going to converge, provide useful structures for the initial guess or maybe use more structures for the path.

Restart

1. If you have stopped the calculation adding a ‘stop’ file, you can just run woelfling-job again.

2. If you have run the maximum number of cycles, just increase maxit and run woelfling-job again. It will then run from the old maxit to the new maxit.

If the calculation crashed ‘on its own’ it is likely that the SCF failed to converge - improve the corresponding options. If the optimization crashed in the middle of the run, it is most likely that it crashed during a SCF or gradient step, since basically all cpu time is spent there. In that case, remove at least the folder where the SCF and gradient program had been running when the program crashed. The files necessary for woelfling should be intact and you can simply restart it.

Restart - more details

If the files are not intact, one can still use the optimized coordinates in one way or the other. There are basically three phases which are explicitly indicated by the number of iterations $riter =$:

0 woelfling reads control, coords to generate an initial guess path.xyz

1 woelfling reads control, gradients to compute an optimized path.xyz. Hessian are initialized and written to hessians-new

>1 woelfling reads control, gradients, oldgradients and hessians to update hessians and compute an optimized path.xyz. Hessian are updated and written to ‘hessians-new’.

Therefore repeated execution of woelfling will yield the same output (unless new energies/gradients are computed). The woelfling-job script takes care of the file-handling and should enable restart at any time. If files are damaged it will be hard to gather gradients and corresponding hessian- and oldgradient-information. If you have an intact gradients or oldgradients-file (necessary condition: number of lines=$(3 + 2 \times \text{natoms}) \times \text{number of structures}$), name it gradients, set $riter$ in the
control-file to 1 and restart. If those file are not intact, you can extract whatever structure information you have obtained and use it to provide a better initial guess: Modify file coords and ncoord in the control-file accordingly, set riter to 0 and restart.
Energy and gradient calculations at the Hartree–Fock (HF) and DFT level can be carried out in two ways: \texttt{dscf} and \texttt{grad} perform conventional calculations based on four–center two–electron repulsion integrals (ERI’s); \texttt{ridft} and \texttt{rdgrad} employ the RI–J approximation, as detailed below.

\texttt{dscf} and \texttt{grad} are modules for energy and gradient calculations at the HF and DFT level, which use an efficient semi–direct SCF algorithm. Calculation of the Coulomb and HF exchange terms is based on the conventional method employing four–center two–electron repulsion integrals (ERI’s). These modules should be used for HF and DFT calculations with exchange-correlation functionals including HF exchange contribution, e.g. B3–LYP, if further approximations (RI–J) are to be avoided. All functionalities are implemented for closed–shell RHF and open–shell UHF reference wavefunctions. Restricted open shell treatments (ROHF) are supported on the HF level only, i.e. not for DFT.

The most important special features of the \texttt{dscf} and \texttt{grad} modules are:

- Selective storage of the most time consuming and frequently used integrals. The integral storage is controlled by two threshold parameters, \texttt{$thize$} and \texttt{$thime$}, related to integral size and computational cost.
- Efficient convergence acceleration techniques for energy calculations. They include standard methods for convergence acceleration (DIIS), which reduce the number of SCF iterations needed as well as methods to reduce the effort within each iteration when the calculation is almost converged (integral prescreening and differential density scheme).
ridft and rdgrad are modules for very efficient calculation of energy and gradient at the Hartree–Fock (HF) and DFT level [55]. Both programs employ the Resolution of the Identity approach for computing the electronic Coulomb interaction (RI–J). This approach expands the molecular electron density in a set of atom–centered auxiliary functions, leading to expressions involving three–center ERI's only. This usually leads to a more than tenfold speedup for non–hybrid DFT compared to the conventional method based on four–center ERI's (for example the dscf or grad module).

The combination of RI–J for Coulomb–interactions with a case–adapted conventional exchange treatment reduces the scaling behaviour of the (conventional) exchange evaluation required in HF–SCF and hybrid DFT treatments. Usage of ridft and rdgrad for HF and hybrid DFT is of advantage (as compared to dscf and grad) for larger systems, where it reduces computational costs significantly.

The most important special features of the ridft and rdgrad modules are:

- A very efficient semi-core algorithm for energy calculation. The most expensive three–center integrals are kept in memory which significantly reduces the computational time for small and middle sized molecules. The amount of stored integrals is controlled by simply specifying the amount of free memory using the keyword $ricore$.

- Multipole accelerated RI for Coulomb (MARI–J) linear scaling ($O(N)$) method for large molecules. It significantly reduces calculation times for molecules with more than 1000 basis functions.

All algorithms implemented in dscf, grad, ridft, and rdgrad modules can exploit molecular symmetry for all finite point groups. Typically, the CPU time is proportional to $1/N_G$, where $N_G$ is the order of the nuclear exchange group. Another important feature is a parallel implementation using the MPI interface.

Additionally dscf and ridft modules include the following common features:

- An UHF implementation [56] with automatic generation of optimal start vectors by solving the HF instability equations [57] in the AO basis (see the keyword $scfinstab$ for detailed information).

- Occupation number optimization using (pseudo–Fermi) thermal smearing.

RI-techniques can also be used for the Hartree–Fock exchange part of the Fock matrix (RI-HF). This is done by the ridft-module, if the keyword $rik$ is found in the control file. In this case ridft performs a Hartree–Fock-SCF calculation using the RI-approximation for both $J$ and $K$, if suitable auxiliary basis sets (which differ from that used for fitting of the Coulomb part only) are specified. This is efficient only for comparably large basis sets like TZVPP, cc-pVTZ and larger.

HF-exchange can also be calculated semi-numerically [58]. The calculation of 4c-2e-Integrals is split into an analytical and a numerical part. The latter is evaluated on a dft-type integration grid. The semi-numerical calculation scales better with system size than RIK and is suitable for large molecules and large basis sets.
Prerequisites

Both dscf and ridft require the control file and starting orbitals obtained from the extended Hückel guess using define.

Energy calculations using dscf can be performed in a direct or semi-direct mode. In the direct mode all four-center ERI’s are recalculated at each SCF iteration. The semi-direct mode uses a selective storage of the most time consuming and frequently used integrals. The amount of integrals stored is controlled by the keywords $thize and $thime, related to integral size and computational cost. The semi-direct mode requires a separate dscf statistics run to estimate the disk space needed for integral storage. The statistics run requires the keyword $statistics dscf to be present in the control file. It can be set either manually or using the tool Stati.

For ridft and rdgrad following additional prerequisites are required:

1. An auxiliary basis defined in the data group $jbas. This group is created automatically when using ri menu of define.
2. The maximum core memory the program is allowed to allocate should be defined in the data group $ricore; the recommended value is 75–85\% of the available (physical) core memory.
3. Calculations using MARI-J method require the keyword $marij.
4. For RI-HF-calculations auxiliary bases defined in the data group $jkbas are needed. This group is created by the rijk menu in define.

How to Perform a Calculation

Single point calculations

Call the dscf or ridft program after running define.

Geometry optimizations and molecular dynamics

For HF or DFT calculations using dscf and grad simply invoke jobex. For DFT calculations using ridft and rdgrad type jobex -ri; see Section 5.1 for additional options and parameters for geometry optimizations and ab initio molecular dynamics calculations.

6.1 Background Theory

In Hartree–Fock theory, the energy has the form,

\[ E_{HF} = h + J - K + V_{nuc}, \]

where \( h \) is the one-electron (kinetic plus potential) energy, \( J \) is the classical Coulomb repulsion of the electrons, \( K \) is the exchange energy resulting from the quantum (fermion) nature of electrons, and \( V_{nuc} \) is the nuclear repulsion energy.
In density functional theory, the exact Hartree–Fock exchange for a single determinant is replaced by a more general expression, the exchange-correlation functional, which can include terms accounting for both exchange energy and the electron correlation which is omitted from Hartree–Fock theory. The DFT energy is expressed as a functional of the molecular electron density \( \rho(r) \),

\[
E_{\text{DFT}}[\rho] = T[\rho] + V_{\text{ne}}[\rho] + J[\rho] + E_x[\rho] + E_c[\rho] + V_{\text{nuc}},
\]

(6.2)

where \( T[\rho] \) is the kinetic energy, \( V_{\text{ne}}[\rho] \) is the nuclei-electron interaction, \( E_x[\rho] \) and \( E_c[\rho] \) are the exchange and correlation energy functionals.

The exchange and correlation functionals normally used in DFT are integrals of some function of the density and possibly the density gradient. In addition to pure DFT methods, \texttt{dscf} and \texttt{grad} modules support hybrid functionals in which the exchange functional includes the Hartree–Fock exchange, e.g. B3-LYP.

### 6.2 Exchange-Correlation Functionals Available

The following exchange-correlation functionals are available:

- LDAs: S-VWN, PWLDA
- GGAs: B-VWN, B-LYP, B-P, PBE
- MGGA: TPSS; M06 (using XCFun)
- hybrid functionals: BH-LYP, B3-LYP, PBE0, TPSSh; M06-2X (using XCFun)
- double–hybrid functional: B2-PLYP (energy calculations only!)

For EXX and LHF, see Chapter 19

The XCFun library (Arbitrary-Order Exchange-Correlation Functional Library) by Ulf Ekström and co-workers has been included \cite{59} and some of the functionals implemented there can now be utilized. Among them are the empirically fitted MGGA\#s M06 and M06-2X from the Truhlar group \cite{60}. XCFun functionals are available for energy, gradient, vib. frequencies and TDDFT excited state energy calculations - with and without RI approximation. For details and the license of XCFun please refer to its web site https://repo.ctcc.no/projects/xcfun/wiki

See the next chapter for available functionals from XCFun.

In detail, the Turbomole own functional library consists of:

- The Slater–Dirac exchange functional only (S) \cite{61,62}.
- The 1980 correlation functional (functional V in the paper) of Vosko, Wilk, and Nusair only (VWN) \cite{63}.
- A combination of the Slater–Dirac exchange and Vosko, Wilk, and Nusair 1980 (functional V) correlation functionals (S-VWN) \cite{61–63}. 

6.2. EXCHANGE-CORRELATION FUNCTIONALS AVAILABLE

- The S-VWN functional with VWN functional III in the paper. This is the same functional form as available in the Gaussian program [61–63].
- A combination of the Slater–Dirac exchange and Becke’s 1988 exchange functionals (B88) [61,62,65].
- Lee, Yang, and Parr’s correlation functional (LYP) [66].
- The B-LYP exchange-correlation functional (B88 exchange and LYP correlation functionals) [61,62,65,66].
- The B-VWN exchange-correlation functional (B88 exchange and VWN (V) correlation functionals) [61–63,65].
- The B-P86 exchange-correlation functional (B88 exchange, VWN(V) and Perdew’s 1986 correlation functionals) [61–63,65,67].
- The Perdew, Burke, and Ernzerhof (PBE) exchange-correlation functional [61,62,64,68].
- The Tao, Perdew, Staroverov, and Scuseria functional (Slater–Dirac, TPSS exchange and Perdew-Wang (1992) and TPSS correlation functionals) [61,62,64,69].

Additionally, for all four modules (dscf, grad, ridft, and rdgrad) following hybrid functionals are available (a mixture of Hartree–Fock exchange with DFT exchange-correlation functionals):

- The BH-LYP exchange-correlation functional (Becke’s half-and-half exchange in a combination with the LYP correlation functional) [61,62,65,66,70].
- The B3-LYP exchange-correlation functional (Becke’s three-parameter functional) with the form,

\[ 0.8S + 0.72B88 + 0.2HF + 0.19VWN(V) + 0.81LYP \]  \hspace{1cm} (6.3)

where HF denotes the Hartree-Fock exchange [61,62,65,66,71].
- The B3-LYP exchange-correlation functional with VWN functional V in the paper. This is the same functional form as available in the Gaussian program.
- The 1996 hybrid functional of Perdew, Burke, and Ernzerhof, with the form,

\[ 0.75(S + PBE(X)) + 0.25HF + PW + PBE(C) \]  \hspace{1cm} (6.4)

where PBE(X) and PBE(C) are the Perdew–Burke–Ernzerhof exchange and correlation functionals and PW is the Perdew–Wang correlation functional [61,62,64,68,72].
The TPSSH exchange-correlation functional of Staroverov, Scuseria, Tao and Perdew with the form,

$$0.9(S + TPSS(X)) + 0.1HF + PW + TPSS(C)$$

where HF denotes the Hartree–Fock exchange [61,62,64,69,73].

The Double-Hybrid Functional B2-PLYP can be used for single point energy calculations. Note that one has to run an MP2 calculation after the DFT step to get the correct B2-PLYP energy!

B2-PLYP is a so-called double-hybrid density functional (DHDF) [74] that uses in addition to a non-local exchange contribution (as in conventional hybrid-GGAs) also a non-local perturbation correction for the correlation part. In the following options/restrictions in the present version of this method:

- single point calculations only (computed with the DSCF/RIDFT and RIMP2/RICC2 modules).
- UKS treatment for open-shell cases.
- can be combined with resolution-of-identity approximation for the SCF step (RI-JK or RI-J option).
- can be combined with the dispersion correction (DFT-D method, $s_6$(B2-PLYP)=0.55).

The non-local perturbation correction to the correlation contribution is given by second-order perturbation theory. The idea is rooted in the ab initio Kohn-Sham perturbation theory (KS-PT) by Görling and Levy [75,76]. The mixing is described by two empirical parameters $a_x$ and $a_c$ in the following manner:

$$E_{XC}(DHDF) = (1-a_x)E_X(GGA) + a_xE_X(HF) + (1-a_c)E_C(GGA) + a_cE_C(KS-PT2),$$

where $E_X(GGA)$ is the energy of a conventional exchange functional and $E_C(GGA)$ is the energy of a correlation functional. $E_X(HF)$ is the Hartree-Fock exchange of the occupied Kohn-Sham orbitals and $E_C(KS-PT2)$ is a Møller-Plesset like perturbation correction term based on the KS orbitals:

$$E_C(KS-PT2) = \frac{1}{2} \sum_{ia} \sum_{jb} \frac{(ia|jb)(ia|jb) - (ib|ja)}{e_i + e_j - e_a - e_b}.$$
6.2. EXCHANGE-CORRELATION FUNCTIONALS AVAILABLE

- during preparation of your input with DEFINE select b2-plyp in the DFT menu.
- carry out a DSCF run. Prepare and run a RI-MP2 calculation with either RIMP2 or RICC2 program modules.
- the RI-MP2 program directly prints the B2PLYP energy if this functional has been chosen before

Or use the b2plypprep script to setup up the calculation.

- define coord and basis set
- (optional: switch on ri or rijk and define jbasis or jkbasis)
- run b2plypprep
- run DSCF (or RIDFT) and RICC2

6.2.1 Exchange-Correlation Functionals from XCFun library

The XCFun library is taken from: https://repo.ctcc.no/projects/xcfun/wiki

The current TURBOMOLE version uses XCFun 1.99 and enables the usage of individual mixtures of the available exchange and correlation functionals.

To trigger the usage of XCFun functionals, use the keyword xcfun in the $dft section:

```
$dft
  functional xcfun set-gga
  functional xcfun <name1> <factor1>
  functional xcfun <name2> <factor2>
```

In addition to the name of the functional, it is necessary to tell TURBOMOLE whether the used functional is of GGA or MGGA type. Pure LDA functionals are currently not supported.

Available settings are:

- `functional xcfun set-gga` – sets a GGA functional
- `functional xcfun set-mgga` – sets a meta-GGA functional
- `functional xcfun set-hybrid 0.2` – defines a hybrid functional with a portion of 0.2 of Hartree-Fock exchange

Add the switch for either GGA or meta-GGA but not both in the same input!

List of available XCFun functionals (copied from XCFun documentation), in arbitrary order:
slaterx -- Slater exchange
beckex -- Becke exchange
beckecorrx -- Becke GGA exchange
ktx -- Keal-Tozer exchange
pbex -- Perdew-Burke-Ernzerhof exchange
tpssx -- TPSS original exchange functional
m05x -- Truhlar M05 exchange
m05x2x -- Truhlar M05-2X exchange
m06x -- Truhlar M06 exchange
m06x2x -- Truhlar M06-2X exchange
m06lx -- Truhlar M06L exchange
m06hfx -- M06-HF Meta-Hybrid Exchange Functional
b97x -- B97 exchange
b97_1x -- B97-1 exchange
b97_2x -- B97-2 exchange
b97_dx -- B97-D exchange ($disp3 in addition required)
optx -- OPTX Handy & Cohen exchange GGA exchange functional
optxcorr -- OPTX Handy & Cohen exchange -- correction part only
brx -- Becke-Roussells exchange with jp dependence
brxc -- Becke-Roussells exchange and correlation with jp dependence
pw86xtot -- Perdew-Wang 86 GGA exchange including Slater part
pw91x -- Perdew-Wang 1991 GGA Exchange Functional
ldaerfx -- Short range exchange LDA functional

vwn5c -- VWN5 correlation
vwn3c -- VWN3 correlation
lypc -- LYP correlation
pw91c -- PW91 Correlation
pw92c -- PW92 LDA correlation
pz81c -- PZ81 LDA correlation
pbec -- PBE correlation functional
vwn_pbec -- PBE correlation functional with VWN LDA correlation
spbec -- Simplified PBE correlation functional for use with the SSB functionals
tpssc -- TPSS original correlation functional
revtpssc -- Revised TPSS correlation functional
p86c -- P86C GGA correlation
m05c -- M05 Meta-Hybrid Correlation Functional
m05x2c -- M05-2X Meta-Hybrid Correlation Functional
**6.2. EXCHANGE-CORRELATION FUNCTIONALS AVAILABLE**

m06c -- M06 Meta-Hybrid Correlation Functional
m06lc -- M06-L Meta GGA Correlation Functional
m06x2c -- M06-2X Meta-Hybrid Correlation Functional
csc -- Colle-Salvetti correlation functional
brc -- Becke-Roussells correlation with jp dependence
b97_1c -- B97-1 correlation
b97_2c -- B97-2 correlation
b97_dc -- B97-D correlation ($disp3$ in addition required)
b97c -- B97 correlation
ldaerfc -- Short range correlation LDA functional

pw91k -- PW91 GGA Kinetic Energy Functional
btk -- Borgoo-Tozer kinetic energy functional
tfk -- Thomas-Fermi Kinetic Energy Functional
vw -- von Weizsaeccker kinetic energy

Some common functionals are pre-defined in XCFun and their individual parts do not have to be set manually. Those aliases can be directly used as names with a following factor of 1.0:
blyp, pbe, bp86, kt1, kt2, kt3, pbe0, b3lyp, m06, m06-2x, m06L, b3lyp-g, b3p86, b97, b97d, olyp and some more.

Note that if the functional needs a portion of HF exchange, this has to be added manually in the control file using `functional xcfun set-hybrid <number>`

Example for B3-LYP using VWN3 instead of VWN5:

```bash
$dft
  functional xcfun set-gga
  functional xcfun b3lyp-g 1.0
  functional xcfun set-hybrid 0.2
```

The functionals described in this section can be used for ground state energies, gradients and frequency calculations as well as TDDFT spectra. TDDFT analytic gradients are not yet supported, please use the TURBOMOLE own functionals instead.

**Notes about DFT-D3, gCP and functionals using those corrections**

For details about the options of DFT-D3 please see section 6.6.

In the original TURBOMOLE implementation of the B97-D functional only energy and gradient calculations are possible due to missing higher derivatives of the functional
itself. Using the XCFun version of B97-D, analytic 2nd derivatives using aoforce and TDDFT excited state energies are possible. The names in the $dft$ section are b97-d for the TURBOMOLE own version and b97d for the XCFun version. However, the total energies of those two flavours are slightly different due to the fact that the parameters used are either the originally published ones (TURBOMOLE) or re-computed (XCFun). For properties like geometries and frequencies the differences are negligible, but one should not mix the total energies.

The PBEh-3c functional needs, besides the functional name pbeh-3c also DFT-D3 dispersion correction including the three-body term and geometrical counterpoise correction method called gCP. For details see: Stefan Grimme, University Bonn. In order to get the full version of PBEh-3c, your control file has to include:

```
$dft
  functional pbeh-3c
  gridsize m4
$disp3 -bj -abc
```

Note: gcp is automatically added if pbeh-3c functional is used, but the D3 part has to be switched on manually by adding $disp3 as given above.

To use HF-3c (R. Sure, S. Grimme, J. Comput. Chem. 2013, 34, 1672–1685), an input without DFT functional but with DFT-D3 correction is required. Note that it is important to use the 'Minix' basis set and to select hf-3c as functional name for DFT-D3:

```
$disp3 -bj func hf-3c
```

The gCP correction (H. Kruse, S. Grimme, J. Chem. Phys. 2012, 136, 154101) will by default be added to the DFT-D3 correction term if pbeh-3c or hf-3c is selected.
6.3 Restricted Open-Shell Hartree–Fock

6.3.1 Brief Description

The spin-restricted open-shell Hartree–Fock method (ROHF) can always be chosen to systems where all unpaired spins are parallel. The TURBOMOLE keywords for such a case (one open shell, triplet $e_2^2$) are:

```plaintext
$open shells type=1
eg 1
$roothaan 1
   a=1  b=2
```

It can also treat more complicated open-shell cases, as indicated in the tables below. In particular, it is possible to calculate the $[xy]$ singlet case. As a guide for expert users, complete ROHF TURBOMOLE input for $O_2$ for various CSFs (configuration state function) is given in Section 22.6. Further examples are collected below.

The ROHF ansatz for the energy expectation value has a term for interactions of closed-shells with closed-shells (indices $k, l$), a term for purely open-shell interactions (indices $m, n$) and a coupling term ($k, m$):

$$E = 2 \sum_k h_{kk} + \sum_{k,l} (2J_{kl} - K_{kl}) + f \left[ 2 \sum_m h_{mm} + f \sum_{m,n} (2aJ_{mn} - bK_{mn}) + 2 \sum_{k,m} (2J_{km} - K_{km}) \right]$$

where $f$ is the (fractional) occupation number of the open-shell part ($0 < f < 1$), and $a$ and $b$ are the Roothaan parameters, numerical constants which depend on the particular configuration of interest.

6.3.2 One Open Shell

Given are term symbols (up to indices depending on actual case and group) and $a$ and $b$ coefficients. $n$ is the number of electrons in an irrep with degeneracy $n_{ir}$. Note that not all cases are Roothaan cases.

All single electron cases are described by:

$$a = b = 0$$
Table 6.1: Roothaan-coefficients $a$ and $b$ for cases with degenerate orbitals.

<table>
<thead>
<tr>
<th>$n_{ir}$</th>
<th>$f$</th>
<th>$e^n$</th>
<th>$\pi^n$</th>
<th>$\delta^n$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1/2</td>
<td>$^3A$</td>
<td>$^3\Sigma$</td>
<td>$^1\Delta$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1E^*$</td>
<td>$^1\Sigma$</td>
<td>$^1\Gamma$</td>
<td>1/2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1A$</td>
<td>$^1\Sigma$</td>
<td>$^1\Sigma$</td>
<td>0</td>
<td>−2</td>
</tr>
<tr>
<td>3</td>
<td>3/4</td>
<td>$^4E$</td>
<td>$^2\Pi$</td>
<td>$^2\Delta$</td>
<td>8/9</td>
<td>8/9</td>
</tr>
</tbody>
</table>

$1 \ n_{ir}=3$: p (O(3)), t (T, O, $I$)$^\dagger$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f$</th>
<th>$p^n$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1/3</td>
<td>$^3P$</td>
<td>3/4</td>
<td>3/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1D^{**}$</td>
<td>9/20</td>
<td>−3/10</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
<td>$^4S$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^2D^{**}$</td>
<td>4/5</td>
<td>4/5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^2P$</td>
<td>2/3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2/3</td>
<td>$^3P$</td>
<td>15/16</td>
<td>9/8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1D^{**}$</td>
<td>69/80</td>
<td>27/40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1S$</td>
<td>3/4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5/6</td>
<td>$^2P$</td>
<td>24/25</td>
<td>24/25</td>
</tr>
</tbody>
</table>

Only irrep $g(I)$ (mainly high spin available)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f$</th>
<th>$g^n$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/8</td>
<td>$^2G$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>$^1f$</td>
<td>2/3</td>
<td>4/3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1A$</td>
<td>0</td>
<td>−4</td>
</tr>
<tr>
<td>3</td>
<td>3/8</td>
<td>$^4G$</td>
<td>8/9</td>
<td>16/9</td>
</tr>
<tr>
<td>4</td>
<td>1/2</td>
<td>$^3A$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5/8</td>
<td>$^4G$</td>
<td>24/25</td>
<td>32/25</td>
</tr>
<tr>
<td>6</td>
<td>3/4</td>
<td>$^1f$</td>
<td>26/27</td>
<td>28/27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1A$</td>
<td>8/9</td>
<td>4/9</td>
</tr>
<tr>
<td>7</td>
<td>7/8</td>
<td>$^2G$</td>
<td>48/49</td>
<td>48/49</td>
</tr>
</tbody>
</table>

Continues on next page
Table 6.1: Roothaan-coefficients \( a \) and \( b \) for cases with degenerate orbitals (continued).

<table>
<thead>
<tr>
<th>( n )</th>
<th>( f )</th>
<th>( d^n )</th>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/10</td>
<td>( ^2D )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1/5</td>
<td>( ^3F ) ( ^3P ) ( ^1S )</td>
<td>5/8</td>
<td>5/4</td>
</tr>
<tr>
<td>3</td>
<td>3/10</td>
<td>( ^4F ) ( ^4P ) ( ^1S )</td>
<td>5/6</td>
<td>5/3</td>
</tr>
<tr>
<td>4</td>
<td>2/5</td>
<td>( ^5D ), ( ^5H )</td>
<td>15/16</td>
<td>15/8</td>
</tr>
<tr>
<td>5</td>
<td>1/2</td>
<td>( ^6S ), ( ^6A )</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3/5</td>
<td>( ^5D ), ( ^5H )</td>
<td>35/36</td>
<td>25/18</td>
</tr>
<tr>
<td>7</td>
<td>7/10</td>
<td>( ^4F ) ( ^4P ) ( ^1S )</td>
<td>95/98</td>
<td>55/49</td>
</tr>
<tr>
<td>8</td>
<td>4/5</td>
<td>( ^5D ), ( ^5H )</td>
<td>125/128</td>
<td>65/64</td>
</tr>
<tr>
<td>9</td>
<td>9/10</td>
<td>( ^2D ), ( ^2H )</td>
<td>80/81</td>
<td>80/81</td>
</tr>
</tbody>
</table>

* except cases (e.g. \( D_{2d} \) or \( D_{4h} \)) where \( e^2 \) gives only one-dimensional irreps, which are not Roothaan cases.
† only \( p^n \) given, the state for groups \( T_d \) etc. follows from \( S \rightarrow A \ (T,O,I) \ P \rightarrow T \ (T,O,I) \ D \rightarrow H \ (I) \), \( E + T \ (T,O) \)
** This is not a CSF in \( T \) or \( O \), \( (a,b) \) describes average of states resulting from \( E + T \)
†† \( (a,b) \) describes weighted average of high spin states, not a CSF.

Example

The \( 4d^9 \) \( 5s^2 \) \( ^2D \) state of Ag, in symmetry \( I \)

$closed shells$

\[
\begin{align*}
& a & 1-5 & (2) \\
& t_1 & 1-3 & (2) \\
& h & 1 & (2)
\end{align*}
\]

$open shells type=1$

\[
\begin{align*}
& h & 2 & (9/5)
\end{align*}
\]

$roothaan$

\[
\begin{align*}
& 1 \\
& a = 80/81 & b = 80/81
\end{align*}
\]
6.3.3 More Than One Open Shell

A Half-filled shell and all spins parallel

All open shells are collected in a single open shell and

\[ a = 1 \quad b = 2 \]

Example: The \( 4d^6 5s^1 7S \) state of Mo, treated in symmetry I

\[
\text {$roothaan } 1 \\
\text { a = 1 \quad b = 2 } \\
\text {$closed shells } \\
\text { a 1-4 ( 2 ) } \\
\text { t1 1-3 ( 2 ) } \\
\text { h 1 ( 2 ) } \\
\text {$open shells type=1 } \\
\text { a 5 ( 1 ) } \\
\text { h 2 ( 1 ) } \\
\]

Two-electron singlet coupling

The two MOs must have different symmetries (not required for triplet coupling, see example 6.3.3). We have now two open shells and must specify three sets of \((a, b)\), i.e. one for each pair of shells, following the keyword $rohf.$

Example: \( \text{CH}_2 \) in the \(^1\text{B}_2 \) state from \((3a_1)^1 (1b_2)^1\), molecule in (x,z) plane.

\[
\text {$closed shells } \\
\text { a1 1-2 ( 2 ) } \\
\text { b1 1 ( 2 ) } \\
\text {$open shells type=1 } \\
\text { a1 3 ( 1 ) } \\
\text { b2 1 ( 1 ) } \\
\text {$roothaan } 1 \\
\text {$rohf } \\
\text { 3a1-3a1 \quad a = 0 \quad b = 0 } \\
\text { 1b2-1b2 \quad a = 0 \quad b = 0 } \\
\text { 3a1-1b2 \quad a = 1 \quad b = -2 } \\
\]
Two open shells

This becomes tricky in general and we give only the most important case:

**shell 1** is a Roothaan case, see 6.3.2

**shell 2** is one electron in an a (s) MO \((n_{ir} = 1)\)

with parallel spin coupling of shells.

This covers e.g. the \(p^5 s^1\) 3P states, or the \(d^4 s^1\) 6D states of atoms. The coupling information is given following the keyword `$rohf`. The \((a, b)\) within a shell are taken from above (6.3.2), the cross term (shell 1)–(shell 2) is in this case:

\[
\begin{align*}
    a &= 1 \quad \text{always} \\
    b &= 2 \quad \text{if } n \leq n_{ir} \\
    b &= \left(\frac{2n_{ir}}{n}\right) \quad \text{if } n > n_{ir}
\end{align*}
\]

where \(n_{ir}\) and \(n\) refer to shell 1.

**Example 1:** The \(4d^4 s^1\) 6D state of Nb, in symmetry I

```plaintext
$closed shells
   a  1-4       ( 2 )
   t1 1-3       ( 2 )
   h  1        ( 2 )
$open shells type=1
   a  5       ( 1 )
   h  2      ( 4/5 )
$roothaan  1
$rohf
  5a-5a  a = 0  b = 0
  5a-2h  a = 1  b = 2
  2h-2h  a = 15/16  b = 15/8
```

**Example 2:** The \(4d^5 5s^1\) 7S state of Mo, symmetry I (see Section 6.3.3) can also be done as follows.

```plaintext
$roothaan  1
$rohf
  5a-5a  a = 0  b = 0
  5a-2h  a = 1  b = 2
  2h-2h  a = 1  b = 2
```
$closed shells$
\begin{align*}
a & : 1-4 & (2) \\
t_1 & : 1-3 & (2) \\
h & : 1 & (2)
\end{align*}

$open shells type=1$
\begin{align*}
a & : 5 & (1) \\
h & : 2 & (1)
\end{align*}

The shells $5s$ and $4d$ have now been made inequivalent. Result is identical to 6.3.3 which is also more efficient.

**Example 3:** The $4d^9 5s^1 3D$ state of Ni, symmetry $I$

$closed shells$
\begin{align*}
a & : 1-3 & (2) \\
t_1 & : 1-2 & (2)
\end{align*}

$open shells type=1$
\begin{align*}
a & : 4 & (1) \\
h & : 1 & (9/5)
\end{align*}

$roothaan 1$

$rohf$
\begin{align*}
4a-4a & a = 0 & b = 0 \\
1h-1h & a = 80/81 & b = 80/81 \\
4a-1h & a = 1 & b = 10/9
\end{align*}

(see basis set catalogue, basis $SV.3D$ requires this input and gives the energy you must get)

### 6.3.4 Miscellaneous

**Valence states**

Valence states are defined as the weighted average of all CSFs arising from an electronic configuration (occupation): $(\text{MO})^n$. This is identical to the average energy of all Slater determinants.

$$a = b = \frac{2n_{ir}(n-1)}{(2n_{ir}-1)n}$$

This covers, e.g. the cases $n = 1$ and $n = 2n_{ir} - 1$: $p^1$, $p^5$, $d^1$, $d^9$, etc, since there is only a single CSF which is identical to the average of configurations.
6.3. RESTRICTED OPEN-SHELL HARTREE–FOCK

Totally symmetric singlets for 2 or \((2n_{ir}-2)\) electrons

\[
\begin{align*}
 n &= 2 & a &= 0 & b &= -n_{ir} \\
n &= (2n_{ir} - 2) & a &= \frac{n_{ir}(n_{ir} - 2)}{(n_{ir} - 1)^2} & b &= \frac{n_{ir}(n_{ir} - 3)}{(n_{ir} - 1)^2}
\end{align*}
\]

This covers the \(^1S\) states of \(p^2, p^4, d^2, d^8\), etc.

Average of high-spin states: \(n\) electrons in MO with degenerate \(n_{ir}\).

\[
\begin{align*}
 a &= \frac{n_{ir}(4k(k + l - 1) + l(l - 1))}{(n_{ir} - 1)n^2} \\
b &= \frac{2n_{ir}(2k(k + l - 1) + l(l - 1))}{(n_{ir} - 1)n^2}
\end{align*}
\]

where: \(k = \max(0, n - n_{ir})\), \(l = n - 2k = 2S\) (spin)

This covers most of the cases given above. A CSF results only if \(n = \{1, (n_{ir} - 1), n_{ir}, (n_{ir} + 1), (2n_{ir} - 1)\}\) since there is a single high-spin CSF in these cases.

The last equations for \(a\) and \(b\) can be rewritten in many ways, the probably most concise form is

\[
\begin{align*}
 a &= \frac{n(n - 2) + 2S}{(n - 2f)n} \\
b &= \frac{n(n - 2) + (2S)^2}{(n - 2f)n}
\end{align*}
\]

This applies to shells with one electron, one hole, the high-spin couplings of half-filled shells and those with one electron more or less. For \(d^2, d^3, d^7\), and \(d^8\) it represents the (weighted) average of high-spin cases: \(3F + 3P\) for \(d^2, d^8\), \(4F + 4P\) for \(d^3, d^7\).
6.4 Relativistic effects

TURBOMOLE provides two different possibilities for the treatment of relativistic effects: Via effective core potentials (ECPs) or via all-electron approaches (X2C, DKH, BSS). Both techniques can be employed in an one-component (scalar relativistic) or two-component (including spin-orbit coupling) framework. The latter is only available in the module ridft.

6.4.1 One- and two-component relativistic methods

Incoporation of scalar-relativistic effects leads to additional contributions to the one-electron integrals (either from ECP or all-electron approach). The program structure is the same as in non-relativistic theory (all quantities are real). Two-component treatments allow for self-consistent calculations including spin-orbit interactions. These may be particularly important for compounds containing heavy elements (additionally to scalar relativistic effects). Two-component treatments require the use of complex two-component orbitals

$$\psi_i(x) = \begin{pmatrix} \psi_i^{\alpha}(r) \\ \psi_i^{\beta}(r) \end{pmatrix}$$

instead of real (non-complex) one-component orbitals needed for non-relativistic or scalar-relativistic treatments. The Hartree-Fock and Kohn-Sham equations are now spinor equations with a complex Fock operator

$$\begin{pmatrix} \hat{F}_{\alpha\alpha} & \hat{F}_{\alpha\beta} \\ \hat{F}_{\beta\alpha} & \hat{F}_{\beta\beta} \end{pmatrix} \begin{pmatrix} \psi_i^{\alpha}(r) \\ \psi_i^{\beta}(r) \end{pmatrix} = \epsilon_i \begin{pmatrix} \psi_i^{\alpha}(r) \\ \psi_i^{\beta}(r) \end{pmatrix}.$$ 

The wavefunction is no longer eigenfunction of the spin operator, the spin vector is no longer an observable.

In case of DFT for open-shell systems rotational invariance of the exchange-correlation energy was ensured by the non-collinear approach. In this approach the exchange-correlation energy is a functional of the particle density and the absolute value of the spin-vector density $\vec{m}(r)$ ($\vec{\sigma}$ are the Pauli matrices)

$$\vec{m}(r) = \sum_i \psi_i^\dagger(x)\vec{\sigma}\psi_i(x).$$

This quantity replaces the spin-density (difference between density of alpha and beta electrons) of non- or scalar-relativistic treatments.

For closed-shell species the Kramers-restricted scheme, a generalization of the RHF-scheme of one component treatments, is applicable.

**Effective core potentials** The most economic way to account for relativistic effects is via effective core potentials by choosing either the one- or the two-component
ECP (and for the latter additionally setting \texttt{$sogbf$} in the control file). The theoretical background and the implementation for the two-component SCF procedure is described in [77]. The background theory of the more fundamental approach, all-electron relativistic electronic structure theory, is given in the next paragraph.

**Relativistic all-electron approaches (X2C, DKH, BSS)**  
Relativistic calculations are based on the Dirac rather than on the Schrödinger Hamiltonian. Since the Dirac Hamiltonian introduces pathological negative-energy states and requires extensive one-electron basis set expansions, methods have been devised which allow one to calculate a matrix representation of that part of the Dirac Hamiltonian, which describes electronic states only. For this, a unitary transformation is employed to block-diagonalize the Dirac Hamiltonian and thus to decouple the negative-energy from the electronic states. In order to be efficient, this transformation is carried out only for the one-electron part of the full Hamiltonian (as a consequence, the two-electron interaction will then be affected by a picture-change effect). The resulting quantum chemical approach has been called an exact two-component (X2C) one. It was developed by many groups starting with formal work in the mid 1980s. X2C is related to the step-wise Douglas-Kroll-Hess (DKH) approach, which also achieves exact decoupling but sequentially. The number of transformation steps is called the order of DKH. X2C is also related to the Barysz-Sadlej-Snijders (BSS) method, that first applies the free-particle Foldy-Wouthuysen transformation (which is the first mandatory step in DKH), and then constructs the one-step exact decoupling transformation of X2C. These three approaches have been reviewed and directly compared in terms of formalism and results, respectively, in Ref. [78] (see also this reference for a complete bibliography on exact-decoupling methods).

Essentially, X2C methods change the one-electron Hamiltonian in basis-set representation. The Schrödinger one-electron Hamiltonian (including the external potential of the atomic nuclei) is replaced by the transformed (upper-left block of the) Dirac Hamiltonian. Since the transformation is carried out in the fully decontracted primitive basis, all matrix operations needed for the generation of the relativistic one-electron Hamiltonian can be cumbersome and even prohibitive if the size of the molecule is large. In order to solve this unfavorable scaling problem, a rigorous local approach, called DLU, has been devised [79] and should be activated for large molecules. However, since the local (atomic) structure is defined by the atom-centered basis functions within DLU, diffuse functions in the basis set should be handled with care and the use of tailored basis sets is recommended (relativistic calculations require refitted basis sets; if these are not available for X2C or X2C-DLU, standard second-order DKH basis sets could be used).

As in DKH theory, X2C and BSS exist in full two-component (spin-(same-)orbit coupling including) and in a one-component scalar relativistic form. Both have been implemented into the TURBOMOLE package and all details on the efficient implementation have been described in Ref. [80], which should be cited when the module is activated.
6.4.2 How to use

The keyword $soghf$ enforces the two-component calculations. Keywords for specification of the method of calculation are the same as for the one-component case. The DIIS scheme for complex Fock operators can be activated by inserting $gdiis$ in the control-file. For closed-shell species a Kramers invariant density functional formalism (only pure density functionals) can be switched on with the keyword $kramers$. These keywords have to be inserted into the control-file manually.

As start wavefunctions Hückel-, UHF- or RHF-wavefunctions may be used. The two-component formalism supports Abelian point group symmetry if $kramers$ is set. Otherwise start wave functions may be transformed to C1 symmetry by define or the script 'uhfuse'. For open-shell molecules it is often helpful to increase the value for $scforbitalshift closedshell; a value of ca. 1.0 may serve as a rough recommendation.

**Effective core potentials**  The two-component formalism may be most easily prepared and applied in the following way:

- Run define: choose C1-symmetry; select ECPs and basis sets with suffices $-2c$ for the respective elements. For spin-orbit treatments two-component ECPs (suffix $-2c$) are required, the use of extended basis sets accounting for the spatial splitting of inner p-shells (also suffix $-2c$) is recommended (see [81]). ECPs and basis sets def2-XVP-2c ($X=\text{S}, \text{TZ}, \text{QZ}$) are available for Ag - I, Au - At, they can be selected within the define session. RI-J and RI-JK auxiliary basis sets of def2-type are of sufficient flexibility for two-component treatments; they are the same with and without suffix $-2c$. The corresponding auxiliary basis sets are provided automatically.

- Insert $soghf$ in the control file (as well as further desired keywords).

- Start the two-component calculation with ridft

- At the end of the SCF procedure real and imaginary parts of spinors are written to files spinor.r and spinor.i, eigenvalues and spinor occupations are collected in the file EIGS, the total energy is added to data group $energy$. The data groups $closed shells (\alpha shells and \beta shells for open shell cases) are no longer significant, but nevertheless kept in the control-file; additionally the spinor occupations are deposited in data group $spinor$.

**All-electron calculations**  The keywords $rx2c$, $rbss$ and $rdkh$ [Order of DKH] are used to activate the X2C, BSS or DKH Hamiltonian. The default order of the DKH Hamiltonian is four. It is not recommended to go beyond, but to use X2C instead. For details on the arbitrary-order DKH Hamiltonians see Ref. [82] for details on the infinite-order DKH theory, [83] for the implementation, and [84] for a conceptual review of DKH theory. The local approach (DLU) can be optionally activated by $rlocal$ for all one- and two-component all-electron Hamiltonians. For
symmetric molecules, point-group symmetry is not exploited by default, but can be used in the one-component case by setting $rsym$. 
6.5 Periodic Electrostatic Embedded Cluster Method

6.5.1 General Information

The Periodic Electrostatic Embedded Cluster Method (PEECM) functionality [85] provides electronic embedding of a finite, quantum mechanical cluster in a periodic, infinite array of point charges. It is implemented within HF and DFT energy and gradient TURBOMOLE modules: dscf, grad, ridft, rdgrad, and escf. Unlike embedding within a finite set of point charges the PEEC method always yields the correct electrostatic (Madelung) potential independent of the electrostatic moments of the point charges field. It is also significantly faster than the traditional finite point charges embedding.

6.5.2 Theoretical Background

Generally, the PEEC method divides the entire, periodic and infinite system into two parts, the inner (I) part, or so called cluster, and the outer (O) part which describes its environment. Thus, unlike "true" periodic quantum mechanical methods, PEECM primarily aims at calculations of structure and properties of localized defects in dominantly ionic crystals. The innermost part of the cluster is treated quantum mechanically (QM), whereas in the remaining cluster part cations are replaced by effective core potentials (ECPs) and anions by ECPs or by simply point charges. Such an "isolating" outer ECP shell surrounding the actual QM part is necessary in order to prevent artificial polarization of the electron density by cations which would otherwise be in a direct contact with the QM boundary. The outer part or the environment of the cluster is described by a periodic array of point charges, representing cationic and anionic sites of a perfect ionic crystal.

The electronic Coulomb energy term arising from the periodic field of point charges surrounding the cluster has the following form

\[ J = \sum_{\mu\nu} \sum_{k} \sum_{\vec{L}\in O} D_{\mu\nu} q_k \int \frac{\mu(\vec{r})\nu(\vec{r})}{|\vec{r'} - \vec{R}_k - \vec{L}|} d\vec{r}, \]

where UC denotes the unit cell of point charges, \( D_{\mu\nu} \) are elements of the density matrix, \( \mu, \nu \) are basis functions, \( q_k, \vec{R}_k \) denote charges and positions of point charges, and \( \vec{L} \) denote direct lattice vectors of the outer part O. It is evaluated using the periodic fast multipole method (PFMM) [86] which, unlike the Ewald method [87], defines the lattice sums entirely in the direct space. In general, PFMM yields a different electrostatic potential than the Ewald method, but the difference is merely a constant shift which depends on the shape of external infinite surface of the solid (i.e. on the way in which the lattice sum converges toward the infinite limit). However, this constant does not influence relative energies which are the same as obtained using the Ewald method, provided that the total charge of the cluster remains constant. Additionally, since the electrostatic potential within a solid is not a well defined
6.5. PERIODIC ELECTROSTATIC EMBEDDED CLUSTER METHOD

quantity, both the absolute total energies and orbital energies have no meaning (i.e. you cannot compare energies of neutral and charged clusters!).

6.5.3 Calculation Setup

There are three key steps in setting up a PEECM calculation. In the first step the periodic field of point charges has to be defined by specifying the point charges unit cell. Next step is the definition of the part infinite of point charges field that will be replaced by the explicit quantum mechanical cluster. Finally, the quantum mechanical cluster together with surrounding ECPs representing cationic sites as well as point charges representing anions is defined and put in place of the point charges. The input preparation steps can be summarized as follows

1. Dimensionality of the system is specified by the keyword periodic in the $embed section: periodic 3 means a bulk three-dimensional system, periodic 2 denotes a two-dimensional surface with an aperiodic z direction.

2. Definition of the unit cell of periodic point charges field is specified in the subsections cell and content of the $embed section.

3. Definition of the values of the point charges by specifying a charge value per species, using the subsection charges, or a charge value for each point charge, using the subsection ch_list. Note that only one of the subsections can be defined.

4. Definition of the part of point charges field that will be replaced by the QM cluster together with the isolating shell (ECPs, explicit point charges) is specified in the subsection cluster of the $embed section.

5. Definition of the quantum mechanical cluster as well as the surrounding ECPs and anionic point charges is included in the usual $coord section.

The following two examples show the definition of the point charges unit cells.

Example 1. Ca$_4$F$_{19}$ cluster embedded in bulk CaF$_2$

In this example a QM cluster with the composition Ca$_4$F$_{19}$, surrounded by 212 ECPs and 370 explicit point charges, representing Ca$^{2+}$ cations and F$^-$ anions is embedded in a periodic field of point charges (+2 for Ca and -1 for F) corresponding to the CaF$_2$ fluorite lattice.

First, the program has to know that this is a three-dimensional periodic system. This is specified by the keyword periodic 3, meaning periodicity in three dimensions. The dimensions of the unit cell for bulk CaF$_2$ are given in the subsection cell of the $embed keyword. By default, the unit cell dimensions are specified in atomic units and can be changed to Å using cell ang. The positions of the point charges in the unit cell are specified in the subsection content. In this example positions are given in fractional crystal coordinates (content frac). You can change this by specifying content for Cartesian coordinates in atomic units or content ang for
Cartesian coordinates in Å. The values of point charges for Ca and F are given in
the subsection charges.

\[$embed\]
\[
\begin{array}{l}
\text{periodic} \hspace{0.5cm} 3 \\
\text{cell} \\
\quad 10.47977 \hspace{0.1cm} 10.47977 \hspace{0.1cm} 10.47977 \hspace{0.1cm} 90.0 \hspace{0.1cm} 90.0 \hspace{0.1cm} 90.0 \\
\text{content frac} \\
\quad \begin{array}{ccc}
F & 0.00 & 0.00 & 0.00 \\
Ca & -0.25 & -0.75 & -0.75 \\
F & 0.50 & -0.50 & 0.00 \\
F & 0.50 & 0.00 & -0.50 \\
F & 0.00 & -0.50 & -0.50 \\
F & 0.50 & -0.50 & -0.50 \\
F & 0.00 & 0.00 & -0.50 \\
F & 0.50 & 0.00 & 0.00 \\
F & 0.00 & -0.50 & 0.00 \\
Ca & -0.25 & -0.25 & -0.25 \\
Ca & 0.25 & -0.75 & -0.25 \\
Ca & 0.25 & -0.25 & -0.75
\end{array}
\end{array}
\]

\[\text{end}\]
\[\ldots\]
\[\text{charges} \]
\[
\begin{array}{l}
F & -1.0 \\
Ca & 2.0
\end{array}
\]
\[\text{end}\]

The above input defines a periodic, perfect, and infinite three-dimensional lattice of
point charges corresponding to the bulk CaF$_2$ structure. In order to use this lattice
for PEECM calculation we have to make “space” for our QM cluster and the isolating
shell. This is done by specifying the part of the lattice that is virtually removed
from the perfect periodic array of point charges to make space for the cluster. The
positions of the removed point charges are specified in the subsection cluster of the
$embed$ keyword. Note, that the position of the QM cluster and the isolating shell
must exactly correspond to the removed part of the crystal, otherwise positions
of the cluster atoms would overlap with positions of point charges in the periodic
lattice, resulting in a “nuclear fusion”.

\[\text{cluster} \]
\[
\begin{array}{cccc}
F & 0.00000000000000 & 0.00000000000000 & 0.00000000000000 \\
Ca & -2.61994465796043 & -2.61994465796043 & -2.61994465796043 \\
Ca & 2.61994465796043 & -2.61994465796043 & 2.61994465796043 \\
Ca & 2.61994465796043 & 2.61994465796043 & -2.61994465796043 \\
Ca & -2.61994465796043 & 2.61994465796043 & 2.61994465796043
\end{array}
\]
By default, the positions of point charges are specified in atomic units as Cartesian coordinates. You can change this by specifying `cluster frac` for fractional crystal coordinates or `cluster ang` for Cartesian coordinates in Å.

Finally, you have to specify the coordinates of the QM cluster along with the surrounding ECPs representing cationic sites and explicit point charges representing anions. This is done in the usual way using the `$coord` keyword.

```
coord
  0.00000000000000  0.00000000000000  0.00000000000000 f
-2.86167504097169  -2.86167504097169  -2.86167504097169 ca
 2.86167504097169   2.86167504097169   2.86167504097169 ca
-2.86167504097169  -2.86167504097169  -2.86167504097169 ca
 2.86167504097169   2.86167504097169   2.86167504097169 ca
 0.00000000000000  -5.24009410923923  0.00000000000000 f
-5.24009410923923  0.00000000000000  0.00000000000000 f
 0.00000000000000   5.24009410923923  0.00000000000000 f
 0.00000000000000  -5.24009410923923  0.00000000000000 f
-5.24009410923923  -5.24009410923923  0.00000000000000 f
 5.24009410923923  -5.24009410923923  0.00000000000000 f
 0.00000000000000  -5.24009410923923  5.24009410923923 f
 0.00000000000000  5.24009410923923 -5.24009410923923 f
 0.00000000000000  0.00000000000000  5.24009410923923 f
... repeated for Ca_{216}F_{389}
$end
This is the standard TURBOMOLE syntax for atomic coordinates. The actual distinction between QM cluster, ECP shell, and explicit point charges is made in the $atoms section.

```
$atoms
f 1,6-23
  basis =f def-TZVP
ca 2-5
  basis =ca def-TZVP
ca 24-235
  basis =none
  ecp =ca ecp-18 hay & wadt
f 236-605
  basis =none
  charge= -1.00000000
```

In the example above the F atoms 1 and 6-23 as well Ca atoms 2-5 are defined as QM atoms with def-TZVP basis sets. The Ca atoms 24-235 are pure ECPs and have no basis functions ($basis =none$) and F atoms 236-605 are explicit point charges with charge -1, with no basis functions and no ECP.

This step ends the input definition for the PEECM calculation.

**Example 2. Al$_8$O$_{12}$ cluster embedded in $\alpha$-Al$_2$O$_3$ (0001) surface**

In this example a QM cluster with the composition Al$_8$O$_{12}$, surrounded by 9 ECPs representing Al$^{3+}$ cations is embedded in a two-dimensional periodic field of point charges (+3 for Al and -2 for O) corresponding to the (0001) surface of $\alpha$-Al$_2$O$_3$.

As in the first example, the program has to know that this is a two-dimensional periodic system and this is specified by the keyword periodic 2. The dimensions of the unit cell for the (0001) $\alpha$-Al$_2$O$_3$ surface are given in the subsection cell of the $embed$ keyword. The aperiodic direction is always the z direction, but you have to specify the unit cell as if it was a 3D periodic system. This means that the third dimension of the unit cell must be large enough to enclose the entire surface in this direction. The unit cell dimensions are specified in Å using cell ang. The positions of the point charges in the unit cell are specified as Cartesian coordinates in Å ($content$ ang). The values of point charges for Al and O are given in the subsection charges.

```
$embed
periodic 2

content ang
  Al  2.402142286  1.386878848  5.918076515
  Al -0.000013520  -0.000003382  7.611351967
```
The above input defines a periodic, perfect, and infinite two-dimensional lattice of point charges corresponding to the (0001) α-Al₂O₃ surface. In order to use the lattice for PEECM calculation we have to make "space" for our QM cluster and the surrounding ECP shell. This is done by specifying the part of the lattice that is virtually removed from the perfect periodic array of point charges to make space for the cluster. The positions of the removed point charges are specified in the subsection \texttt{cluster} of the \texttt{$embed$} keyword. Note, that the position of the QM cluster must \textbf{exactly} correspond to the removed part of the crystal, otherwise positions of the cluster atoms would overlap with positions of point charges in the periodic lattice.
resulting in a “nuclear fusion”.

```plaintext
cluster ang
Al  -0.000012482  5.547518253  9.977437973
Al  2.402141094  6.934002943  8.064809799
Al  2.402144432  4.160642624 10.247499466
Al  4.804287434  5.547518253  9.977437973
Al  2.402250767  6.93436185  11.246870041
Al  -0.000005568  8.321288109 10.247499466
Al  2.402137518  9.708164215  9.977437973
Al  4.804294586  8.32188109  10.247499466
O   0.907584129  4.160642624  8.957920074
O   1.517618299  5.483696461 11.127141953
O   -0.703624666  6.893717766  11.27161026
O   3.145770990  5.457115650  8.957922935
O   3.990177393  4.265182018 11.127140045
O   0.751026928  7.029124260  8.957910538
O   4.100675106  6.893717766 11.127161026
O   7.206440926  6.934002943  8.064809799
O   4.804286480  8.321288109  7.611351967
end
```

The positions of point charges are specified in Å as Cartesian coordinates.

Finally, you have to specify the coordinates of the QM cluster along with the surrounding ECPs. This is done in the usual way using the `$coord` keyword.

```plaintext
$coord
-0.0000235760000  10.48329315900000  18.85463057110000  al
4.5393907480000  13.10412613690000  15.24028611330000  al
4.53939638280000  7.86247730390000  19.36497297520000  al
9.0787906320000  10.48329315900000  18.85463057110000  al
```
This is the standard TURBOMOLE syntax for atomic coordinates. The actual distinction between QM cluster and ECP shell is made in the $atoms section.

$atoms
al 1-8
   basis =al def-SV(P)
o 9-20
   basis =o def-SV(P)
al 21-29
   basis =none
   ecp =al ecp-10 hay & wadt

In the example above the Al atoms 1-8 and O atoms 9-20 are defined as QM atoms with def-SV(P) basis sets. The Al atoms 21-29 are pure ECPs and have no basis functions (basis =none).

This step ends the input definition for the PEECM calculation.
6.6 Dispersion Correction for DFT Calculations

Based on an idea that has earlier been proposed for Hartree-Fock calculations [88, 89], a general empirical dispersion correction has been proposed by Stefan Grimme for density functional calculations [90]. A modified version of the approach with extension to more elements and more functionals has been published in ref. [91]. The most recent implementation [92] is less empirical, i.e. the most important parameters are computed by first principles, and it provides a consistent description across the whole periodic system.

The first version (DFT-D1) can be invoked by the keyword $olddisp$ in the control file. The second version (DFT-D2) is used if the keyword $disp$ is found. For the usage of DFT-D3 just add keyword $disp3$ to the control file. Only one of the three keywords is expected to be present.

If DFT-D3 is used, the total energy is given by

$$E_{DFT-D3} = E_{KS-DFT} - E_{disp}$$

where $E_{KS-DFT}$ is the usual self-consistent Kohn-Sham energy as obtained from the chosen functional and $E_{disp}$ is a dispersion correction given by the sum of two- and three-body energies

$$E_{disp} = E^{(2)} + E^{(3)},$$

with the dominating two-body term

$$E^{(2)} = \sum_{AB} \sum_{n=6,8,10,...} s_n C_{n}^{AB} r_{n} f_{d,n}(r_{AB}).$$

The first sum runs over all atom pair, $C_{n}^{AB}$ denotes the $n$th-order dispersion coefficient for atom pair $AB$, $r_{AB}$ is their interatomic distance, and $f_{d,n}$ is a damping function.

Becke-Johnson (BJ) damping can be invoked by adding the option bj or -bj to the $disp3$ keyword: $disp bj$. If you use this damping option please also cite [93].

The three-body term can be switched on by adding abc to the $disp3$ input line, i.e. to use it in combination with Becke-Johnson damping just add $disp3 bj abc$

It is also possible not to use the functional name given in the control file but to tell the DFT-D3 routines to use the parameters which have been fitted to a specific functional. Just as in the original DFT-D3 routines, this can be selected by adding the func option, for example $disp3 bj func pbe0$. It is recommended to use this option as the last one in the $disp3$ input line.

Please have a look at DFT-D3 homepage, Grimme group Bonn for more detailed information.
Density-based dispersion corrections of non-local vdW-DF type

A non-local, electron density dependent dispersion correction which is based on Vydrov and Van Voorhis’ VV10 [94] has been implemented by the Grimme group [95] and is available for \texttt{ridft} and \texttt{rdgrad}. This correction can either be applied in a post-SCF and non-self-consistent way for energy calculations or self-consistently which is required to compute the gradients.

To switch on DFT-NL in a non-self-consistent way, just add

$\$donl$

$\$to the control file. For a self-consistent treatment of the dispersion correction add

$\$doscnl$

$\$instead. Note that dispersion corrections of DFT-DN and NL–DFT type must not be combined. The grid size for the non-local integration is set automatically by adapting the grid for the quadrature of the functional evaluation.

Currently only $C_1$ symmetry and serial jobs are possible. DFT-NL is an interesting scientific alternative to DFT-D3, but we recommend to use DFT-D3 for applications instead.
6.7 Energy Decomposition Analysis (EDA)

The interaction energy between molecules can be calculated with the supermolecular approach: one performs calculations for the supersystem and for the subsystems with size-consistent methods and derive the interaction energy $\Delta E$ by taking the energy difference. The energy decomposition analysis (EDA) allow a partitioning of the Hartree-Fock (HF) or DFT interaction energy in physically meaningful contributions: the classical electrostatic interaction $\Delta E_{\text{ele}}$, the exchange-repulsion $\Delta E_{\text{exrep}}$, the orbital relaxation energy $\Delta E_{\text{orb}}$ and additionally for DFT the correlation interaction $\Delta E_{\text{cor}}$:

$$\Delta E_{\text{HF}} = \Delta E_{\text{ele}} + \Delta E_{\text{exrep}} + \Delta E_{\text{orb}},$$  \hspace{1cm} (6.11)

$$\Delta E_{\text{DFT}} = \Delta E_{\text{ele}} + \Delta E_{\text{exrep}} + \Delta E_{\text{orb}} + \Delta E_{\text{cor}}.$$  \hspace{1cm} (6.12)

Further details and derivations of the different energy contributions can be found in [96].

6.7.1 How to perform

The EDA scheme is implemented in the module ridft and can be done with RI-Hartree-Fock and with all local, gradient corrected, hybrid and meta density functionals (please note that the functionals included in the XCFun library are not supported!).

- **Calculation of the subsystems:**
  In HF and hybrid DFT calculations please insert `$\text{scfdenapprox} 0$` in the `control` file, at least in a second run. The EDA scheme needs the exchange and Coulomb energies of every system separately. After the subsystem calculation you will find under `$\text{subenergy}$` the different energy contribution to the total energy of the system: the one electron energy, Coulomb- and exchange energy, correlation energy in case of DFT calculations, nuclear repulsion energy and optionally the dispersion energy.

- **Preparation of the supersystem control file:**
  First run `define` for the supersystem and take for consistency the same basis set and the same method (i.e. the same functional and the same grid). Please use in case of DFT calculations not the multiple grids m3 to m5, because this would lead to erroneous orbital relaxation energies. If the subsystems are open-shell species the occupation in the EHT submenu of `define` of the supersystem must be chosen open-shell, too. For open-shell systems the Fermi-smearing is recommended. The sequence of the supersystem coordinates must have the same sequence as the subsystem coordinates. In the case of HF and hybrid DFT calculation use again `$\text{scfdenapprox} 0$`.
  Then please insert in the `control` file:
$\text{subsystems}$

molecule#1 file=sub1/control
molecule#2 file=sub2/control

If you use the supermolecular basis set for the calculation of the monomers please insert after $\text{subsystems}$ the option copo:

$\text{subsystems copo}$

molecule#1 file=sub1/control
molecule#2 file=sub2/control

It is possible to generate orthogonal product wave functions when you use opro instead of copo. But with this choice it is not possible to calculate the different energy contributions of the interaction energy.

You can choose at most ten subsystems.

- Generation of the product molecular wave functions:
The module promova generates RHF and UHF product molecular wave functions. The new (product) start vectors can be found in the files .mos for closed-shell systems or in alpha and beta for open-shell systems. Please note that the molecular orbitals of the different subsystems are not orthogonal to each other.

- Energy decomposition analysis:
After the supersystem ridft calculation you will find the following output with the different contributions of the interaction energy in Hartree:

```
<table>
<thead>
<tr>
<th>* Total Interaction energy =  -0.0058700698</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Electrostatic Interaction =  -0.0134898233 :</td>
</tr>
<tr>
<td>Nuc---Nuc =  18.2843363825 :</td>
</tr>
<tr>
<td>1-electron =  -36.5372802833 :</td>
</tr>
<tr>
<td>2-electron =  18.2394540775 :</td>
</tr>
<tr>
<td>* Exchange-Repulsion =  0.0112325934 :</td>
</tr>
<tr>
<td>Exchange Int. =  -0.0139477002 :</td>
</tr>
<tr>
<td>Repulsion =  0.0251802936 :</td>
</tr>
<tr>
<td>* Orbital Relaxation =  -0.0036128399 :</td>
</tr>
<tr>
<td>------------------------------------------------</td>
</tr>
</tbody>
</table>
```
Chapter 7

DFT Calculations for Molecular and Periodic Systems

7.1 Functionalities of Riper

riper is a module for the calculation of ground state energies for molecular and periodic systems at the DFT level. In this program a combination of the Resolution of Identity (RI) approximation and continuous fast multipole method (CFMM) is applied for efficient evaluation of the electronic Coulomb term. For the exchange-correlation term an octree-based adaptive numerical integration scheme is used. For the molecular systems a low-memory modification of the RI approximation is implemented aimed specifically for calculations on very large molecules. More details are given in Refs [97], [98] and [99].

Current functionalities and limitations of riper are:

- both closed–shell RHF and open–shell UHF calculations
- sequential and parallel runs (OpenMP parallelization, see Sec. 3.2.2)
- all pure GGA exchange-correlation functionals for periodic systems
- hybrid GGA functionals for molecules
- almost $O(N)$ scaling with the system size
- as for now only $C_1$ symmetry point group for molecules and $P_1$ space group for periodic systems

7.2 Theoretical Background

The theory behind riper and implementation details are briefly presented as far as needed for applications and understanding of the main keywords. For more details see Refs [97–99] and references therein.
In periodic systems the translational symmetry of solids leads to Bloch orbitals $\psi_{p\sigma}^k$ and one-particle energies $\varepsilon_{p\sigma}^k$ depending on the band index $p$, spin $\sigma$, and the wave vector $k$ within the Brillouin zone (BZ), which is the unit cell of reciprocal space. The orbitals

$$\psi_{p\sigma}^k(r) = \frac{1}{\sqrt{N_{UC}}} \sum_L e^{ikL} \sum_\mu C_{p\mu\sigma}^k \rho_L(r) \quad (7.1)$$

are expanded in GTO basis functions $\mu(r-R_\mu-L) \equiv \rho_L(r)$ centered at atomic positions $R_\mu$ in direct lattice cells $L$ over all $N_{UC}$ unit cells. This results in the unrestricted Kohn-Sham equations,

$$F_{\sigma}^k C_{\sigma}^k = S_{\sigma}^k C_{\sigma}^k \varepsilon_{\sigma}^k, \quad (7.2)$$

which may be solved separately for each $k$ in the BZ. The same equations hold for the molecular case, where only $L = k = 0$ is a valid choice and $N_{UC}$ is one. Eq 7.2 contains the reciprocal space Kohn-Sham and the overlap matrices $F_{\sigma}^k$ and $S^k$, respectively, obtained as Fourier transforms of real space matrices

$$F_{\mu\nu\sigma}^L = \sum_L e^{ikT_L} F_{\mu\nu\sigma}^L \quad S_{\mu\nu}^L = \sum_L e^{ikT_L} S_{\mu\nu}^L. \quad (7.3)$$

The elements $F_{\mu\nu\sigma}^L$ contain three contributions: elements $T_{\mu\nu}^L$ of the kinetic energy matrix, elements $J_{\mu\nu}^L$ of the Coulomb matrix, and elements $X_{\mu\nu\sigma}^L$ of the exchange-correlation matrix,

$$F_{\mu\nu\sigma}^L = T_{\mu\nu}^L + J_{\mu\nu}^L + X_{\mu\nu\sigma}^L. \quad (7.4)$$

The total energy per unit cell $E$ is calculated as the sum of the kinetic $T$, Coulomb $J$, and exchange-correlation contribution $E_{XC}$,

$$E = T + J + E_{XC} \quad (7.5)$$

Calculations of kinetic energy and corresponding matrix elements are similar as in the molecular case. For the exchange-correlation term an adaptive numerical integration scheme is used [98]. The core of this method is a hierarchical spatial grouping of basis functions based on their spatial extents using an octree.

For the Coulomb term the periodic Resolution of Identity approximation is applied [100]. In this approach the total crystal electron density $\rho_{\text{cryst}}$ is approximated by auxiliary crystal electron density $\tilde{\rho}_{\text{cryst}}$

$$\rho_{\text{cryst}} \approx \tilde{\rho}_{\text{cryst}} = \sum_L \rho_L, \quad (7.6)$$

composed of unit cell auxiliary densities $\rho_L$ with

$$\rho_L = \sum_\alpha c^T \alpha_L, \quad (7.7)$$

where $\alpha$ denotes the vector of auxiliary basis functions. The vector of expansion coefficients $c$ is determined by minimizing the Coulomb repulsion $D$ of the residual density $\delta \rho = \rho - \tilde{\rho}$

$$D = \sum_L (\delta \rho | \delta \rho_L) = \sum_L (\rho - \tilde{\rho} | \rho_L - \tilde{\rho}_L). \quad (7.8)$$

The RI approximation allows to replace four-center electron repulsion integrals (ERIs) by two- and three-center ones. In this formalism, elements of Coulomb matrix $J^{L'}_{\mu\nu}$ are defined as

$$J^{L'}_{\mu\nu} = \sum_L (\mu\nu|L\tilde{\rho}_L - \rho_{nL}), \quad (7.9)$$

where $\rho_n$ denotes the unit cell nuclear charge distribution. The total Coulomb energy including the nuclear contribution is

$$J = \sum_{\mu\nu} D^{L}_{\mu\nu} J^{L}_{\mu\nu} - \frac{1}{2} \sum_L (\tilde{\rho} + \rho_n | L\tilde{\rho}_L - \rho_{nL}). \quad (7.10)$$

with the real space density matrix elements obtained by integration

$$D^{L}_{\mu\nu\sigma} = \frac{1}{V_k} \int_{BZ} D^{k}_{\mu\nu\sigma} e^{ikT} d^k, \quad (7.11)$$

of the reciprocal space density matrix

$$D^{k}_{\mu\nu\sigma} = \sum_p f^{p}_{\rho\sigma} (C^{k}_{\mu\rho\sigma})^* C^{k}_{\nu\rho\sigma} \quad (7.12)$$

over the BZ with volume $V_k$. The reciprocal space integral in Eq. 7.11 is evaluated numerically on a finite grid of k-points.

Eqs. 7.9 and 7.10 as well as RI scheme require evaluating infinite lattice sums of the form

$$\sum_L (\rho_1 | \rho_{2L}), \quad (7.13)$$

where the distribution $\rho_1$ in the central cell interacts with an infinite number of distributions $\rho_{2L}$, i.e., $\rho_2$ translated by all possible $L$. An important property of the RI-CFMM formulation in riper is that these sums are convergent. In the RI-CFMM scheme [86] the sum from Eq. 7.13 is partitioned into Crystal Far-Field (CFF) and Crystal Near-Field (CNF) parts. The CFF part contains summation over all direct space lattice vectors $L$ for which the overlap between the distributions $\rho_1$ and $\rho_{2L}$ is negligible. This part is very efficiently calculated using multipole expansions. The CNF contribution is evaluated using an octree based algorithm. In short, a cubic parent box enclosing all distribution centers of $\rho_1$ and $\rho_2$ is constructed that is large enough to yield a predefined number $n_{\text{targ}}$ of distribution centers per lowest level box. The parent box is successively subdivided in half along all Cartesian axes yielding the octree. In the next step all charge distributions comprising $\rho_1$ and $\rho_2$ are sorted into boxes basing on their extents so that distributions from well-separated boxes do not overlap. Two boxes are considered well-separated if the distance between their centers is greater than sum of their lengths times $0.5 \times wsicl$, where $wsicl$ is a predefined parameter $\geq 2$. Interactions between charges from well-separated boxes are calculated using the hierarchy of multipole expansions. The remaining contribution to the Coulomb term is obtained from direct integration. This approach results in linear scaling of the computational effort with the system size.

For the molecular case, a low-memory modification of the RI approximation is implemented within the riper module [99]. In the RI scheme minimization of the Coulomb repulsion of
the residual density, Eq. 7.8, yields a system of linear equations

\[ \mathbf{Vc} = \gamma, \]  

(7.14)

where \( \mathbf{V} \) is the Coulomb metric matrix with elements \( V_{\alpha\beta} = \langle \alpha | \beta \rangle \) representing Coulomb interaction between auxiliary basis functions and vector \( \gamma \) is defined as

\[ \gamma_{\alpha} = \sum_{\mu\nu} (\alpha | \mu\nu ) D_{\mu\nu}. \]  

(7.15)

In the new approach instead of attempting a direct solution of this inhomogeneous system of linear equations, the conjugate gradient (CG) iterative method is used. In order to decrease the number of CG iterations a preconditioning is employed, i.e., the original system of Eq. 7.14, is transformed using a preconditioner \( \mathbf{P} \) to an equivalent problem

\[ (\mathbf{P}^{-1} \mathbf{V}) \mathbf{c} = \mathbf{P}^{-1} \gamma \]  

(7.16)

with an improved condition number resulting in faster convergence of the CG method. Iterative CG solver employs preconditioners that are formed from the blocks of the \( \mathbf{V} \) matrix

- atomic block preconditioner

\[ P_{\alpha\beta}^{at} = \begin{cases} \langle \alpha I | \beta I \rangle , & I \in A_I, A_I \text{ are all atoms in molecule} \\ 0, & \text{otherwise} \end{cases} \]

- ss block preconditioner - \( P_{\alpha\beta}^{at} \cup P_{\alpha\beta}^{ss} \)

\[ P_{\alpha\beta}^{ss} = \begin{cases} \langle \alpha | \beta \rangle , & \alpha, \beta \in \{S\}, S \text{ are all s auxiliary basis functions} \\ 0, & \text{otherwise} \end{cases} \]

- sp block preconditioner - \( P_{\alpha\beta}^{at} \cup P_{\alpha\beta}^{sp} \)

\[ P_{\alpha\beta}^{sp} = \begin{cases} \langle \alpha | \beta \rangle , & \alpha, \beta \in \{S, P\}, P \text{ are all p auxiliary basis functions} \\ 0, & \text{otherwise} \end{cases} \]

corresponding to the strongest and most important interactions between the auxiliary basis functions such that \( \mathbf{P}^{-1} \mathbf{V} \approx \mathbf{I} \).

The costly matrix-vector products of the \( \mathbf{Vc} \) type that need to be evaluated in each CG iteration are not calculated directly. Instead, the linear scaling CFMM implementation presented above is applied to carry out this multiplication since the elements of the \( \mathbf{Vc} \) vector represent Coulomb interaction between auxiliary basis functions \( \alpha \) and an auxiliary density \( \hat{\rho} \)

\[ (\mathbf{Vc})_{\alpha} = \sum_{\beta} (\alpha | \beta ) c_{\beta} = \left( \alpha | \sum_{\beta} c_{\beta} \beta \right) = (\alpha | \hat{\rho}). \]  

(7.17)

Hence, in contrast to conventional RI neither the \( \mathbf{V} \) matrix nor its Cholesky factors need to be stored and thus significant memory savings are achieved.
7.3 How to Perform a Calculation

7.3.1 Prerequisites

Calculations using riper module require the control file and starting orbitals generated using define. DFT method needs to be specified in the $dft data group. All pure GGA functionals are supported for periodic calculations. For molecular case hybrid functionals are also available. Moreover, an auxiliary basis defined in the data group $jbas is required. For periodic calculations additional keywords specifying the system periodicity and number of k-points (if used) are to be manually added. Finally the $riper group containing all options specific to this module (including the preconditioned conjugate gradient options) needs to be created. The input preparation steps are summarized below. More detailed informations about all options are in section 21.2.7.

7.3.2 Creating the Input File

- Run define: in the geometry menu choose C1 symmetry. Create data groups $dft and $jbas using dft and ri, respectively, in the general menu.
- For periodic calculations specify system dimensionality by adding the keyword $periodic n and cell parameters with the keyword $cell. A 3D unit cell needs to be defined regardless of the system periodicity. Dimensions of the unit cell must be large enough to enclose all atoms specified in the coord data group.
- When k-points are used specify their number using keyword $kpoints.
- Create the $riper data group.

To run riper type

riper > riper.out &

7.3.3 Example

$periodic 3
$cell

18.5911 16.5747 16.5747 90. 90. 90.

$kpoints

nkpoints 3 3 3

$riper

# CFMM control options

northol 5
lmaxmom 20
nctrgt 10
wsicl 3.0
epsbext  1.0d-9  
locmult   on   
locmomadd 2   
pqmatdiag off   
pqstol     1.0d-8   
# PCG control options   
lpcg       off   
lcfmmpcg   on   
lmmpcg     20   
pctol      1.0d-9   
pctyp      sp

In this example the three-dimensional system is defined (\texttt{periodic 3}). The parameters of the unit cell are defined by the \texttt{cell} keyword, with lengths and angles specified in atomic units and degrees. All six parameters are required, regardless the system periodicity. In 2D and 1D systems the aperiodic dimensions need to be large enough to enclose all atoms specified in the \texttt{coord} data group. An uniform grid of 27 ($3 \cdot 3 \cdot 3$) k-points is used. The datagroup \texttt{riper} contains the most important options along with standard values for periodic calculations. More detailed discussion for each keyword is presented in section 21.2.7. If any of the options is missing, the standard value is used. For molecular calculations \texttt{locmult} should be switched to \texttt{off} as well as the \texttt{lpcg} since preconditioned CG iterative solver is implemented only for the molecular systems.
Chapter 8

Hartree–Fock and DFT Response Calculations: Stability, Dynamic Response Properties, and Excited States

8.1 Functionalities of \texttt{ESCf} and \texttt{EGRAD}

\texttt{ESCf} and \texttt{ERAD} are designed as efficient tools for response and excited state calculations on large molecules. \texttt{ESCf} serves to compute the following properties for HF and KS reference states:

- Eigenvalues of the electronic Hessian (stability analysis)
- Frequency-dependent polarizabilities and optical rotations
- Vertical electronic excitation energies
- Transition moments, oscillator and rotatory strengths of electronic excitations
  \[ \Rightarrow \text{UV-VIS and CD spectra} \]

Spin-restricted closed-shell and spin-unrestricted ground states (except for stability analysis) are supported. The RI-J approximation in conjunction with LDA, GGA, and meta-GGA (MGGA) functionals is implemented for all properties. Excitation energies and transition moments can be computed either within the full time-dependent HF (TDHF) or time-dependent DFT (TDDFT) formalisms or within the Tamm-Dancoff approximation (TDA).
8.2. THEORETICAL BACKGROUND

Furthermore, two-component relativistic Kramers-restricted closed-shell ground states are supported (vertical electronic excitation energies as well as transition moments, oscillator and rotatory strengths). LDA and GGA functionals are implemented in combination with the RI-J approximation.

Excited state first order properties can be evaluated analytically using egrad. They include:

- Gradients of the excited state energy with respect to nuclear positions
  ⇒ Excited state equilibrium structures (jobex), adiabatic excitation energies, emission spectra
- Excited state densities ⇒ Charge moments, population analysis
- Excited state force constants by numerical differentiation of gradients (using the script Numforce)

Moreover, analytical gradients of static and frequency-dependent polarizabilities are available from egrad. Together with vibrational normal modes from the aoforce or Numforce they are used to calculate vibrational Raman intensities. Excited state gradients for MGGA functionals are presently unavailable.

Again, ground states may be spin-restricted closed-shell or spin-unrestricted, RI-J is available, and either full TDDFT/TDHF or the TDA can be used. For further details we refer to a recent review [101].

8.2 Theoretical Background

We briefly state the basic working equations in the following, as far as required to understand the program output. For a more detailed treatment of the theory see refs. [24,101–104] and refs. therein. The following discussion is restricted to the one-component (nonrelativistic) treatment, for the sake of convenience.

The first-order frequency dependent response of the density matrix can be expanded as

\[ \gamma(x,x') = \sum_{ai} \{ X_{ai} \varphi_i(x) \varphi_i^*(x') + Y_{ai} \varphi_\alpha(x) \varphi_\alpha^*(x') \}. \]  

(8.1)

The (real) expansion coefficients \( X_{ai} \) and \( Y_{ai} \) are conveniently gathered in a “super-vector”

\[ |X,Y\rangle = \begin{pmatrix} X \\ Y \end{pmatrix} \]  

(8.2)

on \( L \), the linear space of products of occupied and virtual ground state MOs \( \varphi_i(x) \varphi_\alpha^*(x') \) plus their complex conjugates. \( X \) and \( Y \) describe the first-order change of the ground state MOs due to an external perturbation which is represented by \( |P,Q\rangle \) on \( L \). For example, if an oscillating electric dipole perturbation along the \( z \) axis is applied, \( |P,Q\rangle = |\mu_z\rangle \), where \( \mu \) is the electric dipole operator.
Next we define the $2 \times 2$ “super-matrices”

$$
\Lambda = \begin{pmatrix}
A & B \\
B & A
\end{pmatrix}, \quad
\Delta = \begin{pmatrix}
1 & 0 \\
0 & -1
\end{pmatrix},
$$

(8.3)

where the four-index quantities $A$ and $B$ are the so-called “orbital rotation Hessians”. Explicit expressions for the standard $A$ and $B$ can be found, e.g., in ref. [24]. For MGGA functionals, the linear response of the paramagnetic current density leads to additional XC kernel matrix elements, and subsequently to modified definitions of $A$ and $B$ [105]. The vector $|X,Y\rangle$ is determined as the solution of the TDHF/TDDFT response problem,

$$
(\Lambda - \omega \Delta)|X,Y\rangle = -|P,Q\rangle.
$$

(8.4)

If $|X_\alpha,Y_\alpha\rangle$ arises from an electric dipole perturbation $|\mu_\alpha\rangle$, the electronic dipole polarizability at frequency $\omega$ is

$$
\alpha_{\alpha\beta}(\omega) = -\langle X_\alpha,Y_\alpha|\mu_\beta \rangle,
$$

(8.5)

$\alpha, \beta \in \{x,y,z\}$. Similarly, if $|m_\alpha\rangle$ is a component of the magnetic dipole moment operator, the optical rotation is [106]

$$
\delta_{\alpha\beta}(\omega) = -\frac{c}{\omega} \text{Im}\langle X_\alpha,Y_\alpha|m_\beta \rangle,
$$

(8.6)

where $c$ is the light velocity.

Excitation energies $\Omega_n$ are the poles of the frequency-dependent density matrix response. They are thus the zeros of the operator on the left-hand side of Eq. (8.4),

$$
(\Lambda - \Omega_n \Delta)|X_n,Y_n\rangle = 0.
$$

(8.7)

The corresponding eigenvectors $|X_n,Y_n\rangle$ are the transition density matrices for a given excitation (also called “excitation vectors” in the following). They are required to be normalized according to

$$
\langle X_n,Y_n|\Delta|X_n,Y_n\rangle = 1.
$$

(8.8)

Transition moments are evaluated by taking the trace with one-particle operators, e.g.,

$$
\mu^{0n} = \langle X_n,Y_n|\mu \rangle
$$

(8.9)

for the electric and

$$
m^{0n} = \langle X_n,Y_n|m \rangle
$$

(8.10)

for the magnetic transition dipole moments.

The full TDHF/TDDFT formalism is gauge-invariant, i.e., the dipole-length and dipole-velocity gauges lead to the same transition dipole moments in the basis set limit. This can be used as a check for basis set quality in excited state calculations. The TDA can formally be derived as an approximation to full TDHF/TDDFT by constraining the $Y$ vectors to zero. For TDHF, the TDA is equivalent to configuration interaction including all single excitations from the HF reference (CIS). The TDA is not gauge invariant and does not satisfy the usual sum rules [102], but it is somewhat less affected by stability problems (see...
8.2. THEORETICAL BACKGROUND

below). For MGGA functionals, the response of the paramagnetic current density is required to ensure gauge invariance and is included by default.

Stability analysis of closed-shell electronic wavefunctions amounts to computing the lowest eigenvalues of the electric orbital rotation Hessian $A + B$, which decomposes into a singlet and a triplet part, and of the magnetic orbital rotation Hessian $A - B$. Note that $A - B$ is diagonal for non-hybrid and non-MGGA DFT, while $A + B$ generally is not. See refs. [23, 105, 107] for further details.

The two-component relativistic TDDFT eigenvalue problem for excitations of (Kramers-restricted) closed-shell systems taking (approximately) into account the effect of spin-orbit coupling is

$$M(X + Y)_n = \Omega^2_n (X + Y)_n. \quad (8.11)$$

$M$ is a Hermitian (complex) matrix containing spinor energy differences, Coulomb matrix elements as well as matrix elements of the two-component noncollinear exchange-correlation kernel. An explicit expression for $M$ can be found in ref. [108]. $(X + Y)_n$ are complex two-component excitation vectors.

Properties of excited states are defined as derivatives of the excited state energy with respect to an external perturbation. It is advantageous to consider a fully variational Lagrangian of the excited state energy [24],

$$L[X, Y, \Omega, C, Z, W] = E_{GS} + \langle X, Y|\Lambda|X, Y\rangle - \Omega(\langle X, Y|\Delta|X, Y\rangle - 1) + \sum_{ia} Z_{ia} F_{ia} - \sum_{pq} W_{pq}(S_{pq} - \delta_{pq}). \quad (8.12)$$

Here $E_{GS}$ denotes the ground state energy, $F$ and $S$ are the Fock and overlap matrices, respectively, and indices $p, q$ run over all, occupied and virtual MOs.

First, $L$ is made stationary with respect to all its parameters. The additional Lagrange multipliers $Z$ and $W$ enforce that the MOs satisfy the ground state HF/KS equations and are orthonormal. $Z$ is the so-called $Z$-vector, while $W$ turns out to be the excited state energy-weighted density matrix. Computation of $Z$ and $W$ requires the solution of a single static TDHF/TKS response equation (8.4), also called coupled and perturbed HF/KS equation. Once the relaxed densities have been computed, excited state properties are obtained by simple contraction with derivative integrals in the atomic orbital (AO) basis. Thus, computation of excited state gradients is more expensive than that of ground state gradients only by a constant factor which is usually in the range of $1 \ldots 4$.

TDHF/TDDFT expressions for components of the frequency-dependent polarizability $\alpha_{\alpha\beta}(\omega)$ can also be reformulated as variational polarizability Lagrangians [109]

$$L^{\alpha\beta}[X_\alpha, Y_\alpha, X_\beta, Y_\beta, C, Z^{\alpha\beta}, W^{\alpha\beta}](\omega) = \langle X_\alpha, Y_\alpha|\Lambda - \omega\Delta|X_\beta, Y_\beta\rangle + \langle X_\alpha, Y_\alpha|\mu_\beta\rangle + \langle \mu_\alpha|X_\beta, Y_\beta\rangle + \sum_{ia} Z^{\alpha\beta}_{ia} F_{ia} - \sum_{pq} W^{\alpha\beta}_{pq}(S_{pq} - \delta_{pq}). \quad (8.13)$$

The stationary point of $L^{\alpha\beta}(\omega)$ equals to $-\alpha_{\alpha\beta}(\omega)$. The requirement that $L^{\alpha\beta}(\omega)$ be stationary with respect to all variational parameters determines the Lagrange multipliers $Z^{\alpha\beta}$.
and \( W^{\alpha\beta} \). All polarizability components \( \alpha\beta \) are processed simultaneously which allows for computation of polarizability derivatives at the computational cost which is only 2–3 higher than for the electronic polarizability itself.

Within TDDFT and TDHF, the \( X \) and \( Y \) coefficients are normalized as follows:

\[
\sum_{ia} (|X_{ia}|^2 - |Y_{ia}|^2) = 1,
\]

where \( i \) and \( a \) label occupied and virtual MOs, respectively. Thus, the squared "coefficient" of a single electron excitation from orbital \( i \) to orbital \( a \) can be defined as

\[
|c_{ia}|^2 = |X_{ia}|^2 - |Y_{ia}|^2.
\]

\texttt{escf} prints out \(|c_{ia}|^2 \times 100\) starting with the largest coefficient, until the sum of the coefficients is 0.9 or greater. TDA is contained as special case with \( Y_{ia}=0 \).

### 8.3 Implementation

Without giving details, we discuss features of the implementation in \texttt{escf} and \texttt{egrad} that matter for applications. The interested reader is referred to the refs. given in the program headers as well as ref. [110].

**Simultaneous vector iteration.** The solutions of Eqs. (8.4) and (8.7) (Eq. (8.11)) are expanded in a subspace of \( L \) which is iteratively expanded (Davidson method [111]). The iteration is stopped when the Euclidean norm of the residual vector is smaller than \( 10^{-k} \). The default for \( k \) is 5, which usually gives excitation energies accurate to \( 8 - 10 \) digits and properties accurate to \( 4 - 5 \) digits \( (k \) can be changed by specifying \$\texttt{rpaconv} \( k \)). Several roots, i.e., several excited states or frequencies can be treated simultaneously, which is very effective and permits the calculation of whole excitation spectra and dispersion curves. During the iteration, the vectors are kept on scratch files \texttt{vfile_<IR>}, \texttt{wfile_<IR>}, and/or \texttt{rhs_<IR>}, where \( \texttt{IR} \) denotes an IRREP of the point group (see below). Before the programs terminate, the converged vectors are written onto formatted files \( \langle \text{type} \rangle \langle \text{IR} \rangle \), where \( \text{type} \) is an abbreviation for the type of response calculation performed (cf. \$\texttt{scfinstab}). Given these files in the working directory, \texttt{escf} and \texttt{egrad} calculations can be restarted or continued, e.g., with a larger number of roots.

**Integral direct algorithm.** In the iterative method outlined above, the super-matrices \( A \) and \( B \) never need to be set up explicitly; only the products of \( A \) and \( B \) with some suitable basis vectors are required. These matrix-vector-products are evaluated very efficiently in the AO basis, because the required four-index integrals can be computed "on the fly" and need not be transformed or stored on disk. In addition, prescreening techniques based on rigorous bounds are straightforward to apply. This leads to a low-order scaling \( O(N^2) - O(N) \) for the time-determining steps. Due to the similarity to ground state fock matrix construction, the same keywords are used to control these steps as in semi-direct SCF, namely \$\texttt{sthime}, \$\texttt{qsDen}, \$\texttt{thgamma}, and \$\texttt{ehi}.
8.4. HOW TO PERFORM

$thize, $scfintunit, see Chapter 6. The same is true for DFT and RI keywords such as $dft, $ridft, $ricore.

Point group symmetry. escf and egrad can exploit point group symmetry for all finite point groups (with up to 99-fold symmetry axes, $\rightarrow$ symmetry). The response and eigenvalue problems (8.4) and (8.7) decompose into separate problems for each IRREP that are solved independently. For excited state and instability calculations, it is thus necessary to specify the IRREPs to be treated ($ssoes$, see below). For response calculations, the perturbation is automatically subdued into irreducible components. The overall speedup compared to $C_1$ symmetry is approximately $1/g$, where $g$ denotes the point group order. For spin-restricted closed-shell ground states, spin symmetry is used to further reduce the dimension of the response and eigenvalue problems by a factor of 2. Point group symmetry cannot be exploited in two-component calculations.

Other features. escf and egrad fully support external fields (using the keyword $electrostatic field; specify geofield on in $fldopt), point charges (using the keyword $point_charges), and effective core potentials (using $ecp). In escf calculations, occupied and virtual MOs can be frozen (using $freeze).

8.4 How to Perform

The most convenient way to set up an escf or egrad calculation is to use the ex option of the last ("general") define menu, see Chapter 4. define will automatically provide most of the keywords discussed below.

A large number of (not necessarily realistic) sample inputs is contained in the escf and egrad subdirectories of the test suite (TURBOTESE directory).

8.4.1 Preliminaries

All response calculations require a complete set of converged (occupied and virtual) SCF MOs. It is strongly recommended to use well converged MOs, since the error in the ground-state wavefunction enters linearly in all response properties. Thus, before starting escf or egrad, specify the keywords

$scfconv 7
$denconv 1d-7

in control, perform a dscf statistics run, if semi-direct integral processing is to be used (see Chapter 3.1), and (re-)run dscf or ridft,

dscf > dscf.out & or
ridft > ridft.out & in case of RI-J.
The above tight convergence criteria are also recommended for excited state geometry optimizations. To perform a two-component TDDFT calculation, the two-component version of \texttt{ridft} has to be run before (see Chapter 6.4) using the keywords $\texttt{soghf}$ and $\texttt{kramers}$.

### 8.4.2 Polarizabilities and Optical Rotations

The calculation of dynamic polarizabilities is controlled by the keyword

\texttt{$\texttt{scfinstab dynpol unit}$}  
\texttt{list of frequencies}

\texttt{unit} specifies the unit of the following frequencies and may be \texttt{ev}, \texttt{nm}, \texttt{1/cm}, or \texttt{a.u.} (default).

The frequencies may be either purely real or purely imaginary. For example, to calculate dynamic polarizabilities at 590 nm and 400 i nm (i is the imaginary unit), specify

\texttt{$\texttt{scfinstab dynpol nm}$}  
\texttt{590}  
\texttt{400 i}

and run \texttt{escf},

\texttt{escf > escf.out &}

The resulting polarizabilities and rotatory dispersions are given in a.u. in the program output (\texttt{escf.out} in the above example).

The conversion of the optical rotation in a.u. to the specific rotation \([\alpha]_\omega\) in \text{deg} \cdot \text{dm} \cdot (\text{g/cc})^{-1} is given in Eq. (15) of ref. [106].

\[ [\alpha]_\omega = C \cdot \delta(\omega) \quad (8.16) \]

where \(C = 1.343 \cdot 10^{-4} \omega^2 / M\) with \(M\) being the molar mass in g/mol, \(\omega\) the frequency in cm\(^{-1}\), and \(\delta(\omega)\) is 1/3 trace of the electronic rotatory dispersion tensor given in atomic units.

Please note, that \(\delta(\omega)\) has the wrong sign in older TURBOMOLE versions. It has been corrected in version 6.2.

Note that convergence problems may occur if a frequency is close to an electronic excitation energy. This is a consequence of the (physical) fact that the response diverges at the excitation energies, and not a problem of the algorithm.

Static polarizabilities are calculated most efficiently by specifying

\texttt{$\texttt{scfinstab polly}$}

before starting \texttt{escf}. 
8.4. HOW TO PERFORM

8.4.3 Stability Analysis

Stability analysis of spin-restricted closed-shell ground states is enabled by

\$scfinstab singlet
  for singlet instabilities,

\$scfinstab triplet
  for triplet instabilities (most common), and

\$scfinstab non-real
  for non-real instabilities.

After that, it is necessary to specify the IRREPs of the electronic Hessian eigenvectors ("orbital rotations") to be considered. Without additional knowledge of the system one usually needs to calculate the lowest eigenvalue within every IRREP:

\$soes all 1

Positivity of the lowest eigenvalues in all IRREPs is sufficient for stability of the ground state solution. If one is interested in, say, the lowest eigenvalues in IRREPs \(\text{eg}\) and \(\text{t}_2\text{g}\) only, one may specify:

\$soes
  eg 1
  t2g 1

Triplet instabilities in the totally symmetric IRREP indicate open shell diradical states (singlet or triplet). In this case, start MOs for spin-symmetry broken UHF or UKS ground state calculation can be generated by specifying

\$start vector generation

\$escf will provide the start MOs (\(\rightarrow\) \$uhfmo_alpha, \$uhfmo_beta) as well as occupation numbers (\(\rightarrow\) \$alpha shells, \$beta shells) for a spin-unrestricted calculation with equal numbers of \(\alpha\) and \(\beta\) electrons (pseudo-singlet occupation).

8.4.4 Vertical Excitation and CD Spectra

The calculation of excited states within the TDHF(RPA)/TDDFT approach is enabled by

\$scfinstab rpas
  for closed-shell singlet excitations,

\$scfinstab rpat
  for closed-shell triplet excitations, and
CHAPTER 8. HF AND DFT RESPONSE CALCULATIONS

$\textit{scfinstab urpa}$

for excitations out of spin-unrestricted reference states.

If it is intended to use the TDA instead, specify

$\textit{scfinstab ciss}$

for closed-shell singlet excitations,

$\textit{scfinstab cist}$

for closed-shell triplet excitations,

$\textit{scfinstab ucis}$

for excitations out of spin-unrestricted reference states, and

$\textit{scfinstab spinflip}$

for spin-flip ($z$-component of the total spin changes by $\pm 1$) excitations out of spin-unrestricted reference states. For details concerning the theory see ref. [112]. In practice, this functionality can be used for the calculation of triplet-singlet, quartet-doublet, ... excitations (see ref. [113] also for further information about the implementation). It is only available within the TDA in combination with LDA functionals and the HF exchange. It is strongly recommended to increase $\textit{escfiterlimit}$.

In the two-component case, specify

$\textit{scfinstab soghf}$

for two-component excitation energy calculations on Kramers-restricted closed-shell systems. [108] The keywords $\textit{soghf}$ and $\textit{kramers}$ also have to be set. The implementation is only available in combination with LDA and GGA functionals.

Next, the IRREPs of the excitations need to be defined, which is again accomplished using $\textit{ssoes}$. For example, to calculate the 17 lowest excitations in IRREP $b_1g$, the 23 lowest excitations in IRREP $e_u$, and all excitations in IRREP $t_2g$, use

$\textit{ssoes}$

\begin{verbatim}
b1g 17
eu 23
t2g all
\end{verbatim}

and run $\texttt{escf}$. Since point group symmetry cannot be exploited in two-component calculations, there is only the totally symmetric IRREP $a$.

Note that $\textit{ssoes}$ specifies the IRREP of the excitation vector which is not necessarily identical to the IRREP of the excited state(s) involved. In general, the IRREP(s) of the excitation(s) from the ground to an excited state is given by the direct product of the IRREPs of the two states. For example, to calculate the first $A_2$ state in a $C_{2v}$-symmetric molecule with a $B_2$ (open-shell) ground state, it is necessary to specify

$\textit{ssoes}$

\begin{verbatim}
b1 1
\end{verbatim}
The number of excitations that have to be calculated in order to cover a certain spectral range is often difficult to determine in advance. The total number of excitations within each IRREP as provided by the define ex menu may give some hint. A good strategy is to start with a smaller number of excitations and, if necessary, perform a second scf run on a larger number of states using the already converged excitation vectors as input.

To compute absorption and CD spectra, it is often sufficient to include optically allowed transitions only. This leads to substantial reduction of computational effort for molecules with higher symmetry. For example, in the UV-VIS spectrum of an $O_h$ symmetric molecule, only $t_{1u}$ excitations are optically allowed. The IRREPs of the electric and magnetic dipole moments as well as of the electric quadrupole moment are displayed automatically in the define ex menu.

If a large number of states is to be calculated, it is highly recommended to provide extra memory by specifying

```
$rpacor m
```

the integer $m$ being the core memory size in megabytes (default is 20). The larger $m$, the more vectors can be processed simultaneously without re-calculation of integrals. As a rule of thumb, $m$ should be ca. 90% of the available main memory. If RI-J is used ($$ridft$$), it is recommended to set $$ricore$$ to a small value and $$rpacor$$ to a large value if the number of states is large, and vice versa if it is small. Since two-component calculations are more demanding concerning computation time and required memory it is strongly recommended to increase $$rpacor$$.

By specifying

```
$spectrum unit and/or
$cdspectrum unit
```

a list of excitation energies and oscillator and/or rotatory strengths of the optically allowed transitions is written onto file spectrum and/or cdspectrum. As above, unit specifies the energy unit and may be ev, nm, 1/cm, or a.u. (default). The files spectrum and cdspectrum may conveniently be used for further processing, e.g., using a plotting program such as Gnuplot.

By specifying

```
$curswitchdisengage
```

inclusion of the current-density response for MGGA calculations is disabled. *Note that the results of calculations using this flag will no longer be gauge-invariant and will differ from results obtained with the standard gauge-invariant implementation.*
8.4.5 Excited State Geometry Optimizations

The input for computing excited state gradients and properties using `egrad` is exactly the same as for an excited state calculation using `escf`, see the previous section. Gradients and properties are calculated only for one state at a time. By default, this is the highest excitation specified by `$soes` (only one IRREP is allowed). Sometimes, e.g. close to excited state intersections, it may be necessary to include higher excited states in the initial excitation vector calculation to prevent root flipping. This is accomplished using

```
$exopt n
```

which explicitly enforces treatment of the n-th state; n must be less or equal the number of states specified in `$soes`.

After the input for the ground and excited state calculations has been set up, an excited state geometry optimization can be started by issuing the command

```
nohup jobex -ex &
```

The option `-ex` forces `jobex` to call `egrad` instead of `grad` (or `rdgrad` if `-ri` is also specified). In each geometry step, the excitation energy is written on the fourth column in `$energy`, and the data group `$last excitation energy change` is updated. Otherwise, the excited state optimization proceeds in exactly the same way as a ground state optimization (see Chapter 3.1).

8.4.6 Excited State Force Constant Calculations

Excited state vibrational frequencies can be calculated by numerical differentiation of analytic gradients using `Numforce` (see Chapter 14). A `Numforce` calculation for an excited state may be started by the command

```
nohup NumForce -ex n > force.out &
```

where n is the number of the excited state in $C_1$ symmetry. In order to determine n, it is recommended to perform an `escf` calculation in $C_1$ symmetry. Note that numerical calculation of excited state force constants is likely to fail if there are other states nearby (in $C_1$), because the roots may flip when the molecule is distorted. Note also that it may be necessary to include higher excited states (using `$exopt`, see above) in $C_1$ calculations of molecules with higher symmetry in order to enforce convergence to the correct state. In any case, it should be checked that the energy change due to the displacements (available in the `numforce/KraftWerk/*_log` files) is reasonably small.

For a `Numforce` run, the convergence criteria should be tightened. It is recommended to use at least

```
$scfconv 8
```
in all Numforce calculations. Other Numforce options such as -central, -d, -np work in exactly the same way as they do for ground states.

8.4.7 Polarizability Derivatives and Raman Spectra

Calculations of polarizability derivatives by the egrad program use the same specifications in the $scfinstab data group as polarizability calculations by escf.

$scfinstab polly

specifies derivatives of the static polarizability, while

$scfinstab dynpol unit

frequency

requests derivatives of the dynamical polarizability at the given frequency. Note that, unlike polarizability calculations, multiple frequencies are not allowed. Polarizability derivatives have to be projected onto vibrational normal modes to obtain Raman intensities, see Chapter 14 for further details.
Chapter 9

Second-order Møller–Plesset Perturbation Theory

Preliminary note

TURBOMOLE offers three programs for MP2 energy and gradient calculations. A "conventional" implementation [114], mpgrad, based on the calculation of four-center integrals, an implementations which uses the resolution-of-the-identity (RI) approximation [115] as part the RI-CC2 program [8], ricc2, and an implementation which is meant for very large (> 100 atoms and > 3000 basis functions) single point MP2 energy calculations. The latter program used in addition to a local RI also a hybrid OSV-PNO approximation reduce the scaling of the computational costs with the system size to (at most) $O(N^3)$ but has for small and medium-sized systems a larger prefactor than ricc2.

9.1 Functionalities of MPGRAD, and RICC2, and PNOCCSD

Functionalities of mpgrad:

- Calculation of MP2 energies and/or MP2 gradients for RHF and UHF wave functions.
- The frozen core approximation (possibility to exclude low-lying orbitals from the MP2 treatment) is implemented only for MP2 energies.
- Exploitation of symmetry of all point groups.
- Can be used sequentially or parallel.
- Can be combined with the COSMO solvation model (see chapter 20 for details).

(Presently restricted to sequential calculations.)

Functionalities of ricc2 at the MP2-level:
9.1. FUNCTIONALITIES OF MPGRAD, RICC2, AND PNOCCSD

- Calculation of MP2 energies and/or gradients for RHF and UHF wave functions within the efficient RI-approximation (RI-MP2). In geometry optimizations and vibrational frequency calculations (with NumForce) it can be combined with RI-JK-SCF for the Hartree-Fock reference calculation.

- The frozen core approximation is implemented for both energies and gradients.

- RI-MP2 needs optimised auxiliary basis sets, which are available for most standard basis sets as e.g. SVP, TZVP, TZVPP, QZVPP as well as for the (aug-)cc-p(wC)VXZ (X = D, T, Q, 5) basis set series (for Al–Ar also for the (aug-)cc-p(wC)V(X+d)Z series and for p-block elements Ga–Rn also the respective ECP basis set series (-pp)).

- Exploitation of symmetry for all point groups for MP2 energies and gradients.

- Can be combined with the COSMO solvation model (see chapter 20 for details).

- Runs sequentially and parallel (with MPI or OpenMP)

- Contains an implementation of explicitly correlated MP2-F12 methods (presently restricted to energies and the $C_1$ point group).

- Can for open-shell calculations be used with UHF and single-determinant high-spin ROHF reference wavefunctions. (ROHF-MP2 presently limited to energies.)

- Energies and gradients for the spin-component scaled SCS- and SOS-MP2 approaches, including a Laplace-transformed implementation of SOS-MP2 with $O(N^4)$ scaling computation costs.

- Static polarizabilities (currently restricted to closed-shell reference wavefunctions and the sequential and SMP versions; cannot yet be combined with spin-component scaling), see Chapter 10.5 for a description of the input

- See Chapter 10 for further details.

Functionalities of pnoccad:

- Currently restricted to MP2 and DFT double hybrid (e.g. B2PLYP) single point energy calculations with a closed shell reference determinant and $C_1$ the point group.

- Runs sequentially and parallel (with MPI or OpenMP and hybrid MPI/OpenMP).

- Contains an implementation of explicitly correlated PNO-MP2-F12 methods.

- See Section 9.7 for further details.

9.1.1 How to quote

- For calculations with mpgrad:
  
• For calculations with ricc2:
  CC2 excitation energy calculations on large molecules using the resolution of the identity approximation. C. Hättig and F. Weigend;
• for MPI parallel calculations with ricc2 in addition:
• for MP2-F12 calculations in addition:
• for $O(N^4)$-scaling LT-SOS-MP2 calculations:
• for SCS-MP2 calculations:
• for RI-MP2 polarizabilities:

9.2 Some Theory

Second-order Møller–Plesset Perturbation Theory (MP2) corrects errors introduced by the mean-field ansatz of the Hartree–Fock (HF) theory, the perturbation operator is just the difference of the exact and the HF Hamiltonian. One straightforward obtains the MP2 energy:

$$E_{\text{MP2}} = \frac{1}{4} \sum_{ia,jb} t^2_{ab}(ij||ab),$$

(9.1)
with the t-amplitudes
\[ t_{ij}^{ab} = \frac{\langle ij|ab \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \] (9.2)
i and j denote occupied, a and b virtual orbitals, \( \epsilon_p \) are the corresponding orbital energies, \( \langle ij|ab \rangle = \langle ij|ab \rangle - \langle ij|ba \rangle \) are four-center-two-electron integrals in a commonly used notation.

MP2 gradients (necessary for optimisation of structure parameters at the MP2 level) are calculated as analytical derivatives of the MP2 energy with respect to nuclear coordinates; calculation of these derivatives also yields the first order perturbed wave function, expressed as "MP2 density matrix", in analogy to the HF density matrix. MP2 corrections of properties like electric moments or atomic populations are obtained in the same way as for the HF level, the HF density matrix is just replaced by the MP2 density matrix.

The "resolution of the identity (RI) approximation" means expansion of products of virtual and occupied orbitals by expansions of so-called "auxiliary functions". Calculation and transformation of four-center-two-electron integrals (see above) is replaced by that of three-center integrals, which leads to computational savings of RI-MP2 compared to a conventional MP2 calculations by a factor of ca. 5 (small basis sets like SVP) to ca. 10 (large basis sets like TZVPP) and more (for quadruple-\( \zeta \) and larger basis sets). The RI errors (error due to the RI approximation) are with optimised auxiliary basis sets small and well documented [116,117]. The use of the mpgrad program is therefore only recommended for reference calculations or if suitable auxiliary basis sets are not available.

9.3 How to Prepare and Perform MP2 Calculations

Prerequisites

Calculations with mpgrad, ricc2 or pnoccsd require

- a converged SCF calculation with the one-electron density convergence threshold set to $\$denconv 1.d-7 or less
- the maximum core memory the program is allowed to allocate should be defined in the data group $\$maxcor (in MB); the recommended value is ca. 3/4 of the available (physical) core memory at most.
- orbitals to be excluded from the correlation treatment have to be specified in data group $\$freeze
- for mpgrad the calculation of gradients is omitted by adding the flag $\$mp2energy to the control file; in this case only the MP2 energy is calculated.

Calculations with ricc2 and pnoccsd moreover use

- an auxiliary basis defined in the data groups $\$atoms and $\$cbas. If not specified before the start of ricc2 or pnoccsd the programs will assign the default auxiliary basis which have the same names as the one-electron basis sets and will try to extract
them from the TURBOMOLE basis set library. If this succeeds the assignment and auxiliary basis sets are stored in the control file else the programs will stop.

This is not needed for mpgrad, but here one needs

- a specification for scratch files and their size in data group $mointunit (see Section 21.2.16)
- and the number of passes for integral evaluations and transformations in data group $straloop

For explicitly-correlated MP2-F12 calculations one needs—depending the details of the applied approximations—additionally a so-called complementary auxiliary basis set (CABS, defined in $cabs) and a RI-SCF auxiliary basis set defined in $jkbas.

Calculations with ricc2 and pnoccsd

1. RI-MP2 calculations require the specification of auxiliary basis sets ($cbas) and a converged SCF calculation with the one-electron density convergence threshold set to $denconv 1.d-7 or less. In addition, the options $freeze (frozen core approximation) and $maxcor (maximum core memory usage) should be set. All these settings can be done during the input generation with the program define under the entries mp2/cc or pnocc of the last main menu.
2. The ricc2 program requires the data group:
   $ricc2
   mp2
   geoopt model=mp2
   (Where the last line should only be included if the calculation of gradients is needed e.g. in geometry optimizations.) This can be prepared with define in the menu cc.
3. The pnoccsd program does not require any special input for a MP2 calculation with default thresholds but it is recommended to specify PNO truncation threshold explicitly in the data group:
   $pnoccsd
   mp2
   tolpno= 1.0E-7
   This can be prepared with define in the menu pnoc.
4. For explicitly-correlated MP2-F12 calculations with ricc2 or pnoccsd also the data groups $rir12 and $lcg are needed.
5. Start a single ricc2 and pnoccsd calculations respectively with the commands ricc2 and pnoccsd.
6. For optimisation of structure parameters at the RI-MP2 level with ricc2 use the command jobex -level cc2. For geometry optimizations with RI-JK-SCF as reference
9.3. HOW TO PREPARE AND PERFORM MP2 CALCULATIONS

for RI-MP2 with the ridft and ricc2 binaries the additional option -rijk has to be given.

7. The combination of RI-MP2 with RI-JK-SCF can lead to significant computational savings in particular for geometry optimizations for small and medium sized molecules with large basis sets (quadruple-ζ and beyond) or basis sets with diffuse functions (e.g. the aug-cc-pVXZ basis set families). For large molecules with TZVPP or similar basis sets, conventional direct SCF calculations are usually more efficient.

8. With ricc2 and pnoccsd spin-component scaled SCS- or SOS-RI-MP2 calculations can be carried out by adding in the $ricc2 (for ricc2) or $pnoccsd (for pnoccsd data group the line

\[ \texttt{scs cos=1.2d0 css=0.3333d0} \]

where the two parameters are the scaling factors for, respectively, the opposite- and same-spin contribution. The specification of the scaling factors is optional; the default values are cos=6/5 and css=1/3 as recommended by S. Grimme in J. Chem. Phys. 118 (2003) 9095. The abbreviation sos can be used for SOS-MP2 calculations with cos=1.3 and css=0.0 (Y., Jung, R.C. Lochan, A.D. Dutoi, and M. Head-Gordon, J. Chem. Phys. 121 (2004) 9793.). SOS-MP2 is in ricc2 implemented with $O(N^4)$ scaling costs. For such calculations the data group $laplace$ has to be added.

9. For technical recommendations and additional options for parallel RI-MP2 and PNO-MP2 calculations with the ricc2 and pnoccsd programs see Secs. 3.2 and 10.6 and 9.7.

MP2 calculations with a ROHF reference state

With the program ricc2 it is possible to compute MP2 (and spin-component scaled) MP2 energies with single-determinant restricted open-shell reference wavefunctions. No additional input is required apart from the usual ROHF input for the dscf and ridft programs and a standard MPn or CC input for ricc2.

TURBOMOLEs Hartree-Fock codes can handle within the ROHF framework many cases, which include beside common high- and low-spin configuration state functions also weighted averages of high-spin CSFs (see Sec. 6.3 for further details). The Møller-Plesset perturbation theory and coupled cluster functionalities implemented in ricc2 require a single-determinant reference state and can thus only deal with high-spin open-shell cases (not averaged):

- The spins of all \( n_{\text{open}} \) unpaired electrons are parallel (\( \alpha \) spin will be assumed) so that the ROHF reference state has the spin multiplicity \( n_{\text{open}} + 1 \).
- There must be only one type of open shells and all orbitals in this shell must have the occupation number 1.
- For the single electron case (i.e. doublets) the Roothaan parameters are \( a = b = 0 \), for high-spin cases with more than one unpaired electron the Roothaan parameters must be set to \( a = 1 \) and \( b = 2 \).
For non-abelian points groups this implies that shells with degenerate orbitals (as e.g. \( t_1 \) in point group \( I \)) must be half-filled. An average over the different (symmetry-equivalent or inequivalent) high-spin determinants that are obtained when a shell of degenerate orbitals is less than or more than half-filled is not possible with single-point \textit{ricc2} calculations.

For states with less than or more than half-filled shells of degenerate orbitals the calculations half to be done in a point group that lifts the degeneracy such that it becomes possible to assign integer occupation numbers. A symmetry-breaking of the orbitals can be avoided by doing the Hartree-Fock calculation in the full point group. The input (and MO coefficients) can then be transformed to a lower point group using define only for the \textit{ricc2} calculation.

Calculations with \textit{mpgrad}

1. Add \texttt{denconv 1.d-7} to the \textit{control} file and perform a \texttt{dscf} run.

2. If any orbitals are decided to be excluded from MP2 treatment, add data group \texttt{$freeze$} manually to the \textit{control} file, see also Section 21.2.16.

3. For preparation of an \textit{mpgrad} run use the script \texttt{Mp2prep}:
   \begin{verbatim}
   mp2prep -e/g -m memory -p discspace [scratch file directory]
   \end{verbatim}
   As an example, with the command
   \begin{verbatim}
   mp2prep -e -m 100 -p 1000 /work
   \end{verbatim}
   an MP2-energy calculation is prepared, the amount of available core memory is restricted to 100 MB, the MOs are blocked, so that integral scratch files—located in the directory /work—do not need more than 1000 Mb. The number of blocks, i.e. the number of passes with repeated integral evaluations, is written to the \textit{control} file ($\texttt{straloop}$) as well as the specification of scratch files ($\texttt{mointunit}$, see Section 21.2.16). Note: less disc space means more passes and thus lower efficiency of \textit{mpgrad}, but due to the technical limitations \textit{discspace} should be limited to values < 16Gb to avoid integer overflow errors. Settings obtained by \texttt{mp2prep} may be changed manually. You may change the number of passes in $\texttt{straloop}$ by editing the \textit{control} file (e.g. if the originally intended disc space is not available). To adapt the size of scratch files add \texttt{$statistics mpgrad} to \textit{control} file and start an \textit{mpgrad} statistics run with the command \texttt{mpgrad}.

4. Start a single \textit{mpgrad} calculation with the command \texttt{mpgrad}.

5. For optimisation of structure parameters at the (non-RI-) MP2 level use the command \texttt{jobex -level mp2}. Note, that the frozen core approximation is ignored in this case.
9.4 General Comments on MP2 Calculations, Practical Hints

Recommendations

- It is well-known, that perturbation theory yields reliable results only, if the perturbation is small. This is also valid for MP2, which means, that MP2 improves HF results only, if HF already provides a fairly good solution to the problem. If HF fails, e.g. in case of partially filled \( d \)-shells, MP2 usually will also fail and should not be used in this case.

- MP2 results are known to converge very slowly with increasing basis sets, in particular slowly with increasing \( l \)-quantum number of the basis set expansion. Thus for reliable results the use of TZVPP basis sets (or higher) is recommended. When using SVP basis sets a qualitative trend can be expected at the most. Basis sets much larger than TZVPP usually do not significantly improve geometries of bonded systems, but still can improve the energetic description. For non–bonded systems larger basis sets (especially, with more diffuse functions) are needed.

- It is recommended to exclude all non-valence orbitals from MP2 calculations, as neither the TURBOMOLE standard basis sets SVP, TZVPP, and QZVPP nor the cc-pVXZ basis set families (with \( X=\text{D,T,Q,5,6} \)) are designed for correlation treatment of inner shells (for this purpose polarisation functions for the inner shells are needed). The default selection for frozen core orbitals in Define (orbitals below -3 a.u. are frozen) provides a reasonable guess. If core orbitals are included in the correlation treatment, it is recommended to use basis sets with additional tight correlation functions as e.g. the cc-pwCVXZ and cc-pCVXZ basis set families.

- RI-MP2: We strongly recommend the use of auxiliary basis sets optimized for the corresponding (MO) basis sets.

RI-MP2 calculations with the ricc2 program: All what is needed for a RI-MP2 gradient calculation with the ricc2 program is a $ricc2$ data group with the entry geoopt model=mp2. If you want only the RI-MP2 energy for a single point use as option just mp2. To activate in MP2 energy calculations the evaluation of the \( D_1 \) diagnostic (for details see Sec. 10.1.), use instead mp2 d1diag. (Note that the calculation of the \( D_1 \) diagnostic increases the costs compared to a MP2 energy evaluation by about a factor of three.)

Comments on the Output

- Most important output for ricc2, pnoccsd, and mpgrad are of course the MP2(\(^{+}\)HF) energies (written to standard output and additionally to the file energy) and MP2(\(^{+}\)HF) gradients (written to the file gradient).
• In case of MP2 gradient calculations the modules also calculate the MP2 dipole moment from the MP2 density matrix (note, that in case of mpgrad a frozen core orbital specification is ignored for gradient calculations and thus for MP2 dipole moments).

Further output contains indications of the suitability of the (HF+MP2) treatment.

• As discussed above, reliable (HF+MP2) results are in line with small MP2 corrections. The size of the MP2 correction is characterised by the t-amplitudes, as evident from the above equations. mpgrad by default plots the five largest t-amplitudes as well as the five largest norms of t-amplitudes for fixed $i$ and $j$, rimp2 does the same upon request, if $tplot$ is added to control file. More or less than five t-amplitudes will be plotted for $tplot n$, where $n$ denotes the number of largest amplitudes to be plotted. It is up to the user to decide from these quantities, whether the (SCF+MP2) treatment is suited for the present problem or not. Unfortunately, it is not possible to define a threshold, which distinguishes a "good" and a "bad" MP2-case, since the value of individual t-amplitudes are not orbital-invariant, but depend on the orbital basis (and thereby under certain circumstances even on the orientation). Example: the largest norm of t-amplitudes for the Cu-atom ($d^{10} s^1$, "good" MP2-case) amounts to ca. 0.06, that of the Ni-atom ($d^8 s^2$, "bad" MP2 case) is ca. 0.14.

• A more descriptive criterion may be derived from the MP2 density matrix. The eigenvalues of this matrix reflect the changes in occupation numbers resulting from the MP2 treatment, compared to the SCF density matrix, where occupation numbers are either one (two for RHF) or zero. Small changes mean small corrections to HF and thus suitability of the (HF+MP2) method for the given problem. In case of gradient calculations rimp2 displays by default the largest eigenvalue of the MP2 density matrix, i.e. the largest change in occupation numbers (in %). All eigenvalues are shown, if $mp2occ$ is added to the control file. For main group compounds largest changes in occupation numbers of ca. 5 % or less are typical, for $d^{10}$ metal compounds somewhat higher values are tolerable.

• A similar idea is pursued by the $D_2$ and $D_1$ diagnostics [118,119] which is implemented in ricc2. $D_2$ is a diagnostic for strong interactions of the HF reference state with doubly excited determinants, while $D_1$ is a diagnostic for strong interactions with singly excited determinants.

### 9.5 RI-MP2-F12 Calculations

To obtain the F12 correction to the MP2 energy, the data group $rir12$ must be added to the control file. A typical run will include the input:

```
$ricc2
  mp2 energy only
$rir12
```
The MP2-F12 ground-state energy is

\[ E_{\text{MP2-F12}} = E_{\text{MP2}} + E_{\text{F12}}, \]  

(9.3)

where \( E_{\text{MP2}} \) is the conventional MP2 energy and \( E_{\text{F12}} \) the correction from explicitly-correlated theory. The second term contains contributions from explicitly-correlated geminal basis functions of the form

\[ \hat{Q}_{12} f_{12} |ij\rangle, \]  

(9.4)

where \(|ij\rangle\) is a two-electron determinant of occupied (semi-)canonical Hartree–Fock spin orbitals, \( f_{12} \) is a correlation factor, which can be either linear \( r_{12} \) (in this case, the approach is denoted MP2-R12 instead of MP2-F12) or a function of \( r_{12} \), and \( \hat{Q}_{12} \) defines the doubles excitation space covered by the geminals (it also ensures strong orthogonality to the occupied orbitals). Usually \( \hat{Q}_{12} \) is chosen to be \( \hat{Q}_{12} = (1 - \hat{O}_1)(1 - \hat{O}_2) - \hat{V}_1 \hat{V}_2 \), where \( \hat{O}_\mu = \sum_k |\varphi_k(\mu)\rangle\langle \varphi_k(\mu)| \) is the projection operator onto the space spanned by the occupied spin orbitals \( \varphi_k \) and \( \hat{V}_\mu = \sum_a |\varphi_a(\mu)\rangle\langle \varphi_a(\mu)| \) is the projector onto the virtual spin orbitals.

The F12 correction is obtained by minimizing the functional

\[ F_{\text{F12}} = \sum_{ij} \left\{ c^T_{ij} B_{ij} c_{ij} + 2 c^T_{ij} v_{ij} \right\} \]  

(9.5)

with respect to the amplitudes collected in the vector \( c_{ij} \). The vectors \( v_{ij} \) and the matrices \( B_{ij} \) are defined as

\[ v_{ij}(kl) = \langle kl | f_{12} \hat{Q}_{12} r_{12}^{-1} |ij\rangle, \]  

(9.6)

\[ B_{ij}(kl,mn) = \langle kl | f_{12} \hat{Q}_{12} (\hat{f}_1 + \hat{f}_2 - \varepsilon_i - \varepsilon_j) \hat{Q}_{12} f_{12} |mn\rangle, \]  

(9.7)

in the spin-orbital formalism (\( m, n \) denote spin orbitals and \(|mn\rangle\) is a two-electron determinant). \( \hat{f}_\mu \) is the Fock operator for electron \( \mu \) and \( \varepsilon_k \) is a (semi-)canonical Hartree–Fock orbital energy.

A MP2-F12 calculation is defined through a number of choices concerning the nature of the geminals \( (f_{12} \) and \( \hat{Q}_{12} \)), the geminal excitation space (ijkl or ijij) and approximations in computing the \( B \) matrix (GBC, EBC, \([T,f_{12}]\)). These choices correspond to keywords in the \$rir12 data group, explained below.

To run a MP2-F12 calculation, one has to select the auxiliary basis sets \( \text{cbas} \), \( \text{cabs} \) and optionally \( \text{jkbas} \). The \texttt{ricc2} program uses the robust fitting techniques of Ref. [120] for the F12 integrals and the \( \text{cbas} \) basis is used for both the F12 and the usual MP2 Coulomb integrals. For the density fitting of the Coulomb and exchange matrices of the Fock matrix, the \( \text{jkbas} \) will be used instead of the \( \text{cbas} \) basis if it is included in the control file (this is recommended and is achieved using the \texttt{rijk} menu in \texttt{define}). For the RI approximation of the 3- and 4-electron integrals as sums of products of 2-electron integrals, intrinsic to the F12 method, the complementary auxiliary basis (CABS) approach is used [121]. If \texttt{define} is used to set up the \( \text{cabs} \) basis, the library \texttt{cabasen} is searched. This library contains the optimised \( \text{cabs} \) basis sets [122] for the cc-pVXZ-F12 basis sets of Peterson \textit{et al.} [123]. For
other basis sets, the auxiliary basis in the library cabasen is identical with the auxiliary basis in the library cbas.

The $rir12$ data group may be set by choosing the f12 option in the cc menu when running define. This command activates the f12 menu, where the default options may be changed if desired:

Orbital basis : cc-pVTZ-F12
Cardinal number : T
Recommended exponent: 1.0000
Actual exponent: 1.0000

INPUT MENU FOR MP2-F12 CALCULATIONS

ansatz : CHOOSE ANSATZ 2 [1,2*,2]
r12model : CHOOSE MODEL B [A,B]
comaprox : COMMUTATOR APPROXIMATION F+K [F+K,T+V]
cabs : CABS ORTHOGONALIZATION svd 1.0D-08 [cho,svd]
examp : CHOOSE FORMULATION fixed noflip [inv, fixed, noinv, flip, noflip]
ri2orb : CHOOSE GEMINAL ORBITALS hf [hf, rohf, boys, pipek]
corrfac : CHOOSE CORRELATION FACTOR LCG [R12, LCG]
cabsingles: CABS SINGLE EXCITATIONS on [on, off]
pairenergy: PRINT OUT PAIRENERGIES off [on, off]
slater : SLATER EXPONENT 1.0000

* / end : write $rir12$ to file and leave the menu
& : go back - leaving $rir12$ unchanged...

ansatz corresponds to the choice of $\hat{Q}_{12}$. Almost all modern MP2-F12 calculations use ansatz 2 (default), which gives much improved energies over ansatz 1 (see Ref. [124] for details). The principal additional cost of using ansatz 2 over ansatz 1 is concerned with the coupling between the F12 and conventional amplitudes. This is avoided by choosing 2*, which corresponds to neglecting EBC (Extended Brillouin Condition) terms in the Fock matrix elements.

r12model is the method of computing the matrices $B_{ij}$ (see Ref. [124] for details). The cost and accuracy increases from A to B. It is recommended to use B (default). The energies computed using A are then also printed out in the output.

comaprox is the method for approximately computing the integrals for the operator $[\hat{T}, f_{12}]$, where the matrix representations of F+K or T+V are used. F+K (the core Hamiltonian plus Coulomb term) is recommended and is the default.
9.5. RI-MP2-F12 CALCULATIONS

**cabs** refers to the method of orthogonalising the orbitals in the complementary auxiliary basis. Singular-value decomposition (**svd**) or Cholesky decomposition (**cho**) are available. **svd** is recommended and is the default, with a threshold of $1.0d-08$. The basis set used for CABS is set from the **cc** menu.

**examp** refers to the choice of excitation space. **inv** is the orbital-invariant method of Ref. [125], with amplitudes $c_{ij}(kl)$. **noinv** is the original orbital-dependent diagonal "ijij" method of Ref. [125], with amplitudes $c_{ij}(ij)$ (not recommended, unless in combination with localised orbitals). **fixed** is the (diagonal and orbital-invariant) rational generator approach of Ref. [126], where the F12 amplitudes are not optimised but predetermined using the coalescence conditions (default). An additional keyword **noflip** suppresses the use of spin-flipped geminals in open-shell calculations; by default spin-flipped geminals are used as described in Ref. [19].

**r12orb** controls which orbitals are used in the F12 energy contribution. **hf** means that (semi-)canonical Hartree–Fock orbitals are used (default). **rohf** means that ROHF orbitals are used (any frozen orbitals will then also implicitly be ROHF). For calculations on closed-shell systems, localised orbitals may be used. Both the Boys [127] and Pipek–Mezey [128] methods are available for localisation of the orbitals.

**corrfac** corresponds to the choice of correlation factor $f_{12}$ in the geminal basis functions. **R12** results in a calculation using linear-r$_{12}$ and **LCG** results in a calculation using the Slater-type correlation factor with exponent 1.4 $a_0^{-1}$, represented as a linear combination of six Gaussians (see Ref. [129]). Note that the exponents 0.9, 1.0 and 1.1 $a_0^{-1}$ are recommended for use with the cc-pVXZ-F12 basis sets [123].

**cabsingles** switches on/off the calculation of a second-order correction to the Hartree–Fock energy by accounting for single excitations into the complementary auxiliary basis set (CABS). The single excitations into the CABS basis can be computed without extra costs if the CABS Fock matrix elements are required anyway for the F12 calculation (i.e., for ansatz 2, approximation B or comaprox F+K). The computation of CABS singles cannot be switched off if it is free of costs.

**pairenergy** controls whether or not the F12 contribution to the MP2 pair energies appear in the output (default off).

Further options:

**corrfac LCG** refers to a further data group for the definition of the correlation factor. When **define** is used, the default is

$\$1cg$
The nature of the LCG correlation factor may be changed by editing this data group in the control file. For example, to use a Slater-type correlation factor with exponent 1.0 $a_0^{-1}$, represented as a linear combination of three Gaussians, use

```ncf
$lcg
nlcg 3
slater 1.0000
```

Alternatively, the exponents and coefficients of the fit may be given explicitly:

```ncf
$lcg
nlcg 3
expo1 coef1
expo2 coef2
expo3 coef3
```


```ncf
$ricc2
mp2 energy only
scs
```

In this case, the SCS parameters cos=6/5 and css=1/3 are used. Also individual scaling factors for the same-spin and opposite-spin contributions may be defined, see Section 10.7.

For open-shell calculations, two choices of the `examp fixed` method are available. These are controlled by a keyword in the `$rir12` data group

```ncf
ump2fixed full [diag,full]
```

These differ in the treatment of the $\alpha\beta$ block, where either only the diagonal excitations enter (with amplitude 0.5) `diag`, or the equivalent of the spin-adapted singlet and triplet pair excitations enter (as far as possible) `full`. Note that the `diag` method with UMP2-F12 yields a result different to that of `fixed` MP2-F12, even for identical RHF and UHF determinants. However, the `diag` method is somewhat less expensive than the `full` method.

Recommendations for orbital and auxiliary basis sets:

The best orbital basis sets to use for MP2-F12 calculations are probably the cc-pVXZ-F12 basis sets, specially optimised for MP2-F12 calculations [123] for the atoms H, He, B–Ne and Al–Ar. In conjunction with these cc-pVXZ-F12 basis sets, we recommend to use the optimised cc-pVXZ-F12 sets of Yousaf and Peterson [122] as cabs. Furthermore, cbas and
jkbas basis sets can be selected from the cbasen and jkbasen libraries, respectively, using the alias cc-pVXZ-F12 (a jkbas is currently not available for He, Ne and Ar). This alias points to the corresponding aug-cc-pwCV(X+1)Z cbas and aug-cc-pV(X+1)Z jkbas. These recommendations are on the side of caution and are likely to be refined as more experience is gained [130–132].

For atoms other than H, He, B–Ne and Al–Ar, optimised F12 basis sets are not yet available. In this case, basis sets must be selected and/or optimised carefully. It is advised to contact the Theoretical Chemistry Group in Karlsruhe for support (e-mail to: klopper@kit.edu).

9.6 Laplace-transformed SOS-RI-MP2 with $O(N^4)$ scaling costs

The ricc2 module contains an implementation of SOS-MP2 which exploits the RI approximation and a Laplace transformation of the orbital energy denominators

$$\frac{1}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} = \int_0^\infty e^{-(\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)t} dt \approx \sum_{\alpha=1}^{n_L} w_{\alpha} e^{-(\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)t_{\alpha}}, \quad (9.8)$$

to achieve an implementation with $O(N^4)$ scaling costs, opposed to the conventional $O(N^5)$ scaling implementation. In particular for large molecules the Laplace-transformed implementation can reduce a lot the computational costs of SOS-MP2 calculations without loss in accuracy.

The Laplace-transformed implementation for SOS-MP2 calculations is activated with the input

```
$laplace
conv=5
```

where the parameter conv is a convergence threshold for the numerical integration in Eq. (9.8). A value of conv=5 means that the numerical integration will be converged to a root mean squared error of $\approx 10^{-5}$ a.u.

Whether the conventional or the Laplace-transformed implementation will be more efficient depends firstly on the system size (the number of occupied orbitals) and secondly on the required accuracy (the number of grid points for the numerical integration in Eq. (9.8)) and can be understood and estimated from the following considerations:

- The computational costs for the most expensive step in (canonical) RI-MP2 energy calculations for large molecules requires $\frac{1}{2}O^2V^2N_z$ floating point multiplications, where $O$ and $V$ are, respectively, the number occupied and virtual orbitals and $N_z$ is the number of auxiliary functions for the RI approximation. For the LT-SOS-RI-MP2 implementation the most expensive step involves $n_LOVN_z^2$ floating point multiplications, where $n_L$ is the number of grid points for the numerical integration. Thus, the
ratio of the computational costs is approximately
\[
\text{conv} : LT \approx \frac{\frac{1}{2}O^2V^2N_x}{n_LOVN_x^2} = \frac{OV}{2n_LN_x} \approx O : 6n_L,
\]
where for the last step \(N_x \approx 3V\) has been assumed. Thus, the Laplace-transformed
implementation will be faster than the conventional implementation if \(O > 6n_L\).

The number of grid points \(n_L\) depends on the requested accuracy and the spread of the or-
bital energy denominators in Eq. (9.8). The efficiency of Laplace-transformed SOS-RI-MP2
calculations can therefore (in difference to conventional RI-MP2 calculations) be enhanced
significantly by a careful choice of the thresholds, the basis set, and the orbitals included
in the correlation treatment:

- The threshold \text{conv} for the numerical integration is by default set to the value of \text{conv}
specified for the ground state energy in the data group $\text{ricc2}$ (see Sec. 21.2.17), which
is initialized using the threshold $\text{denconv}$, which by default is set conservatively to
the tight value of \(10^{-7}\).
  - For single point energy calculations \text{conv} in $\text{laplace}$ can safely be set to 4,
    which gives SOS-MP2 energies converged within \(\approx 10^{-4}\) a.u. with computational
costs reduced by one third or more compared to calculations with the default
settings for these thresholds.
  - For geometry optimizations with SOS-MP2 we recommend to set \text{conv} in $\text{laplace}$
to 5.

- The spread of the orbital energy denominators depends on the basis sets and the
orbitals included in the correlation treatment. Most segmented contracted basis sets
of triple-\(\zeta\) or higher accuracy (as e.g. the TZVPP and QZVPP basis sets) lead to
rather high lying “anti core” orbitals with orbital energies of 10 a.u. and more.
  - For the calculation of SOS-MP2 valence correlation energies it is recommended
to exclude such orbitals from the correlation treatment (see input for $\text{freeze}$
in Sec. 21).
  - Alternatively one can use general contracted basis sets, as e.g. the correlation
consistent cc-pVXZ basis sets. But note that general contracted basis sets in-
crease the computational costs for the integral evaluation in the Hartree-Fock
and, for gradient calculations, also the CPHF equations and related 4-index
integral derivatives.
  - Also for the calculation of all-electron correlation energies with core-valence basis
sets which include uncontracted steep functions it is recommended to check if
extremely high-lying anti core orbitals can be excluded.

Note that for large molecules it is recommended to disable for geometry optimizations (or for
gradient or property calculations in general) the preoptimization for the Z vector equations
with the $\text{nozpreopt}$ option in the $\text{response}$ data group (see Sec. 21.2.17).
Restrictions:

- The Laplace-transformed SOS-MP2 implementation in ricc2 is presently only parallelized with MPI. The OpenMP parallelization is not (yet) recognized by the LT-SOS-RI-MP2 related program parts.
- It is presently not compatible with the calculation of the $D_1$ and $D_2$ diagnostics. The respective options will be ignored by program if the Laplace-transformed implementation is used.

9.7 Low-scaling MP2 and MP2-F12 calculations with a hybrid OSV-PNO approximation

The pnoccsd program works with localized molecular occupied orbitals (LMOs) uses in addition to the RI approximation for the two-electron integrals truncations of the virtual orbital space which are determined separately for each LMO and each LMO pair. The pair natural orbital (PNO) methods belong to the family of local correlation methods which have in common the key idea that the correlation energy can for all wavefunction methods be written as a sum over pair contributions

$$E_{corr} = \sum_{i \leq j} e_{ij} = \frac{1}{4} \sum_{a_jb} (t_{ab}^{ij} + t_{ab}^{ji}) \langle ij|ab \rangle$$

and that with localized molecular orbitals the pair contributions $e_{ij}$ decrease fast (approximately with $1/|r_{ij}|^6$) with the distance between the LMOs $i$ and $j$ so that for large molecules the number of pairs that need to be included to compute the correlation energies within a given accuracy increases only linearly with the number of atoms. Furthermore, only such virtual orbitals contribute significantly the correlation energy $e_{ij}$ of a pair $ij$ which are spatially close to the either $i$ or $j$.

The OSV-PNO hybrid approach implemented in the TURBOMOLE’s pnoccsd program starts with the determination of truncated subsets of orbital-specific virtual (OSV) orbitals for each LMO $i$ as the eigenvectors of diagonal density contributions $D_{ii}^{ij}$ computed from (approximate) MP2 doubles amplitudes:

$$D_{ab}^{ij} = 2 \sum_c t_{ac}^{ij} t_{cb}^{ij} = \sum_{\tilde{c}} d_{ac}^{ij} \tilde{n}_{\tilde{c}}^{ij} d_{bc}^{ij}$$

with occupation numbers $n_{\tilde{c}}^{ij}$ not smaller than a predefined threshold $tol_{osv}$. The OSVs are first used to calculate for each pair $ij$ an an estimate for the pair correlation energy contribution $e_{ij}$ and to set up a list of pairs $ij$ for which significant contributions to the total correlation energy are expected. Only for this list of pairs the OSVs of the two LMOs are merged together to an orthonormal set of pre-PNOs $\tilde{c}_{ij}$ which are used to compute the contribution of the pair $ij$ to the density matrix via approximated MP2 amplitudes and as eigenvectors of these densities the coefficients of the pair natural orbitals:

$$D_{\tilde{a} \tilde{b}}^{ij} = 2 \sum_{\tilde{c}_{ij}} t_{\tilde{a} \tilde{c}_{ij}}^{ij} t_{\tilde{b} \tilde{c}_{ij}}^{ij} = \sum_c d_{ac}^{ij} n_{\tilde{c}}^{ij} d_{bc}^{ij}.$$
Similar as for the OSVs also the PNOs with occupation numbers \( n_{ij} \) below the PNO truncation threshold \( \text{tolpno} \) are discarded.

For OSV-PNO-MP2 the MP2 equations are then solved with the amplitudes and two-electron integrals projected onto the so determined set of pair natural orbitals. For the explicitly-correlated variant OSV-PNO-MP2-F12 similar sets of orbital and pair specific auxiliary orbitals (OSX) are determined for the occupied and the complementary virtual orbital spaces which appear in the calculation of the three-electron integrals for the additional contributions from the geminals.

The thresholds for discarding PNOs and OSVs and the subset of auxiliary orbitals for the local RI approximation for computing two-electron integrals play an important role for the accuracy and the computational costs of calculations with the \texttt{pnoccsd} program. Per default all thresholds are coupled to the PNO truncation threshold \( \text{tolpno} \) such that the errors due to the OSV, local RI and pair truncation approximations and due to the screening of (AO) integrals and contributions to the residual are about an order of magnitude smaller than the PNO truncation error. The default value for \( \text{tolpno} \) is \( 10^{-7} \) which gives PNO truncation errors in the correlation energy \(<\ 1\%\). Test calculations showed that the relative errors in the correlation energy decrease typically with \( 30 \times \sqrt{\text{tolpno}} \).

For a PNO-MP2 calculation with default thresholds and standard basis sets the \texttt{pnoccsd} program does not require any special input apart from \$\text{freeze} \) and \$\text{maxcor} \) and the input needed for the Hartree-Fock calculation. It is, however, recommended to specify the PNO truncation threshold in the data group \$\text{pnoccsd} \).

**Running \texttt{pnoccsd} parallel**  The \texttt{pnoccsd} program is parallelized with OMP for shared-memory and with MPI for distributed memory architectures. Important for performance of the parallel \texttt{pnoccsd} calculations are the settings for the core memory usage and for the directories where large integral and scratch files are stored.

The keyword \$\text{maxcor} \) defines for \texttt{pnoccsd} (as for \texttt{ricc2} and \texttt{ccsdf12}) the core memory usage in megabytes per process. In an OMP calculation the amount of memory defined with \$\text{maxcor} \) is shared between all threads of a process. In a MPI calculation every process (node) will allocate \$\text{maxcor} \) megabytes of core memory. (Note, however, that \$\text{maxcor} \) defines only the memory controled by the electronic structure code. Additional memory can be allocated by the math and MPI libraries linked into the program and by operating and I/O system.)

For MPI parallel \texttt{pnoccsd} calculations it is very important to set with the \$\text{tmpdir} \) keyword the path to directories in fast local file systems to avoid that large integral and scratch files are stored in the NFS file system where the calculation is started.
Chapter 10

Second-Order Approximate Coupled-Cluster (CC2) Calculations

ricc2 is a module for the calculation of excitation energies and response properties at a correlated \textit{ab initio} level, in particular the second-order approximate coupled-cluster model CC2 [133]. All calculations employ the resolution-of-the-identity (RI) approximation for the electron repulsion integrals used in the correlation treatment and the description of excitation processes. At present the following functionalities are implemented:

**ground state energies** for MP2 and CC2 and spin-component scaled variants thereof; the MP2 results are identical with those obtained with rimp2 (but usually the calculations are somewhat faster).

**excitation energies** for the models CIS/CCS, CIS(D), CIS(D$_\infty$), ADC(2), and CC2 including spin-component scaled SCS and SOS version of of the latter four methods

**transition moments** for ground state—excited and excited—excited state transitions for the models CCS and CC2; for ADC(2) only moments for ground state—excited state transitions are available

**first-order properties** for the ground state with SCF (CCS), MP2, and CC2 and for excited states with CCS, CC2, ADC(2) and CIS(D$_\infty$)

**geometric gradients** for the electronic ground state at the MP2 and the CC2 level; for electronically excited states at the CIS(D$_\infty$), ADC(2), and CC2 level

**second-order properties** for the ground state with MP2 and CC2 and a closed-shell RHF reference wavefunction (currently restricted to the sequentical and SMP parallel versions)

**gradients for auxiliary basis sets** for RI-MP2, -CC2, etc. calculations based on the RI-MP2 error functional
F12 corrections to RI-MP2; MP2 ground-state energies can be computed (in $C_1$ symmetry) using explicitly-correlated two-electron basis functions in the framework of the MP2-F12 model \cite{130,134}.

**Solvent effects** for the methods and states for which (orbital–relaxed) densities are available equilibrium solvent effects can be included in the framework of the cosmo mode (for details see Chapter 20).

All functionalities at the MP2 and CC2 level are implemented for closed-shell RHF and open-shell UHF reference wavefunctions (with the exception of second-order properties, which are only available for a closed-shell RHF reference). Ground state energies for MP2, MP2-F12 and CC2 and excited state energies for CC2 are also implemented for single determinant restricted open-shell Hartree-Fock (ROHF) reference wavefunctions (cmp. Sec. 9.3). (Note, that no gradients are available for MP2 and CC2 with ROHF reference wavefunctions.) For a two-component GHF reference wavefunction energies for the CCS, MP2/ADC(2), CIS(D$_\infty$) and CC2 methods as well as ground state—excited state transition moments for ADC(2) and CC2 are available.

The second-order models MP2, CIS(D), CIS(D$_\infty$), ADC(2) and CC2 can be combined with a spin-component scaling (SCS or SOS). (Not yet available for second-order properties.) For the SOS variants one can switch to an implementation with $O(N^4)$-scaling costs by setting the keyword for the numerical Laplace transformation (LT) ($\$laplace$).

For calculations with CCSD, CCSD(T) and other higher-order models beyond CC2 see Chapter 11.

**Prerequisites**

Calculations with the ricc2 module require (almost) the same prerequisites as RI-MP2 calculations:

1. a converged SCF calculation with the one-electron density convergence threshold set to $\$denconv 1.d-5$ or less
2. if non-standard basis sets used: an auxiliary basis defined in the data group $\$cbas$ (for standard basis sets, where a corresponding auxiliary basis set is found in the basis set library, the program will automatically use this if $\$cbas$ is not set)
3. if orbitals should be excluded from the correlation treatment (and excitation processes) the data group $\$freeze$ has to be set
4. the maximum core memory which the program is allowed to allocate should be defined in the data group $\$maxcor$; the recommended value is 66–75% of the available (physical) core memory.
5. depending on the type of calculations that should be carried out, additionally the data groups $\$ricc2$, $\$excitations$, $\$response$, $\$laplace$, $\$rir12$ and $\$lcg$ have to be set (see below and Section 21.2.17).
For calculations with the ricc2 program it is recommended to use the cc2 submenu of the define program to set the data groups $denconv, $freeze, $cbas, $maxcor. MP2-F12 calculations require in addition the data groups $rir12, $cabs, $jkbas and $lcg. The exponent of the Slater function in the interelectronic distance $r_{12}$, which appears in the geminals used MP2-F12 is defined in the data group $lcg and should be adapted to the one-electron basis set which is used.

Note, that the implementation of non-Abelian point groups in ricc2 is limited to the electronic ground state (but comprises all of the RI-MP2 functionality included in ricc2). In the present version ricc2 can for excited states only deal with real Abelian point groups ($C_1, C_s, C_2, C_i, C_{2h}, C_{2v}, D_2, D_{2h}$). The F12 correction can only be calculated in the $C_1$ point group.

**How To Perform a Calculation**

**Single point calculations:**

Call the ricc2 program after a converged SCF calculation, which can be carried either with the dscf or the ridft program.

**Geometry optimizations and molecular dynamics:**

Invoke jobex with the -level cc2 option; see Section 5.1 for additional options and parameters of the jobex script that might be needed or useful for geometry optimizations and ab initio molecular dynamics calculations.

**Force constants and vibrational frequencies:**

Force constants can be calculated by numerical differentiation of the gradients. Invoke for this NumForce with the -level cc2 option; see Chapter 14 for details about Numforce. The usage of the Numforce interface for excited states is restricted to $C_1$ symmetry.

**Note:** using ricc2 in connection with jobex or Numforce requires that the method and the electronic state, for which the gradient should be calculated and written to the interface files, is specified in the option geoopt (see Section 10.3.1) in datagroup $ricc2 (see Section 21.2.17). For calculations on excited states this state has in addition to be included in the input for excitation energies in datagroup $excitations.

**RI-SCF reference wavefunctions:** The ricc2 program can be used in combination with conventional SCF or with the RI-J and RI-JK approximations for SCF, with the exception that the calculation of gradients for reference wavefunctions which employ only the RI-J approximation for the Coulomb matrix but 4-index integrals for the exchange matrix is presently not supported. The implementation of gradients in ricc2 assumes that the reference wavefunction has either been calculated without RI-J approximation (using dscf) or with the RI-JK approximation (using ridft).

See Chapter 6 for a discussion of the RI approximations in SCF calculations and 21.2.6 for the required input. In geometry optimizations with jobex and for the calculation of force
constants and vibrational spectra with NumForce, the ricc2 program is used in combination with the RI-JK approximation for the Hatree-Fock calculation (using ridft) if jobex and NumForce are invoked with the \texttt{-rijk} option.

**How to quote**

If results obtained with the ricc2 program are used in publications, the following citations should be included if you have used the methods, program parts, auxiliary basis sets, or results reported in therein:

**Methods:**

- for the approximate coupled-cluster singles-and-doubles model CC2:

- for CI singles with a perturb. correct. for connected double excitations, CIS(D):
  and for the iterative CIS(D$_\infty$) variant:

- for the algebraic diagrammatic construction through second order ADC(2):

- for MP2-F12:

- for the SCS and SOS variants of MP2:

- for the SCS and SOS variants of CC2 and ADC(2):

- for the two-component CCS, ADC(2), CIS(D$_\infty$) and CC2 methods:

**Implementation:**

- please, include always a reference to the publication reporting the implementation of the core part of the ricc2 program:

- for transition moments and excited state first order properties:
• for triplet excited states include:
• for ground state geometry optimizations include:
• for geometry optimizations for excited states include:
• for calculations with RI-ADC(2), RI-CIS(D), RI-CIS(D∞) include:
• if the parallel version of ricc2 is used include a reference to:
• for transition moments between excited states:
• for RI-MP2-F12 calculations:
  R. A. Bachorz, F. A. Bischoff, A. Glöß, C. Hättig, S. Höfener, W. Klopper,
• for O(N^4)-scaling calculations using the Laplace transformation:
  – ground-state and excitation energies:
  – transition moments, first-order properties and gradients:
• for second-order properties (relaxed or unrelaxed):

Auxiliary basis sets:
• the appropriate reference for the auxiliary SVP, TZVP and TZVPP basis sets
  (for calculations with RI-MP2, RI-CC2 and related methods) is:
  143.
• for the auxiliary cc-pVXZ (cc-pV(X+d)Z), aug-cc-pVXZ (aug-cc-pV(X+d)Z)
  basis sets with X = D, T, or Q cite:
• for the auxiliary cc-pV5Z (cc-pV(5+d)Z), aug-cc-pV5Z (aug-cc-pV(5+d)Z), cc-
  pwCVXZ with X = D, T, Q, 5 and QZVPP basis sets the reference is:
This reference should also be included if you employ the analytic basis set gra-
  dients implemented in the ricc2 program for the optimization of your own
  auxiliary basis set(s).
CHAPTER 10. RI-CC2


(For more details on the references for the basis sets included in the basis set libraries of the TURBOMOLE distribution see Sec. 1.3 and the library files.)

10.1 CC2 Ground-State Energy Calculations

The CC2 ground-state energy is—similarly to other coupled-cluster energies—obtained from the expression

$$E_{CC} = \langle HF | H | CC \rangle = \langle HF | H \exp(T) | HF \rangle ,$$

(10.1)

$$E_{SCF} + \sum_{iajb} \left[ t_{ij}^{ab} + t_{ia}^{t} \right] \left[ 2(ia|jb) - (ja|ib) \right] ,$$

(10.2)

where the cluster operator $T$ is expanded as $T = T_1 + T_2$ with

$$T_1 = \sum_{ai} t_{ai}^{i} \tau_{ai},$$

(10.3)

$$T_2 = \frac{1}{2} \sum_{abj} t_{ab}^{ij} \tau_{abj},$$

(10.4)

(for a closed-shell case; in an open-shell case an additional spin summation has to be included). The cluster amplitudes $t_{ai}^{i}$ and $t_{ab}^{ij}$ are obtained as solution of the CC2 cluster equations [133]:

$$\Omega_{\mu_1} = \langle \mu_1 | \hat{H} + [\hat{H}, T_2] | HF \rangle = 0 ,$$

(10.5)

$$\Omega_{\mu_2} = \langle \mu_2 | \hat{H} + [F, T_2] | HF \rangle = 0 ,$$

(10.6)

with

$$\hat{H} = \exp(-T_1) H \exp(T_1),$$

where $\mu_1$ and $\mu_2$ denote, respectively, the sets of all singly and doubly excited determinants.

The residual of the cluster equations $\Omega(t_{ai}, t_{abj})$ is the so-called vector function. The recommended reference for the CC2 model is ref. [133], the implementation with the resolution-of-the-identity approximation, RI-CC2, was first described in ref. [8].
Advantages of the RI approximation: For RI-CC2 calculations, the operation count and thereby the CPU and the wall time increases—as for RI-MP2 calculations—approximately with $O(O^2V^2N_x)$, where $O$ is the number of occupied and $V$ the number of virtual orbitals and $N_x$ the dimension of the auxiliary basis set for the resolution of the identity. Since RI-CC2 calculations require the (iterative) solution of the cluster equations (10.5) and (10.6), they are about 10–20 times more expensive than MP2 calculations. The disk space requirements are approximately $O(2V + N_xN_x + N_x^2)$ double precision words. The details of the algorithms are described in ref. [8], for the error introduced by the RI approximation see refs. [117,135].

Required input data: In addition to the above mentioned prerequisites ground-state energy calculations with the ricc2 module require only the data group $\texttt{ricc2}$ (see Section 21.2.17), which defines the methods, convergence thresholds and limits for the number of iterations etc. If this data group is not set, the program will carry out a CC2 calculation. With the input

```plaintext
$\texttt{ricc2}
mp2
cc2
conv=6
```

the ricc2 program will calculate the MP2 and CC2 ground-state energies, the latter converged to approximately $10^{-6}$ a.u. The solution for the single-substitution cluster amplitudes is saved in the file CCR0--1--1---0, which can be kept for a later restart.

Ground-State calculations for other methods than CC2: The MP2 equations and the energy are obtained by restricting in the CC2 equations the single-substitution amplitudes $t_{ai}$ to zero. In this sense MP2 can be derived as a simplification of CC2. But it should be noted that CC2 energies and geometries are usually not more accurate than MP2.

For CCS and CIS the double-substitution amplitudes are excluded from the cluster expansion and the single-substitution amplitudes for the ground state wavefunction are zero for closed–shell RHF and open–shell UHF reference wavefunctions and thus energy is identical to the SCF energy.

For the Methods CIS(D), CIS(D∞) and ADC(2) the ground state is identified with the MP2 ground state to define total energy of the excited state, which is needed for the definition of gradients and (relaxed) first-order properties which are obtained as (analytic) derivatives the total energy.

Diagnostics: Together with the MP2 and/or CC2 ground state energy the program evaluates the $D_1$ diagnostic proposed by Janssen and Nielsen [118], which is defined as:

$$D_1 = \sqrt{\max \left( \lambda_{\max} \left[ \sum_i t_{ai}t_{bi} \right], \lambda_{\max} \left[ \sum_a t_{ai}t_{aj} \right] \right)}$$  

(10.7)
where $\lambda_{\text{max}}[M]$ is the largest eigenvalue of a positive definite matrix $M$. (For CC2 the $D_1$ diagnostic will be computed automatically. For MP2 is must explicitly be requested with the $d1diag$ option in the $\text$dricc2 data group, since for RI-MP2 the calculation of $D_1$ will contribute significantly to the computational costs.) Large values of $D_1$ indicate a multireference character of the ground-state introduced by strong orbital relaxation effects. In difference to the $T_1$ and $S_2$ diagnostics proposed earlier by Lee and coworkers, the $D_1$ diagnostic is strictly size-intensive and can thus be used also for large systems and to compare results for molecules of different size. MP2 and CC2 results for geometries and vibrational frequencies are, in general, in excellent agreement with those of higher-order correlation methods if, respectively, $D_1(\text{MP2}) \leq 0.015$ and $D_1(\text{CC2}) \leq 0.030$ [11, 118]. For $D_1(\text{MP2}) \leq 0.040$ and $D_1(\text{CC2}) \leq 0.050$ MP2 and/or CC2 usually still perform well, but results should be carefully checked. Larger values of $D_1$ indicate that MP2 and CC2 are inadequate to describe the ground state of the system correctly!

The $D_2$ diagnostic proposed by Nielsen and Janssen [119] can also be evaluated. This analysis can be triggered, whenever a response property is calculated, e.g. dipole moment, with the keyword $\text{D2-diagnostic}$. Note that the calculation of $D_2$ requires an additional $O(N^5)$ step! $D_2(\text{MP2/CC2}) \leq 0.15$ are in excellent agreement with those of higher-order correlation methods, for $D_2(\text{MP2/CC2}) \geq 0.18$ the results should be carefully checked.

10.2 Calculation of Excitation Energies

With the $\text{ricc2}$ program excitation energies can presently be calculated with the RI variants of the methods CCS/CIS, CIS(D), CIS(D$_\infty$), ADC(2) and CC2. The CC2 excitation energies are obtained by standard coupled-cluster linear response theory as eigenvalues of the Jacobian, defined as derivative of the vector function with respect to the cluster amplitudes.

$$A_{\mu\nu}^{\text{CC2}} = \frac{d\Omega}{d\nu} = \begin{pmatrix} \langle \mu_1 | [[\hat{H}, T_2], \tau_{\nu_1}] | \text{HF} \rangle & \langle \mu_1 | [\hat{H}, \tau_{\nu_2}] | \text{HF} \rangle \\ \langle \mu_2 | [\hat{H}, \tau_{\nu_1}] | \text{HF} \rangle & \langle \mu_2 | [F, \tau_{\nu_2}] | \text{HF} \rangle \end{pmatrix}$$

(10.8)

Since the CC2 Jacobian is a non-symmetric matrix, left and right eigenvectors are different and the right (left) eigenvectors $E^i_\nu$ ($\bar{E}^i_\mu$) are not orthogonal among themselves, but form a biorthonormal basis (if properly normalized):

$$\bar{E}^i E^j = \bar{E}^i_{\mu_1} E^j_{\nu_1} + \bar{E}^i_{\mu_2} E^j_{\nu_2} = \delta_{ij} .$$

(10.9)

To obtain excitation energies only the right or the left eigenvalue problem needs to be solved, but for the calculation of transition strengths and first-order properties both, left and right, eigenvectors are needed (see below). A second complication that arises from the non-symmetric eigenvalue problem is that in the case of close degeneracies within the same irreducible representation (symmetry) it can happen that instead of two close lying real roots a degenerate complex conjugated pair of excitation energies and eigenvectors is obtained. CC2 (and also other standard coupled-cluster response methods) are thus not suited for the description of conical intersections etc. For the general theory behind coupled cluster response calculations see e.g. ref. [136, 137] or other reviews.
10.2. CALCULATION OF EXCITATION ENERGIES

The \texttt{ricc2} program exploits that the doubles/doubles block of the CC2 Jacobian is diagonal and the (linear) eigenvalue problem in the singles and doubles space can be reformulated as a (non-linear) eigenvalue problem in single-substitution space only:

\[
A_{\mu_1 \nu_1}^{\text{eff}}(t, \omega) = A_{\mu_1 \nu_1}^{\text{CC2}}(t) - A_{\mu_1 \gamma_2}^{\text{CC2}}(t)(A_{\gamma_2 \nu_1}^{\text{CC2}} - \omega)A_{\gamma_2 \nu_1}^{\text{CC2}}(t)
\]

This allows to avoid the storage of the double-substitution part of the eigen- or excitation vectors \(E_{\nu_2}, \bar{E}_{\nu_2}\). The algorithms are described in refs. [8,9], about the RI error see ref. [135].

The solution of the CC2 eigenvalue problem can be started from the solutions of the CCS eigenvalue problem (see below) or the trial vectors or solutions of a previous CC2 excitation energy calculation. The operation count per transformed trial vector for one iteration for the CC2 eigenvalue problem is about 1.3–1.7 times the operation count for one iteration for the cluster equations in the ground-state calculation—depending on the number of vectors transformed simultaneously. The disk space requirements are about \(O(V + N)N_x\) double precision words per vector in addition to the disk space required for the ground state calculation.

CCS excitation energies are obtained by the same approach, but here double-substitutions are excluded from the expansion of the excitation or eigenvectors and the ground-state amplitudes are zero. Therefore the CCS Jacobian,

\[
A_{\mu \nu}^{\text{CCS}} = \frac{d\Omega_{\mu}}{dt_{\nu}} = \langle \mu_1 | [H, \tau_{\nu_1}] | \text{HF} \rangle,
\]

is a symmetric matrix and left and right eigenvectors are identical and form an orthonormal basis. The configuration interaction singles (CIS) excitation energies are identical to the CCS excitation energies. The operation count for a RI-CIS calculation is \(O(ON^2N_x)\) per iteration and transformed trial vector.

The second-order perturbative correction CIS(D) to the CIS excitation energies is calculated from the expression

\[
\omega_{\text{CIS(D)}} = \omega_{\text{CIS}} + \omega_{(D)} = E_{\text{CIS}} A_{\text{eff}}(t^{\text{MP1}}, \omega_{\text{CIS}})E_{\text{CIS}}
\]

(Note that \(t^{\text{MP1}}\) are the first-order double-substitution amplitudes from which also the MP2 ground-state energy is calculated; the first-order single-substitution amplitudes vanish for a Hartree–Fock reference due to the Brillouin theorem.) The operation count for a RI-CIS(D) calculation is similar to that of a single iteration for the CC2 eigenvalue problem. Also disk space requirements are similar.

**Running excitation energy calculations:** The calculation of excitation energies is initiated by the data group \$excitations\$ in which at least the symmetries (irreducible representations) and the number of the excited states must be given (for other options see Section 21.2.17). With the following input the \texttt{ricc2} program will calculate the lowest two roots (states) for the symmetries \(A_1\) and \(B_1\) of singlet multiplicity * at the CIS, CIS(D) and

---

*Provided that it is not an unrestricted open shell run. In this case the wavefunctions will not be spin eigenfunctions and multiplicities are not well defined.
CC2 level with default convergence thresholds. Ground-state calculations will be carried out for MP2 (needed for the CIS(D) model and used as start guess for CC2) and CC2.

$\text{ricc2}\\text{cis}\\text{cis(d)}\\text{cc2}\\$\text{excitations}\\text{irrep}=a1 \text{ nexc}=2\\text{irrep}=b1 \text{ nexc}=2\\$

The single-substitution parts of the right eigenvectors are stored in files named CCRE0-$s--m--xxx$, where $s$ is the number of the symmetry class (irreducible representation), $m$ is the multiplicity, and $xxx$ the number of the excitation within the symmetry class. For the left eigenvectors the single-substitution parts are stored in files named CCLE0-$s--m--xxx$. These files can be kept for later restarts.

**Trouble shooting:** For the iterative second-order methods CIS(D$_{\infty}$), ADC(2), and CC2 the solution of the nonlinear partitioned eigenvalue problem proceeds usually in three steps:

1. solution of the CCS/CIS eigenvalue problem to generate reasonable start vectors; the eigenvectors are converged in this step only to a remaining residual norm $< \text{preopt}$
2. pre-optimization of the eigenvectors by a robust modified Davidson algorithm (see ref. [8]) using the LINEAR CC RESPONSE SOLVER until the norm of all residuals are below $\text{preopt}$, combined with a DIIS extrapolation for roots assumed to be converged below the threshold $\text{thrdiis}$.
3. solution of the nonlinear eigenvalue problem with a DIIS algorithm using the DIIS CC RESPONSE SOLVER until the norm of the residuals are below the required threshold $\text{conv}$

This procedure is usually fairly stable and efficient with the default values for the thresholds. But for difficult cases it can be necessary to select tighter thresholds. In case of convergence problems the first thing do is to verify that the ground state is not a multireference case by checking the D1 diagnostic. If this is not the case the following situations can cause problems in the calculation of excitation energies:

- almost degenerate roots in the same symmetry class
- complex roots (break down of the CC approximation close to conical intersections)
- large contributions from double excitations

The first two reasons can be identified by running the program with a print level $\leq 3$. It will then print in each iteration the actual estimates for the eigenvalues. If some of these are very close or if complex roots appear, you should make sure that the DIIS procedure is not
10.2. CALCULATION OF EXCITATION ENERGIES

switched on before the residuals of the eigenvectors are small compared to the differences in the eigenvalues. For this, \texttt{thrdiis} (controlling the DIIS extrapolation in the linear solver) should be set about one order of magnitude smaller than the smallest difference between two eigenvalues and \texttt{preopt} (controlling the switch to the DIIS solver) again about one order of magnitude smaller then \texttt{thrdiis}.

Tighter thresholds or difficult situations can make it necessary to increase the limit for the number of iterations \texttt{maxiter}.

In rare cases complex roots might persist even with tight convergence thresholds. This can happen for CC2 and CIS(D) close to conical intersections between two states of the same symmetry, where CC response can fail due to its non-symmetric Jacobian. In this case one can try to use instead the ADC(2) model. But the nonlinear partitioned form of the eigenvalue problem used in the \texttt{ricc2} program is not well suited to deal with such situations.

Diagnostics for double excitations: As pointed out in ref. \cite{10}, the $\%T_1$ diagnostic (or $\%T_2 = 100 - \%T_1$) which is evaluated directly from the squared norm of the single and double excitation part of the eigenvectors $\%T_1 = 100 \cdot T_1/(T_1 + T_2)$ with $T_i = \sum \mu_i E^i_{\mu i}$, where the excitation amplitudes are for spin-free calculations in a corresponding spin-adapted basis (which is not necessarily normalized) has the disadvantage that the results depend on the parameterization of the (spin-adapted) excitation operators. This prevents in particular a simple comparison of the results for singlet and triplet excited states if the calculations are carried out in a spin-free basis. With the biorthogonal representation for singlet spin-coupled double excitations \cite{136} results for $\%T_1$ also differ largely between the left and right eigenvectors and are not invariant with respect to unitary transformations of the occupied or the virtual orbitals.

The \texttt{ricc2} module therefore uses since release 6.5 an alternative double excitation diagnostic, which is defined by $\%T_1 = 100 \cdot T_1/(T_1 + T_2)$ with $T_1 = \sum \mu_i E^i_{\mu i}$ and $T_2 = \sum_{i>j} \sum_{a>b} E^2_{abij}$ with $E_{\mu i}$ and $E_{abij}$ in the spin-orbital basis. They are printed in the summaries for excitation energies under the headings $\%t1$ and $\%t2$. For spin-adapted excitation amplitudes $T_1$ and $T_2$ have to be computed from respective linear combinations for the amplitudes which reproduce the values in the spin-orbital basis. For ADC(2), which has a symmetric secular matrix with identical left and right normalized eigenvectors $T_1$ and $T_2$ are identical with the contributions from the singles and doubles parts for the eigenvectors to the trace of the occupied or virtual block of the (orbital unrelaxed) difference density between the ground and the excited state, i.e. the criterium proposed in ref. \cite{10}. Compared to the suggestion from ref. \cite{10} $T_1$ and $T_2$ have the additional advantage of that they are for all methods guaranteed to be positive and can be evaluated with the same insignificantly low costs as $T_1$ and $T_2$. They are invariant with respect to unitary transformations of the occupied or the virtual orbitals and give by construction identical results in spin-orbital and spin-free calculations. For CC2 and CIS(D) the diagnostics $T_1$ and $T_2$ agree for left and right eigenvectors usually with a few 0.01%, for CIS(D) and ADC(2) they are exactly identical. For singlet excitations in spin-free calculations, $\%T_2$ is typically by a factors of 1.5–2 larger than $\%T_2$. The second-order methods CC2, ADC(2), CIS(D) and CIS(D) can usually be trusted for $\%T_2 \leq 15\%$. 

For compatibility, the program can be switched to use of the old $\%T_1$ and $\%T_2$ diagnostics (printed with the headers $||T_1||$ and $||T_2||$) by setting the flag oldnorm in the data group $\$excitations$. Note that the choice of the norm affects the individual results left and right one- and two-photon transition moments, while transition strengths and all other observable properties independent of the individual normalization of the right and left eigenvectors.

The $\%T_2$ and $\%T_2$ diagnostics cannot be monitored in the output of the (quasi-) linear solver. But it is possible to do in advance a CIS(D) calculation. The CIS(D) results for the $\%T_2$ and $\%T_2$ correlate usually well with the results for this diagnostic from the iterative second-order models, as long as there is clear correspondence between the singles parts of the eigenvectors. Else the DIIS solver will print the doubles diagnostics in each iteration if the print level is set $> 3$. States with large double excitation contributions converge notoriously slow (a consequence of the partitioned formulation used in the ricc2 program). However, the results obtained with second-order methods for doubly excited states will anyway be poor. It is strongly recommended to use in such situations a higher-level method.

**Visualization of excitations:** An easy way to visualize single excitations is to plot the natural transition orbitals that can be obtained from a singular value decomposition of the excitation amplitudes. See Sec. 17.1 for further details.

Another, but computationally more involved possibility is plot the difference density between the ground and the respective excited state. This requires, however, a first-order property or gradient calculation for the excited state to obtain the difference density. For further details see Sec. 10.3.3.

### 10.3 First-Order Properties and Gradients

For the ground state first-order properties (expectation values) are implemented at the SCF, MP2 and CC2 level. Note that for the ground state CCS and CIS are equivalent to SCF. For excited states first-order properties are implemented only at the CCS and CC2 level. Gradients are presently only available for the ground state at the MP2 and the CC2 and for excited states only at the CC2 level.

#### 10.3.1 Ground State Properties, Gradients and Geometries

For CC2, one distinguishes between orbital-relaxed and unrelaxed properties. Both are calculated as first derivatives of the respective energy with respect to an external field corresponding to the calculated property. They differ in the treatment of the SCF orbitals. In the orbital-relaxed case the external field is (formally) already included at the SCF stage and the orbitals are allowed to relax in the external field; in the orbital-unrelaxed case the external field is first applied after the SCF calculation and the orbitals do not respond to the external field. *Orbital-unrelaxed* CC2 properties are calculated as first derivatives of the
10.3. FIRST-ORDER PROPERTIES AND GRADIENTS

real part of the unrelaxed Lagrangian \[133\]

\[
L_{\text{ur \ CC2}}(t, \bar{t}, \beta) = \langle HF| H| \text{CC} \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{H} + [\hat{H}, T_2]| HF \rangle + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | \hat{H} + [F_0 + \beta \hat{V}, T_2]| HF \rangle
\]

with \( H = H_0 + \beta V \)—where \( V \) is the (one-electron) operator describing the external field, \( \beta \) the field strength, and \( H_0 \) and \( F_0 \) are the Hamiltonian and Fock operators of the unperturbed system—by the expression:

\[
\langle V \rangle_{\text{ur \ CC2}} = \Re \left( \frac{\partial L_{\text{ur \ CC2}}(t, \bar{t}, \beta)}{\partial \beta} \right)_0 = \sum_{pq} D_{pq}^{\text{ur}} V_{pq}, \tag{10.13}
\]

where \( \Re \) indicates that the real part is taken. Relaxed CC2 properties (and gradients) are calculated from the the full variational density including the contributions from the orbital response to the external perturbation, which are derived from the Lagrangian \[11,137\]

\[
L_{\text{rel \ CC2}}(t, \bar{t}) = \langle HF| H| \text{CC} \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{H} + [\hat{H}, T_2]| HF \rangle + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | [\hat{V}, T_2]| HF \rangle + \sum_{\mu_0} \bar{\kappa}_{\mu_0} F_{\mu_0},
\]

where \( F \) is the Fock operator corresponding to the Hamiltonian of the perturbed system \( H = H_0 + \beta V \). One-electron properties are then obtained as:

\[
\langle V \rangle_{\text{rel \ CC2}} = \Re \left( \langle HF| \hat{V}| HF \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{V} + [V, T_2]| HF \rangle + \sum_{\mu_2} \bar{t}_{\mu_2} \langle [\hat{V}, T_2]| HF \rangle + \sum_{\mu_0} \bar{\kappa}_{\mu_0} V_{\mu_0} \right),
\tag{10.16}
\]

The calculation of one-electron first-order properties requires that in addition to the cluster equations also the linear equations for the Lagrangian multipliers \( \bar{t}_\mu \) are solved, which requires similar resources (CPU, disk space, and memory) as the calculation of a single excitation energy. For orbital-relaxed properties also a CPHF-like linear equation for the Lagrangian multipliers \( \bar{\kappa}_{\mu_0} \) needs to be solved and the two-electron density has to be build, since it is needed to set up the inhomogeneity (right-hand side). The calculation of relaxed properties is therefore somewhat more expensive—the operation count for solving the so-called Z-vector equations is similar to what is needed for an SCF calculation—and requires also more disk space to keep intermediates for the two-electron density—about \( O(2V + 2N)N_x + N_x^2 \) in addition to what is needed for the solution of the cluster equations. For ground states, orbital-relaxed first-order properties are standard in the literature.
The calculation of the gradient implies the calculation of the same variational densities as needed for relaxed one-electron properties and the solution of the same equations. The construction of the gradient contributions from the densities and derivative integrals takes about the same CPU time as 3–4 SCF iterations and only minor extra disk space. For details of the implementation of CC2 relaxed first-order properties and gradients and a discussion of applicability and trends of CC2 ground-state equilibrium geometries see ref. [11]. The following is an example input for a MP2 and CC2 single point calculation of first-order properties and gradients:

```
$rcc2
  mp2
  cc2
$response
  static relaxed operators=diplen,qudlen
  gradient
```

A different input is required for geometry optimizations: in this case the model for which the geometry should be optimized must be specified in the data group `$rcc2` by the keyword `geoopt`:

```
$rcc2
  mp2
  cc2
  geoopt model=cc2
```

For CC2 calculations, the single-substitution part of the Lagrangian multipliers $\tilde{t}_\mu$ are saved in the file `CCL0--1--1---0` and can be kept for a restart (for MP2 and CCS, the single-substitution part $\tilde{t}_\mu$ vanishes).

For MP2 only relaxed first-order properties and gradients are implemented (unrelaxed MP2 properties are defined differently than in CC response theory and are not implemented). For MP2, only the CPHF-like Z-vector equations for $\tilde{\kappa}_\mu$ need to be solved, no equations have to be solved for the Lagrangian multipliers $\tilde{t}_\mu$. CPU time and disk space requirements are thus somewhat smaller than for CC2 properties or gradients.

For SCF/CIS/CCS it is recommended to use the modules `grad` and `rdgrad` for the calculation of, ground state gradients and first-order properties.

### 10.3.2 Excited State Properties, Gradients and Geometries

Also for excited states presently unrelaxed and relaxed first-order properties are available in the `ricc2` program. These are implemented for CCS and CC2. Note, that in the unrelaxed case CIS and CCS are not equivalent for excited-states first-order properties and no first-order properties are implemented for CIS in the `ricc2` program.
Orbital-unrelaxed first-order properties

The unrelaxed first-order properties are calculated from the variational excited states Lagrangian \[ L_{ur}^{CC2, ex}(E, \bar{E}, t, \bar{t}^{(ex)}, \beta) = \langle HF|H|CC \rangle + \sum_{\mu \nu} E_\mu A_{\mu \nu}(t, \beta) E_\nu \] (10.18) where it is assumed that the left and right eigenvectors are normalized such that \( \sum_{\mu \nu} \bar{E}_\mu \langle \mu | \tau_\nu \rangle E_\nu = 1 \) and \( H = H_0 + \beta V \). The first-order properties are calculated as first derivatives of \( L_{ur}^{CC2, ex}(E, \bar{E}, t, \bar{t}^{(ex)}, \beta) \) with respect to the field strength \( \beta \) and are evaluated via a density formalism:

\[ \langle V \rangle_{ur,ex} = R \left( \frac{\partial L_{ur,ex}^{CC2}(E, \bar{E}, t, \bar{t}^{(ex)}, \beta)}{\partial \beta} \right)_0 = \sum_{pq} D_{pq}^{ur,ex} V_{pq} \] (10.19)

(Again \( R \) indicates that the real part is taken.) The unrelaxed excited-state properties obtained thereby are related in the same way to the total energy of the excited states as the unrelaxed ground-state properties to the energy of the ground state and the differences between excited- and ground-state unrelaxed properties are identical to those identified from the second residues of the quadratic response function. For a detailed description of the theory see refs. [137,138]; the algorithms for the RI-CC2 implementation are described in refs. [10,135]. ref. [135] also contains a discussion of the basis set effects and the errors introduced by the RI approximation.

The calculation of excited-state first-order properties thus requires the calculation of both the right \( (E_\mu) \) and left \( (\bar{E}_\mu) \) eigenvectors and of the excited state Lagrangian multipliers \( \bar{t}^{(ex)}_\mu \). The disk space and CPU requirements for solving the equations for \( \bar{E}_\mu \) and \( \bar{t}^{(ex)}_\mu \) are about the same as those for the calculation of the excitation energies. For the construction of the density matrices in addition some files with \( \mathcal{O}(n_{root} N^2) \) size are written, where \( n_{root} \) is the number of excited states.

The single-substitution parts of the excited-states Lagrangian multipliers \( \bar{t}^{(ex)}_\mu \) are saved in files named CCONLO-ss-xxxxxx.

For the calculation of first-order properties for excited states, the keyword \texttt{exprop} must be added with appropriate options to the data group \$excitations; else the input is same as for the calculation of excitation energies:

\$ricc2
cc2
$response
fop unrelaxed_only operators=diplen,qudlen
Orbital-relaxed first-order properties and gradients

To obtain orbital-relaxed first-order properties or analytic derivatives (gradients) the Lagrange functional for the excited state in Eq. (10.18) is—analogously to the treatment of ground states—augmented by the equations for the SCF orbitals and the perturbations is also included in the Fock operator:

\[ L^{\text{rel CC2, ex}}(E, \hat{E}, t, \tilde{t}^{(ex)}, \beta) = \langle \text{HF} | H | \text{CC} \rangle + \sum_{\mu \nu} E_{\mu} A_{\mu \nu}(t, \beta) E_{\nu} \]

\[ + \sum_{\mu_1} \bar{t}_{\mu_1}^{(ex)} \langle \mu_1 | \hat{H} + [\hat{H}, T_2] | \text{HF} \rangle \]

\[ + \sum_{\mu_2} \bar{t}_{\mu_2}^{(ex)} \langle \mu_2 | \hat{H} + [F, T_2] | \text{HF} \rangle + \sum_{\mu_0} \bar{\kappa}_{\mu_0}^{(ex)} F_{\mu_0} \, . \]

Compared to unrelaxed properties, the calculation of relaxed properties needs in addition for each excited state the solution of a CPHF equations for the Lagrangian multipliers \( \bar{\kappa}_{\mu_0}^{(ex)} \), for which the computational costs are similar to those of a Hartree-Fock calculation.

Orbital-relaxed properties are requested by adding the flag \texttt{relaxed} to the input line for the \texttt{exprop} option. The following is an example for a CC2 single point calculation for orbital-relaxed excited state properties:

\`
$ricc2$
cc2
$excitations$
  irrep=a1 nexc=2
  exprop states=all relaxed operators=diplen,qudlen
``

Note that during the calculation of orbital-relaxed excited-state properties the corresponding unrelaxed properties are also automatically evaluated at essentially no additional costs. Therefore, the calculation of unrelaxed properties can not be switched off when relaxed properties have been requested.

Again the construction of gradients requires the same variational densities as needed for relaxed one-electron properties and the solution of the same equations. The construction of the gradient contributions from one- and two-electron densities and derivative integrals takes approximately the same time as for ground states gradients (approx. 3–4 SCF iterations) and only minor extra disk space. The implementation of the excited state gradients for the RI-CC2 approach is described in detail in Ref. [12]. There one can also find some information about the performance of CC2 for structures and vibrational frequencies of excited states.

For the calculation of an excited state gradient with CC2 at a single point (without geometry optimization and if it is not a calculation with \texttt{NumForce}) one can use the input:

\`
$ricc2$
cc2
$excitations$
  irrep=a1 nexc=2
  exprop states=all relaxed operators=diplen,qudlen
``
$ricc2
cc2
$excitations
  irrep=a1 nexc=2
  xgrad states=(a1 1-2)

For geometry optimizations or a numerical calculation of the Hessian with NumForce the wavefunction model and the excited state for which the geometry should be optimized have to be specified in the data group $ricc2 with the keyword geoopt:

$ricc2
  geoopt model=cc2 state=(a1 2)
$excitations
  irrep=a1 nexc=2

If the geometry optimization should carried out for the lowest excited state (of those for which an excitation energy is requested in $excitation), one can use alternatively state=(s1).

Since the calculation of unrelaxed and relaxed first-order properties can be combined gradient calculations without significant extra costs, a request for excited state gradients will automatically enforce the calculation of the relaxed and unrelaxed dipole moments. If the keyword geoopt is used, the relaxed dipole moment for the specified excited state and wavefunction model will be written to the control file and used in calculations with NumForce for the evaluation of the IR intensities.

10.3.3 Visualization of densities and Density analysis

As most other programs which allow for the calculation of wavefunctions and densities also the ricc2 module is interfaced to wavefunction analysis and visualization toolbox described in chapter 17. From ricc2 module this interface can used in two different ways

1. If through the geoopt keyword in $ricc2 a unique method and state has been specified for which the density, gradient and properties are evaluated, the density analysis and visualization routines will called by default with the (orbital-relaxed) density for this state and method similar as in dscf, ridft, mpgrad, etc.

2. The ricc2 program can be called in a special analysis mode which allows to analyse densities and combination (e.g. differences) of densities evaluated in preceeding ricc2 calculations.

Default density analysis and visualization:

As in a single calculations with the ricc2 program one–electron densities can be calculated for more than one method and/or electronic state, the interface to the analysis and visualization routines require the specification of a unique level of calculation and a unique state.
This is presently done through the `geoopt` flag which determines the method/state for which results are written to interface files (e.g. `control`, `gradient`, or `xxx.map`).

In ground state calculations `ricc2` will pass to the density analysis routines the correlated total (and for UHF based calculations also the spin) density and the canonical SCF orbitals from which the SCF (spin) density is constructed. All options described in chapter 17 are available from within the `ricc2` program apart from the evaluation of electrostatic moments, which would interfere with the calculation of expectation values requested through the `fop` option in `$response$`.

In excited state calculation `ricc2` will pass the excited state total (and for UHF based calculation in addition the spin) density. But no ground state densities and/or uncorrelated densities or orbitals. Thus, for excited states the `ricc2` program does, in difference to `egrad` not print out a comparison with the ground state SCF density. Also, all some options which require orbitals (as e.g. the generation and visualization of localized orbitals or some population analysis options) and not available for excited states in `ricc2`.

As other modules, also `ricc2` provides the `-proper` flag to bypass a re-calculation of the density and gradient to enter immediately the density analysis routines with a previously calculated density. The `ricc2` program will then pass the densities found on the interface file for the density analysis routines without further check on the method and state for which they have been evaluated. If both, ground and excited state densities are found on file, both will be passed to the density analysis, thereby providing a shortcut to the `-fanal` and the `$anadens$` keyword for the analysis of differences between ground and excited state densities.

**The general density analysis option:**

In general `ricc2` saves by default all relaxed densities generated during a calculation in files named `cc1td-<type>-<mult><irrep>-<number>`, where `cc1td` stands for “coupled-cluster one-electron total density”. `<type>` is one of `mp2-gs` (MP2 ground state), `cc2-gs` (CC2 ground state), `ccs-xs` (CCS excited state), `cc2-xs` (CC2 excited state), or `adc2-xs` (ADC(2) excited state) and the other entries specify multiplicity, irreducible representation and the number of the state. Having specified the calculation of relaxed densities—e.g. by requesting relaxed one-electron properties or as a by-product of a gradient calculation—you will end up with two files named like

```
cc1td-cc2-gs-1a1-001
cc1td-cc2-xs-3a2-001
```

In case of open shell molecules, additional files with names `cc1sd...` (for one-electron spin-densities) will be generated.

These files are (currently) in a binary format, similar as the files `dens`, `mdens` and `edens`. Therefore be aware that a transfer between different computer architectures may result in trouble.

The densities on these files can be analysed with the tools and interfaces provided by Moloch (see Section 17.2). This can be done by calling `ricc2` with the option `-fanal` which by-
10.4. TRANSITION MOMENTS

passes the usual wavefunction calculation and triggers the program into an analysis mode for densities. In this mode the program interprets $\texttt{anadens}$ and the keywords described in Section 17.2. To plot, for example, the difference density of the two above mentioned total densities you have to add the following lines in your control file

```plaintext
$\texttt{anadens}
  \texttt{calc my_favourite_diffden from}
  1d0 cc1td-cc2-xs-3a2-001
  -1d0 cc1td-cc2-gs-1a1-001
$\texttt{pointval}
```

and invoke

```
ricc2 -fanal
```

This will generate the files $\texttt{my_favourite_diffden}$ and $\texttt{my_favourite_diffden.map}$. The latter can be converted into gOpenMol format as described in Section 17.2.

10.3.4 Fast geometry optimizations with RI-SCF based gradients

If geometry optimizations on MP2 or CC2 level are performed with large basis set, especially with diffuse basis functions, the $N^4$-steps might become the dominant part of the overall timings. In these cases, the integral screening in the Hartree–Fock part often becomes inefficient. The resolution–of–the–identity can be applied here to speed up the calculation of the HF reference wavefunction, as well as the solution of the coupled–perturbed Hartree–Fock (CPHF) equations in the MP2 or CC2 gradient calculation.

An additional auxiliary basis (denoted $\texttt{jkbas}$) set has to be assigned via the General Options Menu in the define program. In the submenu $\texttt{rijk}$ choose on and select your auxiliary basis set. Then, run the $\texttt{jobex}$ script the additional $\texttt{rijk}$-flag:

```
> jobex -level cc2 -rijk
```

10.4 Transition Moments

Transition moments are presently implemented for excitations out of the ground state and for excitations between excited states for the coupled cluster models CCS and CC2. Transition moments for excitations from the ground to an excited state are also available for ADC(2), but use an additional approximation (see below). Note, that for transition moments (as for excited-state first-order properties) CCS is not equivalent to CIS and CIS transition moments are not implemented in the ricc2 program.
10.4.1 Ground to excited state transition moments

In response theory, transition strengths (and moments) for transitions from the ground to excited state are identified from the first residues of the response functions. Due to the non-variational structure of coupled cluster different expressions are obtained for the CCS and CC2 “left” and “right” transitions moments $M_{0\rightarrow f}$ and $M_{f\rightarrow 0}$. The transition strengths $S_{V_1V_2}^{0f}$ are obtained as a symmetrized combinations of both \[139]\):

$$S_{V_1V_2}^{0f} = \frac{1}{2} \left\{ M_{0\rightarrow f}^{V_1} M_{f\rightarrow 0}^{V_2} + \left( M_{0\rightarrow f}^{V_2} M_{f\rightarrow 0}^{V_1} \right)^* \right\}$$  \hspace{1cm} (10.21)

Note, that only the transition strengths $S_{V_1V_2}^{0f}$ are a well-defined observables but not the transition moments $M_{0\rightarrow f}$ and $M_{f\rightarrow 0}$. For a review of the theory see refs. [137, 139]. The transition strengths calculated by coupled-cluster response theory according to Eq. (10.21) have the same symmetry with respect to an interchange of the operators $V_1$ and $V_2$ and with respect to complex conjugation as the exact transition moments. In difference to SCF (RPA), (TD)DFT, or FCI, transition strengths calculated by the coupled-cluster response models CCS, CC2, etc. do not become gauge-independent in the limit of a complete basis set, i.e., for example the dipole oscillator strength calculated in the length, velocity or acceleration gauge remain different until also the full coupled-cluster (equivalent to the full CI) limit is reached.

For a description of the implementation in the ricc2 program see refs. [11, 135]. The calculation of transition moments for excitations out of the ground state resembles the calculation of first-order properties for excited states: In addition to the left and right eigenvectors, a set of transition Lagrangian multipliers $\bar{M}_\mu$ has to be determined and some transition density matrices have to be constructed. Disk space, core memory and CPU time requirements are thus also similar.

The single-substitution parts of the transition Lagrangian multipliers $\bar{N}_\mu$ are saved in files named CCMEO-<s>-<m>-xxx.

To obtain the transition strengths for excitations out of the ground state the keyword \texttt{spectrum} must be added with appropriate options (see Section 21.2.17) to the data group \texttt{$\$excitations}; else the input is same as for the calculation of excitation energies and first-order properties:

```
$ricc2
c2
$excitations
  irrep=a1 nexc=2
  spectrum states=all operators=diplen,qudlen
```

For the ADC(2) model, which is derived by a perturbation expansion of the expressions for exact states, the calculation of transition moments for excitations from the ground to an excited state would require the second-order double excitation amplitudes for the ground state wavefunction, which would lead to operation counts scaling as $\mathcal{O}(N^6)$, if no further
10.4. TRANSITION MOMENTS

approximations are introduced. On the other hand the second-order contributions to the transition moments are usually not expected to be important. Therefore, the implementation in the ricc2 program neglects in the calculation of the ground to excited state transition moments the contributions which are second order in ground state amplitudes (i.e. contain second-order amplitudes or products of first-order amplitudes). With this approximation the ADC(2) transition moments are only correct to first-order, i.e. to the same order to which also the CC2 transition moments are correct, and are typically similar to the CC2 results. The computational costs for the ADC(2) transition moments are (within this approximation) much lower than for CC2 since the left and right eigenvectors are identical and no lagrangian multipliers need to be determined. The extra costs (i.e. CPU and wall time) for the calculations of the transitions moments are similar to the those for two or three iterations of the eigenvalue problem, which reduces the total CPU and wall time for the calculation of a spectrum (i.e. excitation energies and transition moments) by almost a factor of three.

10.4.2 Transition moments between excited states

For the calculation of transition moments between excited states a set of Lagrangian multipliers $\bar{\mu}^\mu$ has to be determined instead of the $\bar{\mu}^\mu$ for the ground state transition moments. From these Lagrangian multipliers and the left and right eigenvectors one obtains the “right” transition moment between two excited states $i$ and $f$ as

$$M^V_{f-i} = \sum_{pq} \left\{ D^x_{pq}(\tilde{N}_f^i) + D^A_{pq}(\tilde{E}_f^i, E^i) \right\} \tilde{V}_pq. \quad (10.22)$$

where $\tilde{V}$ are the matrix elements of the perturbing operator. A similar expression is obtained for the “left” transition moments. The “left” and “right” transition moments are then combined to yield the transition strength

$$S^V_{Vi_1V_2} = \frac{1}{2} \left\{ M^V_{i-f}M^V_{f-i} + \left( M^V_{i-f}M^V_{f-i} \right)^* \right\} \quad (10.23)$$

As for the ground state transitions, only the transition strengths $S^V_{Vi_1V_2}$ are a well-defined observables but not the transition moments $M^V_{i-f}$ and $M^V_{f-i}$.

The single-substitution parts of the transition Lagrangian multipliers $\bar{\mu}^\mu$ are saved in files named ccnc0-s-m-xxx.

To obtain the transition strengths for excitations between excited states the keyword tmexc must be added to the data group $\$excitations$. Additionally, the initial and final states must be given in the same line; else the input is same as for the calculation of excitation energies and first-order properties:

\$ricc2
  cc2
$excitations
  irrep=al nexc=2
10.5 Ground State Second-order Properties with MP2 and CC2

For closed-shell restricted Hartree-Fock reference states second-order properties for one-electron perturbation can be computed at the MP2 and the CC2 level. For MP2, second-order properties are computed as derivatives of the SCF + MP2 total energy. This approach includes the relaxation of the SCF orbitals in the presence of the perturbation and is restricted to the static (i.e. frequency-independent) limit.

For coupled-cluster model CC2, second-order properties can, similar as the first-order properties, calculated in orbital-unrelaxed or orbital-relaxed approach as derivatives of the Lagrange functions in Eqs. 10.12 and 10.15. As for MP2, the orbital-relaxed calculations are restricted to the static limit. Frequency-dependent second-order properties as e.g. dipole polarizabilities can be computed with the orbital-unrelaxed approach.

Note that second-order properties are currently not yet available in the MPI parallel version or for spin-component scaled variants of MP2 and CC2. Furthermore, non-Abelian point groups are not implemented for second-order properties.

In addition to the standard input, second-order properties require that the data group for the numerical Laplace transformation \$laplace and that the sop s option in the data group \$response is set. Frequency-dependent dipole polarizabilities with the CC2 model are obtained with the input:

\[
\texttt{$ricc2 cc2$laplace conv=4$response}
\]

The frequency has to be given in atomic units. Static orbital relaxed polarizabilities are obtained with

\[
\texttt{$response sop operators=(diplen,diplen) relaxed$}
\]

10.6 Parallel RI-MP2 and RI-CC2 Calculations

The ricc2 program is partially parallelized for distributed memory architectures (e.g. clusters of Linux boxes) based on the message passing interface (MPI) standard. In the present
version parallel calculations can be carried out for ground state and excitation energies for all wavefunction models available in ricc2. The analytic gradients for RI-MP2 and RI-CC2 in the ground state and RI-CC2 in excited states are also parallelized.

While in general the parallel execution of ricc2 works similar to that of other parallelized Turbomole modules (as e.g. dscf and grad), there are some important differences concerning in particular the handling of the large scratch files needed for RI-CC2 (or RI-MP2). As the parallel version dscf also the parallel version of ricc2 assumes that the program is started in a directory which is readable (and writable) on all compute nodes under the same path (e.g. a NFS directory). The directory must contain all input files and will at the end of a calculation contain all output files. Large scratch files (e.g. for integral intermediates) will be placed under the path specified in the control file with $tmpdir$ (see Section 21.2.17) which should point to a directory in a file system with a good performance. All large files will be placed on the nodes in these file systems. (The local file system must have the same name on all nodes) Note that at the end of a ricc2 run the scratch directories specified with $tmpdir$ are not guaranteed to be empty. To avoid that they will fill your file system you should remove them after the ricc2 calculation is finished.

Another difference to the parallel HF and DFT (gradient) programs is that ricc2 will communicate much larger amounts of data between the compute nodes. With a fast network interconnection (Gigabit or better) this should not cause any problems, but with slow networks the communication might become the limiting factor for performance or overloading the system. If this happens the program can be put into an alternative mode where the communication of integral intermediates is replaced by a reevaluation of the intermediates (at the expense of a larger operation count) wherever this is feasible. Add for this in the control the following data group:

```
$mpi_param
  min_comm
```

### 10.7 Spin-component scaling approaches (SCS/SOS)

By introducing individual scaling factors for the same-spin and opposite-spin contributions to the correlation energy most second-order methods can be modified to achieve a (hopefully) better performance. SCS-MP2 has first been proposed by S. Grimme and SOS-MP2 by Y. Jung et al. (see below). The generalization of SCS and SOS to CC2 and ADC(2) for ground and excited states is described in [14]. It uses the same scaling factors as proposed for the original SCS- and SOS-MP2 approaches (see below). In the ricc2 program we have also implemented SCS and SOS variants of CIS(D) for excitation energies and of CIS(D∞) for excitation energies and gradients, which are derived from SCS-CC2 and SOS-CC2 in exactly the same manner as the unmodified methods can be derived as approximations to CC2 (see Sec. 10.2 and ref. [140]). Please note, that the SCS-CIS(D) and SOS-CIS(D) approximations obtained in this way and implemented in ricc2 differ from the spin-component scaled SCS- and SOS-CIS(D) methods proposed by, respectively, S. Grimme and E. I. Ugorodina in [141].
and Y. M. Rhee and M. Head–Gordon in [142].

A line with scaling factors has to be added in the $ricc2$ data group:

```
$ricc2
scs cos=1.2d0 css=0.3333d0
```

cos denotes the scaling factor for the opposite–spin component, css the same–spin component.

As an abbreviation scs can be inserted in $ricc2$. In this case, the SCS parameters cos=6/5 and css=1/3 proposed S. Grimme (S. Grimme, J. Chem. Phys. 118 (2003) 9095.) are used. These parameters are also recommended in [14] for the SCS variants of CC2, CIS(D), CIS(D∞), and ADC(2) for ground and excited states.

Also, just sos can be used as a keyword, to switch to the SOS approach proposed by the Head-Gordon group for MP2 with scaling factors of cos=1.3 and css=0.0 (Y., Jung, R.C. Lochan, A.D. Du- 

toi, and M. Head-Gordon, J. Chem. Phys. 121 (2004) 9793.,) which are also recommended for the SOS variants of CC2, CIS(D), CIS(D∞), and ADC(2). The Laplace-transformed algorithm for the SOS variants are activated by the additional data group $laplace$:

```
$laplace
  conv=4
```

For further details on the Laplace-transformed implementation and how one can estimated whether the $O(N^4)$-scaling Laplace-transformed or $O(N^5)$-scaling conventional RI imple-
mentation is efficient see Sec. 9.6.

Since Version 6.6 the $O(N^4)$-scaling Laplace-transformed implementation is available for ground and excited state gradients with CC2 and ADC(2).

Restrictions:

- the spin ($S^2$) expectation value for open-shell calculation can not be evaluated in the SCS or SOS approaches
- for LT-SOS-CC2 (and the related CIS(D) and ADC(2) versions) the following further limitations apply:
  - only parallelized with MPI (no OpenMP parallelization)
  - incompatible with the calculation of the $D_1$ and $D_2$ diagnostics
Chapter 11

CCSD, CCSD(F12*) and CCSD(T) calculations

Since Release V7.0 then coupled-cluster singles-and-doubles method CCSD and its explicitly-correlated variants CCSD(F12) and CCSD(F12*) are implemented in the ccdf12 program. CCSD and the F12 variants can be combined with a perturbative correction for connected triple excitations, CCSD(T).* As perturbative approximations beyond MP2 also the approximations MP3, MP3(F12), MP4, and MP4(F12*) are available. Presently the implementation of the F12 variants and of connected triple excitations is restricted to ground state energies and the CCSD implementation to ground–state and excitation energies. Closed-shell (RHF), unrestricted (UHF) or single determinant restricted (ROHF) open-shell reference wavefunctions can be used for CCSD and CCSD(T), but no gradients or properties are (yet) available for these wavefunction models. The MP3 and MP4 approximations can currently not be combined with ROHF reference wavefunctions.

Further limitations:

no MPI parallelization, calculations at these levels can presently only carried out on a single compute node, only the OpenMP (see Sec. 3.2.2) parallelization is available for calculations beyond CC2.

use of symmetry restricted to \( D_{2h} \) and its subgroups for the conventional implementation; no symmetry can be used for the F12 methods

Please note that calculations with MP3, MP4, CCSD and methods beyond CCSD require considerably more disc space and core memory than MP2 or CC2 calculations. (See section below for more details and recommendations.)

*Note that for the explicitly correlated CCSD variants the explicitly-correlated double excitations are neglected for the calculation of the triples corrections.
Prerequisites

MP3, MP4, CCSD and CCSD(T) calculations with the ccsdf12 module require the same prerequisites as RI-CC2 calculations,

1. a converged SCF calculation with the one-electron density threshold set to $\text{denconv} = 1 \cdot 10^{-5}$ or less
2. an auxiliary basis defined in the data group $\text{cbas}$
3. if orbitals should be excluded from the correlation treatment the data group $\text{freeze}$ has to be set
4. the maximum core memory which the program is allowed to allocate should be defined in the data group $\text{maxcor}$; the recommended value is 66–75% of the available (physical) core memory.
5. the data group $\text{ricc2}$ with a specification of the coupled-cluster model

Calculations with the CCSD(F12*) and CCSD(F12) methods require in addition:

- the data group $\text{rir12}$ with the definition of the standard approximations for the explicitly-correlated contributions (see Sec. 9.5 for details)
- the data group $\text{lcg}$, which define the correlation function (here it is in particular important to choose for F12 calculations the exponent; recommended values are 0.9 for cc-pVDZ-F12, 1.0 for cc-pVTZ-F12 and 1.1 for cc-pVQZ-F12 basis sets)
- a complementary auxiliary (CABS) basis set

Furthermore it is recommended to select in addition an auxiliary JK basis set for the evaluation of the Fock matrix elements. (The rijk menu of define can be used for this.)

How To Perform a Calculation

As presently no gradients are available, only single-point calculations are possible:

1. Select in define within the menu cc
   
   the wavefunction model (submenu ricc2),
   
   frozen core options (submenu freeze),
   
   an auxiliary basis for $\text{cbas}$ (submenu cbas),
   
   the amount of main memory (option memory),
   
   and for CCSD(F12) calculations in addition
   
   the F12 options (submenu f12), and
   
   a CABS basis (submenu cabs).
11.1. CHARACTERISTICS OF THE IMPLEMENTATION AND COMPUTATIONAL DEMANDS

By default a CCSD(F12) with ansatz 2 and geminal amplitudes fixed by the cusp conditions is performed. To switch to the computationally more efficient recommended CCSD(F12*) approximation add to the input group \$rir12 the line:

\[
\text{ccsdapprox ccsd(f12*)}
\]

The auxiliary JK basis must be chosen in menu \texttt{rijk} and the exponent for the correlation function must be set by editing the \$lcg data group of the control file.

2. Do an SCF calculations using either the \texttt{dscf} or the \texttt{ridft} module.
3. Invoke the \texttt{ccsdf12} program on the command line or with a batch script.

How to quote:


11.1 Characteristics of the Implementation and Computational Demands

In CCSD the ground–state energy is (as for CC2) evaluated as

\[
E_{\text{CC}} = \langle \text{HF}|H|\text{CC}\rangle = \langle \text{HF}|H \exp(T)|\text{HF}\rangle ,
\]

where the cluster operator \(T = T_1 + T_2\) consist of linear combination of single and double excitations:

\[
T_1 = \sum_{ai} t_{ai} \tau_{ai} ,
\]

\[
T_2 = \frac{1}{2} \sum_{aibj} t_{aibj} \tau_{aibj} .
\]

\[\text{†For other available approximation and the corresponding input options see Sec. 21.2.17.}\]
In difference to CC2, the cluster amplitudes $t_{ai}$ and $t_{aibj}$ are determined from equations which contain no further approximations apart from the restriction of $T$ to single and double excitations:

$$\Omega_{\mu_1} = \langle \mu_1 | \hat{\tilde{H}} + [\hat{\tilde{H}}, T_2] | \text{HF} \rangle = 0,$$

$$\Omega_{\mu_2} = \langle \mu_2 | \hat{\tilde{H}} + [\hat{\tilde{H}}, T_2] + [[\hat{\tilde{H}}, T_2], T_2] | \text{HF} \rangle = 0,$$

where again

$$\hat{\tilde{H}} = \exp(-T_1) \hat{H} \exp(T_1),$$

and $\mu_1$ and $\mu_2$ are, respectively, the sets of all singly and doubly excited determinants. For MP3 the energy is computed from the first-order amplitudes ($t^{(1)}_{iajb}$) as

$$E_{\text{MP3,tot}} = E_{\text{HF}} + E_{\text{MP2}} + E_{\text{MP3}} = \langle \text{HF}, \hat{H} + [\hat{\tilde{H}}, T_2^{(1)}] | \text{HF} \rangle + \sum_{\mu_2} t^{(1)}_{\mu_2} \langle \mu_2 | [\hat{\tilde{W}}, T_2^{(1)}] | \text{HF} \rangle$$

with $\hat{\tilde{W}} = \hat{\tilde{H}} - \hat{\tilde{F}}$. To evaluate the fourth-order energy one needs in addition to the first-order also the second-order amplitudes, which are obtained from the solution of the equations

$$\langle \mu_1 | [\hat{\tilde{F}}, T_2^{(2)}] + [\hat{\tilde{W}}, T_2^{(1)}] | \text{HF} \rangle = 0,$$

$$\langle \mu_2 | [\hat{\tilde{F}}, T_2^{(2)}] + [\hat{\tilde{W}}, T_2^{(1)}] | \text{HF} \rangle = 0,$$

$$\langle \mu_3 | [\hat{\tilde{F}}, T_3^{(2)}] + [\hat{\tilde{W}}, T_2^{(1)}] | \text{HF} \rangle = 0.$$

From these the fourth-order energy correction is computed as:

$$E_{\text{MP4}} = \sum_{\mu_2} t^{(1)}_{\mu_2} \langle \mu_2 | [\hat{\tilde{W}}, T_1^{(2)} + T_2^{(2)} + T_3^{(2)}] + [[\hat{\tilde{W}}, T_2^{(1)}], T_2^{(1)}] | \text{HF} \rangle.$$

Eqs. (11.5) and (11.7) – (11.11) are computational much more complex and demanding than the corresponding doubles equations for the CC2 model. If $N$ is a measure for the system size (e.g., the number of atoms), the computational costs (in terms of floating point operations) for CCSD calculations scale as $O(N^6)$. If for the same molecule the number of one-electron basis functions $N$ is increased the costs scale with $O(N^4)$. (For RI-MP2 and RI-CC2 the costs scale with the system size as $O(N^5)$ and with the number of basis functions as $O(N^3)$.) The computational costs for an MP3 calculations are about the same as for one CCSD iteration. For MP4 the computational costs are comparable to those for two CCSD iteration plus the costs for the perturbation triples correction (see below).

**Explicitly-correlated CCSD(F12) methods:** In explicitly-correlated CCSD calculations the double excitations into products of virtual orbitals, described by $T_2 = \frac{1}{2} \sum_{aibj} t_{aibj} t_{aibj}$, are augmented with double excitations into the explicitly-correlated pairfunctions (geminals) which are described in Sec. 9.5:

$$T = T_1 + T_2 + T_2',$$

$$T_2' = \frac{1}{2} \sum_{ijkl} c_{ijkl} \tilde{T}_{ijkl},$$
where $\tau_{kij} = \hat{Q}_{ij} f_{12} [kl]$ (for the definition $\hat{Q}_{ij}$ and $f_{12}$ see Sec. 9.5). This enhances dramatically the basis set convergence of CCSD calculations (\cite{17}). Without any further approximations than those needed for evaluating the necessary matrix elements, this extension of the cluster operator $T$ leads to the CCSD-F12 method. CCSD(F12) is an approximation (\cite{17,143}) to CCSD-F12 which neglects certain computationally demanding higher-order contributions of $\hat{T}_{2'}$. This reduces the computational costs dramatically, while the accuracy of CCSD(F12) is essentially identical to that of CCSD-F12 \cite{144,145}. In the CCSD(F12) approximation the amplitudes are determined from the equations:

\begin{align}
\Omega_{\mu_1} &= \langle \mu_1 | \hat{H} + [\hat{H}, T_2 + T_{2'}] | \text{HF} \rangle = 0 , \quad (11.14) \\
\Omega_{\mu_2} &= \langle \mu_2 | \hat{H} + [\hat{H}, T_2 + T_{2'}] + [[\hat{H}, T_2 + 2T_{2'}], T_2] | \text{HF} \rangle = 0 , \quad (11.15) \\
\Omega_{\mu_2'} &= \langle \mu_2' | [\hat{F}, T_{2'}] + \hat{H} + [\hat{H}, T_2] | \text{HF} \rangle = 0 . \quad (11.16)
\end{align}

Similar as for MP2-F12, also for CCSD(F12) the coefficients for the doubles excitations into the geminals, $c_{kl}^{ij}$ can be determined from the electronic cusp conditions using the rational generator (also known as SP or fixed amplitude) approach. In this case Eq. (11.16) is not solved. To account for this, the energy is then computed from a Lagrange function as:

\begin{equation}
E_{\text{CCSD(F12)-SP}} = L_{\text{CCSD(F12)}} = \langle \text{HF} | H | \text{CC} \rangle + \sum_{\mu_2'} c_{\mu_2'} \Omega_{\mu_2'} \quad (11.17)
\end{equation}

This is the recommended approach which is used by default if not any other approach has been chosen with the examp option in $\texttt{rir12}$ (see Sec. 9.5 for further details on the options for F12 calculations; note that the examp noinv option should not be combined with CCSD calculations). CCSD(F12)-SP calculations are computationally somewhat less expensive than CCSD(F12) calculations which solve Eq. (11.16), while both approaches are approximately similar accurate for energy differences.

The SP approach becomes in particular very efficient if combined with the neglect of certain higher-order explicitly-correlated contributions which have a negligible effect on the energies but increase the costs during the CC iterations. The most accurate and recommended variant is the CCSD(F12*) approximation \cite{18}, which gives essentially identical energies as CCSD(F12). Also available are the CCSD[F12] (Ref. \cite{18}), CCSD-F12a (Ref. \cite{146}) and CCSD-F12b (Ref. \cite{147}) approximations as well as the perturbative corrections CCSD(2)$^{\text{PTT}}$ and CCSD(2*)$^{\text{F12}}$ (see Refs. \cite{18,148,149}). Note that these approximations should only be used with ansatz 2 and the SP approach (i.e. fixed geminal amplitudes).

For MP3 the approximations (F12*), and (F12) to a full F12 implementation become identical: they include all contributions linear in the coefficients $c_{kl}^{ij}$. The explicitly-correlated MP4 method MP4(F12*) is defined as fourth-order approximation to CCSD(F12*)(T). Note that MP4(F12*) has to be used with the SP or fixed amplitude approach for the geminal coefficients $c_{kl}^{ij}$. MP3(F12*) and MP4(F12*) are currently only available for closed-shell or unrestricted Hartree-Fock reference wavefunctions.

The CPU time for a CCSD(F12) calculation is approximately the sum of the CPU time for an MP2-F12 calculation with the same basis sets plus that of a conventional CCSD calculation multiplied by $(1 + N_{CABS}/N)$, where $N$ is the number of basis and $N_{CABS}$ the number
of complementary auxiliary basis (CABS) functions (typically \( N_{CABS} \approx 2 - 3N \)). If the

geminal coefficients are determined by solving Eq. (11.16) instead of using fixed amplitudes, the costs per CCSD(F12) iteration increase to \( \approx (1 + 2N_{CABS}/N) \) the costs for conventional CCSD iteration. Irrespective how the geminal coefficients are determined, the disc space for CCSD(F12) calculations are approximated a factor of \( \approx (1 + 2N_{CABS}/N) \) larger than the disc space required for a conventional CCSD calculation. Note that this increase in the computational costs is by far outweighted by the enhanced basis set convergence.

In combination with the CCSD(F12*) approximation (and also CCSD[12], CCSD-F12a, CCSD-F12b, CCSD(2)_{F12} and CCSD(2*)_{F12}) the CPU time for the SP approach is only about 20% or less longer than for a conventional CCSD calculation within the same basis set.

**CC calculations with restricted open-shell (ROHF) references:** The MP2 and all CC calculations for ROHF reference wavefunctions are done by first transforming to a semi-canonical orbital basis which are defined by the eigenvectors of the occupied/occupied and virtual/virtual blocks of the Fock matrices of alpha and beta spin. No spin restrictions are applied in the cluster equations. This approach is sometimes also denoted as ROHF-UCCSD.

Note that if a frozen-core approximation is used, the semicanonical orbitals depend on whether the block-diagonalization of the Fock matrices is done in space of all orbitals or only in the space of the correlated valence orbitals. The two approaches lead thus to slightly different energies, but none of two is more valid or more accurate than the other. The ccdf12, pnoccsd and ricc2 programs uses the former scheme with the block-diagonalization done in the space of all molecular orbitals. The same scheme is used e.g. in the CFOUR program suite, but other codes as e.g. the implementation in MOLPRO use a block-diagonalization restricted to the active valence space.

**Perturbative triples corrections:** To achieve ground state energies a high accuracy which systematically surpasses the accuracy MP2 and DFT calculations for reaction and binding energies, the CCSD model should be combined with a perturbative correction for connected triples. The recommended approach for the correction is the CCSD(T) model

\[
E_{CCSD(T)} = E_{CCSD} + E_{DT}^{(4)} + E_{ST}^{(5)}
\]

(11.18)

which includes the following two terms:

\[
E_{DT}^{(4)} = \sum_{\mu_2} t_{\mu_2}^{CCSD} \langle \mu_2 | [H, T_3^{(2)}] | \text{HF} \rangle
\]

(11.19)

\[
E_{ST}^{(5)} = \sum_{\mu_1} t_{\mu_1}^{CCSD} \langle \mu_1 | [H, T_3^{(2)}] | \text{HF} \rangle
\]

(11.20)

where the approximate triples amplitudes evaluated as:

\[
t_{\text{abjc}}^{(2)} = -\frac{\langle \delta jk \rangle_{\text{abc}} | \hat{H}, T_2 | \text{HF} \rangle}{\epsilon_a - \epsilon_i + \epsilon_b - \epsilon_j + \epsilon_c - \epsilon_k}
\]

(11.21)
11.1. COMPUTATIONAL DEMANDS

In the literature one also finds sometimes the approximate triples model CCSD[T] (also denoted as CCSD+T(CCSD)), which is obtained by adding only $E_{DT}^{(4)}$ to the CCSD energy. Usually CCSD(T) is slightly more accurate than CCSD[T], although for closed-shell or spin-unrestricted open-shell reference wavefunctions the energies of both models, CCSD(T) and CCSD[T] model, are correct through 4th order perturbation theory. For a ROHF reference, however, $E_{ST}^{(5)}$ contributes already in 4th order and only the CCSD(T) model is correct through 4th order perturbation theory.

Integral-direct implementation and resolution-of-the-identity approximation: The computationally most demanding (in terms floating point operations) steps of a CCSD calculation are related to two kinds of terms. One of the most costly steps is the contraction

$$\Omega_{aibj}^B = \sum_{cd} t_{cidj}(ac|bd) \quad (11.22)$$

where $a$, $b$, $c$, and $d$ are virtual orbitals. For small molecules with large basis sets or basis sets with diffuse functions, where integral screening is not effective, it is time-determining step and can most efficiently be evaluated with a minimal operation count of $\frac{1}{2}O^2V^2$ (where $O$ and $V$ are number of, respectively occupied and virtual orbitals), if the 4-index integrals $(ac|bd)$ in the MO are precalculated and stored on file before the iterative solution of the coupled-cluster equation, 11.4 and 11.5. For larger systems, however, the storage and I/O of the integrals $(ac|bd)$ leads to bottlenecks. An alternatively, this contribution can be evaluated in an integral-direct was as

$$t_{\kappa\lambda ij} = \sum_{cd} t_{ci,dj}C_{\kappa C_{\mu}C_{\nu}C_{\lambda}}$$
$$\Omega_{\mu\nu ij}^B = \sum_{\kappa\lambda} t_{\kappa\lambda ij}(\mu\kappa|\nu\lambda), \quad \Omega_{aibj}^B = \sum_{\mu\nu} \Omega_{\mu\nu ij}^B C_{\mu a} C_{\nu b} \quad (11.23)$$

which, depending on the implementation and system, has formally a 2–3 times larger operation count, but allows to avoid the storage and I/O bottlenecks by processing the 4-index integrals on-the-fly without storing them. Furthermore, integral screening techniques can be applied to reduce the operation count for large systems to asymptotic scaling with $O(N^4)$. In TURBOMOLE only the latter algorithm is presently implemented. (For small systems other codes will therefore be faster.)

The other class of expensive contributions are so-called ring terms (in some publications denoted as C and D terms) which involve contractions of the doubles amplitudes $t_{aibj}$ with several 4-index MO integrals with two occupied and two virtual indeces, partially evaluated with $T_1$-dependent MO coefficients. For these terms the implementation in TURBOMOLE employs the resolution-of-the-identity (or density-fitting) approximation (with the cbas auxiliary basis set) to reduce the overhead from integral transformation steps. Due this approximation CCSD energies obtained with TURBOMOLE will deviate from those obtained with other coupled-cluster programs by a small RI error. This error is usually in the same order or smaller the RI error for a RI-MP2 calculation for the same system and basis sets.

The RI approximation is also used to evaluate the 4-index integrals in the MO basis needed for the perturbative triples corrections.
Disc space requirements: In difference to CC2 and MP2, the CCSD model does no longer allow to avoid the storage of double excitation amplitudes (\( t_{aibj} \)) and intermediates of with a similar size. Thus, also the disc space requirements for the CCSD calculation are larger than for RI-MP2 and RI-CC2 calculation for the same system. For a (closed-shell) CCSD ground state energy calculations the amount of disc space needed can be estimated roughly as

\[
N_{\text{disk}} \approx \left( 4N^3 + (4 + m_{DIIS})O^2N^2 \right) / (128 \times 1024) \text{ MBytes} , \tag{11.24}
\]

where \( N \) is the number of basis functions, \( O \) the number of occupied orbitals and \( m_{DIIS} \) the number of vectors used in the DIIS procedure (by default 10, see Sec. 21.2.17 for details).

For (closed-shell) CCSD(T) calculations the required disc space is with

\[
N_{\text{disk}} \approx \left( 5N^3 + 5O^2N^2 + ON^3 \right) / (128 \times 1024) \text{ MBytes} , \tag{11.25}
\]

somewhat larger.

For calculations with an open-shell (UHF or ROHF) reference wavefunctions the above estimates should be multiplied by factor of 4.

Memory requirements: The CCSD and CCSD(T) implementation in Turbomole uses multi-pass algorithms to avoid strictly the need to store any arrays with a size of \( N^3 \) or \( O^2N^2 \) or larger as complete array in main memory. Therefore, the minimum memory requirements are relatively low—although is difficult to give accurate estimate for them.

On should, however, be aware that, if the amount of memory provided to the program in the data group \$\text{maxcor}\$ becomes too small compared to \( O^2N^2 / (128 \times 1024) \) MBytes, loops will be broken in many small batches at the cost of increased I/O operations and a decrease in performance. As mentioned above, it is recommended to set \$\text{maxcor}\$ to 66–77% of the physical core memory available for the calculation.

Important options: The options to define the orbital and the auxiliary basis sets, the maximum amount allocatable core memory (\$\text{maxcor}\$), and the frozen-core approximation (\$\text{maxcor}\$) have been mentioned above and described in the previous chapters on MP2 and CC2 calculations. Apart from this, CCSD and CCSD(T) calculations require very little additional input.

Relevant are in particular some options in the \$\text{ricc2}\$ data group:

\[
\begin{align*}
\$\text{ricc2} \\
\text{ccsd} \\
\text{ccsd(t)} \\
\text{conv}=7 \\
\text{oconv}=6 \\
m\text{dii}=10 \\
m\text{axiter}=25
\end{align*}
\]
The options \texttt{ccsd} and \texttt{ccsd(t)} request, respectively, CCSD and CCSD(T) calculations. Since CCSD(T) requires the cluster amplitudes from a converged CCSD calculation, the option \texttt{ccsd(t)} implies the \texttt{ccsd} option.

The number given for \texttt{mxdiis} defines the maximum number of vectors included in the DIIS procedure for the solution of the cluster equations. As mentioned above, it has some impact on the amount of disc space used by a CCSD calculation. Unless disc space becomes a bottleneck, it is not recommended to change the default value.

With \texttt{maxiter} one defines the maximum number of iterations for the solution of the cluster equations. If convergence is not reached within this limit, the calculation is stopped. Usually 25 iterations should be sufficient for convergence. Only in difficult cases with strong correlation effects more iterations are needed. It is recommended to increase this limit only if the reason for the strong correlation effects is known. (Since one reason could also be an input error as e.g. unreasonable geometries or orbital occupations as a wrong basis set assignment.)

The two parameters \texttt{conv} and \texttt{oconv} define the convergence thresholds for the iterative solution of the cluster equations. Convergence is assumed if the change in the energy (with respect to the previous iteration) has is smaller than $10^{-\text{conv}}$ and the euclidian norm of the residual (the so-called vector function) is smaller than $10^{-\text{oconv}}$. If \texttt{conv} is not given in the data group $\$ricc2$ the threshold for changes in the energy is set to value given in $\$denconv$ (by default $10^{-7}$). If \texttt{oconv} is not given in the data group $\$ricc2$ the threshold for the residual norm is by default set to 10 times the threshold changes in the energy. With the default settings for these thresholds, the energy will thus be converged until changes drop below $10^{-7}$ Hartree, which typically ensures an accuracy of about 1 $\mu$H. These setting are thus rather tight and conservative even for the calculation of highly accurate reaction energies. If for your application larger uncertainites for the energy are tolerable, it is recommended to use less tight thresholds, e.g. \texttt{conv=6} or \texttt{conv=5} for an accuracy of, respectively, at least 0.01 mH (0.03 kJ/mol) or 0.1 mH (0.3 kJ/mol). The settings for \texttt{conv} (and \texttt{oconv}) have not only an impact on the number of iterations for the solution of the cluster equations, but as they determine the thresholds for integral screening also to some extend on the costs for the individual iterations.

**CCSD(T) energy with a second-order correction from the interference-corrected MP2-F12:** The error introduced from a CCSD(T) calculation with a finite basis set can be corrected from second-order corrections of the the interference-corrected MP2-F12 (INT-MP2-F12) (Ref. \cite{150}). The approximate CCSD(T)-INT-F12 at the basis set limit is given from

\[
E_{\text{CCSD(T)/CBS}} \approx E_{\text{CCSD(T)}} + \sum_{i<j} F_{ij}^{\text{MP2-F12}} \left[ \epsilon_{ij}^{\text{MP2}} - \epsilon_{ij}^{\text{MP2}} \right]. \quad (11.26)
\]

From define, in the submenu $\$ricc2$ select the \texttt{ccsd(t)} method and add the keyword \texttt{intcorr}

$\$ricc2$
ccsd(t)
intcorr

Then, switch on the f12 method (approximation A or B, inv or fixed). The corrected CCSD(T)-INT-F12 energy will be printed in the end of the calculation. It is highly recommended to start the CCSD(T)-INT-F12 calculation from a converged SCF calculation with symmetry, which is transformed to $C_1$. It is furthermore recommended to use Boys localized orbitals in the \$rir12 submenu. A table with the corrected second-order pair-electron energies and the corresponding interference factors can also be printed in the output by using the keyword intcorr all instead of intcorr.

**Excitation energies with CCSD:** Since release V6.5 electronic excitation energies can also be computed at the (conventional) CCSD level. For this the data group \$excitations has be added (the same keyword as for CC2 apply). The implementation is currently restricted to vertical excitation energies (no transition moments or properties available) and in the closed-shell case to singlett excited states.

Note that for single-excitation dominated transitions CCSD is as CC2 correct through second-order in $H$ and does not necessarily more accurate than CC2. It is, however, for double excitations still correct through first-order, while CC2 describes double excitations only in a zero-order approximation. Therefore, CCSD results are more robust with respect to double excitation contributions to transitions and are thus useful to check if CC2 is suitable for a certain problem.
Chapter 12

Random Phase Approximation Calculations: Energy and First-Order Properties

Ground state energies and analytic first-order properties (e.g., 'gradients' for structure optimizations) can be computed within the random phase approximation (RPA) using the \texttt{rirpa} module. Theory and development of the \texttt{rirpa} module is published in references [151, 152] for the energy and reference [153] for the first-order properties. In case of two-component relativistic RPA energy calculations see reference [154]. For energy and gradients, the resolution-of-the-identity (RI) approximation is used to approximate the two-electron repulsion integrals in the correlation treatment and is combined with an imaginary frequency integration. The RI approximation is also employed by default for the computation of the Coulomb integrals for the HF energy. For the energy, it is optional to use RI for the Fock exchange integrals ('RI-K'), while RI-K for the gradients is not available yet. Open shell systems and the frozen core approximation may be used in RPA energy calculations but are not presently available in gradient calculations. Two-component RPA energy calculations are only possible for Kramers-restricted closed-shell systems. ECPs are presently not compatible with RIRPA gradients. Neither RPA energy nor gradients support symmetry at the moment. The gradients may be used together with the scripts \texttt{jobex} (for structure optimizations) and \texttt{NumForce} (for numerical harmonic vibrational frequencies).

12.1 Ground State Energy Theory

The RPA energy

\[ E^{\text{RPA}} = E^{\text{HF}} + E^{\text{C RPA}} \]  

(12.1)
consists of the Hartree-Fock exact exchange energy $E_{\text{HF}}$ and a correlation energy piece $E_{\text{RPA}}^{C}$. \texttt{rirpa} computes Eq. (12.1) non-selfconsistently from a given set of converged input orbitals. The correlation energy

$$E_{\text{RPA}}^{C} = \frac{1}{2} \sum_n (\Omega_{n}^{\text{RPA}} - \Omega_{n}^{\text{TDARPA}})$$

(12.2)

is expressed in terms of RPA excitation energies at full coupling $\Omega_{n}^{\text{RPA}}$ and within the Tamm-Dancoff approximation $\Omega_{n}^{\text{TDARPA}}$. The further discussion is restricted to the one-component (nonrelativistic) treatment, for the sake of convenience. For the derivation of the two-component RPA theory see ref. [154]. The excitation energies are obtained from time-dependent DFT response theory and are eigenvalues of the symplectic eigenvalue problem [155, 156]

$$(\Lambda - \Omega_{0n} \Delta)|X_{0n}, Y_{0n}\rangle = 0.$$  

(12.3)

The super-vectors $X_{0n}$ and $Y_{0n}$ are defined on the product space $L_{\text{occ}} \times L_{\text{virt}}$ and $L_{\text{occ}} \times L_{\text{virt}}$, respectively, where $L_{\text{occ}}$ and $L_{\text{virt}}$ denote the one-particle Hilbert spaces spanned by occupied and virtual static KS molecular orbitals (MOs). The super-operator

$$\Lambda = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$$

(12.4)

contains the so-called orbital rotation Hessians,

$$(A + B)_{ijab} = (\epsilon_a - \epsilon_i) \delta_{ij} \delta_{ab} + 2 (i |a j b),$$

(12.5)

$$(A - B)_{ijab} = (\epsilon_a - \epsilon_i) \delta_{ij} \delta_{ab}.$$  

(12.6)

$\epsilon_i$ and $\epsilon_a$ denote the energy eigenvalues of canonical occupied and virtual KS MOs. \texttt{rirpa} computes so-called direct RPA energies only, i.e. no exchange terms are included in Eqs. (12.5) and (12.6).

In RIRPA the two-electron integrals in Eqs (12.5) are approximated by the resolution-of-the-identity approximation. In conjunction with a frequency integration this leads to an efficient scheme for the calculation of RPA correlation energies [151]

$$E_{\text{RIRPA}}^{C} = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} F_{\text{C}}^{C}(\omega),$$

(12.7)

where the integrand contains $N_{\text{aux}} \times N_{\text{aux}}$ quantities only,

$$F_{\text{C}}^{C}(\omega) = \frac{1}{2} \text{tr} \left( \ln (I_{\text{aux}} + Q(\omega)) - Q(\omega) \right).$$

(12.8)

$N_{\text{aux}}$ is the number of auxiliary basis functions. The integral is approximated using Clenshaw-Curtiss quadrature.

**Prerequisites**

Calculations with \texttt{rirpa} require
12.2 GRADIENTS THEORY

- a converged SCF calculation
- **rirpa**-options may be included by adding them in the lines below the keyword **$rirpa** in the control file. Possible options are:
  - `npoints` (integer) - Number of frequency integration points (default is 60).
  - `nohxx` - HF energy is skipped, \((HXX = \text{Hartree} + e\text{Xact (Fock) eXchange})\).
  - `rpaprof` - Generates profiling output.
- the maximum core memory the program is allowed to allocate should be defined in the data group **$maxcor** (in MB); the recommended value is ca. 3/4 of the available (physical) core memory at most.
- orbitals to be excluded from the correlation treatment have to be specified in data group **$freeze**
- an auxiliary basis defined in the data group **$cbas**
- an auxiliary basis defined in the data group **$jbas** for the computation of the Coulomb integrals for the Hartree-Fock energy
- (optional) an auxiliary basis defined in the data group **$jkbas** for the computation of the exchange integrals for the Hartree-Fock energy. **$rik** should be added to the control file for RI-JK to be effective.

Effective core potentials (ECPs) are not presently compatible with the HF energy at the KS reference as computed in **rirpa**. The **nohxx** option *must* therefore be included for systems where ECPs were used to obtain the reference KS orbitals in order to skip the HF energy calculation and compute solely the correlation energy.

To perform a two-component relativistic RIRPA calculation \([154]\) on (Kramers-restricted) closed-shell systems taking into account spin-orbit coupling, the two-component version of **ridft** has to be run before (see Chapter 6.4) using the keywords **$soghf** and **$kramers**. The implementation is currently only available in combination with the **nohxx** option.

### 12.2 Gradients Theory

All details on the theory and results are published in \([153]\). The RI-RPA energy is a function of the MO coefficients \(C\) and the Lagrange multipliers \(\epsilon\) and depends parametrically (i) on the interacting Hamiltonian \(\hat{H}\), (ii) on the AO basis functions and the auxiliary basis functions. All parameters may be gathered in a supervector \(X\) and thus

\[
E^{RIRPA} = E^{RIRPA}(C, \epsilon|X) .
\]  

(12.9)

\(C\) and \(\epsilon\) in turn depend parametrically on \(X\), the exchange-correlation matrix \(V^{XC}\), and the overlap matrix \(S\) through the KS equations and the orbital orthonormality constraint. First-order properties may be defined in a rigorous and general fashion as total derivatives
of the energy with respect to a “perturbation” parameter $\xi$. However, the RI-RPA energy is not directly differentiated in our method. Instead, we define the RI-RPA energy Lagrangian

$$L_{\text{RIRPA}}(C, \mathcal{E}, D^\Delta, W|X, V^{XC}, S) = E_{\text{RIRPA}}(C, \mathcal{E}|X) + \sum_\sigma \left( \langle D^\Delta_\sigma (C_\sigma^T F_\sigma C_\sigma - \mathcal{E}_\sigma) \rangle - \langle W_\sigma (C_\sigma^T S C_\sigma - 1) \rangle \right).$$

$C, \mathcal{E}, D^\Delta,$ and $W$ are independent variables. $L_{\text{RIRPA}}$ is required to be stationary with respect to $C, \mathcal{E}, D^\Delta,$ and $W.$ $D^\Delta$ and $W$ act as Lagrange multipliers enforcing that $C$ and $\mathcal{E}$ satisfy the KS equations and the orbital orthonormality constraint,

$$\left( \frac{\partial L_{\text{RIRPA}}}{\partial D^\Delta_\sigma} \right)_\text{stat} = C_\sigma^T F_\sigma C_\sigma - \mathcal{E}_\sigma = 0,$$

$$\left( \frac{\partial L_{\text{RIRPA}}}{\partial W_\sigma} \right)_\text{stat} = C_\sigma^T S C_\sigma - 1 = 0.$$  

$D^\Delta$ and $W$ are determined by the remaining stationarity conditions,

$$\left( \frac{\partial L_{\text{RIRPA}}}{\partial \mathcal{E}} \right)_\text{stat} = 0,$$

$$\left( \frac{\partial L_{\text{RIRPA}}}{\partial C} \right)_\text{stat} = 0.$$  

It turns out from eqs (12.13) and (12.14) that the determination of $D^\Delta$ and $W$ requires the solution of a single Coupled-Perturbed KS equation. Complete expressions for $D^\Delta$ and $W$ are given in [153]. At the stationary point “stat = ($C = C, \mathcal{E} = \epsilon, D^\Delta = D^\Delta, W = W$),” first-order RI-RPA properties are thus efficiently obtained from

$$\frac{dE_{\text{RIRPA}}(C, \mathcal{E}|X)}{d\xi} = \left\langle \left( \frac{\partial L_{\text{RIRPA}}}{\partial X} \right)_\text{stat} \frac{dX}{d\xi} \right\rangle + \left\langle \left( \frac{\partial L_{\text{RIRPA}}}{\partial V^{XC}} \right)_\text{stat} \left( \frac{\partial V^{XC}}{\partial \xi} \right)_\text{stat} \right\rangle + \left\langle \left( \frac{\partial L_{\text{RIRPA}}}{\partial S} \right)_\text{stat} \frac{dS}{d\xi} \right\rangle.$$  

Finally, the RPA energy gradients may be explicitly expanded as follows:

$$\frac{dE_{\text{RIRPA}}(C, \mathcal{E}|X)}{d\xi} = \left\langle D_{\text{RIRPA}} \frac{d\mathbf{h}}{d\xi} \right\rangle + \left\langle \Gamma^{(4)} \frac{d\Pi^{(4)}}{d\xi} \right\rangle + \left\langle D^\Delta \left( \frac{\partial V^{XC}}{\partial \xi} | D \right) \right\rangle + \left\langle \Gamma^{(3)} \frac{d\Pi^{(3)}}{d\xi} \right\rangle + \left\langle \Gamma^{(2)} \frac{d\Pi^{(2)}}{d\xi} \right\rangle \right\rangle - \left\langle W \frac{dS}{d\xi} \right\rangle.$$  

where $D_{\text{RIRPA}}$ is the KS ground state one-particle density matrix $D$ plus the RI-RPA difference density matrix $D^\Delta$ which corrects for correlation and orbital relaxation effects. $\mathbf{h}$ is the one-electron Hamiltonian; $\Pi^{(2/3/4)}$ are 2-, 3-, and 4-centre electron repulsion integrals and the $\Gamma^{(2/3/4)}$ are the corresponding 2-, 3-, and 4-index relaxed 2-particle density matrices; $W$ may be interpreted as the energy-weighted total spin one-particle density matrix.

This result illustrates the key advantage of the Lagrangian method: Total RI-RPA energy derivatives featuring a complicated implicit dependence on the parameter $X$ through the variables $C$ and $\mathcal{E}$ are replaced by partial derivatives of the RI-RPA Lagrangian, whose computation is straightforward once the stationary point of the Lagrangian has been fully determined.
12.3. FURTHER RECOMMENDATIONS

Gradients Prerequisites

Geometry optimizations and first order molecular property calculations can be executed by adding the keyword `rpagrad` to the `@rirpa` section in the control file. RPA gradients also require

- an auxiliary basis defined in the data group `@jbas` for the computation of the Coulomb integrals for the Hartree-Fock energy
- an auxiliary basis defined in the data group `@cbas` for the ERI’s in the correlation treatment.
- zero frozen core orbitals; RIRPA gradients are not compatible with the frozen core approximation at this time.

The following gradient-specific options may be further added to the `@rirpa` section in the control file

- `drimp2` - computes gradients in the DRIMP2 limit.
- `niapblocks ⟨integer⟩` - Manual setting of the integral block size in subroutine rirhs.f; for developers.

In order to run a geometry optimization, `jobex` must be invoked with the level set to `rirpa`, and the `-ri` option (E.g. `jobex -ri -level rirpa`).

In order to run a numerical frequency calculation, `NumForce` must be invoked with the level set to `rirpa`, e.g., `NumForce -d 0.02 -central -ri -level rirpa`.

12.3 Further Recommendations

- The direct RPA correlation energy is defined in a Kohn-Sham context without inclusion of exchange integrals and therefore the use of self-consistent KS orbitals obtained from (semi-)local functionals is recommended. HF-orbitals or KS-orbitals obtained form hybrid functionals lead to inferior results.
- Experience has demonstrated that the difference in RPA correlation energies obtained from different (semi-)local functionals is very small (much smaller than the inherent error of the method).
- Like MP2, RIRPA results are known to converge very slowly with increasing basis set size, in particular slowly with increasing $l$-quantum number of the basis set. For reliable results the use of QZVP basis sets (or higher) is recommended. For non-covalently bound systems larger basis sets (especially, with more diffuse functions) are needed.
- It is recommended to exclude all non-valence orbitals from RIRPA calculations, as neither the TURBOMOLE standard basis sets SVP, TZVPP, and QZVPP nor the cc-pVXZ
basis set families (with X=D,T,Q,5,6) are designed for correlation treatment of inner shells (for this purpose polarisation functions for the inner shells are needed). The default selection for frozen core orbitals in Define (orbitals below -3 a.u. are frozen) provides a reasonable guess. If core orbitals are included in the correlation treatment, it is recommended to use basis sets with additional tight correlation functions as e.g. the cc-pwCVXZ and cc-pCVXZ basis set families.

- We recommend the use of auxiliary basis sets optimized for the corresponding (MO) basis sets. The auxiliary basis sets optimized for RI-MP2 and RI-CC2 are suitable for rirpa \cite{151} correlation energy calculations.

- For systems where ECPs are required as well as within the two-component relativistic implementation, RIRPA total energies (HF@KS + correlation) must be computed in two steps. RIRPA correlation energies can be obtained using the nohxx option, and the HF energy can then be computed separately, e.g., in ridft if the RI-J approximation is used for the Coulomb integrals. To compute the HF@KS energy, compute the KS orbitals first; then disable $dft$ and set $scfiterlimit 1$ in the control file to perform a single SCF iteration. Finally add the total HF@KS energy from ridft to the correlation energy from the nohxx-rirpa calculation to obtain the total RIRPA energy. Note: the molecular orbitals are altered by ridft after a single-iteration, so the HF@KS energy must be computed after the RIRPA correlation energy.

- Tight SCF ($scfconv 7$) and one-electron density matrix ($denconv 1d-7$) convergence criteria, large basis sets (QZVP), and large frequency grids which ensure a sensitivity measure of no more than $1d-4$ should be used in combination with rpagrad for accurate results.

### 12.4 Comments on the Output

- The most important output for rirpa are the Hartree-Fock (HXX) energy and the RIRPA correlation energy which are written to the standard output.

- The optimal scaling parameter for the quadrature grid is printed together with a sensitivity parameter. The sensitivity parameter provides a numerical estimate for the error in the numerical integration used to evaluate $E^{C_{RIRPA}}$. Experience demonstrates that the sensitivity parameter correlates well with the condition number of the matrix $Q$. Small gap systems have large condition numbers and therefore require large grids. An estimate of the number of eigenvalues smaller than $0.05 H$ is given and if necessary, a warning to increase the grid size is printed to standard output.

- The molecular dipole and quadrupole moments are written to the standard output whenever a gradient calculation is carried out.

- Rotational constants are provided in the rirpa output below the coordinates section.

- Enabling the option rpaprof will output additional timings information.
Chapter 13

Many body perturbation theory in
the \textit{GW} approximation

13.1 Theoretical background.

A method to systematically improve upon DFT-estimates of single particle excitation spectra, i.e., ionization potentials and electron affinities is the \textit{GW-method}. Its central object is the single particle Green’s function \(G\); its poles describe single particle excitation energies and lifetimes. In particular, the poles up to the Fermi-level correspond to the primary vertical ionization energies. The \textit{GW}-approach is based on an exact representation of \(G\) in terms of a power series of the screened Coulomb interaction \(W\), which is called the \textit{Hedin equations}. The \textit{GW}-equations are obtained as an approximation to the Hedin-equations, in which the screened Coulomb interaction \(W\) is calculated neglecting so called \textit{vertex corrections}. In this approximation the self–energy \(\Sigma\), which connects the fully interacting Green’s function \(G\) to a reference non-interacting Green’s function \(G_0\), is given by \(\Sigma = GW\).

This approach can be used to perturbatively calculate corrections to the Kohn-Sham spectrum. To this end, the Green’s function is expressed in a spectral representation as a sum of quasi particle states.

\[
G(r, r'; z) = \sum_n \frac{\Psi_{r,n}(r, z)\Psi_{l,n}^\dagger(r', z)}{z - \varepsilon_n(z) + i\eta \text{sgn}(\varepsilon_n - \mu)},
\]

Under the approximation that the KS-states are already a good approximation to these quasi–particle states \(\Psi_{l,n}\) the leading order correction can be calculated by solving the zeroth order quasi–particle equation:

\[
\varepsilon_n = \varepsilon_n + \langle n | \Sigma | G_{KS}(\varepsilon_n) - V_{xc} | n \rangle
\]
An approximation to the solution of this equation can be obtained by linearizing it:

$$\varepsilon_n = \varepsilon_n + Z_n \langle n| \Sigma(\varepsilon_n) - V_{\text{xc}} |n\rangle$$  \hspace{1cm} (13.3)

Here, $Z_n$ is given by:

$$Z_n = \left[ 1 - \langle n | \frac{\partial \Sigma(E)}{\partial E} \bigg|_{E=\varepsilon_n} |n\rangle \right]^{-1}$$  \hspace{1cm} (13.4)

reducing the computational effort to a single iteration.

The self-energy $\Sigma$ appearing in Eqn. (13.2) is calculated in the $GW$ approximation from the KS Green’s function and screening. This is the so-called $G_0W_0$ approximation. The self-energy splits in an energy independent exchange part $\Sigma^x$ and a correlation part $\Sigma^c(E)$ that does depend on energy. Their matrix elements are given by:

$$\langle n| \Sigma^x |n'\rangle = = - \sum_i (ni|in')$$ \hspace{1cm} (13.5)

and

$$\langle n| \Sigma^c(\varepsilon_n) |n\rangle = \sum_m \sum_n \frac{|(nn|m\rangle|^2}{\varepsilon_n - \varepsilon_n - Z_m \text{sgn}(\varepsilon_n - \mu)}.$$ \hspace{1cm} (13.6)

Where $Z_m = \Omega_m - i\eta$ are the excitation energies shifted infinitesimally into the complex plane. The $\rho_m$ are the corresponding excitation densities. More details, tests and benchmark calculations are can be found in Ref. 157.

### 13.2 $GW$ features.

$G_0W_0$ is implemented in TURBOMOLE in the `escf` module supporting the following features.

- LDA, GGA and Hybrid functionals can be used for the underlying DFT calculation.
- RI approximation.
- In $G_0W_0$, the linearized, Eqn. (13.3), and solved, Eqn. (13.2), quasi-particle equation.
- Both RPA and TDDFT response functions can be used to screen the coulomb interaction in constructing $W$.
- Closed shell systems with rpas excitations and open shell systems with urpa excitations.

### 13.3 General recipe for $G_0W_0$ calculations

The general recipe for a $G_0W_0$ calculation is as follows:

1. define session
2. Provide additional $GW$ control flags

3. $dscf$ or $ridft$ calculation

4. $escf$ calculation

Ad 1) Symmetry has not yet been implemented for $GW$, sensible calculations can only be done in $C_1$. The def2-TZVPP basis seems to be the most useful, it comes for all tested systems within 0.1 eV of the def2-QZVP result with about half the number of basis functions. In the define session the calculation of the response function needs to be defined. In the final define menu select the ex menu and select the calculation of RPA singlet excitations, or $urpa$ in case of open shell. Select "all all" to get all excitations. For systems and basis set having less than 4000 rpa excitations just set for all excitations. For large systems start to run $G_0W_0$ with ~4000 rpa excitations. In subsequent runs add more excitations until a converged result is reached. $escf$ will keep the converged roots, so not much time is lost using this restart approach.

Ad 2) If $gw$ is set in the control file the quasi particle energies will be evaluated according to equation 13.3. Additional options are described in the keyword section, section 21.2.13.

Ad 4) In the $escf$ run the response function is calculated which is needed to determine the screened coulomb interaction. At the end of this run the actual $GW$ calculation is performed.

Possible source of errors: When $dscf$ or $ridft$ is repeated after $escf$ the sing\_a file may not be correct anymore, this may happen when degenerate levels are present. $escf$ will however not recognize this and continue using the previously converged data in sing\_a leading to nonsense values for $\Sigma_c$. Before running $escf$ the old sing\_a file has to be removed.
Chapter 14

Calculation of Vibrational Frequencies and Vibrational Spectra

Calculation of second derivatives of total energies leads to the molecular Hessian, which enables prediction of vibrational frequencies and infrared spectra (within the harmonic approximation) as well as the application of improved algorithms for geometry optimization and transition state search.

The aoforce module calculates analytically harmonic vibrational frequencies within the HF- or (RI)DFT-methods for closed-shell- and spin-unrestricted open-shell-systems. Broken occupation numbers would lead to results without any physical meaning. Note, that RI is only used partially, which means that the resulting Hessian is only a (very good) approximation to exact second derivatives of the RIDFT-energy expression. Apart from a standard force constant calculation which predicts all (symmetry allowed and forbidden) vibrational transitions, it is also possible to specify certain irreps for which the calculation has to be done exclusively or to select only a small number of lowest eigenvalues (and eigenvectors) that are generated at reduced computational cost.

Furthermore, the Numforce script allows the calculation of second derivatives for all methods for which a program for analytic gradients is available in TURBOMOLE, i.e. the main use of this script is the prediction of vibrational spectra at the MP2 level and for excited states using RI-CC2 or TDDFT.

If force constant calculations result in imaginary frequencies, molecular distortions along these normal modes should lower the energy. To distort the molecule, use the interactive module vibration, output of the new coordinates is done to the general input file on $newcoord.

Vibrational frequencies also enable calculation of the molecular partition function and thus prediction of thermodynamic functions at temperatures other than 0 K and finite pressure.
(within the assumption of an ideal gas and no coupling between degrees of freedom). These functions can be obtained with the interactive module Freeh, results are printed to standard I/O.

Prerequisites

1. Both aoforce and even more Numforce require well converged SCF-/DFT-calculations (e.g. $scfconv 8$ and jobex [-ri] -gcart 4).

2. The maximum core memory the program aoforce is allowed to allocate should be defined in the data group $\maxcor$; the recommended value is about 50% of the available (physical) core memory (in case of RI-calculations subtract the memory specified in $\ricore$).

3. To start aoforce in the lowest eigenvalue search mode, use the keyword $\les$. For its use as well as other keywords dealing with the calculation of only some irreps, see the Referenceguide part of this manual.

4. Numforce additionally requires the file gradient and will not work, if the calculation is not done at a stationary point of the molecular total energy. For reliable results, always use Numforce with the option -central (i.e. central differences) and be aware of effects due to the step length (option -d real; default value is 0.02 a.u.). It is strongly recommended to use Numforce in DFT calculations only with the option weight derivatives in $\dft$, since this provides more accurate gradients and thus frequencies, see Section 21.2.10.

5. The Numforce script can be run for different levels of theory, which means that the binaries it calls have to be specified additionally. To perform calculations using the RI approximation, call Numforce with the option -ri. MP2 and CC2 calculations are requested via the options -level mp2 and -level cc2, respectively. NumForce works also on the RI-RPA level with -level rirpa (note: the -ri option must be used in this case). To select the correct option(s), use the explanations you get by calling NumForce -h.

For a review of theory and implementation see refs. [158,159].

Limitations

The aoforce code has presently a number of limitations one should be aware of:

- It can only handle basis sets up to at most g functions.
- Point groups with reducible E-representations (such as $C_n$ and $C_{nh}$ with $n \geq 3$, $S_n$ with $n \geq 5$, or $T$ and $T_d$)
- Frozen internal or cartesian coordinates are not recognized. aoforce will always evaluate the full hessian matrix.
14.1 Analysis of Normal Modes in Terms of Internal Coordinates

A note in advance: The analysis of normal modes can (at nearly no computational cost) always be redone as long as you keep a copy of the file `hessian`.

A general prerequisite for this option is that you have defined a set of non-redundant coordinates for all 3N-6 (3N-5) degrees of freedom of your molecule. To make sure that this is the case, you should switch off redundant coordinates (currently, this is only possible by manually removing the data group `$redundant` and also removing the entry `redundant on` in `$optimize`). Run `define` to generate non-redundant coordinates by using the `iaut` command in the internal coordinate menu (or by creating them manually via `idef`). We recommend to use the `irem` command first to delete all previous definitions of internal coordinates. See Section 4 for further details. If the molecules point group is not $C_1$, `define` will set some of the coordinate to status d (display) or i (ignore). Use the `ic` command to change all coordinates to k. You can also achieve this by editing in the `$intdef` data-group manually.

The analysis in internal coordinates is switched on by adding a line in the data-group `$drvopt` that has the following syntax:

```
analysis [only] intcoord [print print-level]
```

Keywords in square brackets are optional. If `only` is added, the program assumes that the file `hessian` exists and runs only the analysis part of `aoforce`. The program will give the following output (controlled by the print level given in parenthesis):

- diagonal elements of the Hessian in internal coordinates (force constants of bonds, angles, etc.) (print level 0)
- complete force constant matrix in internal coordinates (print level 2)
- normal modes in terms of internal coordinates (print level 1)
- Potential energy contributions $\tilde{V}_{ij}^n$, defined as

$$\tilde{V}_{ij}^n = L_i^n L_j^n F_{ij}/\omega^n$$

where $L_i^n$ are the elements of the normal coordinate belonging to mode $n$ and $F_{ij}$ are the elements of the force constant matrix, both expressed in the internal coordinate basis; $\omega$ is the related eigenvalue. The program will list the diagonal contributions $\tilde{V}_{ii}^n$ (print level 1), the off-diagonal contributions $\tilde{V}_{ij}^n + \tilde{V}_{ji}^n = 2\tilde{V}_{ij}^n$ (print level 2 for up to 10 atoms, else print level 10) and the brutto contributions $\sum_i \tilde{V}_{ij}^n$ (print level 1).

- Based on these quantities, the program will give an assignment of normal modes by listing all internal coordinates with large diagonal or brutto contributions (print level 0).
14.2 Calculation of Raman Spectra

Vibrational Raman scattering cross sections are computed in the approximation of the polarizability theory from derivatives of the frequency-dependent polarizability tensor with respect to normal modes of vibration,

\[
\left( \frac{d\sigma}{d\Omega} \right) = k_\omega \left( c_i \alpha'^2(\omega) + c_a \gamma'^2(\omega) \right).
\]

Here, \( \alpha'^2(\omega) \) and \( \gamma'^2(\omega) \) denote the isotropic part and the anisotropy of the differentiated polarizability tensor, respectively. The coefficients \( c_i \) and \( c_a \) depend on the scattering geometry and the polarization of the incident and scattered radiation. The factor

\[
k_\omega = \frac{\hbar}{4\pi\epsilon_0^2 c^4} \frac{(\omega - \omega_v)^4 g_v}{2\omega_v}
\]

includes the frequency \( \omega_v \) and the degeneracy \( g_v \) of the vibration. \( c \) is speed of light and \( \epsilon_0 \) stands for the dielectric constant of vacuum.

Computation of Raman spectra with TURBOMOLE is a three-step procedure. First, vibrational frequencies and normal modes are calculated by aoforce. Cartesian polarizability derivatives are computed in the second step by egrad, see Section 8.4.7. Finally, the program intense is used to project the polarizability derivatives onto vibrational normal modes and to compute Raman scattering cross sections which are written out along with vibrational frequencies and normal modes. The script Raman can be used to perform all these steps automatically.

14.3 Vibrational frequencies with fixed atoms using NumForce

The NumForce script provides with the option -frznuclei a possibility to do a vibrational analysis with fixed atoms. The atoms for which the cartesian coordinates should frozen have to be marked in $coord with a "f" behind the atom type. The frozen coordinates will be skipped during the numerical evaluation of the force constant matrix; instead all off-diagonal elements of the force constant matrix which refer to one or two frozen coordinates will be set to zero, while the diagonal elements for the frozen coordinates will be set to an arbitrarily chosen large value.

This feature is mainly intended to allow for a vibrational analysis in embedded cluster calculations e.g. for defects in ionic crystals. The vibrational analysis uses a kind of "frozen phonon" approximation which corresponds to setting the masses of the fixed atoms to infinity, i.e. decoupling the fixed atoms mechanically from the "mechanically active" subsystem. The resulting vibrational frequencies will thus only provide good approximations to the true
(harmonic) frequencies for such modes for which the mechanical coupling to the embedding environment is negligible. In particular the frequencies of stretch modes which involve bonds between the ‘mechanically active’ subsystem and atoms with frozen coordinates will be strongly affected by this approximation.

Note:

- The `-frznuclei` option is not compatible with the polyhedral difference algorithm. It can only be used with central differences which should be enforced with the `-central` option.

- If the option `-frznuclei` is switched on, the program assumes that the constraints enforced by fixing coordinates remove the six external degrees of freedom for an overall rotation or translation of the system and therefore the hessian matrix is not projected onto the subspace of internal coordinates. Fixing the coordinates of only one or two atoms might do lead to some artificial small, but non-zero frequencies.

- Zero-point vibrational energies calculated with the `-frznuclei` option are only meaningful for comparison of systems with the same mechanically active atoms and similar embedding, as the contributions from the frozen coordinates are not included.

### 14.4 Interface to hotFCHT

`aoforce` supports the generation of input files for the hotFCHT code (version 2.0 and later) of R. Berger and co-workers, see [hotFCHT](#), which allows for the calculation of Franck-Condon factors. Just include the keyword `@hotfcht` in the control file. The option is also active in analysis mode, that is as long as you still have the data group `$hessian` (in the control file or in a file referenced in the control file) you can always use `aoforce` (in analysis mode) to quickly generate the hotFCHT input. The program will write three files. The first one, `hotfcht_header.inp` contains a collection of the most important keywords of hotFCHT (set to some default values, please adapt to your needs) and list of all atomic masses (either TURBOMOLE’s default masses or the ones given in the `$atoms` data group). The other two, `hotfcht_data_i.inp` and `hotfcht_data_f.inp` contain the vibrational frequencies, normal modes and the names of the irreducible representations of the normal modes. In the former file, these data are associated with the hotFCHT keywords for the initial state, while the latter file contains the same data, but associated with the keywords for the final state.

In order to run a hotFCHT calculation, you need to optimize the structures of two electronic states (usually the electronic ground state and an excited or ionized state) and obtain the harmonic force fields for both, using either `aoforce` or NumForce. In order to generate the hotFCHT input, just concatenate the `hotfcht_header.inp` file (from any of the two calculations) and the `hotfcht_data_i.inp` file from the calculation that refers to the initial state (e.g. the ground state in case of an absorption spectrum) and the `hotfcht_data_f.inp` file from the calculation of the final state (the excited state in case of an absorption spectrum). Carefully edit the keywords in the header of the resultant file and run hotFCHT (please, refer to the hotFCHT documentation for further information).
Chapter 15

First order electron-vibration coupling

15.1 Theoretical background

At the effective single-particle level, the Hamiltonian of the coupled system of electrons and vibrations is given by [160]

\[
\hat{H} = \hat{H}_e + \hat{H}_v + \hat{H}_{ev},
\]

(15.1)

where the first term $\hat{H}_e$ describes the electronic system and the second term $\hat{H}_v$ the vibrational degrees of freedom. The last term in the Hamiltonian

\[
\hat{H}_{ev} = \sum_{\mu\nu} \sum_\alpha \hat{d}_\mu^\dagger \lambda_{\mu\nu}^\alpha \hat{d}_\nu (\hat{b}_\alpha^\dagger + \hat{b}_\alpha)
\]

(15.2)

describes the first order electron-vibration (EV) interaction. The EV coupling constants are given as

\[
\lambda_{\mu\nu}^\alpha = \left( \frac{\hbar}{2\omega_\alpha} \right)^{1/2} \sum_\chi \langle \mu | \frac{d\hat{H}_e}{d\chi} | \nu \rangle A_\chi^\alpha,
\]

(15.3)

where $\chi = (k, u)$ is a shorthand notation that refers both to the displacement of atom $k$ from the equilibrium value of the position $\vec{R}_k$ along the Cartesian component $R_{k,u}$ with $u = x, y, z$ as well as the index pair itself. Furthermore, $A_\chi^\alpha = C_\chi^\alpha / \sqrt{M_k}$ are the mass-normalized normal modes, obtained from the eigenvectors $C_\chi^\alpha$ of the dynamical matrix as calculated from the aoforce module [160].
CHAPTER 15. EVIB CALCULATIONS

15.2 evib features

The evib module, implemented in TURBOMOLE, allows to calculate the matrix elements of the first order derivative of the Kohn-Sham operator with respect to atomic displacements \( \chi \)

\[
H_{\mu\nu,\chi}^c = \langle \mu | \frac{dH^c}{d\chi} | \nu \rangle,
\]

which are required to obtain the first order EV coupling constant, as given in Eq. (15.3) [160].

Features and limitations:

- LDA, GGA and mGGA functionals supported,
- RHF and UHF,
- SMP parallelization,
- only c1 symmetry at the moment,
- derivatives of the quadrature weights are not supported, use of large grids recommended,
- just direct calculation, no ri-approximation at the moment.

15.3 General usage of evib

Calculating the matrix elements given in Eq. (15.4) consists of two steps. First a force constant calculation using aoforce is performed, where the following control flags have to be added:

\$nosalc
\$sijuai_out

This will save the derivative of the density matrix.

The subsequent run of evib requires the following control flags:

\texttt{no weight derivatives}

in the \texttt{$dft$} section, usage of large grids is therefore recommended and the multiple grids m3 to m5 should be avoided.

The elements of the matrix in Eq. (15.4) are stored in binary format in dfdxi.dat (dfdxi_a.dat and dfdxi_b.dat for UHF), using formatted Fortran output with a record length of 8 bytes for each matrix element. The matrix elements \( H_{\mu\nu,\chi}^c \) are stored in the atomic orbital (AO) basis in blocks ordered as (i) \((N_{ao}+1) \ast N_{ao}/2\) matrix elements of the upper part of the triangular matrix for the \( \mu\nu \) indices at a certain \( \chi = (k,u) \), (ii) Cartesian component \( u = x, y, z \), (iii) atom number \( k \).

Optionally,
$dfdxitextout

can be used to generate text output of the matrix elements.

For bigger systems this option can, however, generate very large output files.
Chapter 16

Calculation of NMR Shieldings

The program mpshift calculates nuclear magnetic shielding constants using the GIAO (Gauge Including Atomic Orbital) method.

At present the following methods are implemented:

- **HF-SCF**: the coupled perturbed Hartree–Fock (CPHF) equations in the AO basis are solved using a semi-direct iterative algorithm [161] similar to dscf.
- **DFT**: using either non-hybrid functionals where no iterations are needed [162] or hybrid functionals where the same algorithm as at the HF-SCF level is used.
- **MP2**: semi-direct method, see ref. [25].

### 16.1 Prerequisites

1. mpshift needs converged MO vectors from a SCF or DFT run (dscf or ridft)
2. for SCF or DFT calculations, no specifications have to be made in the control file but currently only closed shell cases are implemented in mpshift.
3. to perform an MP2 calculation of the NMR shieldings you have to prepare the input with mp2prep -c

### 16.2 How to Perform a SCF of DFT Calculation

All you have to do for running mpshift is typing mpshift at the shell level.

The results of a SCF or DFT calculation (the trace of the total shielding tensors, its anisotropy and the CPHF contribution for each symmetry distinct atom) are written into the control file after the keyword $nmr <rhf/dft> shielding constants.
16.3. **HOW TO PERFORM A MP2 CALCULATION**

This data group is write only for mpshift, but you can utilize it for graphical rendering of the calculated NMR spectra and for a quick overview of the results. A more detailed output with the complete shielding tensors can be found in the output of mpshift, so it is recommended to put the output in a file when calling the program.

### 16.3 How to Perform a MP2 calculation

To perform an MP2 calculation of the NMR shieldings you have to prepare the input with `mp2prep -c`.

mpshift will then calculate both the SCF and MP2 shielding constants. The result is written into the control file after the keyword `$nmr mp2 shielding constants`.

The script `mp2prep` will create the keywords

```
$csmp2
$thize .10000000E+10
$mointunit
    type=intermed unit=61 size=0 file=halfint
    type=1112 unit=63 size=0 file=moint#1
    type=1122 unit=64 size=0 file=moint#j
    type=1212 unit=65 size=0 file=moint#k
    type=1212a unit=70 size=0 file=moint#a
    type=gamma#1 unit=71 size=0 file=gamma#1
    type=gamma#2 unit=72 size=0 file=gamma#2
    type=dtdb#1 unit=76 size=0 file=dtdb#1
    type=dtdb#2 unit=77 size=0 file=dtdb#2
$traloop 1
$statistics mpshift
```

and starts a statistics run of mpshift (by calling mpshift). If the resulting disk space requirement exceeds the automatically detected free disk space on your system, it will increase `$traloop` and run a statistics run again. This will be done as long as your free disk space is not sufficient for the calculation.

If the `mp2prep` script fails to run on your system, try to use the `-p` option or do the procedure described above by hand. Call `mp2prep -h` for more informations about `mp2prep`.

### 16.4 Chemical Shifts

NMR shifts are obtained by comparing nuclear shieldings of your test compound with a reference molecule ($\delta_{\text{subst}} = \delta_{\text{ref}} + \sigma_{\text{ref}} - \sigma_{\text{subst}}$). Therefore you have to choose a reference molecule with a well-known shift for which you can easily calculate the absolute shielding
constant. This implies a certainty about the geometry, too. Furthermore you have to use
the very same basis set for corresponding atoms to minimize the basis set influence.

Keywords for the module Mpshift

A list of keyword for the module mpshift can be found in Section 21.2.24.

16.5 Other Features and Known Limitations

- the mpshift program can be restarted at any stage of computing, since all intermediate
  results are written into the file restartcs. In case of an external program abort you
  have to remove the $actual step flag (by the command actual -r or using an
  editor). mpshift analyses this file and decides where to continue
- ECPs can not be used since the electrons in the ECP cores are not taken into account
- only closed shell calculations are currently supported
- molecular point groups that contain reducible e representations are not supported
  ($\text{C}_n$, $\text{C}_{n\text{h}}$ with $n > 2$)
- as in mpgrad, basis sets with a contraction that is greater than 10 are currently not
  supported
- PBE and PBE0 DFT functionals are not implemented in mpshift
Chapter 17

Molecular Properties,
Wavefunction Analysis, and Interfaces to Visualization Tools

17.1 Wavefunction analysis and Molecular Properties

Molecular properties (electrostatic moments, relativistic corrections, population analyses for densities and MOs, construction of localized MOs, NTOs, etc.) can be calculated with *proper*. This program is menu-driven and reads the input that determines which properties are evaluated from standard input (i.e. the terminal or an input file if started as `proper < inputfile`). The control file and files referenced therein are only used to determine the molecular structure, basis sets, and molecular orbitals and to read results computed before with other programs.

*proper* is a post-processing tool, mainly intended for an interactive use. Several functionalities are also integrated in the programs that generate MOs or densities and can be invoked directly from the modules dscf, ridft, rimp2, mpgrad, ricc2 and egrad, if corresponding keywords are set in the control file. If one wants to skip the MO- or density generating step for dscf, ridft, rimp2, mpgrad, ricc2 it is possible to directly jump to the routine that carries out the analyses by starting the program with "<program> -proper". (For ricc2 it is, however, recommended to use instead ricctools -proper.) Currently, the respective keywords have to be inserted by hand (not with define) in the control file.

Here we briefly present the functionalities. A detailed description of the keywords that can be used in combination with the -proper flag is found Section 21.2.22.
CHAPTER 17. PROPERTIES AND ANALYSIS, AND GRAPHICS

Selection of densities: The proper program tries on start to read all densities that have been pre-calculated with any of the other programs of the TURBOMOLE package, prints a list with the densities that have been found and selects one of them for the calculation of properties. The default choice can be changed in the menu that is entered with the option dens. Here one can also list or edit additional attributes of the densities or build linear combinations of the available densities to form differences and superposition of densities.

The feature can thus be used to evaluate difference densities between the ground and excited electronic states or differences between densities calculated with different electronic structure methods.

The selected density can then be used in the subsequent menus to evaluate a variety of properties as expectation values or on a grid of points to generate interface files for visualization.

Electrostatic moments: Use the mtps option in the eval menu in proper, or add the $moments keyword control file when you start a program with the -proper flag to evaluate electrostatic moments. Up to quadrupole moments are calculated by default, on request also octupole moments are available. By default unnormalized traced cartesian moments are calculated which are defined as

$$Q_{\alpha\beta\cdots\nu}^{(n)} = \int d\rho(r)r_\alpha r_\beta \cdots r_\nu,$$

where $\rho(r)$ is the charge density as position $r$. With the option Buckingham one can request in addition the computation of cartesian traceless (Buckingham) multipole moments defined as:

$$M_{\alpha\beta\cdots\nu}^{(n)} = \frac{(-1)^n}{n!} \int d\rho(r)r^{2n} \frac{d^n}{dr_\alpha dr_\beta \cdots dr_\nu} \frac{1}{r}$$

Relativistic corrections: The option relcor in the eval menu of proper or the keyword $mvd (when starting a program with the -proper flag) initiate the calculation of relativistic corrections. With the -proper flag they are calculated for the SCF or DFT total density in case of dscf and ridft, for the SCF+MP2 density in case of rimp2 and mpgrad and for that of the calculated excited state in case of egrad. Quantities calculated are the expectation values $\langle p^2 \rangle$, $\langle p^4 \rangle$ and the Darwin term $\langle \sum_A 1/Z_A \ast \rho(R_A) \rangle$. Note, that at least the Darwin term requires an accurate description of the cusp in the wave function, thus the use of basis sets with uncontracted steep basis functions is recommended. Moreover note, that the results for these quantities are not really reasonable if ECPs are used (a respective warning is written to the output).

Population analyses: For population analyses enter the pop menu of proper. The available options and parameters that can be specified are the same as those for the $pop keyword (vide infra). If an electronic structure program is started with the -proper flag the population analyses is requested with the keyword $pop.
mulliken or $\texttt{pop}$ without any extension start a Mulliken population analysis (MPA). For 
$\texttt{-proper}$ the analysis is carried out for all densities present in the respective program, e.g.
total (and spin) densities leading to Mulliken charges (and unpaired electrons) per atom in
RHF(UHF)-type calculations in $\texttt{dscf}$ or $\texttt{ridft}$, SCF+MP2 densities in $\texttt{rimp2}$ or $\texttt{mpgrad}$,
excited state densities in $\texttt{egrad}$. Suboptions (see Section 21.2.22) also allow for the calculation
of Mulliken contributions of selectable atoms to selectable MOs including provision of data for graphical output (simulated density of states).

With $\texttt{pop nbo}$ a Natural Population Analysis (NPA) [163] is done. Currently only the
resulting charges are calculated.

With $\texttt{pop paboon}$ a population analyses based on occupation numbers [164] is performed
yielding "shared electron numbers (SENs)" and multicenter contributions. For this method
always the total density is used, i.e. the sum of alpha and beta densities in case of UHF, the
SCF+MP2-density in case of MP2 and the GHF total density for (two-component-)GHF.
Note that the results of such an analysis may depend on the choice of the number of modified
atomic orbitals ("MAOs"). By default, the number of MAOs is chosen such that they are
reasonable in most cases (see Section 21.2.22). Nevertheless it is recommended to read
carefully the information concerning MAOs given in the output before looking at the results
for atomic charges and shared electron numbers. For different ways of selecting MAOs see
Section 21.2.22.

**Generation of localized MOs:** The option $\texttt{lmos}$ in the $\texttt{mos}$ menu of $\texttt{proper}$ and
the keyword $\texttt{$\texttt{localize}$}$ with the $\texttt{-proper}$ flag trigger the calculation of localized molecular
orbitals. Per default a Boys localization including all occupied MOs is carried out (i.e.
the squared distance of charge centers of different LMOs is maximized). As output one
gets localized MOs (written to files $\texttt{lmos}$ or $\texttt{lalp/lbet}$ in UHF cases), informations about
dominant contributions of canonical MOs to LMOs and about location of LMOs (from
Mulliken PA) are written to standard output.

**Natural transition orbitals** For excited states calculated at the CIS (or CCS) level
the transition density between the ground and an excited state

$$E_{ia} = \langle \Psi_{ex}|a^\dagger_i a_i|\Psi_{ex}\rangle$$

(17.1)
can be brought to a diagonal form through a singular value decomposition (SVD) of the
excitation amplitudes $E_{ia}$:

$$[O^\dagger EV]_{ij} = \delta_{ij} \sqrt{\lambda_i}$$

(17.2)

The columns of the matrices $O$ and $V$ belonging to a certain singular value $\lambda_i$ can be
interpreted as pairs of occupied and virtual natural transition orbitals [165, 166] and the
singular values $\lambda_i$ are the weights with which this occupied-virtual pair contributes to the
excitation. Usually electronic excitations are dominated by one or at least just a few NTO
transitions and often the NTOs provide an easier understanding of transition than the
excitation amplitudes $E_{ia}$ in the canonical molecular orbital basis.
From excitation amplitudes computed with the `ricc2` program NTOs and their weights (the singular values) can be calculated with the `ntos` option in the `mos` menu of `proper` or with `ricctools`. E.g. using the right eigenvectors for the second singlet excited state in irrep 1 with:

```
ricctools -ntos CCRE0-2--1---1
```

Both programs store the results for the occupied and virtual NTOs in files named, respectively, `ntos_occ` and `ntos_vir`. The option `nto` in the `grid` menu of the `proper` program can used to evaluate NTOs for visualization on a grid of points.

Note that the NTO analysis ignores for the correlated methods (CIS(D), ADC(2), CC2, CCSD, etc.) the double excitation contributions and correlation contributions to the ground state. This is no problem for single excitation dominated transition out of a “good” single reference ground state, in particular if only a qualitative picture is wanted, but one has to be aware of these omissions when using NTOs for states with large double excitation contributions or when they are used for quantitative comparisons.

**Difference densities based on natural transition orbitals** If the excitation vectors have been obtained starting from a GHF reference, the NTOs are complex and contain contributions from both spin function. Moreover, the transitions are usually dominated by two NTOs at least. Thus, the interpretation of 2c-NTOs may become difficult. To get a simple picture of the transition at hand still, approximate difference densities can be computed according to

\[
\rho(r)_n = \Re \left( \sum_{ab} N_{\text{vir}} \phi_a(r) \phi_b(r)^\dagger \sum_i N_{\text{occ}} C_i^{a*} C_i^b - \sum_{ij} N_{\text{occ}} \phi_i(r) \phi_j(r)^\dagger \sum_a N_{\text{vir}} C_i^{a*} C_j^a \right).
\]

(17.3)

The first term corresponds to the increase of the occupation of the virtual NTOs, while the second term corresponds to the decrease of the occupation of the occupied NTOs.

This approximate difference density is available for excitation vectors obtained with the following methods: CCS/CIS, CIS(D∞), ADC(2) and CC2. Symmetry other than C₁ is currently not supported. Note that the approximate difference densities are based on the same approximations as the NTOs, namely ignoring correlation and double excitation contributions.

From excitation amplitudes computed with the `ricc2` program the approximate difference densities are computed with `ricctools`. E.g. using the right eigenvectors for the second singlet excited state in irrep 1:

```
ricctools -diffden CCRE0-1--1---2
```

This resulting density file can be visualized using the analysis mode of the `ricc2` program as described in Section 10.3.3, e.g. by adding the following lines to the `control` file.
$anadens
    calc my_approx_diffden from
    1d0 cc2-1a-002-approxdiffden.cao
$pointval

and running

    ricc2 -fanal

**Fit of charges due to the electrostatic potential:** $esp\_fit$ fits point charges at the positions of nuclei to electrostatic potential arising from electric charge distribution (for UHF cases also for spin density, also possible in combination with $soghf$). For this purpose the ("real") electrostatic potential is calculated at spherical shells of grid points around the atoms. By default, Bragg-Slater radii, $r_{BS}$, are taken as shell radii.

A parametrization very close to that suggested by Kollman (a multiple-shell model with shells of radii ranging from $1.4\times r_{vdW}$ to $2.0\times r_{vdW}$; $r_{vdW}$ is the van-der-Waals radius; U.C. Singh, P.A. Kollman, J. Comput. Chem. 5(2), 129-145 (1984)) is used if the keyword is extended:

$esp\_fit kolman

# 17.2 Interfaces to Visualization Tools

There are several possibilities to visualize structures, vibrational frequencies, molecular orbitals, densities and a large number of properties. The easiest one is to use the graphical user interface of TURBOMOLE TmoleX. TmoleX is either included in your TURBOMOLE distribution or can be downloaded for free from the COSMOlogic web site TmoleX client version. Use the COSMObuild part of TmoleX to read in the 3D data from file (plt or plv format).

**Visualization of Molecular Geometries**

The tool $t2x$ can be used to convert the atomic coordinates stored in the $\$\grad$ and $\$\coord$ data groups into the xyz-format, which is supported by most viewers, e.g. jmol. Typing

$\texttt{t2x > opt.xyz}$

in a directory containing the control file generates a series of frames using the information in $\$\grad$. Note $t2x$ writes to standard output which here is redirected to a file. If you are only interested in the most recent structure, type

$\texttt{t2x -c > str.xyz}$

which only extracts the information on $\$\coord$. 


Visualization of Densities, MOs, Electrostatic Potentials and Fields

Note that the easiest way to visualize orbitals, densities and electrostatic properties is to use the graphical user interface TmoleX.

There are several other possibilities to visualize molecular orbitals or densities. The molden option in the export menu of proper converts the MO and geometry information to the molden format. Alternatively one can use the conversion program tm2molden, which is interactive and self-explanatory. The generated file can be read in by the molden program (see molden web site).

For larger systems this may become very time-consuming, as plotting data (values on grids) are calculated by the respective programs (e.g. molden). It is more efficient to calculate the data for plots (MO amplitudes, densities, etc.) with TURBOMOLE and to use a visualization tool afterwards, a way, that is described in the following.

Calculation of data on grids to be used for plots with visualization tools (e.g. gOpenMol, available via gopenmol download) can be generated with the options in the grid menu of proper. Alternatively on can use the keyword $pointval in combination with the -proper option. This keyword is obeyed by all TURBOMOLE programs that generate densities as dscf, ridft, rimp2, mpgrad, ricc2 (see Section 10.3.3) and egrad. Note, that with -proper and $pointval all of the following quantities may be calculated simultaneously, and that for programs dscf, ridft, rimp2 and mpgrad the density matrix generating steps may be skipped by typing "<program> -proper".

Electron densities The option dens in the grid menu of proper or with the above mentioned programs setting of the keyword $pointval dens

or simply $pointval

results in the calculation of densities

$$\rho(\vec{R}_P) = \sum_{\nu\mu} D_{\nu\mu} \phi_\nu(\vec{R}_P) \phi_\mu(\vec{R}_P)$$

(17.4)

on an orthogonal grid of point \(\vec{R}_P\). The size of the grid is automatically adjusted to the size of the molecule and the resolution is adjusted to yield acceptable plots (for specification of non-default grid types (planes, lines) and non-default output formats see Section 21.2.22).

The names of the output files are:

td.plt total density (UHF: \(\alpha\) density plus \(\beta\) density )

sd.plt spin density (\(\alpha\) density minus \(\beta\) density )

mp2d.plt MP2 density
17.2. INTERFACES TO VISUALIZATION TOOLS

mp2sd.plt  MP2 spin density
ed.plt      differential density for excited state
esd.plt     differential spin density for excited state
<mynname>.plt general density passed e.g. by the ricc2 program.

The .plt files may be visualized with gOpenMol; the file coord.xyz, which is also necessary for gOpenMol, is generated by the above programs, if $pointval$ is set in the control-file.

For two-component wavefunctions (only module ridft with $soghf$ is set): Total density is on file td.plt like for one-component wave functions; this is also true for all other quantities depending only on the density matrix (electrostatic potential etc.). sd.plt contains the absolute value of the spin vector density, which is the absolute value of the following vector:

\[ s_i(r) = \left( \Psi^*_\alpha \Psi^*_\beta \right) \sigma_i \left( \begin{array}{c} \Psi_\alpha \\ \Psi_\beta \end{array} \right) \]  

$pointval$ fmt=txt

leads to a file containing the spin density vectors, which can be used by gOpenMol. It is advisable to choose ca. one Bohr as the distance between two gridpoints. The options for visualizing two-component wavefunctions are not yet available in the proper program.

**Available output formats:** To get a list of the available output formats of the proper program invoke the format option with the flag help. For the direct generation of graphics output different formats are available for 3D and 2D plots. They depend furthermore on the graphics libraries linked to proper. Use the format option with the help flag for a 2D and a 3D grid to get the list of formats available for the two cases. For the output formats available in combination with -proper keyword to export data on grids see Section 21.2.22.

**Electrostatic potentials** In an analogous way electrostatic potentials can be calculated on grids. The option ($pointval$) pot leads to the calculation of the electrostatic potential of electrons and nuclei (and external constant electric fields and point charges $Q$ if present).

\[ V(\vec{R}_P) = - \int \frac{\rho(\vec{r})}{r_{PR}} d^3r + \sum_A \frac{Z_A}{R_{PA}} + \left( \vec{R}_P \vec{E} + \sum_Q \frac{Q}{R_{PQ}} \right) \]  

(17.5)

In order to prevent the calculation of singularities at the positions of nuclei, for gridpoints that are closer to a nucleus than $10^{-6}$ a.u. the charge of the respective nucleus is omitted in the calculation of the electrostatic potential for these points. The output files are termed tp.plt, sp.plt, etc.

**Electric fields** (as derivatives of potentials) are calculated with ($pointval$) fld. The absolute values of electric fields are written to files tf.plt, sf.plt, etc. For non-default grid types and outputs that allow also for displaying of components of electric fields see Section 21.2.22.
Exchange-correlation potentials  (Only for DFT) Computation of the Kohn-Sham exchange-correlation potential on a grid is requested with 
\$pointval xc
This functionality is not (yet) available in the proper program.

Canonical molecular orbitals.  Visualization of molecular orbitals, i.e. generation of .plt-files containing amplitudes of MOs $i$
\[ A_i(\vec{R}_P) = \sum_\nu c_{i\nu} \phi_\nu(\vec{R}_P) \quad (17.6) \]
or in the two-component case
\[ A_i^\Gamma(\vec{R}_P) = \sum_\nu c_{i\nu} \phi_\nu(\vec{R}_P) \quad (17.7) \]
with $\Gamma$ as a part of the coefficient matrix ($\text{Re}(\alpha)$, $\text{Im}(\alpha)$, $\text{Re}(\beta)$, $\text{Im}(\beta)$), is achieved e.g. by ($\$pointval$) mo 10-12,15. This triggers the calculation of amplitudes for the MOs/spinors 10-12 and 15 on the grid. The numbering of MOs refers to that you get from the first column of the output of the tool Eiger, the one for spinors refers to the file EIGS. The filenames contain the type of the irreducible representation (irrep) of the MO, the current number within this irrep and in case of UHF calculations also the spin, e.g. 2a1g_a.plt contains amplitudes for the second alpha-spin MO of $a_{1g}$ type. For more-dimensional irreps columns are written to separate files, e.g. 1t2g1_a.plt, 1t2g2_a.plt and 1t2g3_a.plt contain the amplitudes of the three columns of the first irrep (alpha spin) of type $t_{2g}$.

Two-component wavefunctions (only module ridft and only if $soghf$ is set): By default only the density of the chosen spinors is written in files named e.g. 10a_d.plt. Visualization of the amplitudes of the different spinor parts is achieved e.g. by
\$pointval mo 10-12,15 minco real,
where real is a plotting threshold that may take values between zero and one. The corresponding part $\Gamma$ of the spinor ($\text{Re}(\alpha)$, $\text{Im}(\alpha)$, $\text{Re}(\beta)$, $\text{Im}(\beta)$) will be written to file, if $N^\Gamma$ (see below) is larger than that threshold.
\[ N^\Gamma = \text{tr}(D^\Gamma S) \]
\[ D^\Gamma_{\mu\nu} = \sum_i c^*_{i\nu} c^\Gamma_{i\mu} \]
The filenames consist of the number of the spinor according to file EIGS and an additional number for the respective part $\Gamma$ of the spinor (1 for $\text{Re}(\alpha)$, 2 for $\text{Im}(\alpha)$, 3 and 4 for the corresponding $\beta$-parts) e.g. 10a_4.plt for the $\text{Im}(\beta)$ of spinor 10.

Localised molecular orbitals  If one has generated localized molecular orbitals (LMOs, see above) they can also be visualized.
\$pointval lmo 3-6,8
as an example, leads to calculation of amplitudes for LMOs 3-6 and 8. The coefficients are
read from file `lmos` (UHF: `lalp` and `lbet`), the numbering refers to the output from the
localization section. For an UHF case the \( \beta \) spinorbitals get an offset of \( N_{MO} \), where \( N_{MO} \)
is the total number molecular orbitals for each spin case. If one has e.g. 22 orbitals per spin
case and is interested in plotting the first 3 \( \beta \)-type LMOs only, one have to type

\[
\texttt{$pointval lmo 23-25$}
\]

**Natural molecular orbitals for two-component wavefunctions** (only module
\texttt{ridft} and only if \texttt{$soghf$} is set): In two-component calculations it is often useful to visu-
alize natural molecular orbitals. In contrast to one-component calculations the occupation
numbers are no longer close to zero, one or two, but can take any value between zero and
two. Therefore

\[
\texttt{$natural orbitals file=natural$
\texttt{$natural orbital occupation file=natural$}
\]

has to be set additionally to \texttt{$soghf$} (also possible via \texttt{define}).

By setting

\[
\texttt{$pointval nmo 9$
\]

in control-file a gOpenMol-compatible file named `nmo_9.plt` is written which can also be
visualized with \texttt{TmoleX}

**Natural atomic orbitals** If one has generated natural molecular orbitals (NAOs, see
above) they can be visualized with the following command in the control file:

\[
\texttt{$pointval nao 7-9,12$
\]

where the numbers of the NAOs are in the output of the population analysis.

**Natural transition orbitals** If natural transition orbitals (NTOs) for electronic ex-
citations are available in files named `nto_nocc` and `nto_vir` for, respectively, the occupied
and virtual NTOs, plot files for visualizing them can be generated by setting

\[
\texttt{$pointval nto 1-5$
\]

This will generate plot files for the first five occupied and virtual NTOs. The plot file are
named `nto_vir_n.plt`, where \( n \) is the NTO index.

**Non-default grids** are decribed in detail in Sections 21.2.22. In the \texttt{proper} program
non-default grids can be specified with the \texttt{grid} option. Calculation of the above quantities
at single points is needed quite often, thus an example is given here.

\[
\texttt{$pointval geo=point$
7 5 3
0 0 7
\]
calculates densities at points (7,5,3), (0,0,7) and (1,2,3). Output is (x,y,z, density), output file suffix is .xyz.

We note in passing that calculation of electrostatic potential at positions of nuclei may be used as an efficient tool to distinguish atoms of similar atomic numbers thus providing a complement to X-Ray Structure Analysis (details see ref. [167]).
Chapter 18

Frozen Density Embedding calculations

18.1 Background Theory

In the subsystem formulation of the density-functional theory a large system is decomposed into several constituting fragments that are treated individually. This approach offers the advantage of focusing the attention and computational cost on a limited portion of the whole system while including all the remaining environmental effects through an effective embedding potential. Here we refer in particular to the (fully-variational) Frozen Density Embedding (FDE) [168] with the Kohn-Sham Constrained Electron Density (KSCED) equations [169, 170].

In the FDE/KSCED method the embedding potential required by an embedded subsystem with density $\rho_A$ to account for the presence of another (frozen) subsystem with density $\rho_B$ is:

$$v_{emb}(\mathbf{r}) = v^B_{ext}(\mathbf{r}) + v_J[\rho_B](\mathbf{r}) + \frac{\delta T_{nadd}[\rho_A;\rho_B]}{\delta \rho_A(\mathbf{r})} + \frac{\delta E_{xc}^{nadd}[\rho_A;\rho_B]}{\delta \rho_A(\mathbf{r})}$$

(18.1)

where $v^B_{ext}(\mathbf{r})$ and $v_J[\rho_B](\mathbf{r})$ are the electrostatic potentials generated by the nuclei and electron density of the subsystem B, respectively, and

$$T_{nadd}[\rho_A;\rho_B] = T_s[\rho_A + \rho_B] - T_s[\rho_A] - T_s[\rho_B],$$

(18.2)

$$E_{xc}^{nadd}[\rho_A;\rho_B] = E_{xc}[^{\rho_A + \rho_B}] - E_{xc}[\rho_A] - E_{xc}[\rho_B]$$

(18.3)

are the non-additive non-interacting kinetic energy and exchange-correlation energy functionals, respectively. In the expressions above $T_s[\rho]$ is the (unknown) non-interacting kinetic energy density functional and $E_{xc}[\rho]$ is the exchange-correlation energy functional. Note
that, while the first two terms in Eq. (18.1) refer to classical electrostatics (and could be described by e.g. external point-charges), the last two terms are related to quantum-mechanical effects.

Using freeze-and-thaw cycles, the role of the frozen and the embedded subsystem is iteratively exchanged, till convergence. If expressions (18.2) and (18.3) are computed exactly, then the density $\rho_A + \rho_B$ will coincide with the exact density of the total system.

Because the FDE/KSCED was originally developed in the Kohn-Sham framework, using standard GGA approximations for $E_{xc}[\rho]$, the non-additive exchange-correlation potential $(\delta E_{xc}^{\text{add}}/\delta \rho_A(r))$ can be computed exactly as a functional of the density, leaving the expression of the non-additive kinetic energy term as the only approximation (with respect the corresponding GGA calculation of the total system), because the exact explicit density dependence of $T_s$ from the density is not known. Using GGA approximations for the kinetic energy functional $T_s \approx \tilde{T}_{\text{GGA}}$, we have:

$$T_{nadd}^s[\rho_A;\rho_B] \approx \tilde{T}_{\text{GGA}}^s[\rho_A + \rho_B] - \tilde{T}_{\text{GGA}}^s[\rho_A] - \tilde{T}_{\text{GGA}}^s[\rho_B]$$

and

$$\frac{\delta T_{nadd}^s[\rho_A;\rho_B]}{\delta \rho_A(r)} \approx \tilde{v}_{\text{GGA}}^s[\rho_A + \rho_B](r) - \tilde{v}_{\text{GGA}}^s[\rho_A](r).$$

where $\tilde{v}_{\text{GGA}}^s(r) = \delta \tilde{T}_{\text{GGA}}^s/\delta \rho(r)$.

The FDE total energy of total system is:

$$E_{FDE}^T[\tilde{\rho}_A, \tilde{\rho}_B] = T_s[\tilde{\rho}_A] + T_s[\tilde{\rho}_B] + T_{nadd}^s[\tilde{\rho}_A; \tilde{\rho}_B] + V_{\text{ext}}[A + B] + J[\tilde{\rho}_A + \tilde{\rho}_B] + E_{xc}[\tilde{\rho}_A + \tilde{\rho}_B].$$

Note that this energy differs from the KS total energy of the total system due to the approximation in Eq. (18.4) as well as the approximated kinetic potential (see Eq. 18.5) which lead to approximated embedded densities ($\tilde{\rho}_A \approx \rho_A$ and $\tilde{\rho}_B \approx \rho_B$). With the current state-of-the-art GGA kinetic approximations, the error in the binding energy for weakly interacting systems is close to chemical accuracy.

Using the Generalized Kohn-Sham (GKS) theory, also hybrid exchange-correlation functionals can be used in embedding calculations. To obtain a practical computational method, the obtained embedding potential must be approximated by a local expression as shown in Ref. [172]. This corresponds to performing for each subsystem hybrid calculations including the interaction with other subsystems through an embedding potential derived at a semilocal level of theory. When orbital dependent exchange-correlation functionals (e.g. hybrid functional and LHF) are considered within the FDE method, the embedding potential includes a non-additive exchange-correlation term of the form

$$E_{xc}^{\text{add}}[\rho_A; \rho_B] = E_{xc}[\Phi^{A+B}[\rho_A + \rho_B]] - E_{xc}[\Phi^A[\rho_A]] - E_{xc}[\Phi^B[\rho_B]]$$

where $\Phi^{A+B}[\rho_A + \rho_B]$ denotes the Slater determinant which yields the total density $\rho_A + \rho_B$. Since such a determinant is not easily available, the non-additive exchange-correlation contribution cannot be determined directly and the non-additive exchange-correlation term...
18.2 FROZEN DENSITY EMBEDDING CALCULATIONS USING THE FDE SCRIPT

The shell script FDE controls and executes automatically FDE calculations. The script FDE prepares the input files (running define), runs the calculations (only dscf is supported in the present version), and combines the results (running fdetools). Because the FDE equations are coupled sets of one-electron equations (one for each subsystem), full relaxation of the electron densities of both subsystems is obtained by using a freeze-and-thaw \cite{171} procedure until convergence.

The converged FDE calculations are store in the subdirectories \texttt{STEPN/SUBSYSTEM\_A} and \texttt{STEPN/SUBSYSTEM\_B}, where \texttt{N} is the number of the FDE iteration. The subdirectory \texttt{ISOLATED\_SUBSYSTEM\_A} and \texttt{ISOLATED\_SUBSYSTEM\_B} contain instead the calculations for isolated subsystems (see also Section \ref{18.2.1}).

Current functionalities and limitations of FDE are:

- only \textit{C\textsubscript{1}} point group;
- only for closed-shell systems that consist of two closed-shell subsystems (e.g. weakly interacting closed-shell dimers);
- only total and binding energy calculations (no gradients);
- serial and OMP dscf runs (no MPI);
- monomolecular and supermolecular basis set approach;
- LDA/GGA kinetic energy functionals (for weakly interacting systems);
- full or pure electrostatic embedding;
- LDA/GGA, hybrid or orbital-dependent exchange-correlation potentials;
- multilevel FDE calculation;
- energy-decomposition;
- FDE calculation with subsystem B taken frozen.

In order to perform a FDE calculation, the files \texttt{coord} and \texttt{control} for the total system are necessary to take informations on atomic coordinates and basis sets. The input file for the total system can be generated, as usual, with define but no calculation on the total system is required. \texttt{\$denconv 1.d-7} option should be defined in file \texttt{control} in order to better converge the embedded densities and better describe the dipole moment.

Given a closed-shell supramolecular system with a GGA/LDA exchange-correlation functional, the command

\[
E_{xc}^{\text{nadd}}[\rho_A, \rho_B] \approx E_{xc}^{\text{GGA}}[\rho_A + \rho_B] - E_{xc}^{\text{GGA}}[\rho_A] - E_{xc}^{\text{GGA}}[\rho_B].
\]
FDE -p 3

invokes an iterative resolution of the KSCED equations with revAPBEk \([174, 175]\) as approximation of the non-additive kinetic potential (see Eq. 18.5) in the monomolecular basis set approach. The two subsystems are defined via an integer \(m = 3\) in the example above which identifies the first atom of the subsystem B in the file coord of the supramolecular system with \(n\) atoms, where the atoms \(1...m - 1\) belong to the subsystem A while the atoms \(m...n\) to the B one. Thus the file coord must contains first all the atoms of the system A and then all the atoms of the system B.

As an example we report here the FDE -p 3 output for the HF dimer:

```
FDE  Version 1.02
Frozen Density Embedding Main Driver

Scf-like procedure for
closed-shell interacting systems (dimers)

program development: Savio Laricchia
Eduardo Fabiano
Fabio Della Sala

S. Laricchia, E. Fabiano, L. A. Constantin, F. Della Sala,
S. Laricchia, E. Fabiano, F. Della Sala,
L. A. Constantin, E. Fabiano, S. Laricchia, F. Della Sala,
S. Laricchia, E. Fabiano, F. Della Sala,

Sun Mar 25 23:00:01 CEST 2012

Monomolecular basis set approach...

Serial calculation will be performed...
running /home/fabiods/REDO/branch64/TURBOMOLE/bin/em64t-unknown-linux-gnu/dscf

b-lyp exchange-correlation potential in KS supermolecular calculation...
revapbek kinetic energy approximation will be used...
Default convergence criterion on the system dipole: 0.005
Default value of starting damping parameter is 0.45
Default value of step damping parameter is 0.10
Default value of maximum damping parameter is 0.90
18.2. FROZEN DENSITY EMBEDDING CALCULATIONS USING THE FDE SCRIPT

Default value of maximum fde iterations is 20
Saving options in fde.input

+-----------------------------------------------------------+  
| Subsystem A atomic coordinates and basis set information |
| x    y    z   atom   basis set   ecp   |  
+-----------------------------------------------------------+  

2.5015  -0.1705  -0.0000  f  def2-TZVP  none  
3.2889   1.3859   0.0000  h  def2-TZVP  none

+-----------------------------------------------------------+  
| Subsystem B atomic coordinates and basis set information |
| x    y    z   atom   basis set   ecp   |  
+-----------------------------------------------------------+  

-2.7537  0.0364  -0.0000  f  def2-TZVP  none  
-1.0191  -0.1789  0.0003  h  def2-TZVP  none

Running Isolated subsystems:

************************
* ISOLATED SUBSYSTEM A *
************************
Done!

************************
* ISOLATED SUBSYSTEM B *
************************
Done!

Saved isolated subsystems data in:
isolated_energy.ks
mos_A.ks
mos_B.ks

************************
* FDE - step 1 *
************************

FDE ENERGY (TOTAL SYSTEM): -200.96417090754 Ha
FDE BINDING ENERGY: 5.865327 mHa
3.680548 kcal/mol
Dipole convergence: 0.138071, Damping: 0.45

********************
* FDE - step 2  *
********************

FDE ENERGY (TOTAL SYSTEM): -200.96418098234 Ha
FDE BINDING ENERGY: 5.875401 mHa
3.686870 kcal/mol
Dipole convergence: 0.009246, Damping: 0.35

********************
* FDE - step 3  *
********************

FDE ENERGY (TOTAL SYSTEM): -200.96418289036 Ha
FDE BINDING ENERGY: 5.877309 mHa
3.688067 kcal/mol
Dipole convergence: 0.004395, Damping: 0.25

See embedded subsystems calculations in:
STEP3/SUBSYSTEM_A
STEP3/SUBSYSTEM_B
See total system in:
STEP3/ENERGY_SYSTEM

Sun Mar 25 23:00:21 CEST 2012
Total time: 20 secs.

The final energies are stored in the file fde_energy. The directory STEP3/ENERGY_SYSTEM contains the total system with density $\rho_A + \rho_B$; this directory can (only) be used for density analysis.

### 18.2.1 Options

All the options for the FDE can be specified as commandlines, and are described below. The options can be also be specified in file fde.input, which is read by the FDE script. If fde.input is not present it is created by the FDE script. Command lines options overwrites options found in the fde.input file.

#### Subsystem definition

The flag `-p integer` is required or it must be present in the fde.input file.
Equivalent command: \texttt{--pos-cut integer} \\
\texttt{fde.input} option: \texttt{pos-cut= integer}

\textbf{Kinetic-energy functionals}

In order to use different GGA approximations of the non-additive kinetic potential, the flag 
\texttt{-k string} must be used. Here \textit{string} is the acronym used to identify a given GGA kinetic  
energy approximation, that can be selected among the following functionals:

- \textit{string}=\texttt{revapbek}: generalized gradient approximation with a PBE-like enhancement  
  factor, obtained using the asymptotic expansions of the semiclassical neutral atom as  
  reference \cite{174,175} (revAPBEk). This is the default choice;
- \textit{string}=\texttt{lc94}: Perdew-Wang (PW91) exchange functional reparametrized for kinetic  
  energy by Lembarki and Chermette \cite{176} (LC94);
- \textit{string}=\texttt{t-f}: gradient expansion truncated at the zeroth order (GEA0), corresponding  
  to the Thomas-Fermi functional.

For example, the command

\begin{verbatim}
FDE -p 3 -k lc94
\end{verbatim}

approximates the non-additive kinetic contribution to the embedding potential through the  
functional derivative of LC94 kinetic energy functional.

A pure electrostatic embedding can be also performed with \texttt{FDE} script, where the embedding  
potential required by a subsystem A to account for the presence of the B one will be merely:

\begin{equation}
    v_{\text{emb}}(\mathbf{r}) = v^B_{\text{ext}}(\mathbf{r}) + v[J\rho_B](\mathbf{r})
\end{equation}

with $v^B_{\text{ext}}(\mathbf{r})$ and $v[J\rho_B](\mathbf{r})$ the electrostatic potentials generated respectively by the nuclei  
and electron density of the subsystem B. To perform an electrostatic embedding calculation use

\begin{verbatim}
FDE -p 3 -k electro
\end{verbatim}

and can be performed for both Kohn-Sham (only for LDA/GGA exchange-correlation functionals)  
and Hartree-Fock methods.

The electrostatic embedding is implemented only for testing purpose. It resembles an  
electrostatic embedding with external point-charges and/or point-dipoles, but it is “exact” as it  
is based on the whole densities (i.e. it considers all multipole moments of the density and the  
polarizabilities at all orders).

Equivalent command: \texttt{--kin string} \\
\texttt{fde.input} option: \texttt{kin= string}
CHAPTER 18. FROZEN DENSITY EMBEDDING CALCULATIONS

FDE charged subsystems

FDE can perform calculations for charged closed-shell systems whose charge is localized on one or both subsystems. To localize the charge on a given subsystem, \texttt{--chargeA= integer} must be used for the subsystem A and \texttt{--chargeB= integer} for the B one. Here \texttt{integer} denotes the charge added to the neutral subsystem. For example, the command

\begin{verbatim}
FDE -p 3 --chargeA= 2
\end{verbatim}

performs a FDE calculation for a negative charged closed shell system (for example Zn(\text{H}_2\text{O})_2^{2+}) whose subsystem B has charge 2. Note that in this case the starting control file must have a charge +2.

\texttt{fde.input} option: \texttt{chargeA= integer, chargeB= integer}

FDE with subsystem B taken frozen

FDE can perform embedding calculations where the subsystem B is taken frozen, i.e. without \textit{scf} calculation on it using an embedding potential. Therefore only one step will be performed if the flag \texttt{--frozen} will be used

\begin{verbatim}
FDE -p 3 --frozen
\end{verbatim}

The frozen embedding calculation is store in the subdirectory \texttt{STEP1/SUBSYSTEM_A}. The control file is modified with the following keywords:

\begin{verbatim}
$fde read
$fde-input zj file=fde_ZJ.mat
$fde-input kxc file=fde_KXC.mat
\end{verbatim}

The program \texttt{dscf} will read the submatrices \texttt{fde_ZJ.mat} and \texttt{fde_KXC.mat} and add them to the Hamiltonian.

\texttt{fde.input} option: \texttt{frozen=1}

Parallel calculations

If \texttt{PARA_ARCH=SMP} and OMP calculation will be performed. The flag \texttt{-nth nthreads} can be used to specify the number of threads. For example, with the following command

\begin{verbatim}
FDE -p 3 -nth 4
\end{verbatim}

will use 4 threads.

Equivalent command: \texttt{--nthreads integer}

\texttt{fde.input} option: \texttt{nthreads= integer}
Monomolecular and supermolecular basis set approach

The $\rho_A$ and $\rho_B$ densities can be expanded using the supermolecular or monomolecular basis set. In a supermolecular basis set expansion the basis functions $\{\chi\}$ of both subsystems are employed to expand the subsystem electron densities. In a monomolecular basis set expansion, instead, only basis functions $\{\chi^\ell\}$ centered on the atoms in the $\ell$-th subsystem are used to expand the corresponding density.

Both monomolecular and supermolecular basis set expansion of the electron densities are implemented in FDE: with the flag -m a monomolecular expansion is performed, while for a supermolecular one -s is used. In the absence of both flags a monomolecular expansion is performed by default.

For an accurate calculation of binding-energies of weakly interacting molecular systems a supermolecular basis set is required (to avoid the basis-set superposition error). Otherwise a very large monomolecular basis set is necessary.

NOTE: The FDE script supports only basis-set in the TURBOMOLE library.

Equivalent command: --mono or --super

fde.input option: method=mono or method=super

Convergence of the freeze-and-thaw cycles

The script FDE runs a self-consistent calculation when a convergence criterion is fulfilled. The convergence criterion is the change in the total dipole moment. This is a tight convergence criterion, as the dipole moment is highly sensitive to small changes in electron density. The convergence parameter $\varepsilon^j$ for the $j$-th step in the freeze-and-thaw procedure is computed by means the following expression

$$\varepsilon^j = \frac{|\Delta \mu^j_A| + |\Delta \mu^j_B|}{2}$$

(18.10)

where

$$|\Delta \mu^j_i| = |\mu^j_i| - |\mu^{j-1}_i|$$

is the difference between the dipole moments of two consecutive steps for the $i$-th subsystem.

Eq. (18.10) allows to consider changes in both subsystems or one of them because of the relaxation of their electron densities. By default, FDE stops when $\varepsilon^j \leq 0.005$ a.u.. The default value for the convergence criteria can be changed using the flag --epsilon=real where real is a decimal number.

The maximum number of freeze-and-thaw cycles can be specified by --max-iter=integer, and the default value is 20.

In order to make easy the convergence of the iterative solution of the KSCED coupled equations, a damping factor $\eta$ must be used for the matrix elements of the embedding potential $\langle v_{\text{emb}} \rangle_{ij}$ as perturbation to a given subsystem

$$d(v_{\text{emb}})^{j}_{ik} = (1 - \eta)(v_{\text{emb}})^{j}_{ik} + \eta(v_{\text{emb}})^{j-1}_{ik}$$

(18.11)
for the $j$-th iteration. Here $d(v_{\text{emb}})_{ik}^j$ is the matrix element effectively used in the $j$-th iteration after the damping. In FDE the starting value of $\eta$ can be changed using \texttt{--start-damp=real} (default value is 0.45) where \texttt{real} is a decimal number. The damping parameter can also dynamically change at each iterative step (according to the convergence process) of a quantity set by \texttt{--step-damp=real} (default value is 0.10). The minimum value set by \texttt{--max-damp=real} (default value is 0.90).

\texttt{fde.input} options:
\begin{itemize}
  \item \texttt{epsilon=real}
  \item \texttt{max-iter=integer}
  \item \texttt{start-damp=real}
  \item \texttt{max-damp=real}
  \item \texttt{step-damp=real}
\end{itemize}

Embedding energy error

The embedding error in the total energy is computed as
\[
\Delta E = E^{\text{FDE}}[\tilde{\rho}_A; \tilde{\rho}_B] - E^{\text{DFT}}[\rho]
\]  
(18.12)

where $E^{\text{DFT}}$ is the DFT total energy of total system with density $\rho(r)$. In order to compute $\Delta E$ as well as its components, the flag \texttt{--err-energy} must be used. This flag will required also the DFT calculation on the total system. In this case the converged SCF output file must be named \texttt{output.dscf}.

An example of session output for the computation of embedding energy and energy error decomposition, when \texttt{--err-energy} flag is present, is the following:

\begin{verbatim}
FDE ENERGY (TOTAL SYSTEM): -200.99720391651 Ha
FDE BINDING ENERGY: 4.960885 mHa
3.113002 kcal/mol
FDE ENERGY ERROR: 2.003352 mHa
ERROR ENERGY DECOMPOSITION
coulomb contribution: -0.693026 mHa
nuclear contribution: -3.136544 mHa
exchange-correlation contribution: -1.156390 mHa
kinetic contribution: 6.989320 mHa
\end{verbatim}

where the FDE energy ($E^{\text{FDE}}$), the FDE binding energy, the embedding energy error ($\Delta E$) and the error energy decomposition in its coulomb, nuclear, exchange-correlation and kinetic contributions are reported. This output is present at each FDE iteration.

\texttt{fde.input} option: \texttt{err-energy=1}
18.2. FROZEN DENSITY EMBEDDING CALCULATIONS USING THE FDE SCRIPT

Table 18.1: Other options in the shell script FDE

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d or --dipole</td>
<td>dipole=1</td>
</tr>
<tr>
<td>-v or --verbose</td>
<td>shows more informations</td>
</tr>
<tr>
<td>--save-mos</td>
<td>save-mos=1</td>
</tr>
<tr>
<td>--save-matrix</td>
<td>save-matrix=1</td>
</tr>
<tr>
<td>--help</td>
<td>list all commands</td>
</tr>
</tbody>
</table>

**Restarting**

The script FDE checks in the current directory for previous FDE calculations. If these are present, then the FDE calculation will be restarted from the last iteration found. The directories `ISOLATED_SUBSYSTEM_A` and `ISOLATED_SUBSYSTEM_B` will be overwritten by the converged calculations from previous run. The energy and the orbital from the isolated systems are saved in the current directory in the files: `isolated_energy.ks`, `mos_A.ks` and `mos_B.ks`.

Note that a restart is possible only if the same subsystem definition and the same basis set are used (e.g. the same -p flag and the -s or -m flag). Other flags, e.g. kinetic and xc-functionals and convergence parameters, can be instead modified.

As all the options are saved in the `fde.input` file, to restart a FDE calculation the `FDE script can be invoked without any parameters`.

To force a calculation from scratch use:

```
FDE -p 3 --scratch
```

18.2.2 FDE with hybrid and orbital-dependent functionals

In order to use local approximations (18.1) and (18.8) with FDE, the flag `-f string` must be add to the options of the script. Here `string` denotes the local/semilocal approximation to hybrid or orbital-dependent exchange-correlation potentials in $v_{emb}(r)$. All LDA/GGA functionals in TURBOMOLE can be considered as approximations.

For example, the command

```
FDE -p 3 -f b-lyp
```

can be used to approximate bh-lyp or b3-lyp hybrid non-additive potentials, while the command
approximates the \texttt{pbe0} hybrid non-additive potentials. Other combinations of functionals are not recommended (meta-GGA are not supported).

Finally, also calculations with the Local Hartree-Fock (LHF) potential can be performed. In this case the command

\begin{verbatim}
FDE -p 3 -f becke-exchange
\end{verbatim}

can be used to approximate the LHF non-additive potential [173].

Equivalent command: \texttt{--func string}

\texttt{fde.input} option: \texttt{func= string}
Chapter 19

Orbital Dependent Kohn-Sham Density Functional Theory

19.1 Theoretical Background

Approximations to the exchange-correlation (XC) functional of the Kohn-Sham (KS) Density Functional Theory (DFT) can be classified by the so-called “Jacob’s ladder.” The ground on which the ladder lies is the Hartree approximation (XC energy is zero), and the first rung is the local density approximation (LDA) in which the XC energy density is a simple local function of the density. The second rung of the Jacob’s ladder is the generalized gradient approximation (GGA): in this case the XC energy density depends also on the gradient of the density. In the third rung (meta-GGA) an additional variable is used, the Kohn-Sham kinetic energy density which allows, e.g., to construct self-correlation-free functionals. Functionals in the above rungs can have high accuracy for different class of problems in chemistry and solid-state physics, but their main limitation is the self-interaction error (SIE) \[177–180\]. To avoid the SIE the exchange must be treated exactly and this can be achieved by functionals in the fourth rung which depend explicitly on all the occupied KS orbitals. In the KS formalism the EXX (exact-exchange) energy is (for closed-shell systems \(n_s = 2\)) \[177–180\]:

\[
E_x^{\text{EXX}} = -\frac{n_s}{2} \sum_{\text{occ.}} \sum_{\text{occ.}} \int \int d\mathbf{r}d\mathbf{r}' \phi_{\alpha}^{\text{KS}}(\mathbf{r})\phi_{\beta}^{\text{KS}}(\mathbf{r})\phi_{\alpha}^{\text{KS}}(\mathbf{r}')\phi_{\beta}^{\text{KS}}(\mathbf{r}') \frac{1}{\|\mathbf{r} - \mathbf{r}'\|}.
\] (19.1)

i.e. the same functional form of the Hartree-Fock (HF) exchange but computed with KS orbitals which are obtained using a self-consistent local EXX potential. At this point we should recall that hybrid DFT functionals (including HF exchange), doesn’t belong to the KS formalism: in hybrid DFT, in fact, the non-local HF exchange operator \(\hat{v}_x^{\text{NL}}(\mathbf{r}, \mathbf{r}') = \sum a \phi_a(\mathbf{r})\phi_a(\mathbf{r}') / \|\mathbf{r} - \mathbf{r}'\|\) is employed in the self-consistent (Generalized Kohn-Sham) equations determining the orbitals.
While LDA, GGA, meta-GGA and hybrid functionals are implemented (for ground-state calculations) in the \texttt{dscf} and \texttt{ridft}, the \texttt{odft} module considers functionals of the fourth rung. Currently exchange-only orbital-dependent approaches are implemented in the \texttt{odft} module. The EXX KS local potential \((v_x^{\text{EXX}}(r))\) can be obtained using the optimized effective potential (OEP) method (in each self-consistent step): [178–181]:

\[
\int \text{d}r' \chi_s(r, r') v_x^{\text{EXX}}(r) = \sum_{\text{occ}} \sum_{s} 2n_s \langle \phi_a | \hat{v}_{x}^{\text{NL}} | \phi_s \rangle \frac{\phi_s(r) \phi_a(r)}{\epsilon_a - \epsilon_s} \tag{19.2}
\]

where \(\chi_s(r, r') = \sum_{\text{occ}} 2n_s \sum_{s} \phi_i(r) \phi_j(r') \frac{\phi_s(r) \phi_a(r')}{\epsilon_a - \epsilon_s}\) is the non-interacting density response. An effective approximation to the OEP-EXX potential is given by the Localized Hartree–Fock (LHF) potential [177] which is given by

\[
v_x^{\text{LHF}}(r) = -\sum_{ij} n_s \frac{\phi_i(r) \phi_j(r)}{\rho(r)} \int \text{d}r' \frac{\phi_i(r') \phi_j(r')}{||r - r'||} + \sum_{ij} n_s \frac{\phi_i(r) \phi_j(r)}{\rho(r)} \langle \phi_i | v_x^{\text{LHF}} - \hat{v}_{x}^{\text{NL}} | \phi_j \rangle \tag{19.3}
\]

where the first term is called Slater potential and the second term correction term. If terms \(i \neq j\) are neglected in the correction term, the Krieger-Li-Iafrete (KLI) potential [182] is obtained. Note that the Eq. (19.3) depends only on occupied orbitals, whereas Eq. (19.2) depends also on virtual orbitals. The LHF total energy is assumed to be the EXX total energy, even if LHF is not variational (although the deviation from the EXX energy is very small, usually below 0.01\%). The LHF potential is equivalent to the Common Energy Denominator Approximation (CEDA) [183] and to the Effective Local Potential (ELP) [184].

Both OEP-EXX and LHF (in contrast to functionals of the first three rungs) satisfy the HOMO condition [182]

\[
\langle \phi_{\text{HOMO}} | v_x | \phi_{\text{HOMO}} \rangle = \langle \phi_{\text{HOMO}} | \hat{v}_{x}^{\text{NL}} | \phi_{\text{HOMO}} \rangle \tag{19.4}
\]

and the asymptotic relation [185, 186]

\[
v_x(r_l) \xrightarrow{r_l \to \infty} \langle \phi_M | v_x - \hat{v}_{x}^{\text{NL}} | \phi_M \rangle - \frac{1}{r_l} \tag{19.5}
\]

where \(\phi_M\) is the highest occupied orbital which do not have a nodal surface in the asymptotic region along direction \(r_l\). Considering together with condition (19.4), we finally obtain that:

- \(v_x(r)\) will approach \(-1/r\) along all directions where \(\phi_{\text{HOMO}}(r)\) does not have a nodal surface in the asymptotic region (e.g. this is the case of atoms);
- on directions which belong to the nodal surface of the HOMO, the \(v_x(r)\) will approach \(\langle \phi_M | v_x - \hat{v}_{x}^{\text{NL}} | \phi_M \rangle - 1/r\).

Both OEP-EXX and LHF gives total energies very close to the Hartree-Fock one (actually \(E_{\text{LHF}} > E_{\text{EXX}} > E_{\text{HF}}\)), thus, without an appropriate correlation functional, these methods are not suitable for thermochemistry. On the other hand OEP-EXX and LHF give very good
KS orbital spectra. In fact the eigenvalues of the HOMO is very close to the Hartree-Fock and to exact ionization potential (I.P): this is in contrast to functional of the first three rungs which underestimate the HOMO energy by several eVs. In addition a continuum set of bound unoccupied orbitals are obtained. Thus OEP-EXX or LHF KS orbitals are very good input quantities for computing NMR shielding constants [187], energy-levels in hybrid interfaces [188] and TD-DFT excitation energies [189] (the latter using LDA/GGA kernels, not the hybrid ones).

19.2 Implementation

Both the OEP-EXX and LHF methods can be used in spin–restricted closed–shell and spin–unrestricted open–shell ground state calculations. Both OEP-EXX and LHF are parallelized in the OpenMP mode.

19.2.1 OEP-EXX

In the present implementation the OEP-EXX local potential is expanded as [181]:

$$v_{\text{EXX}}^x(r) = \sum_p c_p \int g_p\left(\frac{r'}{|r-r'|}\right) \, dr'$$

(19.6)

where $g_p$ are gaussian functions, representing a new type of auxiliary basis-set (see directory xbasen). Inserting Eq. (19.6) into Eq. (19.2) a matrix equation is easily obtained for the coefficient $c_p$. Actually, not all the coefficients $c_p$ are independent each other as there are other two conditions to be satisfied: the HOMO condition, see Eq. (19.4), and the charge condition

$$\int \sum_p c_p g_p(r) dr = -1 ,$$

(19.7)

which ensures that $v_{\text{EXX}}^x(r)$ approaches $-1/r$ in the asymptotic region. Actually Eq. (19.6) violates the condition (19.5) on the HOMO nodal surfaces (such condition cannot be achieve in any simple basis-set expansion).

Note that for the computation of the final KS Hamiltonian, only orbital basis-set matrix elements of $v_{\text{EXX}}^x$ are required, which can be easily computes as three-index Coulomb integrals. Thus the present OEP-EXX implementation is grid-free, like Hartree-Fock, but in contrast to all other XC-functionals.

19.2.2 LHF

In the LHF implementation the exchange potential in Eq. (19.3) is computed on each grid-point and numerically integrated to obtain orbital basis-sets matrix elements. In this case the DFT grid is needed but no auxiliary basis-set is required. The Slater potential can be computed numerically on each grid point (as in Eq. 19.3) or using a basis-set expansion as [177]:
\[ v_n^{\text{lat}}(r) = \frac{n_s}{\rho(r)} \sum_a u_a^T \chi(r) \chi^T(r) S^{-1} K u_a. \]

(19.8)

Here, the vector \( \chi(r) \) contains the basis functions, \( S \) stands for the corresponding overlap matrix, the vector \( u_a \) collects the coefficients representing orbital \( a \), and the matrix \( K \) represents the non-local exchange operator \( \hat{v}_{NL}^x \) in the basis set. While the numerical Slater is quite expensive but exact, the basis set method is very fast but its accuracy depends on the completeness of the basis set.

Concerning the correction term, Eq. (19.3) shows that it depends on the exchange potential itself. Thus an iterative procedure is required in each self-consistent step: this is done using the conjugate-gradient method.

Concerning conditions (19.4) and (19.5), both are satisfied in the present implementation. KS occupied orbitals are asymptotically continued \[185\] on the asymptotic grid point \( r \) according to:

\[ \tilde{\phi}_i(r) = \phi_i(r_0) \left( \frac{|r|}{|r_0|} \right)^{(Q+1)/\beta_i - 1} e^{-\beta_i (|r| - |r_0|)}, \]

(19.9)

where \( r_0 \) is the reference point (not in the asymptotic region), \( \beta = \sqrt{-2\epsilon_i} \) and \( Q \) is the molecular charge. A surface around the molecule is used to defined the points \( r_0 \).

### 19.3 How to Perform

**OEP-EXX**

To run OEP-EXX calculations select:

```
$dft$
  functional oep
```

As the computation of the OEP functional is completely analytic and grid free, any selection of a grid type or size will not influence the OEP calculation in contrast to other density functionals.

Particular care is instead required to orbital and auxiliary basis set. An arbitrary combination of them can lead to very good total energy (i.e. very close to the Hartree-Fock one) but unphysical OEP potential. In the present release we strongly recommend to use the \texttt{d-aug-cc-pVTZ-oep} basis set and the corresponding auxiliary basis set (directory \texttt{xbasen}).

The following options can modify the quality, time and output of an OEP calculation. All the options can be set by \texttt{define}.

Every option has a reasonable default value so the user does not need to select any of the options below to run a proper OEP calculation.
$\textit{\$oep options}$

Listing of all possible options for the flag $\textit{\$oep}$.

\textit{\textbf{charge vector integer}}

The Charge condition expansion coefficients in auxiliary basis set representation can be calculated in different kinds.

The selection of \textit{integer} = 1 will use the following ansatz to calculate the coefficients:

\[ c_{P_1} = \begin{cases} -\frac{1}{N'_{a_{ux}}} \cdot \frac{1}{G_P} & \text{if } G_P \neq 0 \\ 0 & \text{if } G_P = 0 \end{cases}. \]

\( G_P \) is the integral over a normalized Gaussian auxiliary basis function. \( N'_{a_{ux}} \) is the number of auxiliary basis functions with \( G_P \neq 0 \).

The selection of \textit{integer} = 2 will use the following ansatz to calculate the coefficients:

\[ c_{P_2} = \begin{cases} -\frac{1}{\sum_P G_P} & \text{if } G_P \neq 0 \\ 0 & \text{if } G_P = 0 \end{cases}. \]

The variable \textit{integer} must have an integer value. The default value is 2.

\textit{\textbf{condition [string2] string}}

In the OEP method two constraints can be applied in the OEP equation. This is the HOMO condition and the Charge condition. The variable \textit{string} can have the values \textit{none}, \textit{HOMO}, \textit{Charge} and \textit{both}. No condition is chosen when \textit{none} is elected. The HOMO condition is chosen when \textit{HOMO} is elected. The Charge condition is chosen when \textit{Charge} is elected. The HOMO condition and the Charge condition are chosen when \textit{both} is elected.

The variable \textit{string2} is optional and only electable if a spin-unrestricted calculation is performed. The variable \textit{string2} can have the values \textit{alpha} and \textit{beta}. If \textit{string2} = \textit{alpha} then the condition is defined for the alpha spin channel. If \textit{string2} = \textit{beta} then the condition is defined for the beta spin channel. Both spin channels can have different values.

Example:

\begin{verbatim}
$\textit{\$oep}$
condition alpha HOMO
condition beta Charge
\end{verbatim}

If only one spin channel is defined the other spin channel uses the same condition automatically. The default value in any case is \textit{string = both}.

\textit{\textbf{core memory integer}}

Core memory is the amount of main memory given to the OEP calculation to store the three index integrals calculated during the OEP calculation. The core memory amount is given MB. The calculation runs as fast as possible if
all three index integrals can be stored in the core memory. The variable \textit{integer} must have an integer value. The default value is 200.

\textbf{debug}
Print further information about the OEP calculation especially matrices and vectors used during the OEP calculation. Use this option carefully since a lot of data is written. The default value is \textit{.false.}.

\textbf{eigenvalue difference integer}
Two molecular orbitals are considered as degenerated (due to symmetry or incidentally), if the difference between them is smaller than $10^{-\textit{integer}}$. The variable \textit{integer} must have an integer value. The default value is 6.

\textbf{plot coefficient string}
The expansion coefficients for the auxiliary basis functions which build the local exact exchange potential are written to the file \texttt{oe pcVx.dat} or in case of a spin-unrestricted calculation to the files \texttt{oe pcVxa.dat} and \texttt{oe pcVxb.dat}. If \textit{string} is \texttt{cartesian} the expansion coefficients are given for a cartesian atomic orbital auxiliary basis, if \textit{string} equals \texttt{spherical} the expansion coefficients are given for a spherical atomic orbital auxiliary basis. In any case the expansion coefficients are given for the single atomic orbital auxiliary basis function and contain no information about the symmetry of the system (c1 case). The default value is \texttt{cartesian}.

\textbf{reference potential}
Use the reference potential constructed by the applied conditions to the OEP calculation as exchange potential. The solution of the OEP equation is skipped. The default value is \texttt{.false.}.

\textbf{LHF}

To run a LHF calculations select:

\begin{verbatim}
$df t
  functional lhf
  gridsize  3
\end{verbatim}

This can be done using \texttt{define} (modified grid are not supported) and then run \texttt{od ft}.

A more suitable procedure is the following:

1) Do a Hartree–Fock calculation using \texttt{dscf}.

2) Use the script \texttt{lhfprep} to prepare the \texttt{control} file (the old \texttt{control} file will be saved in \texttt{control.hf} and the molecular orbitals in \texttt{mos.hf} or in \texttt{alpha.hf} and \texttt{beta.hf} for the spin-unrestricted case). See \texttt{lhfprep -help} for options. Actually LHF can be started from any guessed orbitals, but if HF orbitals are used, a much faster convergence is expected. By default the script \texttt{lhfprep} will add/modify the \texttt{control} file with:
19.3. **HOW TO PERFORM**

$\text{dft}$

- **functional** \text{lhf}
- **gridtype** 6
- **gridsize** 3
- **radsize** 3

$\text{lhf}$

- **off-diag on**
- **num-slater off**
- **asymptotic** \text{dynamic}=1.d-3
- **conj-grad conv=1.d-6** \text{maxit}=20 \text{output}=1 \text{asy}=1
- **slater-dtresh** 1.d-9
- **slater-region** 7.0 0.5 10.0 0.5
- **correct-region** 10.0 0.5

$\text{scfdump}$

$\text{scfiterlimit}$ 30

$\text{scfconv}$ 6

$\text{scfdamp}$ \text{start}=0.000 \text{step}=0.500 \text{min}=0.50

$\text{scforbitalshift}$ noautomatic

$\text{correction}$ matrix-elements file=lhfcg

$\text{correction}$ \text{alpha} matrix-elements file=lhfcg_alpha

$\text{correction}$ \text{beta} matrix-elements file=lhfcg_beta

3) Run \text{odft}.

With the LHF potential Rydberg series of virtual orbitals can be obtained. To that end, diffuse orbital basis sets have to be used and special grids are required.

**gridtype 4** is the most diffuse with special radial scaling; **gridtype 5** is for very good Rydberg orbitals; **gridtype 6** (default in Lhfprep) is the least diffuse, only for the first Rydberg orbitals.

Only gridsize 3–5 can be used, no modified grids.

Use \text{test-integ} to check if the selected grid is accurate enough for the employed basis-set, see page 313.

The options in the $\text{lhf}$ group are:

- **off-diag on**
  - The LHF exchange potential is computed (default);

- **off-diag off**
  - The KLI exchange potential is computed (can be selected by \text{lhfprep -kli}).

- **num-slater on**
  - the Slater potential is calculated numerically everywhere: this is more accurate but quite expensive. When ECPs are used, turn on this option. It can be selected by \text{lhfprep -num}. 

CHAPTER 19. ORBITAL DEPENDENT DFT

num-slater off
the Slater potential is computed using basis-sets. This leads to very fast calculations, but accurate results are obtained only for first-row elements or if an uncontracted basis set or a basis set with special additional contractions is used. This is the default.

asymptotic
for asymptotic treatment there are three options:
asymptotic off
No asymptotic-treatment and no use of the numerical Slater. The total exchange potential is just replaced by \(-1/r\) in the asymptotic region. This method is the fastest one but can be used only for the density-matrix convergence or if Rydberg virtual orbitals are of no interest.
asymptotic on
Full asymptotic-treatment and use of the numerical Slater in the near asymptotic-region. It can be selected by lhfprep -asy.
asymptotic dynamic=1.d-3
Automatic switching on (off) to the special asymptotic treatment if the differential density-matrix rms is below (above) 1.d-3. This is the default.
pot-file save
the converged Slater and correction potentials for all grid points are saved in the files slater.pot and corrct.pot, respectively. Using pot-file load, the Slater potential is not calculated but read from slater.pot (the correction potential is instead recalculated). For spin unrestricted calculations the corresponding files are slaterA.pot, slaterB.pot, corrctA.pot and correctB.pot.
homo
allows the user to specify which occupied orbital will not be included in the calculation of correction potential: by default the highest occupied orbital is selected. This option is useful for those systems where the HOMO of the starting orbitals (e.g. EHT, HF) is different from the final LHF HOMO. homob is for the beta spin.
correlation func=functional
a correlation functional can be added to the LHF potential: use func=lyp for LYP, or func=vwn for VWN5 correlation.

For other options see 19.3.

19.4 How to plot the exchange potential
It is recommended to check plots of the exchange potential for both OEP-EXX and LHF potential, to avoid spurious numerical oscillations (which usually originates from too small or too large basis-set). To plot the LHF potential over a line, add to the control file (e.g. for a 2000 points along the z axis):
and run \texttt{odft -proper}. The plotting subroutine reads the file \texttt{lhfcg}, containing the matrix elements of the Correlation potential is already generated by a previous run. The file \texttt{tx.vec} will be generated with four columns (distance, LHF potential, Slater potential, Correction potential).

The procedure to plot the OEP-EXX potential is the same. In this case the expansion coefficients (see Eq. 19.6) are read from the file \texttt{oepcVx.dat} (cartesian format). The file \texttt{tx.vec} will be generated with four columns (distance, EXX potential, EXX potential, zero).

## 19.5 How to quote

- For LHF calculations with \texttt{odft}:

- For OEP-EXX calculations with \texttt{odft}:
Chapter 20

Treatment of Solvation Effects
with COSMO

The Conductor-like Screening Model [190] (COSMO) is a continuum solvation model (CSM), where the solute molecule forms a cavity within the dielectric continuum of permittivity $\varepsilon$ that represents the solvent. The charge distribution of the solute polarizes the dielectric medium. The response of the medium is described by the generation of screening charges on the cavity surface.

CSMs usually require the solution of the rather complicated boundary conditions for a dielectric in order to obtain the screening charges. COSMO instead uses the much simpler boundary condition of vanishing electrostatic potential for a conductor,

$$\Phi^{tot} = 0.$$

This represents an electrostatically ideal solvent with $\varepsilon = \infty$. The vector of the total electrostatic potential on the cavity surface segments is determined by the solute potential $\Phi^{sol}$, which consist of the electronic and the nuclear part, and the vector of the screening charges $q$,

$$\Phi^{tot} = \Phi^{sol} + Aq = 0.$$

$A$ is the Coulomb matrix of the screening charge interactions. For a conductor, the boundary condition $\Phi^{tot} = 0$ defines the screening charges as

$$q = -A^{-1}\Phi^{sol}.$$

To take into account the finite permittivity of real solvents, the screening charges are scaled by a factor.

$$f(\varepsilon) = \frac{\varepsilon - \frac{1}{2}}{\varepsilon + \frac{1}{2}},$$

$$q^* = f(\varepsilon)q.$$
The deviation between the Cosmo approximation and the exact solution is rather small. For strong dielectrics like water it is less than 1%, while for non-polar solvents with $\varepsilon \approx 2$ it may reach 10% of the total screening effects. However, for weak dielectrics, screening effects are small and the absolute error therefore typically amounts to less than one kcal/mol. The dielectric energy, i.e. the free electrostatic energy gained by the solvation process, is half of the solute-solvent interaction energy.

$$E_{\text{dell}} = \frac{1}{2} f(\varepsilon) q^\dagger \Phi^\text{sol}$$

The total free energy of the solvated molecule is the sum of the energy of the isolated system calculated with the solvated wave function and the dielectric energy

$$E = E(\Psi^{\text{sol}}) + E_{\text{dell}}.$$ 

A Cosmo energy calculation starts with the construction of the cavity surface grid. Within the SCF procedure, the screening charges are calculated in every cycle and the potential generated by these charges is included into the Hamiltonian. This ensures a variational optimization of both the molecular orbitals and the screening charges and allows for the evaluation of analytic gradients.

**Radii based Cavity Construction:** In order to ensure a sufficiently accurate and efficient segmentation of the molecular shaped cavity the COSMO implementation uses a double grid approach and segments of hexagonal, pentagonal, and triangular shape. The cavity construction starts with a union of spheres of radii $R_i + RSOLV$ for all atoms $i$. In order to avoid problems with symmetric species, the cavity construction uses de-symmetrized coordinates. The coordinates are slightly distorted with a co-sinus function of amplitude AMPRAN and a phase shift PHSRAN. Initially a basis grid with NPPA segments per atom is projected onto atomic spheres of radii $R_i + RSOLV$. In order to avoid the generation of points in the problematic intersections, all remaining points, which are not in the interior of another sphere, are projected downwards onto the radius $R_i$. In the next step a segment grid of NSPH segments per H atom and NSPA segments for the other atoms is projected onto the surface defined by $R_i$. The basis grid points are associated to the nearest segment grid centers and the segment coordinates are re-defined as the center of area of their associated basis grid points, while the segment area is the sum of the basis grid areas. Segments without basis grid points are discarded. In order to ensure nearest neighbor association for the new centers, this procedure is repeated once. At the end of the cavity construction the intersection seams of the spheres are paved with individual segments, which do not hold associated basis grid points.

**Density based Cavity Construction:** Instead of using atom specific radii the cavity can be defined by the electron density. In such an isodensity cavity construction one can use the same density value for all atoms types or the so-called scaled isodensity values. In the later approach different densities are used for the different atom types. The algorithm implemented in TURBOMOLE uses a marching tetrahedron algorithm for the density based
cavity construction. In order to assure a smooth density change in the intersection seams of atoms with different isodensity specification, this areas are smoothened by a radii based procedure.

**A-Matrix Setup:** The A matrix elements are calculated as the sum of the contributions of the associated basis grid points of the segments \( k \) and \( l \) if their distance is below a certain threshold, the centers of the segments are used otherwise. For all segments that do not have associated basis grid points, i.e. intersection seam segments, the segment centers are used. The diagonal elements \( A_{kk} \) that represent the self-energy of the segment are calculated via the basis grid points contributions, or by using the segment area \( A_{kk} \approx 3.8\sqrt{a_k} \), if no associated basis grid points exist.

**Outlying charge correction:** The part of the electron density reaching outside the cavity causes an inconsistency that can be compensated by the "outlying charge correction". This correction will be performed at the end of a converged SCF or an iterative MP2 calculation and uses an outer surface for the estimation of the energy and charge correction [191]. The outer surface is constructed by an outward projection of the spherical part of the surface onto the radius \( R_i + ROUTF \times RSOLV \). It is recommended to use the corrected values.

**Numerical Frequency Calculation:** The calculation of harmonic frequencies raises the problem of non-equilibrium solvation in the COSMO framework, because the molecular vibrations are on a time scale that do not allow a re-orientation of the solvent molecules. Therefore, the total response of the continuum is split into a fast contribution, described by the electronic polarization, and a slow term related to the orientational relaxation. As can be shown [192] the dielectric energy for the disturbed state can be written as

\[
E_{d\text{iel}}^d = \frac{1}{2} f(\varepsilon) q(P^0) \Phi(P^0) + \frac{1}{2} f(n^2) q(P^\Delta) \Phi(P^\Delta) + f(\varepsilon) q(P^0) \Phi(P^\Delta),
\]

where \( P^\Delta \) denotes the density difference between the distorted state and the initial state with density \( P^0 \). The interaction is composed of three contributions: the initial state dielectric energy, the interaction of the potential difference with the initial state charges, and the electronic screening energy that results from the density difference. The energy expression can be used to derive the correspondent gradients, which can be applied in a numerical frequency calculation. Because the COSMO cavity changes for every distorted geometry the initial state potential has to be mapped onto the new cavity in every step. The mapped potential of a segment of the new cavity is calculated from the distance-weighted potentials of all segments of the old cavity that fulfill a certain distance criterion. The mapped initial state screening charges are re-calculated from the new potential.

**Iterative MP2 COSMO:** For ab initio MP2 calculations within the CSM framework three alternatives can be found in the literature [193]. The first approach, often referred to as PTE, performs a normal MP2 energy calculation on the solvated HF wave function. The
response of the solvent, also called reaction field, is still on the HF level. It is the only of the three approaches that is formally consistent in the sense of second-order perturbation theory [194,195]. In the so-called PTD approach the vacuum MP2 density is used to calculate the reaction field. The third approach, often called PTED, is iterative so that the reaction field reflects the density of the first-order wave function. In contrast to the PTE approach the reaction field, i.e. the screening charges, change during the iterations until self consistency is reached. Gradients are available on the formally consistent PTE level only [196].

**Vertical excitations and Polarizabilities for TDDFT, TDA and RPA:** The escf program accounts for the Cosmo contribution to the excitation energies and polarizabilities. The Cosmo settings have to be defined for the underlying Cosmo dscf or ridft calculation. In case of the excitation energies the solvent response will be divided into the so-called slow and fast term [192,197]. The screening function of the fast term depends on the refractive index of the solvent which can be defined in the input. If only the Cosmo influence on the ground state should be taken into account we recommend to perform a normal Cosmo calculation (dscf or ridft) and to switch off Cosmo (i.e. deactivate $\text{cosmo}$) before the escf calculation.

**The Direct COSMO-RS method (DCOSMO-RS):** In order to go beyond the pure electrostatic model a self consistent implementation of the COSMO-RS model the so-called "Direct COSMO-RS" (DCOSMO-RS) [198] has been implemented in ridft and dscf.

COSMO-RS (COSMO for Real Solvents) [199,200] is a predictive method for the calculation of thermodynamic properties of fluids that uses a statistical thermodynamics approach based on the results of COSMO SCF calculations for molecules embedded in an electric conductor, i.e. using $f(\varepsilon) = 1$. The liquid can be imagined as a dense packing of molecules in the perfect conductor (the reference state). For the statistical thermodynamic procedure this system is broken down to an ensemble of pair wise interacting surface segments. The interactions can be expressed in terms of surface descriptors, e.g. the screening charge per segment area ($\sigma_t = q_t/a_t$). Using the information about the surface polarity $\sigma$ and the interaction energy functional, one can obtain the so-called $\sigma$-potential ($\mu_S(\sigma;T)$). This function gives a measure for the affinity of the system $S$ to a surface of polarity $\sigma$. The system $S$ might be a mixture or a pure solvent at a given temperature $T$. Because the parabolic part of the potential can be described well by the Cosmo model, we subtract this portion form the COSMO-RS potential:

$$\tilde{\mu}_S(\sigma;T) = \mu_S(\sigma;T) - (1 - f(\varepsilon))c_0\sigma^2.$$  

The parameter $c_0$ can be obtained from the curvature of a COSMO-RS $\sigma$-potential of a nonpolar substance, e.g. hexane. Thus, the remaining part of the chemical potential of a compound $i$ with mole fraction $x_i$ in the mixture $S_i$ can be expressed as:

$$\mu^i \approx \sum_{i=1}^{m} f_{pol} a_i \tilde{\mu}_S(\sigma;T) + \mu_{C,S}^i + kT \ln(x_i).$$
where the combinatorial term $\mu_{C,S}$ accounts for effects due to the size and shape differences of the molecules in the mixture and $a_t$ denotes the area of segment $t$. The $kT \ln(x_i)$ can be skipped for infinite dilution. The factor $f_{pol}$ has been introduced to account for the missing solute-solvent back polarization. The default value is one in the current implementation.

The free energy gained by the solvation process in the DCOSMO-RS framework is the sum of the dielectric energy of the Cosmo model and the chemical potential described above:

$$E_{\text{diel},RS} = \frac{1}{2} f(\epsilon) q^i \Phi^{\text{sol}} + \mu^i = E_{\text{diel}} + \mu^i$$

From the above expression the solvent operator $\hat{V}^{RS}$ can be derived by functional derivative with respect to the electron density:

$$\hat{V}^{RS} = -\sum_{t=1}^{m} \frac{f(\epsilon) q_t + q_t^{\Delta RS}}{|\mathbf{r}_t - \mathbf{r}|} = \hat{V}^{\text{cos}} - \sum_{t=1}^{m} \frac{q_t^{\Delta RS}}{|\mathbf{r}_t - \mathbf{r}|}.$$ 

Thus, the solvation influence of the COSMO-RS model can be viewed as a correction of the Cosmo screening charges $q_t$. The additional charges denoted as $q_t^{\Delta RS}$ can be obtained from $q_t^{\Delta RS} = -A^{-1} \Phi^{\Delta RS}$, where the potential $\Phi^{\Delta RS}$ arises from the chemical potential of the solute in the solvent:

$$\phi_t^{\Delta RS} = a_t \left( \frac{\delta \mu_S}{\delta q} \right)_{q=q_t}.$$ 

In order to get a simple and differentiable representation of the COSMO-RS $\sigma$-potential $\mu_S(\sigma; T)$, we use equally spaced cubic splines. An approximate gradient of the method has been implemented. DCOSMO-RS can be used in SCF energy and gradient calculations (geometry optimizations) with dscf, ridft, grad, and rdgrad. Please regard the restriction of the DCOSMO-RS energy explained in the keyword section 21.2.9. Because the DCOSMO-RS contribution can be considered as a slow term contribution in vertical excitations it does not have to be taken into account in response calculations. For the calculation of vertical excitation energies it is recommended to use the mos of a DCOSMO-RS calculation in a Cosmo response calculation (see above).

**Solvation effects on excited states using COSMO in ricc2:** The COSMO approach has been recently implemented into the ricc2 module of TURBOMOLE. It is now possible to equilibrate the solvent charges for any excited state or the ground state as mentioned in the MP2 section above. Using the methods CCS/CIS or ADC(2) the implementation is complete, for CC2 or higher methods, however it still has to be proven if there are terms missing. The recent implementation contains contributions to the off-diagonal elements of the one-electron density. Furthermore the energy contributions for non-equilibrated states can be calculated. Non-equilibrated means in this sense, that the slow part of the solvent charges (described by $f(\epsilon)$) are still equilibrated with a given initial state, while the fast electronic part of the solvent charges (described by $f(n^2)$) are in equilibrium with the target state. To handle this one has to do a macro iteration, like in MP2. This macro iteration can be managed with the script ’cc2cosmo’ which is the same as ’mp2cosmo’ but using ricc2 instead of rimp2 or mpgrad. To set up the basic settings one can reuse the
cosmoprep module, note that one has to specify the refractive index 'refind' when observing excited states. To specify the state to which the solvent charges should be equilibrated one inserts the keyword 'cosmorel state=(x)', where the ground state (x) is used normally but can be replaced by any requested excited state. Make sure to request relaxed properties for any desired state, otherwise the COSMO macro iteration will not work in ricc2. The off-diagonal contributions mentioned above can be switched off by setting the keyword 'nofast' in $cosmo. A typical input might look like:

```
$ricc2
  adc(2)
$excitations
  irrep=a' multiplicity= 1 nexc= 1 npre= 1 nstart= 1
  irrep=a" multiplicity= 1 nexc= 1 npre= 1 nstart= 1
  exprop relaxed states=all
$response
  fop relaxed
$cosmo
  epsilon= 50.000
  rsolv= 1.30
  refind= 3.0000
  cosmorel state=(a" 1)
  # nofast
$cosmo_correlated
$cosmo_atoms
...
```

This would deliver an excited state calculation for the lowest singlet $A'$ and $A''$ excitations using the ADC(2) method. The solvent charges are equilibrated to state $1^1A''$ and the non-equilibrium energy contributions for the MP2 ground state and the $1^1A'$ excited state are calculated furthermore. All contributions to the one-electron density are included since the proper keyword is commented out. Note: when doing solvent relaxations with the CCS/CIS model, no request of relaxed ground-state properties are needed, since the relaxed ground state is identical to the HF ground state. A short summary of the COSMO input is given at the beginning of the ricc2 output as well as a summary of the energy contributions is given at the end of the ricc2 output.
Chapter 21

Keywords in the control file

21.1 Introduction

The file control is the input file for TURBOMOLE which directly or by cross references provides the information necessary for all kinds of runs and tasks. control is usually generated by define, the input generator. This chapter provides a short-hand documentation: a list of the most important key words, the possible parameters for each keyword, default values, and a brief explanation.

21.2 Format of Keywords and Comments

TURBOMOLE input is keyword-directed. Keywords start with a ’$’, e.g. $title. Comments may be given after $dummy, or by a line starting with #; these lines are ignored by TURBOMOLE. Blank lines are also ignored. Keywords may be in any order unless stated otherwise below.

The sample inputs given below should help to give an idea how the keywords are to be used. They are sorted according to program. Complete control files are provided in Chapter 22.

An alphabetical list of all keywords is given in the index.

21.2.1 General Keywords

$operating system unix
$path
$lock off
$suspend off

The four keywords above are set by define, but are not necessary.
$statistics\; dscf \\
or \\
$statistics\; mpgrad

Only a statistics run will be performed to determine file space requirements as specified for\n\texttt{dscf} or \texttt{mpgrad}. On return the statistics option will be changed to \texttt{$statistics\; off}$.

\texttt{$actual\; step\; dscf$} \\
means \textit{current step}. Keyword and data group (as e.g. \texttt{dscf}) is set by every program and removed on successful completion.

\texttt{$last\; step\; relax$} \\
Keyword and data group (as e.g. \texttt{relax}) set by every program on successful completion.

General file cross-references:

\begin{verbatim}
$coord file=coord \\
$intdef file=coord \\
$user-defined\; bonds\; file=coord \\
$basis file=basis \\
$ecp file=ecp \\
$scfmo file=mos \\
$uhfmo_alpha file=alpha \\
$uhfmo_beta file=beta \\
$natural\; orbitals file=natural \\
$natural\; orbital\; occupation\; file=natural \\
$energy file=energy \\
$grad file=gradient \\
$forceapprox file=forceapprox 
\end{verbatim}

It is convenient not to include all input in the \texttt{control} file directly and to refer instead to other files providing the corresponding information. The above cross references are default settings from \texttt{define}; you may use other file names. \texttt{define} will create most of these files. Examples of these files are given below in the samples.

\texttt{$coord$ (and $intdef$ and $user-defined\; bonds$)} \\
contains atom specification—type and location—and the bonds and internal coordinates convenient for geometry optimizations.

\texttt{$basis$} \\
specification of basis sets.

\texttt{$ecp$} \\
specification of effective core potentials.

\texttt{$jbas$} \\
auxiliary (fitting) basis for the Coulomb terms in \texttt{ridft}.
21.2. FORMAT OF KEYWORDS AND COMMENTS

$\text{scfmo}, \text{uhfmo\_alpha}, \text{uhfmo\_beta}$
MO vectors of SCF or DFT calculations for RHF or UHF runs.

$\text{natural orbitals}, \text{natural orbital occupation}$
keywords and data groups set by unrestricted dscf or ridft runs. Contain natural MO vector and orbital occupation.

$\text{energy}, \text{grad}$
energies and gradients of all runs, e.g. for documentation in a geometry optimizations.

$\text{forceapprox}$
approximate force constant for geometry optimizations.

The control file must end with this keyword:
$\text{end}$

21.2.2 Keywords for System Specification

General information defining the molecular system: nuclear coordinates, symmetry, basis functions, number of occupied MOs, etc. which are required by every module.

$\text{title}$
give title of run or project here.

$\text{symmetry d4h}$
Schönflies symbol of the point group. All point groups are supported with the exception of NMR shielding and force constant calculations etc. which do not work for groups with complex irreps ($C_3$, $C_3h$, $T$, etc). Use a lower symmetry group in this case.

$\text{atoms}$
Example:

\begin{verbatim}$atoms
cu 1-4
  basis =cu ecp-18 arep
  jbas =cu ecp-18
  ecp =cu ecp-18 arep
se 5-6
  basis =se ecp-28 arep dzp
  jbas =se ecp-28
  ecp =se arep
\end{verbatim}

note the backslash \ : this is necessary. For each type of atom, one has to specify
- the basis set
- and the auxiliary (fitting) basis for RIDFT calculations
- the ECP if this is used.

The files basis, ecp and jbas must provide the necessary information under the labels specified in \$atoms.

\$pople char

This data group specifies the number of Cartesian components of basis functions (i.e. 5d and 7f in AO-Basis, 6d and 10f in CAO-Basis) for which the SCF calculation should be performed. Possible values for char are AO (default) or CAO. If CAO is used—which is not recommended—a core guess must be used instead of a Hückel guess (see \$scfmo).

RHF

\$closed shells

Specification of MO occupation for RHF, e.g.

\[
\begin{align*}
adg & \quad 1-4 \\
a2g & \quad 1
\end{align*}
\]

(2)

\$open shells type=1

MO occupation of open shells and number of open shells. type=1 here means that there is only a single open shell consisting e.g. of two MOs:

\[
\begin{align*}
b2g & \quad 1 \\
b3g & \quad 1
\end{align*}
\]

(1)

\$roothaan

A = 1  B = 2

\$roothaan

Roothaan parameters for the open shell, here a triplet case. define recognizes most cases and suggests good Roothaan parameters.

For further information on ROHF calculations, see the sample input in Section 22.6 and the tables of Roothaan parameters in Section 6.3.

UHF

\$uhf directs the program to carry out a UHF run, e.g.

\$alpha shells

\[
\begin{align*}
adg & \quad 1-4 \\
a2g & \quad 1
\end{align*}
\]

(1)

\$beta shells

\[
\begin{align*}
adg & \quad 1-4 \\
a2g & \quad 1
\end{align*}
\]

(1)
21.2. FORMAT OF KEYWORDS AND COMMENTS

The specification of MO occupation for UHF, $\text{uhf}$ overwrites closed-shell occupation specification.

21.2.3 Keywords for redundant internal coordinates in $\text{redund\_inp}$

With the parameters in $\text{redund\_inp}$ the generation of redundant internal coordinates can be modified. All entries have to be made in the control file before invoking the $\text{ired}$ option. Important options are:

\text{iprint} \ n

print parameter for debug output: The larger $n$ is, the more output is printed
$n \geq 0, n \leq 5$ (default: 0)

\text{metric} \ n

method for generating and processing of redundant internal coordinates
$n \geq -3, n \leq 3, n \neq 0$ (default: 3)

Values for the \text{metric} option:

\begin{itemize}
  \item $n = 1$ “Delocalized Coordinates”
    The $\text{BmB}^t$ matrix is diagonalized for the complete set of redundant internal coordinates, matrix $\text{m}$ is a unit matrix.
  \item $n = -3$ Delocalized Coordinates obtained with a modified matrix $\text{m}$, the values of $\text{m}$ can be defined by user input (see below).
  \item $n = -1$ “Hybrid Coordinates”
    Natural internal coordinates are defined as in the old $\text{iaut}$ option. If a cage remains, delocalized coordinates (as for $n=1$) are defined for the cage.
  \item $n = -2$ Very similar to the $n = 1$ option, but for the remaining cage delocalized coordinates with modified matrix $\text{m}$ are defined as for $n = -3$.
  \item $n = 2$ “Decoupled coordinates”
    The redundant coordinates are divided into a sequence of blocks. These are expected to have decreasing average force constants, i.e. stretches, angle coordinates, torsions and “weak” coordinates. The $\text{BB}^t$ matrix is diagonalized for each block separately after the columns of $\text{B}$ were orthogonalized against the columns of $\text{B}$ of the preceding blocks.
  \item $n = 3$ “Generalized natural coordinates”
    Natural internal coordinates are defined first, for the remaining cage decoupled coordinates are defined.
\end{itemize}

\text{type} \ r

a positive real number, which is an approximate “force constant”, can be read in for each type of coordinate (see below). The force constants are used for the definition of the matrix $\text{m}$ in $\text{BmB}^t$. 
Types of internal coordinates for the definition of m

The matrix $m$ is assumed to be a diagonal matrix. For each type of coordinate a different value for the force constants $m_{ii}$ can be read in. Types of coordinates are:

- **stre**: bond stretch (default: 0.5)
- **invr**: inverse bond stretch (default: 0.5)
- **bend**: bond angle (default: 0.2)
- **outp**: Out of plane angle (default: 0.2)
- **tors**: dihedral or “torsional” angle (default: 0.2)
- **linc**: Special angle coordinate for collinear chains, bending of the chain a–b–c in the plane of b–c–d (default: 0.2)
- **linp**: bending of the chain a–b–c perpendicular to the plane of b–c–d (default: 0.2)
- **wstr**: stretch of a “weak” bond, i.e. the bond is assumed to have a very low force constant, e.g. a “hydrogen bond” or a “van der Waals bond” (default: 0.05)
- **winv**: inverse stretch of a weak bond (default: 0.05)
- **wbnd**: bond angle involving at least one weak bond (default: 0.02)
- **wout**: Out of plane angle for weak bonds (default: 0.02)
- **wtor**: dihedral angle for weak bonds (default: 0.02)
- **wlinc**: linc coordinate for weak bonds (default: 0.02)
- **wlnp**: linp coordinate for weak bonds (default: 0.02)

### 21.2.4 Keywords for Module uff

One has to specify only the Cartesian coordinates (data group $coord$) to start a uff run. The program uff requires the data groups $uff$, $ufftopology$, $uffgradient$ and $uffhessian$. If these keywords do not exist in the control file the program will generate these data groups.

The data group $uff$ contains the parameters described below. The default values—in the control file—are:

```
1 1 0  ! maxcycle,modus,nseq
1111 1 ! iterm
0.10D-07 0.10D-04  ! econv,gconv
 0.00 1.10  ! qtot,dfac
 0.10D+03 0.10D-04 0.30  ! epssteep,epssearch,dqmax
```
The explanation of the variables are as follows:

**maxcycle**
number of max. optimization cycles (maxcycle=1: single-point calculation).

**modus**
can have the values +1 or -1. If modus = -1 only the topology will be calculated.

**nqeq**
each nqeq cycle the partial charges will be calculated. If nqeq = 0, then the partial charges are calculated only in the first cycle, if the file ufftopology does not exist.

**iterm**
switch for the different types of force field terms:

- 100000: bond terms will be calculated.
- 010000: angle terms will be calculated.
- 001000: torsion terms will be calculated.
- 000100: inversion terms will be calculated.
- 000010: non bonded van der Waals terms will be calculated.
- 000001: non bonded electrostatic terms will be calculated.

**econv, gconv**
convergence criteria for energy and gradient.

**qtot**
total charge of the molecule.

**dfac**
distance parameter to calculate the topology. If the distance between the atoms I and J is less than the sum of the covalent radii of the the atoms multiplied with dfac, then there is a bond between I and J.

**epssteep**
if the norm of the gradient is greater than epssteep, a deepest-descent-step will be done.

**epssearch**
if the norm of the gradient is smaller than epssearch, no line-search step will be done after the Newton step.

**dqmax**
max. displacement in a.u. for a coordinate in a relax step.

**mxls, dhls, ahls**
parameters of linesearch:

- ahls: start value
CHAPTER 21. KEYWORDS IN THE CONTROL FILE

\textit{dhls} \quad \text{increment} \\
\textit{mxls} \quad \text{number of energy calculations} \\

\textit{alpha}, \textit{beta}, \textit{gamma} \\
modification parameter for the eigenvalues of the Hessian (see below): \( f(x) = x \ast (\alpha + \beta \ast \exp(-\gamma \ast x)) \).

\textit{transform} \\
a switch for the transformation in the principal axis system.

\textit{lnumhess} \\
a switch for the numerical Hessian.

\textit{lmd} \quad \text{a switch for a MD calculation.}

\textbf{Input Data Blocks Needed by Uff}

\texttt{$coord$} \\
cartesian coordinates of the atoms (default: \texttt{$coord$ file=coord})

\texttt{$ufftopology$} \\
contains a list of the next neighbours of each atom (see Section 21.2.4). Sometimes it is useful to enter the connectivity (in the input block \texttt{nxtnei12} in the file \texttt{ufftopology}) by hand (not always necessary; default: \texttt{$ufftopology$ file=ufftopology}).

Beyond this \texttt{uff} reads the force field parameters for the atoms from the file \texttt{parms.in}. If this file exists in the directory from which one starts an \texttt{uff} calculation the program will use this file, if not the program reads the data from the file \texttt{$TURBODIR/uff/parms.in}. If one wants own atom types, one has to add these atom types in the file \texttt{parms.in}. For each new atom type one has to specify the \textit{natural} bond distance, the \textit{natural} bond angle, the \textit{natural} non-bond distance, the well depth of the Lennard-Jones potential, the scaling factor \( \zeta \), the effective charge, torsional barriers invoking a pair of sp\(^3\) atoms, torsional barriers involving a pair of sp\(^2\) atoms, generalized Mulliken–Pauling electronegativities, the idem potentials, characteristic atomic size, lower bound of the partial charge, upper bound of the partial charge. Distances, energies and charges are in atomic units and angles are in rad.

\textbf{Uff Output Data Blocks}

\texttt{$coord$} \\
contains the (updated) cartesian coordinates of the atoms (default: \texttt{$coord$ file=coord}).

\texttt{$ufftopology$} \\
contains the full information of the topology of the molecule and the whole force field terms (see below; default: \texttt{$ufftopology$ file=ufftopology}).

\texttt{$uffgradient$} \\
contains the accumulated cartesian analytical gradients (default: \texttt{$uffgradient$ file=uffgradient}).
21.2. FORMAT OF KEYWORDS AND COMMENTS

$uffhessian
contains the cartesian analytical Hessian;
(default: $uffhessian file=uffhessian0-0).

The file $ufftopology

The topology file $ufftopology contains the blocks nxtnei12, nxtnei13, nxtnei14, connectivity, angle, torsion, inversion, nonbond and qpartial. It starts with $ufftopology and ends with $end. The first three blocks (nxtnei12, nxtnei13, nxtnei14) have the same form: they start with the atom number and the number of its neighbours, in the next line are the numbers of the neighbour atoms. Then the connectivity-block follows starting with the number of bond terms. Each line contains one bond term:

\[ I \quad J \quad d \quad BO. \]

Here are \( I \) and \( J \) the number of the atoms, \( d \) the distance in a.u. and BO is the bond order.

The angle terms follow, starting with the number of the angle terms. In each line is one angle term:

\[ J \quad I \quad K \quad wtyp \quad \theta \quad nr_{JI} \quad nr_{IK}. \]

Here are \( J, I \) and \( K \) the atoms number, where atom \( I \) is in the apex. “wtyp” is the angle type and has the following values:

- wtyp = 1 linear case
- wtyp = 2 trigonal planar case
- wtyp = 3 quadratic planar case
- wtyp = 6 octahedral case
- wtyp = 9 all other cases.

\( \theta \) is the angle value in degree. \( nr_{JI} \) and \( nr_{IK} \) are the number of the bonds between \( J \) and \( I \) and the bond between \( I \) and \( K \). The hybridization of atom \( I \) determines “wtyp”.

Then the torsion terms follow, starting with the number of the torsion terms. Each line contains one torsion term:

\[ I \quad J \quad K \quad L \quad nr_{JK} \quad ttyp \quad \phi \quad \theta_{IJK} \quad \theta_{JKL}. \]

Here are \( I, J, K \) and \( L \) the atom numbers. \( nr_{JK} \) is the number of the bond between \( J \) and \( K \). “ttyp” is the torsion type:

- ttyp = 1 \( J (sp^2)-K (sp^3) \)
- ttyp = 11 like ttyp=1, but one or both atoms are in Group 16
- ttyp = 2 \( J (sp^2)-K (sp^3) \) or vice versa
- ttyp = 21 like ttyp=2, but one or both atoms are in Group 16
ttyp = 22 like ttyp=2, but J or K is next a sp² atom
ttyp = 3 J (sp²)–K (sp²)
ttyp = 9 all other cases.

ϕ is the value of the torsion angle in degree. θ_{IJK} is the angle value of (I − J − K) and θ_{JKL} is the c.w. one for J − K − L. The hybridizations of J and K determine “ttyp”.

The inversion terms follow starting with the number of inversion terms (e.g. the pyramidalisation of NH₃). In each line is one inversion term:

\[ I \quad J \quad K \quad L \quad \text{ityp1} \quad \text{ityp2} \quad \text{ityp3} \quad \omega_1 \quad \omega_2 \quad \omega_3. \]

I, J, K and L are the atom numbers. Atom I is the central one. ityp1, ityp2, ityp3 are the types of the inversions:

ityp = 10 atom I is C and atom L is O
ityp = 11 like ityp=10, but L is any atom
ityp = 2 I is P
ityp = 3 I is As
ityp = 4 I is Sb
ityp = 5 I is Bi
ityp = 9 all other cases.

ω₁, ω₂ and ω₃ are the values of the inversion angles in degree.

The nonbond terms follow starting with the number of the non-bonded terms. In each line is one nonbond term:

\[ I \quad J \quad d. \]

Here I and J are the atom numbers, d the distance in a.u. Then the partial charges follow.

If the determination of the molecule connectivity failed, you can specify the block nxtnei12 in the file ufftopology. Then the program calculates the other blocks.

Based on the numbers of the next neighbours (block nxtnei12 in the file ufftopology) the program tries to determine the UFF type of an atom. The following rules are implemented: If the atom has three next neighbours and it is in the nitrogen group, then it has a hybridization three. If it is not in the nitrogen group, it has hybridization two. If the atom has four next neighbours and it is in the carbon group, it has hybridization three. If it is not in the carbon group, it becomes hybridization four. If the number of next neighbours is six, then it gets the hybridization six.

Since the smallest eigenvalues λᵢ of the Hessian has the greatest influence on the convergence of the geometry optimization, one can shift these values with

\[ \tilde{\lambda}_i = \lambda_i \cdot (\alpha + \beta \cdot e^{-\gamma x}) \]

and calculates a new Hessian with these modified eigenvalues.
21.2.5 Keywords for woelfling

Module WOEFLING reads options from data group $woelfling$

The below values of the options are default values with the following meaning:

- $\texttt{ninter} \quad 14$
  
  Number of interpolated structures for optimization.

- $\texttt{ncoord} \quad 2$
  
  Number of input structures provided by user.

- $\texttt{align} \quad 0$
  
  Align input structures by translation/rotation $0=\text{yes}, 1=\text{no}$.

- $\texttt{maxit} \quad 40$
  
  Maximum number of iterations.

- $\texttt{dlst} \quad 3.00000000000000$
  
  Threshold for accuracy of LST-interpolation.

- $\texttt{thr} \quad 1.00000000000000E-004$
  
  Threshold for mean of norms of projected gradients.

- $\texttt{method} \quad \text{q}$
  
  Use standard optimization from initial LST-path ($\text{method q}$) or grow reaction path ($\text{method qg}$).

Furthermore,

- $\texttt{riter} \quad 0$
  
  counts the number of completed iteration (no option).
21.2.6 Keywords for Modules dscf and ridft

$\text{denconv} \ text{real}$

Convergency criterion for the root mean square of the density matrix. If you want to
calculate an analytical MP2 gradient (program mpgrad) real should be 1.d-7 or less.

$\text{dft} \ text{options}$

Listing of all possible sub-keywords for $\text{dft}$ (cross-references are given).
The user normally has to choose only the functional and the grid size, see below. All
other parameters have proven defaults.

**functional name**

For all possible—and useful—functionals, please refer to page 310 and for def-
inition of the functionals the section 6.2 on page 128.

Example (default input):

```
$dft$
  functional b-p
```

**gridsize integer or minteger**

Specification of the spherical grid (see section 21.2.6 on page 310). Default is
gridsize m3.

Example:

```
$dft$
  gridsize m3
```

**gridtype integer** —not recommended for use—

Specification of the mapping of the radial grid.
Possible values for gridtype are 1,...,6, but gridtype 4 to 6 is only for the
use with functional lhf (see page 313). For the definition of gridtype 1 –
3, please refer to Eq. (16), (17) and, (19) in Ref. [201].

Example (default value):

```
$dft$
  gridtype 3
```

**debug integer**

Flag for debugging. debug 0 means no debug output (default), debug 1 means
some output, debug 2 means a lot more output. Be careful!

**nkk integer**

Specification of the sharpness of the partition function as proposed by Becke
[202], default is nkk 3. The usage of nkk makes sense only in the range 1 ≤
nkk ≤ 6.

Example:
21.2. FORMAT OF KEYWORDS AND COMMENTS

\$dft
  nkk 3

\texttt{ntheta} \texttt{integer} —not recommended for use—
\texttt{nphi} \texttt{integer}

Only for user-specified Lobatto grids, i.e. \texttt{gridsize 9}, \texttt{ntheta} specifies the number of $\theta$ points and \texttt{nphi} specifies the number of $\phi$ points. For the fixed Lobatto grid, i.e. \texttt{gridtype 8}, the default value is \texttt{ntheta 25} and \texttt{nphi 48}.

When \texttt{gridsize 9} is given, you have to specify both, \texttt{ntheta} and \texttt{nphi} (see below), otherwise the program will crash. The restriction for user-defined Lobatto grids is the number of grid points, which must not exceed 2000 grid points.

Example:

\$dft
  gridsize 9
  ntheta 30
  nphi 60

\texttt{old_RbCs_xi}

Original grids had not been carefully optimized for all atoms individually. This has now been done, which let to changes of $\xi$ for Rb and Cs only resulting in minor improvements. If you have ongoing projects, which have been started with the old grids, you should continue using them with the keyword \texttt{old_RbCs_xi}.

Example:

\$dft
  old_RbCs_xi

\texttt{radsize} \texttt{integer}

Specifies the number of radial grid points. Default values depend on type of atom and grid (see keyword \texttt{gridsize}). The formula for the radial gridsize is given as,

\[
\text{number of radial grid points} = \text{ioffrad} + (\text{radsize} - 1) \times 5 .
\]

\texttt{ioffrad} is atom-dependent, the more shells of electrons, the larger ioffrad:

<table>
<thead>
<tr>
<th>elements</th>
<th>ioffrad</th>
<th>elements</th>
<th>ioffrad</th>
</tr>
</thead>
<tbody>
<tr>
<td>for H–He</td>
<td>20</td>
<td>for K–Kr</td>
<td>40</td>
</tr>
<tr>
<td>for Li–Ne</td>
<td>25</td>
<td>for Rb–Xe</td>
<td>45</td>
</tr>
<tr>
<td>for Na–Ar</td>
<td>30</td>
<td>for Cs–Lw</td>
<td>50</td>
</tr>
</tbody>
</table>

The radial grid size increases further for finer grids:

<table>
<thead>
<tr>
<th>\texttt{gridsize}</th>
<th>1 2 3 4 5 6 7 8 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{radsize}</td>
<td>1 2 3 6 8 10 14 9 3</td>
</tr>
</tbody>
</table>

If you want to converge results with respect to radial grid size, increase \texttt{radsize} in steps of 5, which is convenient (see equation above).
**diffuse integer**

Serves to increase quadrature grids; this is recommended in case of very diffuse wavefunctions. With the keyword `diffuse` grids are modified by changing the linear scaling factor $\xi$ of radial grid points and the radial gridsize:

$$\text{radsize} \rightarrow \text{radsize} + \text{incr}$$

$$\xi \rightarrow \xi \ast \text{scal}$$

<table>
<thead>
<tr>
<th>diffuse integer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>incr</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>scal</td>
<td>1.5</td>
<td>2.0</td>
<td>2.8</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
</tr>
</tbody>
</table>

For information about the linear scaling parameter $\xi$, see Eq. (16)–(19) and Table 1 in Ref. [201].

In addition, the reduction of spherical grid points near nuclei is suppressed, i.e. `fullshell on` is set (see page 299).

Note: the keyword `radsize integer` overrules the setting of `incr`; for more information, see p. 297.

Recommendation: For diffuse cases use `gridsize m4` (or larger) in combination with `diffuse 2` and check the number of electrons; for more difficult cases use `diffuse 4`. In case of doubt, verify the calculation with a larger grid, i.e. `gridsize 7`.

The test suite example `$TURBODIR/TURBOTEST/dscf/short/H3PO4.DSCF.DIFFUSE` provides an example of usage; this also gives reasonable values for damping and orbital shift to reach convergence in this and similar cases, see `$scfdamp` and `$scforbitalshift` (p. 304 and p. 307).

Example (Recommendation):

```
$df
  gridsize m4
  diffuse 2
  rhostart integer —for developers only—
  rhostop integer
```

Radial grid points have a linear scaling parameter $\xi$, see Eq.(16)–(19) and Table 1 in Ref. [201]. With the following input,

```
$df
  rhostart 50
  rhostop 200
```

one performs a numerical integration for the density and the exchange correlation term for $\xi = 0.5, (0.01), 2.0$ for given MOs and functional. NOTE: only molecules with a single atom type can be used. The results serve to establish
stable, optimal $\xi$ values, see Figure 1 in Ref. [201]. Program stops after this testing.

**reference**

Usage of the reference grid, which is a very fine grid with very tight thresholds. The default values for the different variables are:

- gridsize 7
- radsize 14
- fullshell 1
- dgrenze 16
- fgrenze 16
- qgrenze 16
- fcut 16

Please refer to the corresponding sub-keywords for explanation.

If you want to use the reference grid, you have to skip the keyword `gridsize`, and type instead `reference`. Example:

```
$df
   functional b-p
   reference
```

**test-integ**

Check if the selected grid is accurate enough for the employed basis-set by performing a numerical integration of the norm of all (occupied and virtual) orbitals. Useful for LHF. 313.

**batchsize integer**

Grid points are sorted into batches, which are then processed. This increases efficiency. This should be changed only by developers. Default is `batchsize 100`.

**fullshell**

Standard grids have reduced number of spherical grid points near nuclei. With the keyword `fullshell` this reduction is suppressed. Reference grid (see keyword `reference`) always has full spherical grids with 1202 points. Should be used to checked the influence of spherical grid reduction.

Example for the usage of `fullshell`:

```
$df
   functional b-p
   gridsize m4
   fullshell
```

**symblock1 real** —for developers only—

**symblock2 real**

Values of real effects efficiency of the quadrature, default is `symblock1 0.001` and `symblock2 0.001`, one can try higher or smaller values.
xparameter integer — not recommended for use —
Where xparameter (default) can be: sgrenze (8), fgrenze (10), qgrenze (12),
dgrenze (12) and, fcut (14). These parameters control neglect of near zeros of
various quantities. With xparameter integer one changes the default. integer
larger than defaults will increase the numerical accuracy. Tighter threshold are
set automatically with keyword $scfconv (see section 21.2.6 on page 304).

weight derivatives
Includes the derivatives of quadrature weights to get more accurate results.
Default is that the derivatives of quadrature weights will be not considered, see
section 21.2.10 on page 328.

gridordering
Grid points are ordered into batches of neighbouring points. This increases
efficiency, since now zeros can be skipped for entire batches. gridordering
is default for serial version, not for the parallel one. You cannot use weight
derivatives and gridordering together.
Example for switching off gridordering:

$df
   gridordering 0

$electrostatic field
Specification of external electrostatic field(s). The specification may take place either
by Ex, Ey, Ez or by x, y, z, |E|. See also $fldopt.
Example:

$electrostatic field
   0.1000E-03 0.000 0.000

$fermi tmstrt=<300.0> tmend=<100.0> tmfac=<0.9> hlcrt=<1.0E-01> stop=<1.0E-03> nue=<N>

Requests calculation of occupation numbers at a finite temperature T. For an or-
bital with the energy εi, the occupation number ni ∈ [0, 1] is calculated as

\[ n_i = \frac{1}{2} \text{erfc} \left( \frac{\varepsilon_i - \mu}{fT} \right), \]

where \( \mu \) is the Fermi level. The factor \( f = 4k/\sqrt{\pi} \) is chosen to yield the same slope
at the Fermi level as the Fermi distribution.

Calculation of the fractional occupation numbers starts when the current HOMO-
LUMO gap drops below the value given by hlcrit (default: 0.1). The initial tem-
perature given by tmstrt (default: 300 K) is reduced at each SCF cycle by the factor
tmfac (default: 1.0) until it reaches the value tmend (default: 300 K). Note that the
default values lead to occupation numbers calculated at a constant \( T = 300 \text{ K} \). Cur-
rent occupation numbers are frozen if the energy change drops below the value given
by stop (default: 1·10^-3). This prevents oscillations at the end of the SCF procedure.
Calculation of fractional occupation numbers often requires much higher damping and orbital shifting. Please adjust the values for $scfdamp$ and $scforbitalshift$ if you encounter convergence problems.

In UHF runs this option can be used to automatically locate the lowest spin state. In order to obtain integer occupation numbers $tmend$ should be set to relatively low value, e.g. 50 K.

Calculation of fractional occupation numbers should be used only for single point calculations. When used during structure optimizations it may lead to energy oscillations.

The optional $nue$ value (number of unpaired electrons) can be used to force a certain multiplicity in case of an unrestricted calculation. $nue=0$ is for singlet, $nue=1$ for doublet, etc.

The option $addTS$ adds the entropic contribution of the smearing to the total energy which is important for very high temperatures only.

Finally the option $noerf$ will use the full correct Fermi statistics rather than the term given above.

$firstorder$
Perform first-order SCF-calculation, i.e. perform only one SCF-iteration with the start MOs (which should be the orthogonalized MOs of two independent subsystems as is explained in detail in Chapter 17).

$fdopt$ options
Specification of options related with external electrostatic fields. The following options are available:

1st derivative on/off
Calculate numerical 1st derivative of SCF energy with respect to electrostatic field (default: off), increment for numerical differentiation is $edelt$ (see below).

2nd derivative on/off
Calculate numerical 2nd derivative of SCF energy with respect to electrostatic field (default: off), increment for numerical differentiation is $edelt$.

$edelt$ = real
Increment for numerical differentiation (default: 0.005).

fields on/off
Calculate SCF energy for non-zero external electrostatic fields defined in $electrostatic$ field.

geofield on/off
Calculate SCF energy for one external field definition and dump disturbed MOs onto $scfmo$. This enables to evaluate properties or perform geometry optimizations in the presence of an external field.

Caution: don’t use the RI approximation for all these calculations since this will lead to non-negligible errors!!
$\text{incore \hspace{0.5em} integer}$

By using this option the two-electron integrals are kept in RAM; $\text{integer}$ specifies how many megabytes should be allocated. If the integrals exceed the RAM allocated the program reverts to the standard mode. Supports all methods which process two-electron integrals, i.e. SCF and DFT (including hybrid functionals); RHF and UHF.

The following condition must be met:

$\text{scfdenapprox} \hspace{0.5em} 1$

and rhfshells 1 or 2. It is advisable to set $\text{thize}$ as small as possible (e.g. $\text{thize} \hspace{0.5em} 0.1d-08$) and to remove the keyword $\text{scfdump}$.

**Note:** this keyword does not work for parallel runs.

$\text{mo-diagram \hspace{0.5em} only \hspace{0.5em} nirreps=integer}$

If this keyword is set the energies and symmetry labels of all occupied MOs will be dumped to this data group. This may be helpful to draw mo-diagrams. If $\text{only}$ has been set only the start MOs are dumped and the program quits.

$nirreps$ will hold the total number of displayed orbitals after the successful run.

$\text{mom}$

This keyword enables the use of the maximum overlap method in unrestricted ridft calculations to access excited states self-consistently. [203]

$\text{moprint}$

If this keyword is present all occupied orbitals are dumped to standard output. Be careful about this option as it can create huge output files in case of many basis functions.

$\text{mo output format \hspace{0.5em} format}$

If this line is present, the dscf program is forced to output the MOs using the new FORTRAN format $\text{format}$ regardless of the $\text{format}$-option in data group $\text{scfmo}$. Otherwise the input format will be used.

Example: $\text{mo output format}(3(2x,d15.8))$

$\text{natural orbitals}$

This data group will be written after an UHF calculation (together with $\text{natural \hspace{0.5em} orbital \hspace{0.5em} occupation}$) and contains the natural space orbitals (same syntax as $\text{scfmo}$).

$\text{natural orbital occupation}$

This data group will be written after an UHF calculation (together with $\text{natural \hspace{0.5em} orbitals}$) and contains the occupation of natural orbitals (syntax as any data group related with orbital occupation information, e.g. $\text{closed \hspace{0.5em} shells}$), e.g.

\[
\begin{align*}
\text{a} & \quad 1-5 & \text{ (} & \quad 2.00000000000000 \text{ )} \\
\text{a} & \quad 6 & \text{ (} & \quad 1.99949836999366 \text{ )} \\
\text{a} & \quad 7 & \text{ (} & \quad 1.99687490286069 \text{ )}
\end{align*}
\]
21.2. FORMAT OF KEYWORDS AND COMMENTS

\$point\_charges

Specification of position and magnitude of point charges to be included in the Hamiltonian. Each point charge is defined in the format

\(<x> <y> <z> <q>\)

with \(<x>, <y>, <z>\) being the coordinates and \(<q>\) its charge, e.g.

\$point\_charges thr=<real> selfenergy nocheck list

2. 2. 2. 5.
5. 0. 0. 2.5

In addition the following optional arguments may be given:

\thraeal

distance threshold for discarding redundant point charges, default value \(10^{-6}\).

\selfenergy

if given, the selfenergy of the point charge array will will be included in the energy and the gradient

\nocheck

switches off the check for redundant point charges and the default symmetrization. This option can significantly speed up the point charge treatment if many of them are involved - use only if the point charges are well distributed and symmetry is \(C_1\), e.g. when they stem from proper MM runs

\list

print all point charges in the output (default is to print the point charges only if less than 100 charges given)

\$prediag

concerns the first SCF iteration cycle if start MOs from an EHT guess are used.

The SCF iteration procedure requires control mechanisms to ensure (fast) convergence, in TURBOMOLE these are based on orbital energies \(\epsilon_i\) of the preceeding iteration used for level shifting and damping (besides DIIS, see below). This feature cannot be used in the first iteration if EHT MOs are employed as start, since \(\epsilon_i\) are not available. The keyword \$prediag provides \(\epsilon_i\) of the zeroth iteration' by diagonalization of occ–occ and virt–virt part of the first Fock matrix, to allow for level shifting etc.. See \$scfdiis below.

\$restart dscf twoint

Try a dscf restart. The program will read the data group \$restartd (which must exist, also \$scfmo has to exist!) and continue the calculation at the point where it
ended before. If the additional option \texttt{twoint} is appended, the program will read the two-electron integrals from the files specified in \texttt{$scfintunit$}, so there will be almost no loss of cpu-time.

All this information is normally provided by the previous \texttt{dscf} run if the keyword \texttt{$scfdump$} (see there) was given.

\texttt{$restartd$ data}

Data provided by a previous \texttt{dscf} run that has been interrupted. This keyword is created when \texttt{$scfdump$} was given.

\texttt{$rundimensions$ data}

is set by \texttt{define} so usually no changes are necessary. The dimensions must be greater or equal to those actually required, i.e. you can delete basis functions and keep rundimensions. This keyword is not necessary for small cases.

Example:

\begin{verbatim}
  dim(fock,dens)=6072
  natoms=6
  nshell=34
  nbf(CAO)=108
  nbf(AO)=98
  dim(trafo[SAO<-->AO/CAO])=256
  rhfshells=1
\end{verbatim}

\texttt{$scfconv$ integer}

SCF convergency criterion will be $10^{-\text{integer}}$ for the energy. Gradients will only be evaluated if \texttt{integer} $>$ 6.

\texttt{$scfdamp$ start=<.500> step=<.050> min=<.100>}

Damping parameters for SCF iterations in order to reduce oscillations. The old Fock-operator is added to the current one with weight 0.5 as \texttt{start}; if convergence is good, this weight is then reduced by the \texttt{step} 0.05 in each successive iteration until the \texttt{minimum} of 0.1 is reached. (These are the default settings of \texttt{define} for closed-shell RHF). DSCF automatically tries to adjust the weight to optimize convergence but in difficult cases it is recommended to start with a large weight, e.g. 1.5, and to set the minimum to a larger value, e.g. 0.5.

\texttt{$scfdebug$ options}

Flags for debugging purposes. Following options are available:

\texttt{vectors integer}

Output level concerning molecular orbitals. \texttt{integer}=0 (default) means minimal output, $>$1 will output all start MOs and all MOs in each iteration.

\texttt{density integer}

Output level concerning difference density matrices.

\texttt{debug integer}

\texttt{integer} $>$ 0 will dump a lot of information—be careful!
$\textit{scfdenapproxl}$ integer

Direct SCF procedures build the Fock matrix by exploiting information from previous iterations for better efficiency. With this keyword information from the last integer iterations will be used. This feature is switched on with the default value 20, even if the keyword is absent. The user may reduce the number of iterations employed to smaller values (e.g. 10) in cases were numerical stability could become an issue. With the value 0 this feature is switched off; the Fock matrix is constructed from scratch in each iteration.

$\textit{scfdiis}$ options

Control block for convergence acceleration via Pulay’s DIIS $^P$.

Options are:

\texttt{errvec=} \texttt{char} specifies the kind of error vector to be used (two different kind of DIIS algorithms)

\begin{itemize}
  \item \texttt{char=} ‘FDS’ or ‘SDF’ or ‘FDS–SDF’
    \begin{itemize}
    \item uses (FDS – SDF) as error vector.
    \end{itemize}
  \item \texttt{char=} none
    \begin{itemize}
    \item no DIIS
    \end{itemize}
  \item \texttt{char=} sFDs
    \begin{itemize}
    \item use $S^{-1/2}FDS^{1/2}$ – transposed
    \end{itemize}
\end{itemize}

Further suboptions:

\texttt{maxiter=} \texttt{integer}

maximum number of iterations used for extrapolation.

\texttt{debug=} \texttt{integer}

debug level (default: 0)

\begin{itemize}
  \item \texttt{integer=} 1 print applied DIIS coefficients
  \item \texttt{integer=} 2 print DIIS matrix and eigenvalues, too
\end{itemize}

\texttt{qscal=} \texttt{real}

scaling factor in DIIS procedure: $\texttt{qscal} > 1$ implies some damping, $\texttt{qscal} = 1.0$: straight DIIS.

\texttt{thrd=} \texttt{real}

directs the reduction of qscal to $\texttt{qscal} = 1.0$ (no damping in DIIS), done if $||\texttt{errvec}|| < \texttt{thrd}$.

Defaults for $\textit{sprediag}$ (see above) and $\textit{scfdiis}$

\begin{itemize}
  \item $\texttt{errvec=} \texttt{FDS–SDF}$, $\texttt{maxiter=} 5$, $\texttt{qscal=} 1.2$, $\texttt{thrd=} 0.0$, this implies DIIS damping in all iterations, prediag is switched of.
  \item Recommended: $\texttt{errvec=} \texttt{sFDs}$ leads to the following defaults:
    \begin{itemize}
      \item $\texttt{qscal=} 1.2$, for SCF runs: $\texttt{maxiter=} 6$ and $\texttt{thrd=} 0.3$, prediag is off; for DFT runs:
    \end{itemize}
\end{itemize}

$\text{maxiter}=5$ and $\text{thrd}=0.1$ prediag is on. If you want to switch off prediag put

$\text{prediag none}$.

$\text{scfdump}$

Dump SCF restart information onto data group $\text{restartd}$ and dump SCF MOs in each iteration onto $\text{scfmo}$ ($\text{scfdump = iter}$). Additionally, a data block $\text{scfiterinfo}$ will be dumped containing accumulated SCF total-, one- and two-electron energies of all previous SCF iterations. Information that will allow you to perform a restart if your calculation aborts will be dumped on data group $\text{restartd}$ (see also $\text{restart}$).

$\text{scfintunit options}$

Disc space specification for two-electron integrals. The following suboptions are available (and necessary):

unit=$\text{integer}$

Fortran unit number for this file. Unit numbers 30,31,... are recommended.

size=$\text{integer}$

Filespace in megabytes for this file. size=0 leads to a fully direct run. size is set by a statistics run, see $\text{statistics}$. DSCF switches to direct mode if the file space is exhausted.

file=$\text{char}$

Filename. This may also be a complete path name, if you want to store the integrals in a special directory. Make sure the file is local, otherwise integrals are transmitted over the network.

Thus your data group $\text{scfintunit}$ may look like this:

```
$\text{scfintunit}$
unit=30 size=35 file=twoint1
unit=31 size=35 file=/users/work/twoint2
```

Maximal 30 files may be specified in this way.

$\text{scfiterlimit integer}$

Maximum number of SCF iterations (default: 30).

$\text{scfmo none file=char}$

Input/output data group for SCF MOs. You can specify

none

To perform a calculation without a start vector (i.e. use a core Hamiltonian guess).

file=char

The file where the MOs are written on output (default: mos).

These two options can also be used for $\text{uhfmo_alpha}$ and $\text{uhfmo_beta}$ to use a core guess and write the molecular orbitals to file.

After running define or a TURBOMOLE calculation additional options may appear specifying the origin of the MOs:
expanded
These MOs were obtained by projection from another basis set. They should not be used for wavefunction analysis.

\textbf{scfconv=integer}

The MOs are converged SCF MOs, the convergence criterion applied was $10^{-\text{integer}}$.

\textbf{scfdump=integer}

The MOs are unconverged SCF MOs which were written on this data group after iteration \text{integer}. The latter three options are mutually exclusive.

\textbf{format(format string)}

This specifies the FORTRAN format specification which was used for MO output. The standard format is (4d20.14). (See data group \$mo output format.)

Example:
Your data group \$scfmo could look like this after a successful TURBOMOLE run:

\$scfmo scfconv=7 format(3(1x,d19.13))

1 a1 eigenvalue=-.524127 nsao=6
   .1234567890123d+01 -.1234567890123d+00 .1234567890123d-01
   .1234567890123d+01 -.1234567890123d+00
3 a2 eigenvalue=-.234810
   ...

\$scforbitalorder on/off
Order SCF MOs with respect to their energies (default: on)

\$scforbitalshift options
To assist convergence, either the energies of unoccupied MOs can be shifted to higher energies or, in open-shell cases, the energies of closed-shell MOs to lower energies. In general a large shift may help to get better convergence.

Options are:

\textbf{noautomatic}
Automatic virtual shell shift switched off.

\textbf{automatic real}
Automatic virtual shell shift switched on; the energies of virtual orbitals will be shifted if the HOMO-LUMO gap drops below \textit{real} such that a gap of \textit{real} is sustained. This is the default setting if the keyword is missing with \textit{real}=0.1.

\textbf{closedshell=real}
Option for open-shell cases. Closed shells are shifted to lower energies by \textit{real}. The default shift value is \textit{closedshell}=0.4.

\textbf{Note}: Normally this will disable the automatic shift of energies of virtual orbitals! To override this, you should append an exclamation mark to the 'automatic' switch, i.e. specify 'automatic! real'.
individual

Set shifts for special occupied MOs. To use this option, start the line with
the symmetry label and the list of MOs within this symmetry and append the
desired shift in brackets as in the following example:

\begin{verbatim}
a1 1,2,4-6 (-.34)
b1 8 (+.3)
\end{verbatim}

$scftol real

Integral evaluation threshold. Integrals smaller than real will not be evaluated. Note
that this threshold may affect accuracy and the convergence properties if it is chosen
too large. If $scftol$ is absent, a default value will be taken obtained from $scfconv$
by \( real = 10^{-\left(scfconv^2+1\right)} \cdot \#bf \) (\#bf = number of basisfunctions).

$scratch files

The scratch files allocated by dscf can be placed anywhere in your file systems in-
stead of the working directory by referencing their pathnames in this data group. All
possible scratch files are listed in the following example:

\begin{verbatim}
$scratch files
    dscf dens        path1/file1
    dscf fock        path2/file2
    dscf dfock       path3/file3
    dscf ddens       path4/file4
    dscf statistics  path7/file7
    dscf errvec      path8/file8
    dscf oldfock     path9/file9
    dscf oneint      path10/file10
\end{verbatim}

The first column specifies the program type (dscf stands for SCF energy calculations,
i.e. the dscf program), the second column the scratch file needed by this program
and the third column the pathname of the file to be used as scratch file.

$statistics options

The following options are allowed

\begin{verbatim}
off         Do not perform integrals statistics
dscf        Perform integrals statistics for dscf
mpgrad      see mpgrad
dscf parallel see PARALLEL PROCESSING
\end{verbatim}

Options dscf parallel, grad, mpgrad will be described in the related chapters.
If $statistics dscf$ has been given integral prescreening will be performed (which is
an \( n^4 \)-step and may therefore be time-consuming) and a table of the number of stored
integrals as a function of the two parameters $thize$ and $thime$ will be dumped.
Afterwards, the filespace needed for the current combination of $thize$ and $thime
will be written to the data group ($scfintunit) and $statistics dscf will be replaced by $statistics off.

$thime integer
Integral storage parameter, which is related to the time needed to calculate the integral. The larger integer the less integrals will be stored. The default value is integer = 5. (see also $thize, $statistics)

$thize real
Integral storage parameter, that determines, together with $thime, the number of integrals stored on disc. Only integrals larger than real will be stored. The default value is real = 0.100E-04.

RHF/ROHF

$closed shells
Specification of MO occupation for RHF, e.g.

\[
\begin{align*}
\text{a1g} & \quad 1-4 \\
\text{a2g} & \quad 1
\end{align*}
\]

$open shells type=1
MO occupation of open shells and number of open shells. 'type=1' here means that there is only a single open shell consisting e.g. of two MOs:

\[
\begin{align*}
\text{b2g} & \quad 1 \\
\text{b3g} & \quad 1
\end{align*}
\]

$rohf
This data group is necessary for ROHF calculations with more than one open shell. Example:

\[
\begin{align*}
$rohf & \quad 1 \\
\text{a} -\text{a} & \quad a=0 \quad b=0 \\
\text{h} -\text{h} & \quad a=1 \quad b=2 \\
\text{a} -\text{h} & \quad a=1 \quad b=2
\end{align*}
\]

This example is for the 7S state of chromium ($3d^5 4s^1$) in symmetry group $I$. Note that for this option being activated, $roothaan$ also has to be specified in your control file, although its parameter has no meaning in this case. For more details see Section 6.3.

$roothaan
For ROHF-calculations with only one open shell the Roothaan parameters\(^1\) a and b have to be specified within this data group (see also $rohf). Example:

\[
\begin{align*}
$roothaan & \quad a = 3/4 \quad b = 3/2
\end{align*}
\]

\(^1\)C. C. J. Roothaan: Rev. Mod. Phys. 32 (1960) 179.
This example is for the 3P ground state of carbon \((2p^2)\) in symmetry group I. \texttt{define} recognizes most cases and suggests good Roothaan parameters.

For further information on ROHF calculations (e.g. with more than one open shell), see the sample input in Section 22.6 and the tables of Roothaan parameters in Section 6.3.

Note that this keyword toggles the ROHF mode also for more than one open shell. If it is not given, the open-shell electrons are simply ignored.

**UHF**

\texttt{$alpha shells} and \texttt{$beta shells}

These two data groups specify the occupation of alpha and beta spin UHF MOs (syntax as any data group related with orbital occupation information, e.g. \texttt{$closed shells})

Example:

\begin{verbatim}
$alpha shells
a  1-8   ( 1 )
b  1-2   ( 1 )

$beta shells
a  1-7   ( 1 )
b  1-3   ( 1 )
\end{verbatim}

\texttt{$uhf}

directs the program to carry out a UHF run. \texttt{$uhf} overwrites closed-shell occupation specification.

\texttt{$uhfmo_alpha} and \texttt{$uhfmo_beta}

These two data groups contain the UHF MO vectors for alpha and beta spin respectively (same syntax as \texttt{$scfmo}).

\texttt{$uhfmo_beta}

see \texttt{$uhfmo_alpha}

**DFT**

\texttt{$dft}

\begin{verbatim}
functional b-p
gridsize m3
\end{verbatim}

for DFT calculations one has to specify the functional and the grid (for the quadrature of the exchange correlation part). The settings above are default: both lines can be left out if the B-P86 functional and grid m3 are required. Other useful functionals supported are:

\begin{verbatim}
b-lyp
b3-lyp
\end{verbatim}
b3-lyp\_Gaussian (equivalent to the Gaussian98 keyword B3LYP with VWNIII)
bh-lyp
s-vwn
s-vwn\_Gaussian (equivalent to the Gaussian98 keyword SVWN with VWNIII)
tpss
tpssh

Possible grids are 1–5 and m3–m5 where grid 1 is coarse (least accurate) and 5 most dense. We recommend however the use of so-called multiple grids m3–m5: SCF iterations with grid 1–3, final energy and gradient with grid 3–5. Usually m3 is fine: for large or delicate systems, try m4. For a reference calculation with a very fine grid and very tight thresholds use 'reference' as grid specification instead of 'gridsize xy'.

Note: the functionals b3-lyp\_Gaussian and s-vwn\_Gaussian are made available only for comparability with Gaussian. The functional VWNIII is much less well founded than VWN5 and the TURBOMOLE team does not recommend the use of VWNIII.

RI

dscf does not run with the keyword $rij: you must call the RI modules $ridft and $rdgrad for energy and gradient calculations. However, it does run with the keyword $rik, but it will ignore all RI settings and do a conventional non-RI Hartree–Fock or DFT calculation.

$rij
Enforces an RI-J calculation if module ridft is used, can be used for Hartree-Fock as well as for DFT calculations with pure or hybrid functionals.

$ridft
Obsolete keyword - use $rij instead!

$rik
Enforces a RI-JK calculation if module ridft is used, can be used for Hartree-Fock as well as for DFT calculations with pure or hybrid functionals.

$ricore integer
Choose the memory core available (in megabyte) for special arrays in the RI calculation (the less memory you give the more integrals are treated directly, i.e. recomputed on the fly in every iteration)

$jbas file=auxbasis
Cross reference for the file specifying the auxiliary basis as referenced in $atoms. We strongly recommend using auxbasis sets optimized for the respective MO basis sets, e.g. use SVP (or TZVP) for the basis and the corresponding auxbasis as provided by define (default: file=auxbasis).
$\text{riopop}$

Calculation of atomic charges according to the $s$ partial wave and atomic dipole moments according to the $p$ partial wave as resulting from the auxbasis representation of the density

**RI-JK**

If the keyword $\text{rik}$ is found in the control file, $\text{ridft}$ performs a Hartree–Fock–SCF calculation using the RI-approximation for both Coulomb and HF-exchange (efficient for large basis sets). For this purpose needed (apart from $\text{ricore}$):

$\text{jkbas file=auxbasis}$

Cross reference for the file specifying the JK-auxiliary basis as referenced in $\text{atoms}$. This group is created by the rijk menu in $\text{define}$.

**MARI-J**

Multipole-Accelerated-Resolution-of-Identity-$J$. This method partitions the Coulomb interactions in the near- and far-field parts. The calculation of the far-field part is performed by application of the multipole expansions and the near-field part is evaluated employing the RI-$J$ approximation. It speeds up calculation of the Coulomb term for large systems. It can only be used with the $\text{ridft}$ module and requires setting of the $\text{ridft}$ keyword.

$\text{marij}$

```
$\text{marij}$
  $\text{precision}$  1.0D-06
  $\text{lmaxmom}$   10
  $\text{nbinmax}$   8
  $\text{wsindex}$   0.0
  $\text{extmax}$    20.0
  $\text{thrmom}$    1.0D-18
```

The following options are available:

- **$\text{precision}$**: specifies precision parameter for the multipole expansions. Low-precision MARI-$J$ calculations require $1 \cdot 10^{-6}$, which is the default. For higher precision calculations it should be set to $1 \cdot 10^{-8}$–$1 \cdot 10^{-9}$.

- **$\text{lmaxmom}$**: maximum l-moment of multipole expansions. It should be set to a value equal at least twice the maximum angular momentum of basis functions. Default value is 10 and it should probably never be set higher than 18.

- **$\text{thrmom}$**: Threshold for moment summation. For highly accurate calculations it should be set to $1 \cdot 10^{-24}$.

- **$\text{nbinmax}$**: number of bins per atom for partitioning of electron densities. Default value is 8 and hardly ever needs to be changed.
21.2. FORMAT OF KEYWORDS AND COMMENTS

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wsindex</td>
<td>minimum separation between bins. Only bins separated more than the sum of their extents plus wsindex are considered as far-field. Default is 0.0 and should be changed only by the experts.</td>
</tr>
<tr>
<td>extmax</td>
<td>maximum extent for charge distributions of partitioned densities. Extents with values larger than this are set to extmax. Hardly ever needs to be changed.</td>
</tr>
</tbody>
</table>

Seminumeric HF-Exchange

If the keyword $senex is found in the control file, ridft performs a Hartree–Fock–SCF calculation using the seminumerical approximation for HF-exchange. Standard dft-grids can be used for the numerical integration. Smaller grids (-1,0) and the corresponding m-grids (m1,m2) have been defined as well. Grids of at least size m3 are recommended for heavy atoms. The gridsize can be modified just like in dft-calculations. The keyword $dsenex activates seminumerical gradient calculations. An example using the default grid for SCF (m1) and grid 5 for gradients (default grid: 3) looks like this:

```
$senex
$dsenex
  gridsize 5
```

LHF

Use the Localized Hartree–Fock (LHF) method to obtain an effective Exact-Exchange Kohn–Sham potential (module dscf). The LHF method is a serial implementation for spin-restricted closed-shell and spin-unrestricted ground states.

```
$dft
  functional lhf
  gridsize 6
```

With the LHF potential Rydberg series of virtual orbitals can be obtained. To that end, diffuse orbital basis sets have to be used and special grids are required.

- gridtype 4 is the most diffuse with special radial scaling; gridtype 5 is for very good Rydberg orbitals; gridtype 6 (default in Lhfprep) is the least diffuse, only for the first Rydberg orbitals.

Only gridsize 3–5 can be used, no multiple grids.

Use test-integ to check if the selected grid is accurate enough for the employed basis-set.

How to do LHF runs

1) Do a Hartree–Fock calculation using dscf.
2) Use the script `lhfprep` to prepare the `control` file (the old `control` file will be saved in `control.hf` and the molecular orbitals in `mos.hf` or in `alpha.hf` and `beta.hf` for the spin-unrestricted case). See `lhfprep -help` for options.

3) Run again `dscf`.

Otherwise the LHF functional can be selected in `define`: in this case default options are used.

Options for the LHF potential can be specified as follows (see also `lhfprep -help`)

```bash
$lhf
  off-diag on
  numerical-slater off
  pot-file save
  asymptotic dynamic=1.d-3
  homo 1blu
  homob 1blu # ONLY UNRESTRICTED
  conj-grad conv=1.d-7 maxit=20 output=1 cgasy=1
  slater-dtresh 1.d-9
  slater-region 7.0 0.5 10.0 0.5
  correct-region 10.0 0.5
  slater-b-region 7.0 0.5 10.0 0.5 # ONLY UNRESTRICTED
  correct-b-region 10.0 0.5 # ONLY UNRESTRICTED
  correlation func=lyp

  off-diag off
calculation of the KLI exchange potential. By default the LHF exchange potential is computed (off-diag on).

  numerical-slater on
the Slater potential is calculated numerically everywhere: this is more accurate but much more expensive. When ECP are used, turn on this option.

  numerical-slater off
leads to accurate results only for first-row elements or if an uncontracted basis set or a basis set with special additional contractions is used: in other cases `numerical-slater on` has to be used (this is default).

  asymptotic
for asymptotic treatment there are three options:

  asymptotic off
No asymptotic-treatment and no use of the numerical Slater. The total exchange potential is just replaced by $-1/r$ in the asymptotic region. This method is the fastest one but can be used only for the density-matrix convergence or if Rydberg virtual orbitals are of no interest.
21.2. FORM OF KEYWORDS AND COMMENTS

asymptotic on

Full asymptotic treatment and use of the numerical Slater in the near
asymptotic-region.

asymptotic dynamic=1.d-3

Automatic switching on (off) to the special asymptotic treatment if the
differential density-matrix rms is below (above) 1.d-3. This is the default.

pot-file save

the converged Slater and correction potentials for all grid points are saved in
the files slater.pot and corct.pot, respectively. Using pot-file load, the
Slater potential is not calculated but read from slater.pot (the correction
potential is instead recalculated). For spin unrestricted calculations the corre-
sponding files are slaterA.pot, slaterB.pot, corctA.pot and correctB.pot.

homo

allows the user to specify which occupied orbital will not be included in the
calculation of correction potential: by default the highest occupied orbital is
selected. This option is useful for those systems where the HOMO of the
starting orbitals (e.g. EHT, HF) is different from the final LHF HOMO. homob
is for the beta spin.

correlation func=\textit{functional}

a correlation functional can be added to the LHF potential: use func=lyp for
LYP, or func=vwn for VWN5 correlation.

For expert users

Options for the conjugate-gradient algorithm for the computation of the correction poten-
tial: rms-convergence (conj-grad conv=1.d-7), maximum number of iteration (maxit=20),
output level output=0-3, asymptotic continuation in each iteration (cgasy=1).

With slater-dtresh= 1.d-9 (default) the calculations of the numerical integrals for the
Slater potential is performed only if it changes more than 1.d-9.

Asymptotic regions specification:

corct-region R_F \, \Delta_F
0 \ldots R_F - \Delta_F : basis-set correction potential
R_F - \Delta_F \ldots R_F + \Delta_F : smooth region
R_F + \Delta_F \ldots + \infty : asymptotic correction
Defaults: R_F = 10, \Delta_F = 0.5

slater-region R_N \, \Delta_N \, R_F' \, \Delta_F'
0 \ldots R_N - \Delta_N : basis-set Slater potential
R_N - \Delta_N \ldots R_N + \Delta_N : smoothing region
R_N + \Delta_N \ldots R_F' - \Delta_F' : numerical Slater
R_F' - \Delta_F' \ldots R_F' + \Delta_F' : smoothing region
R_F' + \Delta_F' \ldots + \infty : asymptotic Slater
Note: R_F' - \Delta_F' \leq R_F - \Delta_F
Defaults: $R_N = 7$, $\Delta_N = 0.5$, $R'_F = 10$, $\Delta'_F = 0.5$

Use `correct-b-region` and `slater-b-region` for the beta spin.

### Two-component SCF (GHF)

Self-consistent two-component calculations (e.g. for spin-orbit interactions) can be carried out using the module `ridft`. The following keywords are valid:

- `$soghf$`: enforces two-component-SCF calculations; this option is combinable with `$rij`, `$rik` and `$dft`.
- `$kramers$`: switches on Kramers-restricted formalism
- `$collinear$`: switches on collinear two-component formalism (not rotational invariant)
- `$gdiis$`: enforces DIIS for complex Fock operator.

### All-electron relativistic approaches (X2C, BSS, DKH)

Relativistic *all-electron* calculations can be done employing the X2C, the BSS or the DKH Hamiltonian. Implemented for modules `dscf` and `ridft`.

- `$rx2c$`: switches on X2C calculation.
- `$rbss$`: switches on BSS calculation.
- `$rdkh integer$`: switches on DKH calculation of order `integer`.

- `$dkhparam integer$`: selects parameterization of the DKH Hamiltonian. Valid values are 1 (=default), 2, 3, 4, and 5.
  - `$dkhparam 1$`: Optimum parametrization (OPT)
  - `$dkhparam 2$`: Exponential parametrization (EXP)
  - `$dkhparam 3$`: Square-root parametrization (SQR)
  - `$dkhparam 4$`: McWeeny parametrization (MCW)
  - `$dkhparam 5$`: Cayley parametrization (CAY)
Note in particular that the parametrization does not affect the Hamiltonian up to fourth order. Therefore, as long as you run calculations with DKH Hamiltonians below 5th order you may use any symbol for the parametrization as they would all yield the same results. Higher-order DKH Hamiltonians depend slightly on the chosen parameterization of the unitary transformations applied in order to decouple the Dirac Hamiltonian, but this effect can be neglected. For details on the different parametrizations of the unitary transformations see [204].

$rlocal$
switches on local DLU approach.

$rlocpara \text{ integer}$

selects parameterization of the local approximation. Valid values are 0 or 1. For details on the different parametrizations see [79].

All of these keywords are combinable with $soghf$.

### 21.2.7 Keywords for Module riper

Many of the dscf and ridft keywords are also used by riper. To run the calculations, DFT method needs to be set in the $dft$ data group (see 21.2.6). The list of options specific to the riper module is presented below:

$riper$

# CFMM control options
lmaxmom 20
nctrgt 10
wsicl 3.0
epsbext 1.0d-9
locmult on
locmomadd 2

# PCG control options
lpcg on
lcfmmpcg on
lmxmompcg 20
pcgtol 1.0d-9
pcgtyp sp

# remaining options
northol 5
lchgprj on
kinescr on
pqmatdiag off
pqsingtol 1.0d-8

The following options are available:
# RI-CFMM options

**lmaxmom**  
Maximum $l$-moment of multipole expansions for calculations of Coulomb interactions. Default value is 20 and hardly ever needs to be changed.

**nctrgt**  
Target number of charges per lowest level box of an octree. Default value is 10. For very large systems its increase may lead to a moderate speed-up.

**wsci1**  
The well-separateness criterion. The boxes with centers separated more than sum of their lengths times $0.5 \times wsci1$ are considered as well-separated. The standard value is 3.0 and hardly ever needs to be changed. The usage of $wsci1$ makes sense only for values $\geq 2.0$. For $wsci1 \leq 3.0$ increasing $lmaxmom$ may be necessary.

**epsbext**  
Specifies precision parameter for the multipole expansion, used to determine basis function extents. Default value is $1.0 \cdot 10^{-9}$. Hardly ever needs to be changed.

**locmult**  
If set to on, an additional local multipole scheme is used for the charges from non-well-separated boxes. For periodic systems this leads to a significant speedup of calculations, especially for small unit cells and/or diffuse basis functions. Default value is off and on for molecules and periodic systems, respectively.

**locmomadd**  
If locmult is on increases the default order of local multipole expansions. Default value is 2 and probably never needs to be increased.

# PCG options

**lpcg**  
If set to on the preconditioned conjugate gradient (PCG) algorithm is used for solving the RI equations. It is implemented for the molecular systems only. Default value is off.

**lcfmmcp**  
If lpcg is used, lcfmmcp specifies whether the CFMM is applied for evaluation of the matrix-vector products needed for the iterative PCG solver. Not employing CFMM speeds up the calculations but halves the memory savings since full PQ matrix is stored. Default value is on.

**lmxmompc**  
Maximum $l$-moment of multipole expansions for calculations of Coulomb interactions within the PCG algorithm. Default value is 20. It should be set to the same or larger value than lmaxmom.

**pcgtol**  
Sets the threshold parameter controlling accuracy of the PCG solver (see [99] for details). Default value is $1.0 \cdot 10^{-9}$. For lower-precision calculations it can be set to $1.0 \cdot 10^{-8}$ but values larger than $1.0 \cdot 10^{-7}$ are not allowed as these lead to large errors in Coulomb energies and occasionally to SCF convergence problems.

**pcgtyp char**  
Specifies the type of preconditioner used in the PCG algorithm. Three
types of preconditioners are implemented and are defined explicitly in
Chapter 7. The ‘sp’ preconditioner is a default one performing consistently the best among the preconditioners considered. The ‘at’ preconditioner is less efficient in decreasing the number of CG iterations needed for convergence. However, it has negligible memory requirements and more favorable scaling properties, albeit with a large prefactor. The ‘ss’ preconditioner represents a middle ground between the ’sp’ and ’at’ preconditioners both in terms of the efficiency and memory requirements.

#remaining options

northol Diagonalization of the Fock matrix and orthonormalization of orbital coefficients is done every northol SCF iteration. Default value is 5.
lchgprj Specifies if charge projection is used for molecular systems. If set to off the output energy is identical as in case of ridft calculations for molecules.
kinescr When set to on (default) the kinetic energy criterion is used for integral screening. Probably should never be changed.
pqmatdiag If set to on full diagonalization of PQ matrix is performed when solving the RI equations. When diffuse basis functions are used the Cholesky decomposition may fail due to small eigenvalues. In this case the slower method based on a full diagonalization of the PQ matrix is necessary. Default value is off.
pqsingtol If pqmatdiag is used pqsingtol sets threshold to remove small eigenvalues. Default value is $1.0 \cdot 10^{-8}$.

If the $riper$ group is absent, the default values are used. For periodic systems, the following keywords are used to define the system:

$periodic n$

Specifies the number of periodic direction. Value of $n$ is 3 for a three-dimensional bulk, 2 for a two-dimensional surface slab and 1 for a one-dimensional system.

$cell$

|a| |b| |c| $\alpha \beta \gamma$

Unit cell parameters in form of six real numbers. Here |a|, |b| and |c| are lengths of the appropriate cell vectors, $\alpha$ is the angle between vectors $b$ and $c$, $\beta$ is the angle between vectors $a$ and $c$, and $\gamma$ is the angle between vectors $a$ and $b$. Lengths and angles in atomic units and degrees, respectively. Note that $riper$ assumes that the cell vectors $a$ and $b$ are aligned along the $x$ axis and on the $xy$ plane, respectively. For 1D systems the periodic direction is defined by vector $a$. In case of 2D calculations first and second periodic directions are $a$ and $b$, respectively.

$kpoints$

nkpoints $n_1 \ n_2 \ n_3$
Defines a Γ-centered mesh of points \( k_n = (k_1, k_2, k_3) \) with components

\[
k_j = i/n_j \quad \text{with} \quad i = -(n_j - 1)/2, \ldots, (n_j - 1)/2
\]

in crystal coordinates.

### 21.2.8 Keywords for Periodic Electrostatic Embedded Cluster Method

The Periodic Electrostatic Embedded Cluster Method (PEECM) functionality provides electronic embedding of a finite, quantum mechanical cluster in a periodic, infinite array of point charges. It is implemented within HF and DFT energy and gradient TURBOMOLE modules: *dscf, grad, ridft, rdgrad*, and *escf*. Unlike embedding within a finite set of point charges the PEEC method always yields the correct electrostatic (Madelung) potential independent of the electrostatic moments of the point charges field. It is also significantly faster than the traditional finite point charges embedding.

The basic PEECM settings are defined in the `$embed` block. It can be redirected to an external file using `$embed file=<file_name>$`.

Following keywords are used for the PEECM calculation setup:

**periodic**

Specifies the number of periodic directions. Allowed values for `number` are 3 for a bulk three-dimensional system, 2 for a two-dimensional surface slab, and 1 for a one-dimensional system. Default value is 3.

**cell**

Unit cell parameters in a form of six real values \( |a|, |b|, |c|, \alpha, \beta, \gamma \), where \( |a|, |b|, |c| \) are lengths of the appropriate cell vectors, \( \alpha \) is the angle between vectors \( b \) and \( c \), \( \beta \) is the angle between vectors \( a \) and \( c \), and \( \gamma \) is the angle between vectors \( a \) and \( b \). Default are atomic units and degrees. You can specify unit cell parameters in Å and degrees using `cell ang`.

**content**

\[
\text{label x y z}
\]

Content of the unit cell, where `label` is the label of the point charge Content of the unit cell, where `label` is the label of the point charge type and \( x \ y \ z \) are corresponding Cartesian or fractional crystal coordinates. Defaults are Cartesian coordinates and atomic units. You can specify Cartesian coordinates in Å using `content ang` and fractional coordinates using `content frac`. Note that Cartesian coordinates assume that the cell vector \( a \) is aligned along the \( x \) axis, and the vector \( b \) on the \( xy \) plane.
cluster
  label x y z
end

Atomic coordinates of the piece of the crystal to be replaced by the QM cluster and surrounding isolation shell (ECPs and explicit point charges), where label is the point charge label and x y z are corresponding Cartesian or fractional crystal coordinates. Defaults are Cartesian coordinates and atomic units. You can specify Cartesian coordinates in Å using cluster ang and fractional coordinates using cluster frac.

charges
  label charge
end

Values of point charges (for each atom-type), where label is the point charge label and charge specifies charge in atomic units.

ch_list
  label charge
end

Values of point charges (for each atom), where label is the point charge label and charge specifies charge in atomic units.

Note that charges and ch_list are mutually exclusive. An integer number n can also be appended to charges or ch_list to set the tolerance for charge neutrality violation to $10^{-n}$ (default $n = 5$).

Additionally, the following keywords control the accuracy of PEECM calculation:

lmaxmom
  Maximum order of the multipole expansions in periodic fast multipole method (PFMM). Default value is 25.

potval
  Electrostatic potential at the lattice points resulting from periodic point charges field will be output if this keyword is present. Default is not to output.

wsicl
  Well-separateness criterion for PFMM. Default is 3.0.

epsilon
  Minimum accuracy for lattice sums in PFMM. Default is 1.0d-8.
21.2.9 Keywords for COSMO

The Conductor-like Screening Model (COSMO) is a continuum solvation model, where the solute molecule forms a cavity within the dielectric continuum of permittivity epsilon that represents the solvent. A brief description of the method is given in chapter 20. The model is currently implemented for SCF energy and gradient calculations (dscf/ridft and grad/rdgrad), MP2 energy calculations (rimp2 and mpgrad) and MP2 gradients (rimp2), and response calculations with escf. The ricc2 implementation is described in section 20.

For simple HF or DFT single point calculations or optimizations with standard settings, we recommend to add the $cosmo keyword to the control file and to skip the rest of this section.

Please note: due to improvements in the A matrix and cavity setup the COSMO energies and gradients may differ from older versions (5.7 and older). The use_old_amat option can be used to calculate energies (not gradients) using the old cavity algorithm of TURBOMOLE 5.7.

The basic COSMO settings are defined in the $cosmo and the $cosmo_atoms block.

Example with default values:

```
$cosmo
    epsilon=infinity
    nppa= 1082
    nspa=  92
    disex= 10.0000
    rsolv= 1.30
    routf= 0.85
    cavity closed
    ampran= 0.1D-04
    phsran= 0.0
    refind= 1.3
# the following options are not used by default
    allocate_nps= 140
    use_old_amat
    use_contcav
    no_oc
```

`epsilon=real`

defines a finite permittivity used for scaling of the screening charges.

`allocate_nps=integer`

skips the COSMO segment statistics run and allocates memory for the given number of segments.
21.2. FORMAT OF KEYWORDS AND COMMENTS

no_oc

skips the outlying charge correction.

All other parameters affect the generation of the surface and the construction of the A matrix:

\texttt{nppa= \textit{integer}}

number of basis grid points per atom

(allowed values: \(i = 10 \times 3^k \times 4^l + 2 = 12, 32, 42, 92...\))

\texttt{nspa= \textit{integer}}

number of segments per atom

(allowed values: \(i = 10 \times 3^k \times 4^l + 2 = 12, 32, 42, 92...\))

\texttt{disex= \textit{real}}

distance threshold for A matrix elements (Ångstrom)

\texttt{rsolv= \textit{real}}

distance to outer solvent sphere for cavity construction (Ångstrom)

\texttt{routf= \textit{real}}

factor for outer cavity construction in the outlying charge correction

\texttt{cavity closed}

pave intersection seams with segments

\texttt{cavity open}

leave untidy seams between atoms

\texttt{ampran= \textit{real}}

amplitude of the cavity de-symmetrization

\texttt{phsran= \textit{real}}

phase of the cavity de-symmetrization

\texttt{refind= \textit{real}}

refractive index used for the calculation of vertical excitations and num. frequencies (the default 1.3 will be used if not set explicitly)

\texttt{use_old_amat}

uses A matrix setup of TURBOMOLE 5.7

\texttt{use_contcav}

in case of disjunct cavities only the largest contiguous cavity will be used and the smaller one(s) neglected. This makes sense if an unwanted inner cavity has been constructed e.g. in the case of fullerenes. Default is to use all cavities.

If the \$\texttt{cosmo}$ keyword is given without further specifications the default parameter are used (recommended). For the generation of the cavity, COSMO also requires the definition of atomic radii. User defined values can be provided in Ångstrom units in the data group \$\texttt{cosmo_atoms}$, e.g. for a water molecule:
CHAPTER 21. KEYWORDS IN THE CONTROL FILE

\$cosmo_atoms

# radii in Angstrom units

  o 1 \\
  \hspace{1em} radius = 1.7200

  h 2-3 \\
  \hspace{1em} radius = 1.3000

If this section is missing in the control file, the default values defined in the radii.cosmo file (located in \$TURBODIR/parameter) are used. A user defined value supersedes these defaults. \$cosmo and \$cosmo_atoms can be set interactively with the Cosmo input program cosmoprep after the usual generation of the TURBOMOLE input.

The Cosmo energies and total charges are listed in the result section. E.g.:

SCREENING CHARGE:

  \begin{align*}
  \text{cosmo} & : \ -0.003925 \\
  \text{correction} & : \ \ 0.003644 \\
  \text{total} & : \ -0.000282
  \end{align*}

ENERGIES [a.u.]:

  \begin{align*}
  \text{Total energy} & = \ -76.0296831863 \\
  \text{Total energy + OC corr.} & = \ -76.0297567835 \\
  \text{Dielectric energy} & = \ -0.0118029468 \\
  \text{Diel. energy + OC corr.} & = \ -0.0118765440 \\
  \text{The following value is included for downward compatibility} & \\
  \text{Total energy corrected} & = \ -76.0297199849
  \end{align*}

The dielectric energy of the system is already included in the total energy. OC corr denotes the outlying charge correction. The last energy entry gives the total outlying charge corrected energy in the old definition used in TURBOMOLE 5.7 and older versions. The Cosmo result file, which contains the segment information, energies, and settings, can be set using:

\$cosmo_out file= filename.cosmo

Isodensity Cavity: This option can be used in HF/DFT single point calculations only. The \$cosmo_isodens section defines the settings for the density based cavity setup (see also chapter 20). If the \$cosmo_isodens keyword is given without suboptions, a scaled isodensity cavity with default settings will be created. Possible options are:

\$cosmo_isodens

activates the density based cavity setup. The default values of \text{nspa} and \text{nsph} are changed to 162 and 92, respectively. This values are superseded by the user defined \text{nspa} value of the \$cosmo section. By default the scaled density method is used. The atom type dependent density values are read from the radii.cosmo file (located in \$TURBODIR/parameter).

\text{dx=real}

spacing of the marching tetrahedron grid in Å (default: 0.3Å).
all_dens=real
use one isodensity value for all atom types (value in a.u.)

The outlying charge correction will be performed with a radii based outer cavity. Therefore, and for the smoothing of the density changes in the intersection areas of the scaled density method, radii are needed as for the standard COSMO cavity. Please note: The isodensity cavity will be constructed only once at the beginning of the SCF calculation. The density constructed from the initial mos will be used (file mos or alpha/beta in case of unrestricted calculations). Because the mos of an initial guess do not provide a good density for the cavity construction, it is necessary to provide mos of a converged SCF calculation (e.g. a COSMO calculation with standard cavity). We recommend the following three steps: perform a standard COSMO calculation, add the isodensity options afterwards, and start the calculation a second time.

Radii based Isosurface Cavity: The $cosmo_isorad section defines the radii defined isosurface cavity construction. The option uses the algorithm of the isodensity cavity construction but the objective function used depends on the cosmo radii instead of the electron density. The default values of nspr and nsph are changed to 162 and 92, respectively. This values are superseded by the user defined nspr value of the $cosmo section. The resulting surface exhibits smoother intersection seams and the segment areas are less diverse than the ones of the standard radii bases cavity construction. Because gradients are not implemented, the radii based isosurface cavity can be used in single point calculations only.

$cosmo_isorad

dx=real
spacing of the marching tetrahedron grid in Å (default: 0.3Å).

COSMO in MP2 Calculations: The iterative COSMO PTED scheme (see chapter 20) can be used with the mp2cosmo script. Options are explained in the help message (mp2cosmo -h). Both MP2 modules rimp2 and mpgrad can be utilized. The control file can be prepared by a normal COSMO SCF input followed by a rimp2 or mpgrad input. The PTE gradients can be switched on by using the $cosmo_correlated keyword (rimp2 only). Again a normal SCF COSMO input followed by a rimp2 input has to be generated. The $cosmo_correlated keyword forces dscf to keep the COSMO information needed for the following MP2 calculation and toggles on the COSMO gradient contribution.

COSMO in Numerical Frequency Calculations: NumForce can handle two types of COSMO frequency calculations. The first uses the normal relaxed COSMO energy and gradient. It can be performed with a standard dscf or ridft COSMO input without further
settings. This is the right method to calculate a Hessian for optimizations. The second type, which uses the approach described in chapter 20, is implemented for ridft only. The input is the same as in the first case but Numforce has to be called with the -cosmo option. If no solvent refractive index refind=REAL is given in the $cosmo section of the control file the program uses the default (1.3).

**COSMO in vertical excitations and polarizabilities:** COSMO is implemented in escf and will be switched on automatically by the COSMO keywords of the underlying SCF calculation. The refractive index, used for the fast term screening of the vertical excitations, needs to be defined in the cosmo section of control file (refind=REAL).

**DCCOSMO-RS:** The DCOSMO-RS model (see chapter 20) has been implemented for restricted and unrestricted DFT and HF energy calculations and gradients (programs: dscf/ridft and grad/rdgrad). In addition to the COSMO settings defined at the beginning of this section, the $dcosmo_rs keyword has to be set.

$dcosmo_rs file=filename.pot

activates the DCOSMO-RS method. The file defined in this option contains the DCOSMO-RS $\sigma$-potential and related data (examples can be found in the default potentials in the $TURBODIR/parameter directory).

If the potential file cannot be found in the local directory of the calculation, it will be searched in the $TURBODIR/parameter directory. The following $\sigma$-potential files for pure solvents at 25°C are implemented in the current TURBOMOLE distribution (see parameter subdirectory):

- Water: h2o_25.pot
- Ethanol: ethanol_25.pot
- Methanol: methanol_25.pot
- Tetrahydrofurane: thf_25.pot
- Acetone: propanone_25.pot
- Chloroform: chc13_25.pot
- Tetrachloromethane: cc14_25.pot
- Acetonitrile: acetonitrile_25.pot
- Nitromethane: nitromethane_25.pot
- Dimethylsulfoxide: dimethylsulfoxide_25.pot
- Diethylether: diethylether_25.pot
- Hexane: hexane_25.pot
- Cyclohexane: cyclohexane_25.pot
- Benzene: benzene_25.pot
- Toluene: toluene_25.pot
- Aniline: aniline_25.pot

The DCOSMO-RS energies and total charges are listed in the Cosmo section of the output:
SCREENING CHARGE:

- cosmo: -0.012321
- correction: 0.011808
- total: -0.000513

(correction on the COSMO level)

ENERGIES [a.u.]:

- Total energy = -76.4841708454
- Outlying charge corr. (COSMO) = -0.0006542315
- Outlying charge corr. (DCOSMO-RS) = -0.0011042856
- Combinatorial contribution of the solute = -0.0017627889

(at inf. dil. in the mixture/pure solvent. Not included in the total energy above)

The outlying charge correction cannot be defined straightforward like in the normal COSMO model. Therefore, the output shows two corrections that can be added to the Total energy. The first one is the correction on the COSMO level (COSMO) and the second is the difference of the DCOSMO-RS dielectric energy calculated from the corrected and the uncorrected COSMO charges, respectively (DCOSMO-RS). The charges are corrected on the COSMO level only. The Total energy includes the $E_{\text{dil,}RS}$ defined in section 20. Additionally the combinatorial contribution at infinite dilution of the COSMO-RS model is given in the output. The use of this energy makes sense if the molecule under consideration is different than the used solvent or not component of the solvent mixture, respectively. To be consistent one should only compare energies containing the same contributions, i.e. same outlying charge correction and with or without combinatorial contribution. Please note: the COSMO-RS contribution of the DCOSMO-RS energy depends on the reference state and the COSMO-RS parameterization (used in the calculation of the chosen COSMO-RS potential). Therefore, the DCOSMO-RS energies should not be used in a comparison with the gas phase energy, i.e. the calculation of solvation energies.

21.2.10 Keywords for Modules grad and rdgrad

Many of the dscf and ridft keywords are also used by grad and rdgrad.

$\text{drvopt}$

This keyword and corresponding options are required in gradient calculations only in special circumstances. Just $\text{drvopt}$ is fine, no options needed to compute derivatives of the energy with respect to nuclear coordinates within the method specified: SCF, DFT, RIDFT.

If running a DFT gradient calculation, it is possible to include the derivatives of the quadrature weights, to get more accurate results. In normal cases however those effects are marginal. An exception is numerical calculation of frequencies by Numforce, where it is strongly recommended to use the weight derivatives option. The biggest deviations from the uncorrected results are to be expected if doing gradient calculations for elements heavier
than Kr using all electron basis sets and very small grids. To use the weight derivatives option, add

```
weight derivatives
```

in `$dft`.

The option

```
point charges
```

in `$drvopt` switches on the evaluation of derivatives with respect to coordinates of point charges. The gradients are written to the file `$point_charge_gradients` old gradients will be overwritten.

### 21.2.11 Keywords for Module `aoforce`

This module calculates analytically harmonic vibrational frequencies within the HF- or (RI)DFT-methods for closed-shell and spin-unrestricted open-shell-systems. Broken occupation numbers would lead to results without any physical meaning. Note, that RI is only used partially, which means that the resulting Hessian is only a (very good) approximation to exact second derivatives of the RIDFT-energy expression. Apart from a standard force constant calculation which predicts all (allowed and forbidden) vibrational transitions, it is also possible to specify certain irreps for which the calculation has to be done exclusively or to select only a small number of lowest eigenvalues (and eigenvectors) that are generated at reduced computational cost.

**General keywords**

`$drvopt` is the keyword for non-default options of gradient and second derivative calculations. Possibilities in case of the module `aoforce` are:

- **frequency analysis only**
- **analysis only**
  - to read a complete Hessian from the input file `$hessian` and perform only the frequency analysis
- **analysis [only] intoord [print printlevel]**
  - to perform an analysis of normal modes in terms of internal coordinates. Details about this option and the effect of the printlevel (default is 0) are given in Section 14. The effect of the keyword only is the same as described above.

`$maxcor 50`

fixes the RAM memory to be used by the run (here 50 MB), about 70% of available memory should be fine, because `$maxcor` specifies only the memory used to store derivatives of density and Fock matrices as well as the CPHF-RHS. Default is 200 MB.
$\text{forceconv} \ 7$

sets the convergence criterion for the CPHF-equations to a residual norm of $1.0d-7$. Normally the default value of $1.0d-5$ already provides an accuracy of vibrational frequencies of $0.01 \text{ cm}^{-1}$ with respect to the values obtained for the convergence limit.

$\text{forceiterlimit} \ 10$

fixes the maximum number of Davidson iterations for the solution of the CPHF-equations to a value of ten. Normal calculations should not need more than eight iterations, but as a precaution the default value is 25.

$\text{nosalc}$

forces the program in case of molecules with $C_1$ symmetry not to use $3N - 6(5)$ symmetry adapted but all $3N$ cartesian nuclear displacement vectors. This option may lead to a moderate speed-up for molecules notably larger than 1000 basis functions and 100 atoms.

$\text{noproj}$

forces the program not to project out translations and rotations when forming a basis of symmetry adapted molecular displacements. This option may be needed if a Hessian is required, that contains translation- and rotation-contributions, e.g. for coupling the system with low cost methods. Output of the unprojected hessian is done on $\text{nprhessian}$; format is the same as for conventional $\text{hessian}$. Output of the corresponding eigenvalues and eigenvectors is done analogously on $\text{nprvibrational spectrum}$ and $\text{nprvibrational normal modes}$.

$\text{nomw}$

causes the program to diagonalize a not mass weighted hessian. Output is on $\text{nprhessian}$, $\text{nprvibrational spectrum}$ and $\text{nprvibrational normal modes}$, because projection of rotations is not possible in this case.

$\text{isosub}$

This keyword allows to trace back the effects of isotopic substitution on vibrational frequencies. The atom(s) for which isotopic substitution is to be investigated are specified in subsequent lines of the form (atom index) (mass in special isotope), e.g.

\begin{verbatim}
$\text{isosub}$
3 2.001
5 13
\end{verbatim}

The interpolation then takes place between the mass(es) specified in $\text{atoms}$ (or the default mass(es), if none specified) and the mass(es) in $\text{isosub}$. Take care of symmetry equivalent atoms, otherwise symmetry analysis will fail. This feature can not be used in a lowest eigenvalue search (keyword $\text{les}$).

$\text{isopts} \ 6$

Sets the number of points for interpolation between the two isotopes compared by the $\text{isosub}$ option to six. Default value is 21.
Keywords for the treatment of only selected nuclear displacement vectors:

$\text{ironly}$
CPHF-iteration is done only for distortions, that are IR active.

$\text{ramanonly}$
CPHF-iteration is done only for distortions, that are Raman active.

$\text{les}$
This causes a lowest Hessian eigenvalue search to be performed instead of a complete force constant calculation. The lowest eigenvalue search consists of the calculation of a guess-Hessian and macro-iterations to find the solution vector(s) for the lowest eigenvalue(s). In each macro-iteration the CPHF-equations are solved for the present search vector(s). $\text{les all 1}$ means that one lowest eigenvalue for each irrep will be determined, other numbers of lowest eigenvalues per irrep are admissible too. Different numbers of lowest eigenvalues for different irreps are requested by e.g.

$\text{les}$
\begin{verbatim}
a1 3
a2 all
b2 1
\end{verbatim}

The convergence criterion of the Davidson iterations for the solution of the CPHF-equations as well as the maximal residual norm for the lowest Hessian eigenvalue in the macro-iteration are specified by $\text{forceconv}$ as explained above.

The maximum number of macro-iterations is specified by $\text{lesiterlimit x}$ with the default $x=25$. The maximum number of iterations for each solution of the CPHF-equations is again determined by $\text{forceiterlimit}$ as shown above.

The convergence of the macro-iterations is strongly influenced by the size of the starting search-subspace. Generally all guess-Hessian eigenvectors corresponding to imaginary frequencies and at least two real ones are used as starting search-subspace. However it proved to be necessary to use even more vectors in the case of guess-Hessians with very large conditioning numbers.

$\text{hesscond 8.0d-5}$
means that all eigenvalues with the quotient (eigenvalue)/(max. eigenvalue) lower than 0.00008 are added to the starting search-subspace. Default is $1.0d-4$.

$\text{hotfcht}$
Triggers the generation of input files for hotFCHT (program to calculate Franck-Condon-factors by R. Berger and co-workers). See 14.4.

$\text{sijuai_out}$
Save the derivative of the density matrix for subsequent use in the module evib. See 15

Force constant calculations on the DFT level prove to be numerically reliable only with large integration grids or if one includes the effects of quadrature weights. This is done by
default—to prevent this, insert
no weight derivatives
in $dft$.

21.2.12 Keywords for Module evib

$dfdxi$ textout

can be used to generate text output of the matrix elements of the derivative of the Fock-operator. For bigger systems this can however generate very large output files.

See 15

21.2.13 Keywords for Module escf

TDHF and TDDFT calculations

to perform an escf calculation converged molecular orbitals from a HF, DFT or RIDFT calculation are needed. The HF, DFT or RIDFT method is chosen according to the $dft$ or $ridft$ keywords, specified above. It is recommended to use well-converged orbitals, specifying $scfconv 7$ and $denconv 1d-7$ for the ground-state calculation. The input for an escf calculation can be conveniently generated using the ex menu in define, see Section 4.

In an escf run one of the following properties can be calculated: (please note the 'or' in the text, do only one thing at a time.)

1. RPA and time-dependent DFT singlet or triplet or spin-unrestricted excitation energies (HF+RI(DFT))

   $scfinstab rpas$ or
   $scfinstab rpat$ or
   $scfinstab urpa$

2. TDA (for HF: CI singles) singlet or triplet or spin-unrestricted or spin-flip excitation energies (HF+RI(DFT))

   $scfinstab ciss$ or
   $scfinstab cist$ or
   $scfinstab ucis$ or
   $scfinstab spinflip$

3. Two-component TDDFT excitation energies of Kramers-restricted closed-shell systems

   $scfinstab soghf$
4. Eigenvalues of singlet or triplet or non-real stability matrices (HF+RI(DFT), RHF)

$\texttt{scfinstab singlet}$ or $\texttt{scfinstab triplet}$ or $\texttt{scfinstab non-real}$

5. Static polarizability and rotatory dispersion tensors (HF+(RI)DFT, RHF+UHF)

$\texttt{scfinstab polly}$

6. Dynamic polarizability and rotatory dispersion tensors (HF+(RI)DFT, RHF+UHF)

$\texttt{scfinstab dynpol unit}$

list of frequencies

where $\texttt{unit}$ can be eV, nm, rcm; default is a.u. (Hartree). For example, to calculate dynamic polarizabilities at 590 nm and 400 i nm (i is the imaginary unit):

$\texttt{scfinstab dynpol nm}$

590

400 i

The number and symmetry labels of the excited states to be calculated is controlled by the data group $\texttt{soes}$. Example:

$\texttt{soes}$

b1g 17
eu 23
t2g all

will yield the 17 lowest excitations in IRREP b1g, the 23 lowest excitations in IRREP eu, and all excitations in IRREP t2g. Specify $\texttt{soes all n}$; to calculate the $n$ first excitations in all IRREPS. If $n$ is not specified, all excitations in all IRREPS will be obtained.

During an $\texttt{escf}$ run, a system-independent formatted logfile will be constructed for each IRREP. It can be re-used in subsequent calculations (restart or extension of eigenspace or of $\texttt{rpaconv}$). An $\texttt{escf}$ run can be interrupted by typing “touch stop” in the working directory.

general keywords

$\texttt{rpacor n}$

The maximum amount of core memory to be allocated for the storage of trial vectors
is restricted to \( n \) MB. If the memory needed exceeds the threshold given by \$rpacor\), a multiple pass algorithm will be used. However, especially for large cases, this will increase computation time significantly. The default is 200 MB.

\$spectrum \ unit

The calculated excitation energies and corresponding oscillator strengths are appended to a file named ‘spectrum’. Possible values of unit are eV, nm and cm\(^{-1}\) or rcm. If no unit is specified, excitation energies are given in a.u.

\$cdspectrum \ unit

The calculated excitation energies and corresponding rotatory strengths are appended to a file named ‘cdspectrum’. unit can have the same values as in \$spectrum.

\$start \ vector \ generation \ e

Flag for generation of UHF start MOs in a triplet instability calculation. The option will become effective only if there are triplet instabilities in the totally symmetric IRREP. The optional real number \( e \) specifies the approximate second order energy change in a.u. (default: 0.1).

\$velocity \ gauge

Enables calculation of dipole polarizability/rotatory dispersion in the velocity gauge. Active only for pure DFT (no HF exchange).

\$sum \ rules \ unit

list of frequencies

Enable calculation of oscillator and rotatory strength sum rules at frequencies specified by list of frequencies in unit unit (see \$scfinstab dynpol). Note that the sums will be taken only over the states specified in \$soes.

\$rpaconv \ n

the vectors are considered as converged if the Euclidean residual norm is less than \( 10^{-n} \). Larger values of \( n \) lead to higher accuracy. The default is a residual norm less than \( 10^{-5} \).

\$escfiterlimit \ n

Sets the upper limit for the number of Davidson Iterations to \( n \). Default is \( n = 25 \).

**GW Keywords**

\$gw

The main keyword that switches on a GW calculation. Provided that the response function is calculated setting this keyword will perform a standard \( G_0W_0 \) calculation with default values for the other flags. There are several options which can be added to the \$gw keyword, the syntax is:

\$gw

nl 1
gam 0.001
output qpe.dat
rpa

With the optional entries:

nl <integer>

Default: 1. Number of orbitals to calculate gw for. It is set to the number of occupied orbitals + 5 if set smaller than the number of occupied orbitals.

gam <real>

Default: 0.001. Infinitesimal complex energy shift. Negative value switches to calculating at that value but extrapolating to 0 in linear approximation.

output <filename>

Default: qpe.dat. Output filename for the quasiparticle energies.

rpa

Default: false (not set). If added as option pure rpa response function is calculated. If not added, the TDDFT response function is calculated and used to screen the coulomb interaction.

21.2.14 Keywords for Module rirpa

The keyword $rirpa allows to specify the following options,

npoints n
Number of frequency integration points, n (default is 60).

nohxx
HF energy calculation is skipped, (HXX = Hartree + eXact (Fock) eXchange).

rpaprof
Generates profiling output.

rpagrad
Switches on the gradients calculation for RI-RPA.

drimp2
Computes gradients in the direct RI-MP2 limit.

niapblocks n
Manual setting of the number of integral blocks, n, in subroutine rirhs.f. This is for developers; the default is determined with the $maxcor.
21.2.15 Keywords for Module egrad

egrad uses the same general keywords as escf and grad, see Sections 21.2.10 and 21.2.13. The state to be optimized is by default the highest excited state specified in $soes$. Note that only one IRREP can be treated at the same time in contrast to escf calculations. When the desired excited state is nearly degenerate with another state of the same symmetry it may be necessary to include higher states in the initial calculation of the excitation energy and vector in order to avoid root flipping. This is accomplished by means of the additional keyword

$exopt \ n$

which explicitly enforces that \( n \)-th excited state is optimized. \( n \) must not be larger than the number of states specified in $soes$.

$nacme$

flag to compute Cartesian non-adiabatic coupling vectors between the excited state of interest and the ground state [205]. This option requires the use of weight derivatives in section dft. It is only implemented for \( C_1 \) symmetry.

21.2.16 Keywords for Module mpgrad

If an MP2 run is to be performed after the SCF run, the SCF run has to be done with at least

1) density convergence \( $denconv \ 1.d-7 \)

2) energy convergence \( $scfconv \ 6 \)

$maxcor \ n$

The data group $maxcor$ adjusts the maximum size of core memory (\( n \) in MB) which will be allocated during the MP2 run. Recommendation: 3/4 of the actual main memory at most. If $maxcor$ is not found, its value is set to 200 MB.

$mp2energy$

Calculation of MP2 gradient is omitted, only MP2 energy is calculated.

$freeze$

\( a1g \ 1-2 \)

\( t1u \ 1 \)

The data group $freeze$ specifies frozen orbitals, in the above syntax by irreducible representations. The symmetry-independent and for standard-applications recommended syntax is

$freeze$

implicit core=5 virt=2
CHAPTER 21. KEYWORDS IN THE CONTROL FILE

This will freeze the 5 lowest occupied and 2 highest virtual orbitals (alpha and beta count as one in UHF cases). Note that degenerate orbitals count twice ($e$ representations), thrice ($t$ representations) etc. Note: In case of gradient calculations frozen core orbitals are regarded not by mpgrad, moreover freezing of virtual orbitals is generally not supported by mpgrad.

All essential data groups for mpgrad may be generated by the preparation tool mp2prep, apart from $\text{idmaxcor}$ (see above) these are the following:

\texttt{$straloo} n$

specifies the number of loops (or 'passes') over occupied orbitals, $n$, performed in the mpgrad run: the more passes the smaller file space requirements—but CPU time will go up.

\texttt{$mointunit$}

\begin{verbatim}
type=intermed  unit=61  size=0  file=halfint
type=1111     unit=62  size=0  file=moint#0

type=1112     unit=63  size=0  file=moint#1
type=1122     unit=64  size=0  file=moint#j

type=1212     unit=65  size=0  file=moint#k

type=1212a    unit=70  size=0  file=moint#a

type=gamma#1  unit=71  size=0  file=gamma#1

type=gamma#2  unit=72  size=0  file=gamma#2

type=1212u    unit=73  size=0  file=moint#u

type=1112u    unit=74  size=0  file=moint#v

type=gamma#1u unit=75  size=0  file=gamma#1u
\end{verbatim}

The data group \texttt{$mointunit$} specifies:

- which scratch files are needed,
- where they are located (path name) and
- (after a statistics run, see below) an estimated file size.

\texttt{$statistics$ mpgrad}

statistics run (estimation of disc space needed) for the adjustment of the file sizes will be performed.

\texttt{$mp2pair$}

calculation of MP2 pair correlation energies.

\textbf{21.2.17 Keywords for Module ricc2}

Note that beside the keywords listed below the outcome of the ricc2 program also depends on the settings of most thresholds that influence the integral screening (e.g. \texttt{$\text{denconv}$},
$\textbf{21.2. FORMAT OF KEYWORDS AND COMMENTS}$

$\textbf{scfconv, scftol}$ and for the solution of $Z$ vector equation with 4-index integrals (for relaxed properties and gradients) on the settings for integrals storage in semi-direct SCF runs (i.e. $\texttt{thime, thize, scfintunit}$). For the explanation of these keywords see Section 21.2.6.

$\texttt{cbas file=auxbasis}$

cross reference for the file specifying the auxiliary basis as referenced in $\texttt{atoms}$. We strongly recommend using auxbasis sets optimized for the corresponding MO basis sets.

$\texttt{freeze}$

Freeze orbitals in the calculation of correlation and excitation energies. For details see Section 21.2.16.

$\texttt{printlevel 1}$

Print level. The default value is 1.

$\texttt{tmpdir /work/thisjob}$

Specify a directory for large intermediate files (typically three-index coulomb integrals and similar intermediates), which is different from the directory where the $\texttt{ricc2}$ program is started.

$\texttt{maxcor 20}$

The data group $\texttt{maxcor}$ adjusts the maximum size of core memory in MB which will be allocated during the $\texttt{ricc2}$ run. $\texttt{maxcor}$ has a large influence on computation times! It is recommended to set $\texttt{maxcor}$ to ca. 75–85% of the available (physical) core memory.

$\texttt{spectrum unit}$

The calculated excitation energies and corresponding oscillator strengths are appended to a file named 'spectrum'. Possible values of $\texttt{unit}$ are eV, nm and cm$^{-1}$ or rcm. If no unit is specified, excitation energies are given in a.u.

$\texttt{cdspectrum unit}$

The calculated excitation energies and corresponding rotatory strengths are appended to a file named 'cdspectrum'. $\texttt{unit}$ can have the same values as in $\texttt{spectrum}$.

$\texttt{laplace}$

\begin{verbatim}
conv = 5
\end{verbatim}

The purpose of this data group is twofold: It activates the Laplace-transformed implementation of SOS-MP2 in the $\texttt{ricc2}$ module (if the sos option has been specified in $\texttt{ricc2}$) and it provides the options to specify the technical details for the numerical Laplace-transformation.

\begin{verbatim}
conv
\end{verbatim}

Threshold for the numerical integration used for the Laplace transformation of orbital energy denominators. The grid points for the numerical integration...
are determined such that is the remaining root mean squared error (RMSE) of the Laplace transformation is \( < 10^{-\text{conv}} \). By default the threshold is set to the value of \( \text{conv} \) given in \$\text{ricc2} \) (see below).

\$\text{ricc2} \$

```
  ccs
  cis
  mp2    didiag
  cis(d) energy only
  cis(dinf)
  adc(2)
  cc2
  ccsd
  mp3
  mp4
  ccsd(t)
  restart
  norestart
  hard_restart
  nohard_restart
  conv    = 8
  oconv   = 7
  lindep  = 15
  maxiter = 25
  mxdiis  = 10
  maxred  = 100
  iprint  = 1
  fmtprop = f15.8
  geoopt  model=cc2 state=(a" 2)
  scs cos=1.2d0 css=0.3333d0
  sos
  gsonly
  didiag
  intcorr
```

specifies the \( ab \ initio \) models (methods) for ground and excited states and the most important parameters and thresholds for the solution of the cluster equations, linear response equations or eigenvalue problems. If more than one model is given, the corresponding calculations are performed successively. Note: The CCS ground state energy is identical with the SCF reference energy, CCS excitation energies are identical to CIS excitation energies. The MP2 results is equivalent to the result from the \text{rimp2} \) module. \( \text{cis(dinf)} \) denotes the iterative CIS(D) variant CIS(D\( _\infty \)). The option \text{ccsd(t)} \) request a CCSD calculation with the perturbative triples correction, CCSD(T), and as a side result also the CCSD[T] energy will be printed.
21.2. FORM OF KEYWORDS AND COMMENTS

mp2 d1diag
Request the calculation of the $D_1$ diagnostic in MP2 energy calculations (ignored in MP2 gradient calculations). Note that the evaluation of the $D_1$ diagnostic increases the computational costs of the RI-MP2 energy calculation roughly by a factor of 3.

cis(d) energy only
If the energy only flag is given after the cis(d) keyword, it is assumed that only excitation energies are requested. This switches on some shortcuts to avoid the computation of intermediates needed e.g. for the generation of improved start vectors for CC2.

(no)restart
If the restart flag is set, the program will try to restart the CC2 calculations from previous solution vectors on file. If the norestart flag is set no restart will be done. Default is to do a restart for CC2 if and only if the file CCR0---1---1---0 exists. Note: There is no restart possibility for CCS/CIS or MP2/CIS(D).

(no)hard_restart
If the hard_restart flag is set, the program will try to reuse integrals and intermediates from a previous calculation. This requires that the restart.cc file has been kept, which contains check sums and some other informations needed. The hard_restart flag is switched on by default, if the restart.cc file is present.

conv The conv parameter gives the convergence threshold for the ground state energy for the iterative coupled-cluster methods as $10^{-\text{conv}}$. The default value is taken from the data group $\$\text{deneps}$.

oconv The oconv parameter gives an additional threshold for the residual of the cluster equations (vector function). If this parameter is given, the iterations for the cluster equations are not stopped before the norm of the residual is $< 10^{-\text{oconv}}$. By default the threshold is set to $\text{oconv} = \text{conv} - 1$, or $10 \times \text{deneps}$ if no input for conv is given.

lindep
If the norm of a vector is smaller than $10^{-\text{lindep}}$, the vector is assumed to be zero. This threshold is also used to test if a set of vectors is linear dependent. The default threshold is $10^{-15}$.

maxiter
gives the maximum number of iterations for the solution of the cluster equations, eigenvalue problems or response equations (default: 25).

mxdiis
is the maximum number of vectors used in the DIIS procedures for ground state or excitation energies (default: 10).
maxred
the maximum dimension of the reduced space in the solution of linear equations
(default: 100).

iprint
print level, by default set to 1 or (if given) the the value of the $printlevel
data group.

fmtprop
Fortran print format used to print several results (in particular one-electron
properties and transition moments) to standard output.

geoopt
specify wavefunction and electronic state for which a geometry optimization is
intended. For this model the gradient will be calculated and the energy and
gradient will be written onto the data groups $energy and $grad. Required
for geometry optimizations using the jobex script. Note, that in the present
version gradients are only available for ground states at the MP2 and CC2 and
for excited states at the CC2 level and not for ROHF based open-shell calculations.
Not set by default. The default model is CC2, the default electronic
state the ground state. To obtain gradients for the lowest excited state (of those
included in the excitation energy calculation, but else of arbitrary
multiplicity and symmetry) the short cut s1 can be used. x is treated as synonym for the
ground state.

scs
the opposite–spin scaling factor cos and the same–spin scaling factor css can
be chosen. If scs is set without further input, the SCS parameters cos=6/5 and
css=1/3 are applied. This keyword can presently only be used in connection
with MP2.

sos
the SOS parameters cos=1.3 and css=0.0 are applied. This keyword can
presently only be used in connection with MP2.

d1diag
request the calculation of the $D_1 diagnostic for the ground state wavefunction.
Only needed for MP2 (see above for the alternative input option mp2 d1diag).
For all other correlated methods the $D_1 diagnostic is evaluated by default
(without significant extra costs).

intcorr
calculates the second-order corrections to the CCSD(T) energy from the interference-
corrected MP2-F12 (INT-MP2-F12) if $rir12 is switched on. It can be com-
bined either with the mp2 or the ccsd(t) methods. In the latter case, the
CCSD(T)-INT-F12 energy is printed. The intcorr all keyword writes on
the output all pair energies.

$rir12
ansatz
r12model
comaprox
cabs
examp
r12orb
pairenergy
corrfac
cabsingles
f12metric

ansatz char
  char=1, 2* or 2
  The ansatz flag determines which ansatz is used to calculate the RI-MP2-F12 ground state energy.
  (Ansatz 2 is used if ansatz is absent.)

r12model char
  char=A, A' or B
  The r12model flag determines which approximation model is used to calculate the RI-MP2-F12 ground state energy.
  (Ansatz B is used if r12model is absent.)

comaprox char
  char=F+K or T+V
  The comaprox flag determines the method used to approximate the commutator integrals $[T, f_{12}]$.
  (Approximation T+V is used if comaprox is absent.)

cabs char val
  char=svd or cho
  The cabs flag determines the method used to orthogonalize the orbitals of the CABS basis. val is the threshold below which CABS orbitals are removed from the calculation.
  (svd 1.0d-08 is used if cabs is absent.)

examp char
  char=noinv, fixed or inv with flip or noflip
  The examp flag determines which methods are used to determine the F12 amplitudes. For fixed the amplitudes are optimized using the orbital-invariant method. For fixed and noinv only the diagonal amplitudes are non-zero and are either predetermined using the coalescence conditions (fixed), or optimized (noinv—not orbital invariant). If char=inv, the F12 energy contribution is computed using all three methods. For open-shell calculations noflip supresses the use of spin-flipped geminal functions.
  (The fixed flip method is used if examp is absent.)
**CHAPTER 21. KEYWORDS IN THE CONTROL FILE**

**pairenergy char**

*char=off or on*

If `char=off` (default), the print out of the standard and F12 contributions to the pair energies is suppressed. The summary of the RI-MP2-F12 correlation energies is always printed out.

**corrfac char**

*char=LCG or R12*

The `corrfac` flag determines which correlation factor is used for the geminal basis. LCG requires the data group `$lcg`, which contains the information regarding exponents and coefficients of the linear combination of Gaussians.

**cabsingles char**

*char=off or on*

The `cabsingles` flag determines whether or not the single excitations into the CABS basis are computed. The CABS singles are computed in any case if the CABS Fock matrix elements are computed anyway for the F12 calculation (*i.e.*, for ansatz 2 or r12model B or comaprox F+K).

**r12orb char**

*char=hf, rohf, boys or pipek*

The `r12orb` flag controls which orbitals are used for the F12 geminal basis functions. With `hf` the (semi)-canonical Hartree–Fock orbitals are used (default). For ROHF-based UMP2 calculations `rohf` orbitals can be used, which also implies that the `$freeze` data group options refer to ROHF rather than semi-canonical orbitals. For closed-shell species, localised orbitals can be used with either the Boys or Pipek-Mezey method. For the non-(semi)-canonical options, the `r12orb noinv` F12 energy correction is evaluated using active occupied orbitals transformed to the same basis as the orbitals in the geminal function.

**ccsdapprox label**

defines the approximation to CCSD-F12 which will be used if the MP2-F12 calculation is followed by a CCSD or CCSD(T) calculation. The available approximation and corresponding labels are

- CCSD(F12) \( \text{ccsd(f12)} \)
- CCSD(F12*) \( \text{ccsd(f12*)} \)
- CCSD[F12] \( \text{ccsd[f12]} \)
- CCSD-F12b \( \text{ccsd-f12b} \)
- CCSD(2*)\text{FI2} \( \text{ccsd(2)~/f12} \)
- CCSD(2*)\text{FII} \( \text{ccsd(2)~/f12} \)

It is recommended that these approximations are only used in combination with ansatz 2 and the SP approach (*i.e.* geminal coefficients fixed by the cusp conditions). For CCSD-F12b calculations also the CCSD-F12a energies are calculated as a byproduct. By default a CCSD(F12) calculation is carried out,
but it is recommended that whenever appropriate the computationally more efficient CCSD(F12*) approximation is used.

\texttt{no\_f12metric}

\texttt{f12metric}

If \texttt{no\_f12metric} is selected the coulomb metric is used in the density fitting scheme to calculate the four index integrals over the operators \( f_{12} \), \( f_{12} g_{12} \), \( f_{21} \), and \( f_{12} r_{12} \). If \texttt{f12metric} is selected the operator’s own metric is used. The default for the \texttt{ricc2} program is \texttt{no\_f12metric}, while the \texttt{pnoccsd} program can only be used with \texttt{f12metric}, where it is therefore the default.

\texttt{$excitations$

irrep=au \ multiplicities=1 \ nexc=4 \ npre=6 \ nstart=8
irrep=bg \ multiplicities=3 \ nexc=2 \ npre=4 \ nstart=5
spectrum \ states=all \ operators=diplen,dipvel
mexc \ istates=all \ fstates=all \ operators=diplen,dipvel
exprop \ states=all \ operators=qudlen
xgrad \ states=(ag(3) 1)
conv = 6
thrdiis = 2
preopt = 3
leftopt
bothsides
oldnorm

In this data group you have to give additional input for calculations on excited states:

\texttt{irrep}

the irreducible representation.

\texttt{multiplicity}

spin multiplicity (1 for singlet, 3 for triplet); default: singlet, not needed for UHF.

\texttt{nexc}

the number of excited states to be calculated within this irrep and for this multiplicity.

\texttt{npre}

the number of roots used in preoptimization steps (default: \texttt{npre = nexc}).

\texttt{nstart}

the number of start vectors generated or read from file (default: \texttt{nstart = npre}).

\texttt{spectrum}

This flag switches on the calculation of oscillator strengths for excited state—ground state transitions. Setting the parameter \texttt{states=all} is mandatory for the calculation of transition properties in the present version. The \texttt{operators} flag can be followed by a list of operators (see below) for which the transition
properties will be calculated. Default is to compute the oscillator strengths for all components of the dipole operator.

**tmexc**

This flag switches on the calculation of oscillator strengths for excited state—excited state transitions. Specifying the initial and final states via `istates=all` and `fstates=all` is mandatory for the calculation of transition properties in the present version. The `operators` flag can be followed by a list of operators (see below) for which the transition properties will be calculated. Default is to compute the oscillator strengths for all components of the dipole operator.

**exprop**

require calculation of first-order properties for excited states. For the `states` option see `spectrum` option above; for details for the `operators` input see below.

**xgrad**

request calculation of the gradient for the total energy of an excited state. If no state is specified, the gradient will be calculated for the lowest excited state included in the calculation of excitation energies. The simultaneous calculation of gradients for several state is possible.

**conv** convergence threshold for norm of residual vectors in eigenvalue problems is set to $10^{-\text{conv}}$. If not given, a default value is used, which is chosen as $\max(10^{-\text{conv}}, 10^{-\text{oconv}}, 10^{-6})$, where `conv` refers to the values given in the data group `$\text{ricc2}$`.

**preopt**

convergence threshold used for preoptimization of CC2 eigenvectors is set to $10^{-\text{preopt}}$ (default: 3).

**thrdiis**

threshold ($10^{-\text{thrdiis}}$) for residual norm below which DIIS extrapolation is switched on in the modified Davidson algorithm for the non-linear CC2 eigenvalue problem (default: 2).

**leftopt**

If the flag `leftopt` is set the left eigenvectors are computed (default is to compute the right eigenvectors, for test purposes only).

**bothsides**

The `bothsides` flag enforces the calculation of both, the left and the right eigenvectors (for test purposes only).

**oldnorm**

The `oldnorm` flag switches the program to the old normalization of the eigenvectors and %$T_1$ and %$T_2$ diagnostics which were identical with those used in the CC response code of the Dalton program.

$\response$
21.2. FORMAT OF KEYWORDS AND COMMENTS

fop unrelaxed_only operators=diplen
sop operators=(diplen,diplen) freq=0.077d0
gradient
conv = 6
zconv = 6
semicano
nosemicano
thresemi = 3

In this data group you have to give additional input for the calculation of ground state properties and the solution of response equations:

fop  This flag switches on the calculation of ground state first-order properties (expectation values). The operators flag can be followed by a list of operators (see below) for which the first-order properties will be calculated. Default is to compute the components of the dipole and the quadrupole moment. The option unrelaxed_only suppress the calculation of orbital-relaxed first-order properties, which require solution the CPHF-like Z-vector equations. Default is the calculation of unrelaxed and orbital-relaxed first-order properties. The unrelaxed_only option will be ignored, if the calculation of gradients is requested (see gradient option below and geoopt in data group $ricc2$).

sop  requests the calculation of ground state second-order properties as e.g. dipole polarizabilities. The operators flag has to be followed by a comma separated pair of operators. (If more pairs are needed they have to be given with additional sop commands.) Default is to compute all symmetry-allowed elements of the dipole-dipole polarizability. With the freq flag on can specify a frequency (default is to compute static polarizabilities). The relaxed flag switched from the unrelaxed approach, which is used by default, to the orbital-relaxed approach. Note that the orbital-relaxed approach can not only be used in the static limit (freq=0.0d0). For further restrictions for the computation of second-order properties check Chapter 10.5.

gradient require calculation of geometric gradients. In difference to the geoopt keyword in the data group $ricc2$ this can be used to compute gradients for several methods within a loop over models; but gradients and energies will not be written to the data groups $grad$ and $energy$ as needed for geometry optimizations. Note, that in the present version gradients are only available for MP2 and CC2 and only for a closed-shell RHF reference.

conv  convergence threshold for norm of residual vectors in linear response equations is set to 10$^{-\text{conv}}$. If not given in the $response$ data group, a default value is used, which is chosen as max(10$^{-\text{conv}}$, 10$^{-\text{oconv}}$, 10$^{-6}$), where conv and oconv refer to the values given in the data group $ricc2$. 


**zconv**

Convergence threshold for the norm of the residual vector in the solution of the Z vector equations will be set to $10^{-\text{zconv}}$.

**semicano**

Use semi-canonical formulation for the calculation of (transition) one-electron densities. Switched on by default. The semi-canonical formulation is usually computationally more efficient than the non-canonical formulation. Exceptions are systems with many nearly degenerate pairs of occupied orbitals, which have to be treated in a non-canonical way anyway. (See also explanation for \texttt{thrsemi} below).

**nosemicano**

Use non-canonical formulation for the calculation of (transition) one-electron densities. Default is to use the semi-canonical formulation.

**thrsemi**

The threshold for the selection of nearly degenerate pairs of occupied orbitals which (if contributing to the density) have to be treated in a non-canonical fashion will be set to $10^{-\text{thrsemi}}$. If set to tight the semi-canonical algorithm will become inefficient, if the threshold is too large the algorithm will become numerical unstable.

**zpreopt**

Threshold for preoptimizing the so-called Z vector (i.e. the lagrangian multipliers for orbital coefficients) with a preceding RI-CPHF calculation with the cbas auxiliary basis. The RI-CPHF equations will be converged to a residual error $< 10^{-\text{zpreopt}}$. Default is $\text{zpreopt}=4$. This preoptimization can reduce significantly the computational costs for the solution of the Z vector equations for large basis sets, in particular if they contain diffuse basis functions. For calculations on large molecules with small or medium sized basis sets the preoptimization becomes inefficient compared to the large effects of integral screening for the conventional CPHF equations and should be disabled. This option is automatically disabled for \texttt{ricc2} calculations based on foregoing RI-JK Hartree-Fock calculation.

**nozpreopt**

Disable the preoptimization of the Z vector by a preceding RI-CPHF calculation with the cbas basis set. (Note that the preoptimization is automatically deactivated if the \texttt{ricc2} calculation is based on a foregoing RI-JK Hartree-Fock calculation.)

Common options for keywords in the data groups $\$\texttt{ricc2}$, $\$\texttt{response}$, and $\$\texttt{excitations}$:

**operators=diplen,dipvel**

Input of operator labels for first-order properties, transition moments, etc. Currently implemented operators/labels are
overlaps

overlap (charge) operator: the integrals evaluated in the AO basis are
\( \langle \mu | \nu \rangle \)

dipole

dipole operator in length gauge: \( \langle \mu | r_i^O | \nu \rangle \) with \( i = x, y, z \); the index \( O \) indicates dependency on the origin (for expectation values of charged molecules), which in the present version is fixed to \((0,0,0)\) (all three components, individual components can be specified with the labels \( \text{xdiplen}, \text{ydiplen}, \text{zdiplen} \)).

dipvel

dipole operator in velocity gauge: \( \langle \mu | \nabla_i | \nu \rangle \) (all three components, individual components can be specified with the labels \( \text{xdipvel}, \text{ydipvel}, \text{zdipvel} \)).

quadrupole

quadrupole operator \( \langle \mu | r_i^O r_j^O | \nu \rangle \)
(all six components, individual components can be specified with the labels \( \text{xxqudlen}, \text{xyqudlen}, \text{xzqudlen}, \text{yyqudlen}, \text{yzqudlen}, \text{zzqudlen} \)).

If all six components are present, the program will automatically give the electronic second moment tensor (which involves only the electronic contributions) \( M_{ij} \), the isotropic second moment \( \alpha = \frac{1}{3} \text{tr} M \) and the anisotropy
\[
\beta = \sqrt{\frac{1}{2} \sum_{i=x}^{z} (M_{ii} - M_{i+1,i+1})^2 + 3 \sum_{i=x}^{z} M_{i,i+1}^2}.
\]

Furthermore the traceless quadrupole moment
\[
\Theta_{ij} = \frac{1}{2} \langle 3r_i r_j - r^2 \delta_{ij} \rangle
\]
(including nuclear contributions) is given.

angular momentum

angular momentum \( \langle \mu | L_i^O | \nu \rangle \)
(all three components, individual components can be specified with the labels \( \text{xangmom}, \text{yangmom}, \text{zangmom} \)).

nef

electronic force on nuclei \( \langle \mu | Z_I r_i^I | \nu \rangle \), where \( Z_I \) is the charge of the nucleus \( I \) and \( r_i^I \) is the position vector of the electron relative to the nucleus (all three components for all nuclei: the labels are \( \text{xnef001}, \text{ynef001}, \text{znef001}, \text{xnef002}, \text{ynef002}, \text{znef002} \), etc. where the number depends on the order in the coord file).

states=all

specification of states for which transition moments or first-order properties are to be calculated. The default is \( \text{all} \), i.e. the calculations will be done for all excited states for which excitation energies have been calculated. Alternatively, one can select a subset of these listed in parentheses, e.g. \( \text{states}=(\text{ag}\{3\} 1,3-5; \text{biu}\{1\} 1-3; \text{b2u4}) \). This will select the triplet \( \text{ag} \) states no. 1, 3, 4, 5 and the singlet \( \text{b1u} \) states no. 1, 2, 3 and the singlet (which is default if no \{\} is found) \( \text{b2u} \) state no. 4.

istates=all fstates=all

The specification of initial and final states for transition properties between excited
states is mandatory. The syntax is analog to the states option, i.e. either all or a list of states is required.

$D2$-diagnostic
Calculate the double-substitution-based diagnostics $D_2$.

$cc2$-natocc
Write MP2/CC2 natural occupation numbers and natural orbitals to a file.

$cgrad$ 1000
Calculate the error functional $\delta_{RI}$ for the RI approximation of $(ai|bj)$ integrals

$$\delta_{RI} = \frac{1}{4} \frac{|(ab||ij)_{\text{exact}} - (ab||ij)_{\text{RI}}|^2}{\epsilon_a - \epsilon_i + \epsilon_b - \epsilon_j}$$

and its gradients with respect to exponents and coefficients of the auxiliary basis set as specified in the data group $cbas$. The results are written to $egrad$ scaled by the factor given with the keyword $cgrad$ and can be used to optimize auxiliary basis sets for RI-MP2 and RI-CC2 calculations (see Section 1.5).

21.2.18 Keywords for Module pnoccsd

Note that beside the keywords listed below the outcome of the pnoccsd program also depends on the settings of most thresholds that influence the integral screening (e.g. $denconv$, $scfconv$, $scftol$). For the explanation of these keywords see Section 21.2.6.

$cbas$ file=auxbasis
Auxiliary basis set for RI approximation. For details Section 21.2.17.

$freeze$
Freeze orbitals in the calculation of correlation and excitation energies. For details see Section 21.2.16.

$printlevel$ 1
Print level. The default value is 1.

$tmpdir$ /work/thisjob
Specify a directory for large intermediate files (typically three-index coulomb integrals and similar intermediates), which is different from the directory where the program is started.

$maxcor$ 5000
The data group $maxcor$ adjusts the maximum size of core memory in MB which will be allocated during the pnoccsd run. $maxcor$ has a large influence on computation times! It is recommended to set $maxcor$ to ca. 75–85% of the available (physical) core memory.

$laplace$

conv = 1
Only needed for test purposes. It sets the accuracy for the numerical Laplace-transformation to $10^{-\text{conv}}$. For OSV-PNO-MP2 this threshold is only used for the approximate amplitudes from which the OSVs are computed when the iterative algorithm is enabled. The default value is $10^{-1}$. Decreasing it has little influence on the accuracy of the final results but slows down the OSV generation.

```plaintext
$p noc c s d$

mp2
localize boys
prepno davidson
mxrdim 800
tolpno= 1.00E-7
tolosv= 1.00E-8
tolri= 1.21E-3
tolpair= 4.64E-6
tolopnos on
tolcapno= 1.00E-9 3.16E-8
tolosc= 1.00E-10 3.15E-9
tolopno= 1.00E-9
toloso= 1.00E-10
projector 3
conv = 7
oconv = 3
lindep = 12
maxiter = 25
maxdiis = 10
maxred = 100
scs cos=1.2d0 css=0.3333d0
sos
```

**mp2** specifies the *ab initio* model (method). The current release version is restricted to MP2 which is also the default model.

**localize** specifies the localization method; possible choices are **boys** for Foster-Boys, **pm** for Pipek-Mezey, and **none** for canonical (deprecated, only meant for testing) orbitals. Default are Foster-Boys orbitals. (Pipek-Mezey orbitals become very delocal with diffuse basis sets.)

**prepno** switch between to algorithms for the generation of OSV coefficients. Possible choices are **full** for an $O(N^4)$ scaling direct diagonalization (cmp. [206]) and **davidson** for an $O(N^3)$ scaling iterative diagonalization (cmp. [20]). **davidson** is the default choice and recommended.

**mxrdim** the maximal dimension of trial vectors in the iterative OSV generation (**prepno davidson**). For PNO-MP2 the default is 800. The dimension is bounded by the number of active virtual orbitals and except for small systems a much smaller
value as the number of virutals is sufficient. Smaller dimensions increase the performance, but than the iterative scheme might not converge and the program must be restarted with an adjusted dimension. For PNO-MP2-F12 the Default is 4000 since a larger reduced space is required to construct the OSX (cmp. [206]). Usually there is no need to touch this parameter.

tolpno specifies the PNO truncation threshold. Default value: \(10^{-7}\).

tolosv specifies the OSV truncation threshold. If no given tolosv is set to \(0.1 \times \text{tolpno}\), which is the recommended value.

tolri specifies the threshold for selecting orbital and pair-specific auxiliary basis sets for the local RI approximation. If not given tolri is set to \(10^{7/12} \times \sqrt{\text{tolpno}}\), which is the recommended value.

tolpair specifies the energy threshold for selecting the significant pairs. If not given tolpair is set to \((0.1 \times \text{tolpno})^{2/3}\).

tolcapno sets for F12 calculations the truncation thresholds for the complementary auxiliary PNOs for the virtual spaces (CAPNOs) in the projectors for the three-electron integrals. If not specified, default values are calculated from the threshold tolpno.

tolosc sets for F12 calculations the truncation thresholds for the orbital specific complementary auxiliary virtuals from which the CAPNOs are generated. If not specified, default values are chosen as \(0.1 \times \text{tolcapno}\) thresholds, which is the recommended choice.

tolopno sets for F12 calculations the truncation thresholds for the selection of PNOs for the occupied orbital spaces (OPNOs) in the projectors for the three-electron integrals. If not specified, default values are calculated from the threshold tolpno.

toloso set for F12 calculations the truncation thresholds for the orbital specific auxiliary occupied orbitals from which the OPNOs are generated. If not specified, default values are chosen as \(0.1 \times \text{tolopno}\) threshold, which is the recommended choice.

projector sets for F12 calculations the pair specific projector \(A_{ij}\) for the three-electron integrals. Possible choices are:

- **projector 1** for \(1 - \tilde{O}_1\tilde{O}_2 - \tilde{V}_1\tilde{V}_2 - \tilde{O}_1\tilde{V}_2'' - \tilde{V}_1''\tilde{O}_2\)
- **projector 2** for \(1 - \tilde{O}_1\tilde{O}_2 - \tilde{V}_1\tilde{V}_2 - \tilde{O}_1\tilde{V}_2'' - \tilde{V}_1''\tilde{O}_2\)
- **projector 3** for \(1 - \tilde{O}_1\tilde{O}_2 - \tilde{V}_1\tilde{V}_2 - \tilde{O}_1\tilde{V}_2'' - \tilde{V}_1''\tilde{O}_2\)

With projector 3 \(A_{ij}\) becomes independent of the system size and is therefore the default. But note that for the \(O(N^4)\) scaling algorithm this can not be exploited and here projector 1 is more suited.
The `conv` parameter gives the convergence threshold for the ground state energy as $10^{-\text{conv}}$. The default threshold is $10^{-7}$.

The `oconv` parameter gives an additional threshold for the residual of the ground state equations as $<10^{-\text{oconv}}$. The default threshold is $10^{-3}$.

If the norm of a vector is smaller than $10^{-\text{lindep}}$, the vector is assumed to be zero. This threshold is also used to test if a set of pre-PNOs is linear dependent. The default threshold is $10^{-12}$.

`maxiter` gives the maximum number of iterations for the solution of the cluster equations, eigenvalue problems or response equations (default: 25).

`mxdiis` is the maximum number of vectors used in the DIIS procedures for the ground state equations (default: 10).

`maxred` not used in the current release.

The opposite–spin scaling factor `cos` and the same–spin scaling factor `css` can be chosen. If `scs` is set without further input, the SCS parameters `cos=6/5` and `css=1/3` are applied.

The SOS parameters `cos=1.3` and `css=0.0` are applied.

For the description of this data group see Sec. 21.2.17.

### 21.2.19 Keywords for Module `relax`

#### `$optimize` options

define what kind of nonlinear parameters are to be optimized by `relax` and specify some control variables for parameter update.

Available options are:

- **internal on/off**
  optimize molecular structures in the space of internal coordinates using definitions of internal coordinates given in `$intdef` as described in Section 4.1 (default: on).

- **redundant on/off**
  optimize molecular structures in redundant internal coordinates using definitions of redundant internal coordinates given in `$redundant`. For an optimization in redundant internal coordinates option `internal` has to be switched on too, and option `cartesian` has to be switched off (default: on).

- **cartesian on/off**
  optimize molecular structures in the space of (symmetry-distinct) cartesian coordinates (default: off).

- **basis on/off suboptions**
  optimize basis set exponents (default=off).

Available suboptions are:
logarithm

exponents of uncontracted basis functions will be optimized after conversion into their logarithms (this improves the condition of the approximate force constant matrix obtained by variable metric methods and the behavior of the optimization procedure); scale factors of contracted basis functions will not be affected by the logarithm suboption

scale

ALL basis set exponents will be optimized as scale factors (i.e. contracted blocks and single functions will be treated in the same way); if both suboptions (scale and logarithm) are given the logarithms of the scale factors will be optimized

global on/off

optimize a global scaling factor for all basis set exponents (default: off).

NOTES:

• basis and global have to be used exclusively!
• if $optimize$ has been specified but $forceapprox$ is absent, the option $forceinit$ on is switched on by default.
• specification of the option $interconversion$ on will override $optimize$!

$coordinateupdate$ options

define some variables controlling the update of coordinates.

Available options are:

dqmax $real$

maximum allowed total change for update of coordinates. The maximum change of individual coordinate will be limited to $dq_{max}/2$ and the collective change $dq$ will be damped by $dq_{max}/\langle dq|dq\rangle$ if $\langle dq|dq\rangle > dq_{max}q$

(default: 0.3)

interpolate on/off

calculate geometry update by inter/extrapolation of geometries of the last two cycles (the interpolate option is always switched on by default, but it is only active ANY time if steepest descent update has been chosen, i.e. $forceupdate$ method=none; otherwise it will only be activated if the DIIS update for the geometry is expected to fail)

statistics on/integer/off

provide a statistics output in each optimization cycle by displaying all (the last $integer$, default setting by $define$ is 5) subsequent coordinates, gradient and energy values (default: on).

$gdiihistory$ file=char

the presence of this keyword forces $relax$ to provide informational output about the usage of DIIS for the update of the molecular geometry.
21.2. FORM OF KEYWORDS AND COMMENTS

$\textit{interconversion} \ \textit{options} \ \textit{default}=\textit{off}

special input related to the transformation of atomic coordinates between cartesian and internal coordinate spaces (default: \textit{off}).

Available options are:

\texttt{maxiter=1}

maximum number of iterations for the iterative conversion procedure internal \rightarrow cartesian coordinates (default: 25).

\texttt{qconv}

convergence criterion for the coordinate conversion (default: 1.d-10).

\texttt{on/off \ \textit{options}}

this switch activates special tasks: transform coordinates/grads/ hessians between spaces of internal/cartesian coordinates using the definitions of internal coordinates given in $\textit{intdef}$:

available suboptions are:

\texttt{cartesian -> internal coordinate gradient hessian}

\texttt{cartesian <- internal coordinate}

the direction of the transformation is indicated by the direction of the arrow

\textbf{Note}: specification of $\textit{interconversion} \ \textit{on} \ \textit{will \ override} \ \textit{optimize}!

$\texttt{forceupdate} \ \textit{method \ options}$

this data group defines both the method for updating the approximate force constant matrix and some control variables needed for the force constant update.

Options for \texttt{method}:

\texttt{none} \ \ \no update (steepest descent)

\texttt{ms \ suboptions} \ \ Murtagh–Sargent update

\texttt{dfp \ suboptions} \ \ Davidon–Fletcher–Powell update

\texttt{bfgs \ suboptions} \ \ Broyden–Fletcher–Goldfarb–Shanno update

\texttt{dfp-bfgs \ suboptions} \ \ combined (bfgs+dfp) update

\texttt{schlegel \ suboptions} \ \ Schlegel update

\texttt{ahlrichs \ suboptions} \ \ Ahlrichs update (macro option)

\textit{suboptions} if \texttt{method}=\texttt{ms, dfp, bfgs, schlegel, ahlrichs}

\texttt{nungeo=integer} \ \ \ number of structures used

\texttt{maxgeo=integer} \ \ \ maximum number of geometries (= rank of the update procedure, for \texttt{ahlrichs} only)
**CHAPTER 21. KEYWORDS IN THE CONTROL FILE**

ingeo=integer  
minimum number of geometries needed to start update  

if method=ms, dfp, bfgs:  
maxgeo=2, mingeo=1 as default

additional suboptions if method=ahlrichs

modus= char fmode  
for an explanation see suboptions pulay given below e.g. ahlrichs numgeo=7 mingeo=3 maxgeo=4 modus=<g|d> dynamic

**NOTES:** if the macro option ahlrichs has been chosen and  
n=numgeo, nycl=‘number of geometries available’

- if nycl < n: geometry update by inter/extrapolation using the last two geometries  
- if nycl ≥ n: diagonal update for the hessian by least mean squares fit; pulay update for the geometry (using specified modus, fmode (see pulay below))  
- if (nycl ≥ max(5, n + 3) and max(|g|) < 0.01 and \(\bar{g} < 0.001\)) or \(H_{ij} \neq 0 \forall i \neq j\): diagonal update is replaced by multidimensional BFGS (rank \(n\)) update for the hessian

**pulay suboptions**  
try to find an optimal linear combination of the coordinates of the numpul previous optimization cycles as specified by modus (see below).

Available suboptions are:

numpul=integer  
number of geometries to be utilized  

maxpul=integer  
maximum number of geometries  

minpul=integer  
minimum number of geometries needed to start update  

modus=char fmode  
char=<g|g> or <g|dq> or <dq|dq> defines the quantity to be minimized (dq = internal coordinate change).  
fmode specifies the force constants to be used (only if char=<g|dq> or <dq|dq>)!  
fmode=static: use static force constants  
fmode=dynamic: use updated force constants  

fail=real  
real defines the threshold for the quantity \(g \ast dq/|g| \ast |dq|\) which defines the
angle between gradient vector and coordinate change (default: 0.1). If pulay is
used in connection with a multidimensional BFGS update for the hessian than
the default is real=0.0. If \( \frac{g \cdot dq}{|g| \cdot |dq|} > -\text{real} \) the pulay update for the geometry
is expected to fail and will be ignored. For example:
pulay numpul=4 maxpul=4 minpul=3 modus=<dq|dq> static fail=0.2

**options** for \$forceupdate

**diagonal**
update only the diagonal force constants (update for off-diagonals will be sup-
pressed) (only active if method=ms, dfp, bfgs)

**offdamp** real
this allows to damp off-diagonal force constants by 1/real (compare offreset,
which discards off-diagonals completely). Only values > 1.0 will be accepted.
This option is active only within one relax run and will be disabled automatic-
ally by relax. This is useful in difficult cases, where the non-diagonal update
has lead to too large non-diagonal elements of the hessian.

**offreset**
reset off-diagonal force constants to zero. This option will be active for the
current optimization cycle only, i.e. it will be removed by relax after having
discarded off-diagonals!

**allow=**real
optimization cycle specification of a maximum energy change allowed (given in
mHartree) which will be accepted using the actual approximate force constant
matrix from \$forceapprox; if this energy change will be exceeded, the force
constants will be scaled appropriately
(The default: 0.0 means NO action)

**scale=**real
scaling factor for the input hessian (default: 1.0).

**threig=**real
lower bound for eigenvalues of the approximate hessian (default: 0.005); if any
eigenvalue drops below threig, it will be shifted to a reasonable value defined
by:
reseig=real default: texttt0.005.

**thrbig=**real
upper bound for eigenvalues of the hessian; if any eigenvalue exceeds thrbig,
it will limited to this value (default: 1000.0).

**damping=**real
damp the variable metric update for the hessian by 1/(1+ real) (default: 0.0).

**$forceinit** option
specify initialization of the (approximate) force constant matrix.
Available options are:
on/off
this activates or deactivates initialization; if on has been set, relax will provide an initial force constant matrix as specified by one of the possible initialization options as described below and will store this matrix in data group $forceapprox; after initialization relax resets $forceinit to off!

diag= suboptions
provide a diagonal force constant matrix with:
available suboptions are:
real
this will lead to an assignment of diagonal elements (default: 1.0)).
default
this will lead to an assignment of initial force constant diagonals depending on the coordinate type.
individual
Provide individual defined force constant diagonals for
• internal coordinates (supplied in $intdef ... fdiag=..)
• a global scale factor ( $global ... fdiag=..)
This does not work for basis set optimization. For the correct syntax of ‘fdiag=..’ see descriptions of $intdef, $global

carthess
read a cartesian (e.g. analytical) hessian from $hessian and use it as a start force constant matrix; if $optimize internal has been set: use its transform in internal coordinate space. If large molecules are to be optimized, it may be necessary (large core memory requirements!) to deactivate the numerical evaluation of the derivative of the $B$-matrix with respect to cartesian coordinates, which is needed to transform $H(cart) \rightarrow H(int)$ exactly by specifying no dbdx.

$last SCF energy change = real
$last MP2 energy change = real
These keywords depend on the optimization task to be processed and are updated by the corresponding program (i.e. SCF energy).

$m$-matrix options
This data block contains non-default specifications for the $m$-matrix diagonals. This is of use if some cartesian atomic coordinates shall be kept fixed during optimization.
Available options are:
integer real real real
atomic index followed by diagonal elements of the $m$-matrix for this atom

scratch files
The scratch file ftmp allocated by relax can be placed anywhere in your file systems instead of the working directory by referencing its pathname in this data group as
21.2. FORMAT OF KEYWORDS AND COMMENTS

follows:

\[
\begin{array}{ccc}
\text{$\$scratch$ files} & \text{relax} & \text{ftmp} \\
\text{path/file} & & \\
\end{array}
\]

The first column specifies the program, the second column the scratch file and the third column the pathname of the file to be used as scratch file.

**Input Data Blocks Needed by RELAX**

\texttt{$\$intdef$ or $\$redundant$}

Definitions of internal coordinates and, optionally, values of internal coordinates (\texttt{val=...}, given in a.u. or degrees) or force constant diagonal elements (\texttt{fdiag=...}).

\texttt{$\$grad$}

Cartesian coordinates and gradients calculated in subsequent optimization cycles. Entries are accumulated by one of the gradient programs (\texttt{grad}, \texttt{mpgrad}, \texttt{rimp2}, \texttt{ricc2}, \texttt{egrad}, etc.).

\texttt{$\$egrad$}

Basis set exponents scale factors and their gradients as calculated in subsequent optimization cycles. Entries are accumulated by one of the gradient programs.

\texttt{$\$globgrad$}

Global scale factors and gradients as calculated in subsequent optimization cycles. Entries are accumulated by the \texttt{grad} or \texttt{aoforce} program.

\texttt{$\$corrgrad$}

Allows to augment internal SCF gradients by approximate increments obtained from treatments (e.g. correlation or relativistic) on higher level. See the example below.

\texttt{$\$corrgrad$}

\begin{verbatim}
# coordinate increment
1 0.0600
8 -0.0850
\end{verbatim}

\texttt{$\$forceapprox$ options$}

Approximate force constant matrix (as needed for geometry optimization tasks). The storage format may be specified by the available options:

\texttt{format=}

the default format is \texttt{format=(8f10.5)}, but other 10-digit f10.x formats (e.g. \texttt{x=4,6,...}) are possible and will be used, after being manually specified within \texttt{$\$forceapprox$}. See the example below:

\texttt{$\$forceapprox$ format=(8f10.4)$}
0.9124
-.0108 0.3347
0.2101 0.0299 1.3347
0.0076 0.1088 0.0778 0.6515

$hessian$ (projected)
this data block contains the analytical cartesian force constant matrix (with transla-
tional and rotational combinations projected out) as output by the $aoforce$ program
and may be used to supply a high quality force constant matrix $forceapprox$ for ge-
ometry optimizations (specifying $forceinit$ on carthess, or $interconversion$
cartesian -> internal hessian).

RELAX Output Data Groups

$coord$
either updated cartesian coordinates if a successful coordinate update has been per-
formed, or cartesian coordinates for input internal coordinates if only a conversion
from internal to cartesian coordinates has been performed.

$basis$
updated basis set exponents, basis sets contraction coefficients or scaling factors, if
$optimize$ basis on has been specified.

$global$
updated global scaling factor for all basis set exponents, if $optimize$ global on has
been specified.

$forceapprox$
an approximate force constant matrix to be used in quasi-Newton type geometry
optimizations; this matrix will be improved in subsequent optimization cycles if one
of the variable-metric methods ($forceupdate$) has been chosen. See 5.3.13 and
21.2.19.

$forcestatic$
a static (i.e. never updated) approximate force constant matrix to be used in DIIS-type
geometry optimizations. It will be initialized by relax specifying: $forceupdate$ pulay
...modus=<dq|dq> static.

The next data groups are output by relax (depending on the optimization subject) in order
to control the convergence of optimization procedures driven by the shell script jobex.

$maximum norm of cartesian gradient = real$

$maximum norm of internal gradient = real$

$maximum norm of basis set gradient = real$

real is the absolute value of the maximum component of the corresponding gradient.
21.2. FORMAT OF KEYWORDS AND COMMENTS

Other Input/Output data used by Relax

In order to save the effort for conversion of accumulated geometry and gradient data (as needed for the force constant update or the DIIS update of the geometry) to the optimization space, within which the geometry has to be optimized, one may specify the keyword

$oldgrad

Then the relax program accumulates all subsequent coordinates and gradient as used in optimization in this data group (or a referenced file). This overrides the input of old coordinate and gradient data from data blocks $grad, $egrad, ... as accumulated by the grad program.

degrees

21.2.20 Keywords for Module statpt

$statpt

itrvec 0
update bfgs
hssfreq 0
keeptmode
hssidiag 0.5
radmax 0.3
radmin 1.0d-4
tradius 0.3
threchange 1.0d-6
thrmxdispl 1.0d-3
thrmxgrad 1.0d-3
thrrmsdispl 5.0d-4
thrrmsgrad 5.0d-4

Only non-default values are written in the control file except:

$statpt

itrvec 0

Following options are available:

itrvec

Index of the Hessian eigenvector to follow for transition structure search (transition vector). Eigenpairs are sorted in ascending order, i.e. with increasing eigenvalues and start with index 1. The eigenpairs corresponding to translations and rotations are shifted to the end. For minimization the value 0 has to be specified.
update

Method of hessian update. For minimization default is BFGS, for TS search default is Powell and none is for no update.

hssfreq

Recompute the full Hessian every N’th step during a transition state search. The default is zero and the Hessian is read in or computed in the first step only. If the standard Hessian update methods fail, it can help to use this keyword. Warning: This will make the calculation much more time demanding!

keeptmode

Freezing transition vector index.

hssidiag

diagonal hessian elements for diagonal Hessian guess (default: 0.5).

radmax

Maximum allowed value for trust radius (default: 0.3).

radmin

Minimum allowed value for trust radius (default: 1.0d-4).

tradius

Initial value for trust radius (default tradius: radmax = 0.3).

Convergence criteria

threchange

threshold for energy change (default: 1.0d-6).

thrmxdispl

threshold for maximal displacement element (default: 1.0d-3).

thrmxgrad

threshold for maximal gradient element (default: 1.0d-3).

thrrmsdispl

threshold for RMS of displacement (RMS = root mean square)
(default = 5.0d-4)

thrrmsgrad

threshold for RMS of gradient (default: 5.0d-4).

All values are in atomic units.

21.2.21 Keywords for Module moloch

$properties specifies the global tasks for program moloch by virtue of the following options

$properties

trace off
moments active
potential off
21.2. FORMAT OF KEYWORDS AND COMMENTS

- cowan-griffin: off
- localization: off
- population analyses: off
- plot: off
- firstorder: off
- fit: off

A missing option or a option followed by the flag `off` will not be taken into account. The flag `active` may be omitted. For most of these options (with the only exceptions of `trace` and `cowan-griffin`), there are additional data groups allowing for more detailed specifications, as explained below.

**moments**
- If `moment` is active you need

```
$moments
  0th 1st 2nd 3rd
  point .0 .0 .0
```

To compute the 0th, 1st, 2nd and 3rd moment at the reference point 0 0 0.

**potential**
- If `potential` is active you need

```
$points #1
  pot fld fldgrd shld
  point .0 .0 .0
```

To compute the electrostatic potential (pot) and/or electrostatic field (fld) and/or electrostatic field gradient (fldgrd) and/or the zeroth order contribution to the diamagnetic shielding (shld) at reference point 0 0 0.

**localization**
- If `localization` is active you need `$boys` to perform a boys-localization of orbitals with orbital energies $\geq$ `threshold`=-2 Hartrees; localize with respect to `locxyz`=x, y and z and write resulting orbitals to `lmofile`='lmo'. At the most `sweeps`=10000 orbital rotations are performed. Non-defaults may be specified using the following suboptions:

```
lmofile= filename
locxyz  dir1 dir2 dir3
threshold  real
sweeps  integer
```

**population analyses**
- If `population analyses` is active you need
$mulliken
spdf molap netto irpspd irpmol mommul

to perform a Mulliken population analysis. The options specify the output data:

spdf print molecular orbital contributions to atomic $s$, $p$, $d$, . . . -populations
molap print molecular orbital contributions to overlap populations
netto print atomic netto populations
irpspd print contributions of (irreducible) representations to atomic $s$, $p$, $d$, . . . -populations
irpmol print contributions of (irreducible) representations to overlap populations

or

$loewdin
to perform a Löwdin population analysis (options are invalid here). A Löwdin population analysis is based on decomposing $\sqrt{SD}\sqrt{S}$ instead of $DS$ in case of a Mulliken PA.

or

$spaboon
momao maodump maofile=mao all

to perform a population analysis based on occupation numbers (the options are not necessary and produce some output data concerning the modified atomic orbitals):

momao print MO contributions to occupation numbers of modified atomic orbitals (MAOs).
maodump print all MAOs on standard output
maofile=mao all print all MAOs to file mao.

This kind of population analysis basically aims at so-called shared electron numbers (SEN) between two or more atoms. By default 2-, 3- and 4-center contributions to the total density are plotted if they are larger than 0.01 electrons. Thresholds may be individually chosen, as well as the possibility to compute SENs for molecular orbitals:

$shared
electron
numbers
orbitals
2-center threshold = real
3-center threshold = real
4-center threshold = real

Results of this kind of PA depend on the choice of MAOs. By default, all MAOs with eigenvalues of the atomic density matrices larger than 0.1 will be taken into account. This is a reasonable minimal basis set for most molecules. If modified atomic orbitals
shall not be selected according to this criterion, the data group $mao selection has to be specified

$mao selection threshold = \text{real};

The default criterion for the selection of MAOs is the occupation number, for which a global threshold can be specified within the same line as the keyword $maoselection. If the global criterion or threshold is not desirable for some atoms, lines of the following syntax have to be added for each atom type of these.

\begin{verbatim}
atom symb list nmao=i method=meth threshold=r
\end{verbatim}

The parameters in this definition have the following meaning:

\begin{itemize}
  \item \textbf{symb} atom symbol
  \item \textbf{list} list of all atoms for which this definition should apply. The syntax for this list is as usual in TURBOMOLE, e.g. 2,3,8-10,12
  \item \textbf{nmao=i} means number of MAOs to be included
  \item \textbf{method=meth} means selection criterion for MAOs. \texttt{meth} can be \texttt{occ} (default), \texttt{eig}, or \texttt{man} \texttt{string}, where \texttt{occ} denotes selection of MAOs by occupation numbers, \texttt{eig} selection by eigenvalues and \texttt{man} allows manual selection. In the latter case the string (max. 8 characters) appended to \texttt{man} serves as nickname for the definition of the MAOs to be chosen. This nickname is expected to appear as the leftmost word in a line somewhere within data group $mao selection and is followed by the indices of the modified atomic orbitals which are to be selected.
  \item \textbf{threshold=r} means the threshold to be applied for the selection criteria \texttt{occ} or \texttt{eig} (default: 0.1).
\end{itemize}

Example:

\begin{verbatim}
$mao selection threshold= 0.09
atom c 1,3-5 nmao= 5 method= eig threshold= 0.1
atom o 2 nmao= 3 method= man olabel
olabel 3-5
\end{verbatim}

\textbf{plot}

option \texttt{plot} is out of fashion; to plot quantities on a grid, rather use $pointval in connection with \texttt{dscf}, \texttt{ridft}, \texttt{rimp2} or \texttt{egrad}, as described below. If nevertheless \texttt{plot} is active you need

\begin{verbatim}
$grid #1
mo 4a1g
origin .000000 .000000 .000000
\end{verbatim}
### 21.2.22 Keywords for wave function analysis and generation of plotting data

Properties of RHF, UHF and (two-component) GHF wave functions as well as those of SCF+MP2 densities or such from excited state DFT-calculations can be directly analyzed within the respective programs (`dscf`, `ridft`, `mpgrad`, `rimp2` and `egrad`). In case of spin-unrestricted calculations results are given for total densities ($D^\alpha + D^\beta$) and spin densities ($D^\alpha - D^\beta$). If not explicitly noted otherwise, in the following "D" is the SCF density, D(SCF), in case of `dscf` and `ridft`, the MP2-corrected density, D(SCF)+D(MP2), for `mpgrad` and `rimp2` and the entire density of the excited state in case of `egrad`. For modules `dscf` and `ridft` the analysis of properties may be directly started by calling `dscf -proper`.

```plaintext
vector1  1.000000  .000000  .000000
vector2  .000000   1.000000 .000000
grid1 range -5.000000  5.000000 points 100
grid2 range -5.000000  5.000000 points 100
outfile = 4a1g
```

to obtain two-dimensional plot data of mo 4a1g (the plane is specified by origin and two vectors with grid range and number of grid points) which is written to file 4a1g. Several plots may be obtained (#1, #2 etc.) at the same time. Use tool 'konto' to visualize the plot.

**Note:** This is the old-fashioned way to plot MOs and densities. A new—and easier—one is to use `$pointval`, as described below.

```plaintext
fit
  if fit is active you need
  $vdw_fit
  shell number_of_gridpoints distance_from_vdW_surface
  refine value_of_potential
  shell Each line refers to all atoms, the line specifies a spherical layer of grid points around the atoms. The number of points and their distance from the van der Waals surface [Bohr] are given (the default is 1.0).
  refine one line only, smoothing of the layers of grid points around the molecule: the real number is used to define isopotential surfaces on which the points of the layers have to lie.

  $vdw_radii
  element_symbol van_d_waals_radius

  One line per element has to be specified, it contains the name of the element and the van der Waals radius in [Bohr].
```
21.2. FORMAT OF KEYWORDS AND COMMENTS

(or ridft -proper). In case of mpgrad and rimp2 this is possible only, if the MP2 density has already been generated, i.e. after a complete run of mpgrad or rimp2.

Functionalities of analyses are driven by the following keywords.

$mvd$

leads to calculation of relativistic corrections for the SCF total density in case of dscf and ridft, for the SCF+MP2 density in case of rimp2 and mpgrad and for that of the calculated excited state in case of egrad. Quantities calculated are expectation values $<p^2>, <p^4>$ and the Darwin term ($\sum 1/Z_A * \rho(R_A)$).

$moments$

yields calculation of electrostatic moments arising from nuclear charges and total electron densities. Also without setting this keyword moments up to quadrupole are calculated, with respect to reference point (0,0,0). Supported extensions:

$moments <i>

x1 y1 z1
x2 y2 z2
.
.

By integer i, the maximum order of moments is specified, maximum and default is $i=3$ (octopole moments), real numbers x, y, z allow for the specification of one or more reference points.

$pop$

drives the options for population analyses. By default a Mulliken PA in the basis of cartesian atomic orbitals (CAOs) is performed for the total density ($D^a + D^b$) leading to Mulliken (brutto) charges and, in case of spin-unrestricted calculations also for the spin density ($D^a - D^b$) leading to Mulliken (brutto) numbers for unpaired electrons. Besides total numbers also contributions from $s$, $p$, ... functions are listed separately.

Two-component wavefunctions (only module ridft and only if $soghf$ is set): In two-component calculations instead of $|S_z| (<S_x, S_y, S_z>)$ is written to the output. Additionally a vector-file named spinvec.txt is written, which includes the resulting spinvector for each atom in the system (also the direction).

The following modifications and extensions are supported, if the respective commands are written in the same line as $pop$:

1all Additional information about $p_x, p_y, p_z$ (and analogous for $d$ and $f$ functions) is displayed (lengthy output).

atoms list of atoms

Contributions are plotted only if arising from atoms selected by list.

thrpl=real

Contributions smaller than thrpl are not displayed (default: 0.01).
overlaps: Mulliken atomic overlap matrix is displayed.

netto: Mulliken netto populations (diagonal elements of Mulliken overlap matrix) are calculated.

mosum list of MOs
Summed Mulliken contributions for a group of molecular orbitals defined by numbers referring to the numbering obtained e.g. from the tool eiger. Note that occupancy of MOs is ignored, i.e. all orbitals are treated as occupied.

mo list of MOs
Mulliken contributions for single MOs defined by numbers (independent of whether they are occupied or not). If this option is valid, one may additionally set
dos width=real points=integer
to calculate a (simulated) density of states by broadening the discrete energy levels with Gaussians and superimposing them. The width of each Gaussian may be set by input (default: 0.01 a.u.). The resolution (number of points) may be chosen automatically (default values are usually sufficient to generate a satisfactory plot) or specified by hand. The output files (dos in case of RHF wave functions, and dos_a+b, dos_a-b, dos_alpha, dos_beta; for UHF cases) contain energies (first column), resulting DOS for the respective energy (second column) as well as s-, p-, d-contributions for the respective energy (following columns).

Example:

$pop mo 23-33 dos atoms 2,3,7-8

leads to Mulliken PA (CAO-basis) for each of the eleven MOs 23-33, regarding only contributions from atoms 2-3 and 7-8 (results are written to standard output) and generation of file(s) with the respective simulated density of states.

$pop nbo

to perform a natural population analyses [163]. The possible options (specified in the same line) are

AO
must be provided, the CAO case is not implemented.

tw=real
Threshold \( t_w \) to circumvent numerical difficulties in computing \( O_w \)
(default: \( tw=1.d-6 \)).

idbgl=integer
Debug level
(default: \( idbgl=0 \)).

ab
For UHF cases: Print alpha and beta density results.

short
Print only natural electron configuration and summary.

Example:

$pop nbo AO ab short atoms 1,2,6
leads to a natural population analysis (AO-basis) with printing the results of alpha and beta densities (only the electron configuration and the summary) for the atoms 1, 2 and 6.

To change the NMB set for atoms, one has to add a $nbondmb-block in the control file. Example:

```
$nbondmb
  ni s:4 p:2 d:1
  o s:2 p:1
```

leads to a NMB set for Ni of 4 s-, 2 p- and 1d-functions and for O of 2 s- and 1 p-functions.

$pop paboon
to perform a population analyses based on occupation numbers [164] yielding "shared electron numbers (SEFs)" and multicenter contributions. For this method always the total density is used, i.e. the sum of alpha and beta densities in case of UHF, the SCF+MP2-density in case of MP2 and the GHF total density for (two-component-)GHF.

The results of such an analysis may depend on the choice of the number of modified atomic orbitals ("MAOs"), which can be specified by an additional line; without further specification their number is calculated by the method "mix", see below. Note: One should carefully read the information concerning MAOs given in the output before looking at the numbers for atomic charges and shared electron numbers.

$mao selection options
to specify how MAOs are selected per atom.

Available options are:

a) for the way of sorting MAOs of each atom:

  eig
  MAOs are sorted according to their eigenvalue (those with largest EW finally are chosen). This is the default.

  occ
  MAOs are sorted according to their occupation; note that the number of all occupation is NOT the number of electrons in the system. This option is kept rather for historical reasons.

b) for the determination of the number of MAOs:

  fix
  A fixed number of MAOs is taken for each atom; usually this is the number of shells up to the complete valence shell, e.g. 5 for B-Ne, 6 for Na-Mg, etc. Exceptions are Elements Sc (Y, La), Ti (Zr, Hf), V (Nb, Ta) for which not all five d-shells are included, but only 2, 3 or 4, respectively. This modification leads to better agreement with partial charges calculated by an ESP-fit.
thr \textit{<real>}

All MAOs with an eigenvalue larger than \textit{<real>} are chosen; default is \textit{<real>=0.1}. This and the following two options are not valid in connection with \textit{occ}.

\textbf{max}

Maximum of numbers calculated from \textit{fix} and \textit{thr}=0.1 is taken.

\textbf{mix}

2:1 mixture of \textit{fix} and \textit{thr}=0.1. This choice gives best agreement (statistical) with charges from an ESP-fit. It is the default choice.

c) for additional information about MAOs:

\textbf{info}

Eigenvalues and occupations for each MAO are written to output.

\textbf{dump}

Entire information about each MAO is written to output. Lengthy.

Further for each atom the number of MAOs and the sorting mode can be specified individually in lines below this keyword. Example:

\texttt{atom 1,3-4 eig 7}

leads to choice of the 7 MAOs with largest eigenvalue at atoms 1, 3-4.

\texttt{$\$localize$}

enables the generation of localized molecular orbitals (LMOs) using Boys localization. By default, all occupied orbitals are included, localized orbitals are written (by default in the AO-basis) to file(s) \texttt{lmo} in case of RHF and \texttt{lalpha} and \texttt{lbeta} in case of UHF orbitals. Note, that LMOs usually break the molecular symmetry; so, even for symmetric cases the AO (not the SAO) basis is used for the output. The localized orbitals are sorted with respect to the corresponding diagonal element of the Fock matrix in the LMO basis. In order to characterize these orbitals, dominant contributions of (canonical) MOs are written to standard output as well as results of a Mulliken PA for each LMO (for plotting of LMOs see option \texttt{$\$pointval$}).

The keyword allows for following options (to be written in the same line):

\textbf{mo list of MOs}

Include only selected MOs (e.g. valence MOs) in localization procedure (numbering as available from \texttt{Eiger}).

\textbf{sweeps=integer}

maximum number of orbital rotations to get LMOs; default value is 10000 (sometimes not enough, in particular for highly delocalised systems).

\textbf{thrcont=real}

lower threshold for displaying MO and Mulliken contributions (default: 0.1).

\textbf{CAO}

LMOs are written to file in the CAO basis (instead of AO).
pipmez
Pipek-Meze localization is used.

$wfn$
triggers the generations of a wfn file. It can be used in dscf/ridft single-point calculations or in ricc2/egrad gradient calculations.

$esp\_fit$
fits point charges at the positions of nuclei to electrostatic potential arising from electric charge distribution (also possible for two-component calculations, for UHF cases also for spin density). For this purpose the ("real") electrostatic potential is calculated at spherical shells of grid points around the atoms. By default, Bragg-Slater radii, $r_{BS}$, are taken as shell radii, for each atom the number of points is given by $1000 \cdot r_{BS}^2$, the total number of points is the sum of points for each atom reduced by the number of points of overlapping spheres. Non-default shells (one or more) can be specified as follows:

```
$\text{esp}\_\text{fit}
\text{shell } i1 \, s1
\text{shell } i2 \, s2
:\n```

Integer numbers $i$ define the number of points for the respective shell, real numbers $s$ constants added to radii (default corresponds to one shell with $s=1.0$).

A parameterization very close to that by Kollman (U.C. Singh, P.A. Kollman, J. Comput. Chem. 5(2), 129-145 (1984)) may be obtained by

```
$\text{esp}\_\text{fit kollman}
```

Here five shells are placed around each atom with $r=1.4* r_{vdW} + k$, $k=0$pm, 20pm, 40pm, 60pm, 80pm, and $r_{vdW}$ are the van-der-Waals radii of the atoms.

$pointval$
drives the calculation of space-dependent molecular quantities at 3D grids, planes, lines or single points. Without further specifications the values of densities are plotted on a three-dimensional grid adapted to the molecular size. Data are deposed to output files (suffix .plt) that can be visualized directly with the gOpenMol program. In case of RHF-dscf/ridft calculations you get the total density on file td.plt, for UHF-dscf/ridft calculations one gets both values for the total density ($D^\alpha + D^\beta$) on td.plt and the "spin density" ($D^\alpha - D^\beta$) on sd.plt. For mpgrad/rimp2 calculations one gets in the RHF case the total density ($D^{\alpha\text{(SCF+MP2)}}$) on td.plt and the MP2 contribution on mp2d.plt and in the UHF case one obtains the total density ($D^{\alpha\text{(SCF+MP2)}} + D^{\beta\text{(SCF+MP2)}}$) on td.plt, the "spin density" ($D^{\alpha\text{(SCF+MP2)}} - D^{\beta\text{(SCF+MP2)}}$) on td.plt, and the respective MP2 contributions on files mp2d.plt and mp2sd.plt. For egrad it is similar, just replace in the filenames mp2 by e.
Integration of density (if absolute value greater than $\text{eps}$) within a sphere (origin $x, y, z$, radius $r$) is performed for

\[ \text{$pointval$ integrate $x$ $y$ $z$ $r$ $eps$} \]

By default the origin is at $(0,0,0)$, the radius is chosen large enough to include the whole 3D box and all contributions are regarded ($\text{eps}=0$).

Data different from total and spin densities are generated by following (combinable) settings (to be written in the same line as statement $\text{pointval}$):

- **pot** leads to calculation of electrostatic potential arising from electron densities, nuclei and—if present—constant electric fields and point charges. The densities used for calculation of potentials are the same as above; the respective filenames are generated from those of densities by replacement of the "d" (for density) by a "p" (for potential). By "pot only" only the electronic contribution to the electrostatic potential is calculated.

- **fld** calculation of electric field. Note, that for 3D default output format (`.plt`, see below) only norm is displayed. Densities used are the same as above, filenames are generated from those of densities by replacement of "d" (for density) by "f" (for field).

- **mo list of MO numbers**

  calculation of amplitudes of MOs specified by numbers referring to the numbering obtained e.g. from the tool **eiger** in the same format. The respective filenames are self-explanatory and displayed in the output. Note, that also in MP2 and excited state calculations the HF/DFT ground state orbitals are plotted (and not natural MP2/excited orbitals).

  **Two-component cases**: The density of the spinors specified by numbers referring to the numbering obtained e.g. from the file **EIGS** are visualized. By setting the keyword **minco** also the amplitudes of the spinor-parts are calculated, whose weights (the probability of finding the electron in this part) lie above the threshold.

- **lmo list of LMO numbers**

  calculation of amplitudes of LMOs (previously generated by $\text{localize}$) ordered by the corresponding diagonal element of the Fock matrix in the LMO basis.

- **nmo list of NMO numbers**

  calculation of amplitudes of NMOs (previously generated by $\text{natural orbitals file=natural}$ and $\text{natural orbital occupation file=natural}$

  $\text{dens}$ has to be set, if additionally to one of the above quantities also the density is to be computed.
21.2. FORMAT OF KEYWORDS AND COMMENTS

xc calculation of the Kohn-Sham exchange-correlation potential. It is only valid for DFT calculations and it works for all exchange-correlation functionals, including LHF. Note that for hybrid functionals, only the Kohn-Sham part of the potential will be computed (the HF part is a non-local-operator and can’t be plotted). For GGA functional the full potential will be computed (local and non-local terms).

For line plots the output file is tx.vec. For UHF calculations the output files are tx.vec (alpha-spin potential) and sx.vec (beta-spin potential).

For a line plot the file has three columns: 1: total potential 2: local term (or Slater-potential for LHF) 3: non-local terms or Correction term for LHF.

Output formats may be specified by e.g. fmt=xyz if written to the same line as $pointval. Supported are:

xyz in case of scalars (density, (L)MO amplitudes, electrostatic potential) this format is: \((x,y,z,f(x,y,z))\). In case of vectors components of the vector and its norm are displayed. This format is valid for all types of grid (3D, plane, line, points, see below), it is the default format in case of calculation of values at single points. Output file suffix is .xyz.

plt only for 3D, default in this case. Data are written to binary files that can be directly read by gOpenMol. Note, that this output is restricted to scalar quantities; thus in case of vectors (E-field) only the norm is plotted. Output file suffix is .plt.

map only for 3D. Data are written to ASCII files that can be imported by e.g. gOpenMol. Note, that this output is restricted to scalar quantities; thus in case of vectors (E-field) only the norm is plotted. Output file suffix is .map.

txt a format compatible with gOpenMol for visualization of vectors \(v\). The format is \(x,y,z,v_x,v_y,v_z\).

vec for planes and lines (default in these cases). In case of a line specified by \(\alpha \cdot \vec{v}\) (see below) output is \(\alpha, f(x,y,z)\) for scalars, for vectors components and norm are displayed. vectors. Analogously, in case of planes it is \(\alpha,\beta, f(x,y,z)\). The output (file suffix .vec) may be visualized by plotting programs suited for two-dimensional plots. A command file (termed gnuset) to get a contour plot by gnuplot is automatically generated.

cub only for 3D, writes out data in Cube format which can be imported by many external visualization programs.

For 3D grids non-default boundaries, basis vector directions, origin and resolutions may be specified as follows:

$pointval
grid1 vector 0 3 0 range -2,2 points 200
grid2 vector 0 0 -7 range -1,4 points 300
grid3 vector 1 0 0 range -1,1 points 300
Grid vectors (automatically normalized) now are \((0,1,0),(0,0,-1),(1,0,0)\), the grid is centered at \((1,1,1)\), and e.g. for the first direction 200 points are distributed between -2 and 2.

In particular for 2D plots it is often useful to shift and rotate the grid plane such that some particular atoms are located in the plot plane. This can be achieved with the \texttt{orient} option, which accepts as additional input a list of atoms, e.g. the atoms 4-5 and 9 in the following example:

\begin{verbatim}
$pointval
    grid1 vector 0 3 0 range -2,2 points 200
    grid2 vector 0 0 -7 range -1,4 points 300
    orient 4-5,9
    rotate 25 3
\end{verbatim}

This will shift the origin of the plot into the center of mass of the specified set of atoms and align the grid axes with their principal axes: If two or more atoms are specified which lie on one line grid axis 1 is rotated into this line. If the specified subset consists of three atoms located in a plane the grid axis 1 and 2 are rotated into this plane. If more than three atoms are specified the grid axis 1 and 2 are rotated into a plane which fitted to the position of all specified atoms. With the \texttt{rotate} option the grid can be rotated around a grid axis. It accepts as input the rotation angle the index (1/2/3) of the grid axis around which the grid will be rotated.

Grids of lower dimensionality may be specified (in the same line as \$pointval) by typing either \texttt{geo=plane} or \texttt{geo=line} or \texttt{geo=point} The way to use is best explained by some examples:

\begin{verbatim}
$pointval geo=plane
    grid1 vector 0 1 0 range -2,2 points 200
    grid2 vector 0 0 1 range -1,4 points 300
    origin 1 1 1
\end{verbatim}

Values are calculated at a plane spanned by vectors \((0,1,0)\) and \((0,0,1)\) centered at \((1,1,1)\).

\begin{verbatim}
$pointval geo=line
    grid1 vector 0 1 0 range -2,2 points 50
    origin 0 0 1
\end{verbatim}

Values are calculated at a line in direction \((0,1,0)\) centered at \((0,0,1)\). Output format as above.
21.2. FORMAT OF KEYWORDS AND COMMENTS

Values are calculated at the two points \((7.0, 5.0, 3.0)\) and \((0.0, 0.0, 7.0)\).

Plane-averaged density can be computed by

\[
\begin{align*}
\text{\$pointval dens averagez fmt=vec} \\
\text{grid1 vector 1 0 0 range -10,10 points 100} \\
\text{grid2 vector 0 1 0 range -10,10 points 100} \\
\text{grid3 vector 0 0 1 range -20,20 points 200} \\
\text{origin 0 0 0}
\end{align*}
\]

The generated file \(td.vec\) will contain the quantity

\[
\rho(z) = \int \int dxdy \rho(x, y, z)
\]  

(21.1)

### 21.2.23 Keywords for Module frog

The \textit{ab initio} molecular dynamics (MD) program \textit{frog} needs a command file named \textit{mdmaster}. The interactive \textit{Mdpreg} program manages the generation of \textit{mdmaster} and associated files. It is always a good idea to let \textit{Mdpreg} check over \textit{mdmaster} before starting an MD run. \textit{Mdpreg} has online-help for all menus.

In this implementation of \textit{ab initio} MD, time is divided into steps of equal duration \(\Delta t\). Every step, the energy and its gradient are calculated and these are used by the \textit{frog} to work out the new coordinates for the next step along the dynamical trajectory. Both the accuracy of the trajectory and the total computation time thus depend crucially on the time step chosen in \textit{Mdpreg}. A bad choice of timestep will result in integration errors and cause fluctuations and drift in the total energy. As a general rule of thumb, a timestep \(\Delta t\) should be chosen which is no longer than one tenth of the shortest vibrational period of the system to be simulated.

Note that \textit{Mdpreg} will transform velocities so that the total linear and angular momentum is zero. (Actually, for the Leapfrog algorithm, initial velocities are \(\Delta t/2\) before the starting time).

The following keywords are vital for \textit{frog}:

\texttt{\$nsteps 75}

Number of MD time steps to be carried out. \texttt{\$nsteps} is decreased by 1 every time \textit{frog} is run and \textit{JOBEX -md} stops when \texttt{\$nsteps} reaches 0.

\texttt{\$natoms 9}

Number of atoms in system.

\texttt{\$current file=mdlog.aa}

The file containing the current position, velocity, time and timestep, that is, the input configuration. During an MD run the \texttt{\$current} information is generally kept at the end of the \texttt{\$log} file.
$log file=mdlog.ZZ
The file to which the trajectory should be logged, i.e. the output: t=time (a.u.);
atomic positions x,y,z (Bohr) and symbols at t;
timestep (au) $\Delta t$;
atomic symbols and velocities x,y,z (au) at $t-(\Delta t/2)$;
kinetic energy (H) interpolated at $t$, \textit{ab initio} potential energy (H) calculated at $t$,
and pressure recorded at the barrier surface (atomic units, 1 au = 29.421 TPa) during
the corresponding timestep;
\textit{ab initio} potential energy gradients x,y,z (H/Bohr) at $t$.
This file can be manipulated with log2 tools after the MD run (Section 1.5).

$turbomole file=control$
Where to look for TURBOMOLE keywords $grad$ etc.

$md\_status$
The status of the MD run is a record of the action carried out during the previous MD
step, along with the duration of that step. The format matches that of $md\_action
below.
Canonical dynamics is supported using the Nosé-Hoover thermostat. This option can
be enabled in \texttt{Mdprep} or by the following syntax:

$md\_status
  canonical T=500 t=100
  from t= -25.0000000000 until t= 0.00000000000

Here, T specifies the temperature of the thermostat in K (500 K in the example) and
t specifies the thermostat relaxation time in a.u. (100 a.u. in the example). It is
advisable to choose the thermostat relaxation 2-10 times larger than the time step.
Note that user-defined actions are presently not supported in canonical dynamics
mode.

These are optional keywords:

$seed -123$
Integer random number seed

$title
Arbitrary title

$log\_history
100 mdlog.P
71 mdlog.Q

$ke\_control
  length  50
  response 1
To determine the trends in kinetic energy and total energy (average values and overall drifts) it is necessary to read the history of energy statistics over the recent MD steps. The number of MD steps recorded so far in each log file are therefore kept in the $log_history entry: this is updated by the program each step. The length of records needed for reliable statistics and the number of steps over which changes are made to kinetic energy (response) are specified in $ke_control.

$barrier angstroms
  type  elps
  limits  5.0 10.0 7.5
  constant  2.0
  springlen  1.0
  temperature 300.0

$barrier specifies a virtual cavity for simulating condensed phases. The optional flag, angstroms, can be used to indicate that data will be entered in Ångströms rather than Bohr.

type
  can be one of orth, elps, or none, for orthorhombic, ellipsoidal, or no barrier (the default) respectively.

limits
  are the +x,y,z sizes of the cavity. In this case, an ellipsoid with a major axis of 20 Å along y, semi-major of 15 Å on z, and minor of 10 Å on x.

constant
  is the Hooke’s Law force constant in atomic units of force (H/Bohr) per length unit. Here, it is 2.0 H/Bohr/Ångström, a bastard combination of units.

springlen
  is the effective limit to the restorative force of the barrier. For this system, an atom at 5 Å into the barrier will feel the same force as at 1.0 Å.

temperature
  denotes the temperature of the cavity walls in Kelvin. If the system quasi-temperature is below this setpoint, particles will be accelerated on their return to the interior. Alternately, they will be retarded if the system is too warm. A temperature of 0.0 K will turn off wall temperature control, returning molecules to the system with the same momentum as when they encountered the barrier.

$constraints angstroms
  tolerance  0.05
  adjpercyc  0.25
  type H O 0.9 1.2
  type F C 0.0 1.7
$\text{constraints}$

specifies and/or automatically generates atomic distance constraints. The optional flag, \texttt{angstroms}, can be used to indicate that data will be entered in Ångströms rather than Bohr.

\textbf{tolerance}

is the convergence criterion for application of constraints. All distances must be within +/- \texttt{tolerance} of the specified constraint. Additionally, the RMS deviation of all constrained distances must be below 2/3 of \texttt{tolerance}.

\textbf{adjpercyc}

is the fraction of the total distance correction to be applied on each constraint iteration.

\textbf{type X A const rmax}

commands \texttt{frog} to find the closest A atom to each atom X that is closer than \texttt{rmax} and apply \texttt{const}. The first \texttt{type} line above examines each H atom and looks for any O atoms within 1.2 Å. The shortest distance, if any, is then fixed at 0.9 Å. Similarly, the second \texttt{type} line binds each F to the closest C within 1.7 Å, but since \texttt{const}=0.0, that distance is fixed at the current value. The third \texttt{type} line attaches H atoms to the appropriate nearby C, but at the current average H-C distance multiplied by the absolute value of \texttt{const}.

Explicitly specified constraints are listed by atom index and supercede auto-generated constraints. A positive third number fixes the constraint at that value, while zero fixes the constraint at the current distance, and a negative number unsets the constraint.

The output of \texttt{frog} contains the full list of constrained atom pairs and their current constraints in explicit format.

User-defined instructions allow the user to tell \texttt{frog} to change some aspect of the MD run at some point in time \texttt{t=real number}. The same format is used for $\texttt{md\_status}$ above. Here is an example:

\begin{verbatim}
$md\_action
  fix total energy from t=2000.0
  anneal from t=2500.0
  free from t=3000.0
\end{verbatim}

In this example, starting from the time 2000.0 a.u., velocities are to be scaled every step to keep average total energy constant. Then, from 2500.0 a.u., gradual cooling at the default
rate (annealing) is to occur until the time 3000.0 a.u., when free Newtonian dynamics will resume.

Here are all the possible instructions:

```md_action
fix temperature from t=<real>
fix total energy from t=<real>

These commands cause velocities to be scaled so as to keep the average kinetic energy (i.e. quasi-temperature) or the average total energy approximately constant. This is only possible once enough information about run history is available to give reliable statistics. (Keywords $log_history, $ke_control).
```

```md_action
set temperature at t=<real> to x=<real> K
set total energy at t=<real> to x=<real> H
set kinetic energy at t=<real> to x=<real> H
set position file=<filename> at t=<real>
set velocity file=<filename> at t=<real>
set velocity at t=<real> random
set velocity at t=<real> zero

At some time during the ab initio MD run the user can specify a new value for one of the dynamical variables. The old value is discarded. Single values are given by x=real number. Vectors must be read in frog format from file=file.
```

```md_action
anneal from t=<real>
anneal from t=<real> x=<real>
quench from t=<real>
quench from t=<real> x=<real> file=<file>
relax at t=<real>

In Simulated Annealing MD, the temperature of a run is lowered so as to find minimum-energy structures. Temperature may be lowered gradually by a small factor each step (anneal; default factor 0.905 over 100 steps) or lowered rapidly by reversing all uphill motion (quench; default factor -0.8 each step). The cooling factors may be changed from the default using x=. Another option allows the quenching part of the run to be logged to a separate file. Alternatively, a standard non-dynamical geometry optimization can be carried out in a subdirectory (relax).
```

```md_action
free from t=<real>

Finally, this instruction turns off any previous action and resumes free dynamics. This is the default status of an MD run.
$\textit{surface}_\textit{hopping}$

This keyword allows to carry out Tully-type \textit{fewest switches surface hopping} (SH) \cite{207}. This option is only available in combination with TDDFT. For the TDDFT surface hopping see Tapavicza et al. 2007 \cite{208}; for the current implementation see Tapavicza et al. 2011 \cite{209}. In the current implementation the surface hopping algorithm only allows switches between the first excited singlet state and the ground state. However, total energies of higher excited states can be computed during the MD simulation. The proper functioning of SH has only been tested for the option

$\textit{md}_\textit{action}$

\texttt{fix total energy from t= 0.00000000000}

To carry out SH dynamics simulations, the keyword $\textit{surface}_\textit{hopping}$ has to be added to the control and mdmaster file. In addition several keywords are required in the control file:

$\textit{nacme}$

needed to compute non-adiabatic couplings; this keyword requires the use of \texttt{weight derivatives} in section \texttt{dft}

$\textit{nac}$

keyword needed to collect Cartesian non-adiabatic coupling vectors along the trajectory

$\textit{exopt 1}$

keyword needed to ensure dynamics starting in $S_1$

$\textit{ex_energies file=ex_energies}$

collects excitation energies along the trajectory

$\textit{integral_ex file=integral_ex}$

collects time integration of excitation energies along the trajectory

$\textit{sh_coeffs file=sh_coeffs}$

collects amplitudes of the adiabatic states along the trajectory

$\textit{nac_matrix file=nac_matrix}$

collects NAC elements along the trajectory

Special caution has to be taken to control problems related to conical intersections \cite{210,211}. At geometries where conical intersections between the ground and excited state are present, DFT often exhibits singlet instabilities, which leads to imaginary excitation energies in linear response TDDFT; in this case the MD run is terminated. This problem can be circumvented by the use of the Tamm-Dancoff approximation (TDA) to TDDFT (see \cite{8}). In addition an optional keyword for the md_master file can be used:

$\textit{gap_threshold <real>}$

enforces a switch to the ground state in case the $S_1$-$S_0$ energy gap drops below $<$\texttt{real}> eV. As default a switch to $S_0$ is enforced if the $S_1$ TDDFT-TDA excitation energy becomes negative.
21.2. FORM OF KEYWORDS AND COMMENTS

Often times if a switch is enforced due to a negative TDA excitation energy the potential energy surface is discontinuous and limited numerical precision of the nuclear forces may lead to a loss of total energy conservation. In this case the nuclear velocities are rescaled to obtain a conserved total energy.

21.2.24 Keywords for Module mpshift

In order to control the program execution, you can use the following keywords within the control file:

$\texttt{csmp2}$

Switches on the calculation of the MP2 NMR shieldings. The required SCF shielding step will be performed in the same run. This flag will be set by the script mp2prep.

$\texttt{traloop \ n}$

specifies the number of loops (or ‘passes’) over occupied orbitals \( n \) when doing an MP2 calculation: the more passes the smaller file space requirements—but CPU time will go up. This flag will be set by the script mp2prep.

$\texttt{mointunit}$

Scratch file settings for an MP2 calculation. Please refer to Section 21.2.16 for a description of the syntax. This flag will be set by the script mp2prep.

$\texttt{csconv \ real}$

Sets the convergence threshold for the shielding constant of succeeding CPHF iterations. The unit is ppm and the default value is 0.01.

$\texttt{csconvatom \ integer}$

This selects the atom number for convergence check after each cphf iteration. After this convergence is reached all other atoms are checked, too (default: 1).

$\texttt{thime, thize, scftol, scfintunit, scfmo}$

have the same meaning as in dscf (see Section 21.2.6); Since mpshift works ‘semi-direct’ it uses the same integral storage.

$\texttt{scratch files}$

The scratch files allocated by mpshift can be placed anywhere in your file systems instead of the working directory by referencing their pathnames in this data group. All possible scratch files are listed in the following example:

$\texttt{scratch files}$

```plaintext
mpshift csssmat path1/file1
mpshift chsmat path2/file2
mpshift csdgsmat path3/file3
mpshift csusmat path4/file4
mpshift dens path5/file5
mpshift fock path6/file6
```
$\texttt{trast}, \texttt{trand} traloop-number

stands for traloop start and traloop end. Each loop or pass in MP2 chemical shift calculations can be done individually by providing the keywords $\texttt{trast}$ and $\texttt{trand}$. This can be used to do a simple parallelization of the run:

Create separate inputs for each traloop. Add

\begin{verbatim}
$\texttt{trast} <\text{number}>
$\texttt{trand} <\text{number}>
\end{verbatim}

in the control files, number goes from 1 to the number of $\texttt{traloops}$. Each calculation will create a restart file called $\texttt{restart.mpshift}$. To collect all steps and to do the remaining work, copy all restart files to one directory and rename them to $\texttt{restart.mpshift}.number$, add $\texttt{trast -1}$ and $\texttt{trand number_of_traloops}$ to the control file and start mpshift.

21.2.25 Keywords for Parallel Runs

On all systems the parallel input preparation is done automatically. Details for the parallel installation are given in Section 3.2.1. The following keywords are necessary for all parallel runs:

\begin{verbatim}
$\texttt{parallel_platform} architecture
\end{verbatim}

Currently the following parallel platforms are supported:

\begin{enumerate}
\item \textbf{SMP} for systems with very fast communication; all CPUs are used for the linear algebra part. Synonyms for SMP are:
   \begin{itemize}
   \item HP V-Class, SP3-SMP and HP S/X-Class
   \end{itemize}
\item \textbf{MPP} for systems with fast communication like Fast-Ethernet, the number of CPUs that will be taken for linear algebra part depends on the size of the matrices. Synonyms for MPP are:
   \begin{itemize}
   \item SP3 and linuxcluster
   \end{itemize}
\item \textbf{cluster} for systems with slow communication, the linear algebra part will be done on one single node. Synonyms for cluster are:
   \begin{itemize}
   \item HP Cluster and every platform that is not known by TURBOMOLE
   \end{itemize}
\end{enumerate}
21.2. FORMAT OF KEYWORDS AND COMMENTS

Similar to SMP, but here the server task is treated differently: the MPI implementation on the SGIs would cause this task to request too much CPU time otherwise.

If you want to run mpgrad, $traloop has to be equal to or a multiple of the number of parallel workers.

For very large parallel runs it may be impossible to allocate the scratch files in the working directory. In this case the $scratch files option can be specified; an example for a dscf run is given below. The scratch directory must be accessible from all nodes.

$scratch files

dscf dens  /home/dfs/cd00/cd03_dens

dscf fock  /home/dfs/cd00/cd03_fock

dscf dfock /home/dfs/cd00/cd03_dfock

dscf ddens /home/dfs/cd00/cd03_ddens

dscf xsv   /home/dfs/cd00/cd03_xsv

dscf pulay /home/dfs/cd00/cd03_pulay

dscf statistics /home/dfs/cd00/cd03_statistics

dscf errvec /home/dfs/cd00/cd03_errvec

dscf oldfock /home/dfs/cd00/cd03_oldfock

dscf oneint /home/dfs/cd00/cd03_oneint

For all programs employing density functional theory (DFT) (i.e. dscf/grad and ridft/rdgrad) $pardft can be specified:

$pardft

tasksize=1000

memdiv=0

The tasksize is the approximate number of points in one DFT task (default: 1000) and memdiv says whether the nodes are dedicated exclusively to your job (memdiv=1) or not (default: memdiv=0).

For dscf and grad runs you need a parallel statistics file which has to be generated in advance. The filename is specified with

$2e-ints_shell_statistics file=DSCF-par-stat

or

$2e-ints’_shell_statistics file=GRAD-par-stat

respectively.

The statistics files have to be generated with a single node dscf or grad run. For a dscf statistics run one uses the keywords:

$statistics  dscf parallel

$2e-ints_shell_statistics file=DSCF-par-stat
$parallel_parameters
    maxtask=400
    maxdisk=0
    dynamic_fraction=0.300000

and for a grad statistics run:

$statistics  grad parallel
$2e-ints'_shell_statistics      file=GRAD-par-stat
$parallel_parameters
    maxtask=400

maxtask is the maximum number of two-electron integral tasks,
maxdisk defines the maximum task size with respect to mass storage (MBytes) and
dynamic_fraction is the fraction of two-electron integral tasks which will be allocated
dynamically.

For parallel grad and rdgrad runs one can also specify:

$grad_send_dens

This means that the density matrix is computed by one node and distributed to the other
nodes rather than computed by every slave.

In the parallel version of ridft, the first client reads in the keyword $ricore from the
control file and uses the given memory for the additional RI matrices and for RI-integral
storage. All other clients use the same amount of memory as the first client does, although
they do not need to store any of those matrices. This leads to a better usage of the available
memory per node. But in the case of a big number of auxiliary basis functions, the RI
matrices may become bigger than the specified $ricore and all clients will use as much
memory as those matrices would allocate even if that amount is much larger than the given
memory. To omit this behavior one can use:

$ricore_slave integer

specifying the number of MBs that shall be used on each client.

For parallel jobex runs one has to specify all the parallel keywords needed for the different
parts of the geometry optimization, i.e. those for dscf and grad, or those for ridft and
rdgrad, or those for dscf and mpgrad.
Chapter 22

Sample control files

22.1 Introduction

The file control is the input file for TURBOMOLE which directly or by cross references provides the information necessary for all kinds of runs and tasks. control is usually generated by define, the input generator. The following sample control files cover a variety of methods and systems. The keywords themselves are explained in Chapter 21.
22.2 NH₃ Input for a RHF Calculation

Main File control

$title
NH₃ c3v SVP
$operating system unix
$symmetry c3v
$coord  file=coord
$intdef  file=coord
$atoms
  n  1
    basis =n def-SVP
  h  2-4
    basis =h def-SVP
$people  AO
$basis  file=basis
$rundimensions
  dim(fock,dens)=495
  natoms=4
  nshell=15
  nbf(CAO)=30
  nbf(AO)=29
  dim(trafo[SAO<-->AO/CAO])=69
  rhfshells=1
$scfmo  file=mos
$closed shells
  a1  1-3
     (2)
  e   1
     (2)
$scfiterlimit  30
$scfconv  7
$thize  .10000000E-04
$thime  5
$scfdamp  start=.500  step=.050  min=.100
$scfdump
$scfintunit
  unit=30  size=0  file=twoint
$scfdiis  start=0.5
$drvopt
  cartesian  on
  basis     off
  global    off
  hessian   on
dipole on
nuclear polarizability

\$interconversion\ off
qconv=1.d-10
maxiter=25

\$optimize
internal on
cartesian off
global off
basis off logarithm

\$coordinateupdate
dqmax=0.3
interpolate on
statistics 5

\$forceupdate
ahlrichs numgeo=0 mingeo=3 maxgeo=4 modus=<g|dq> dynamic fail=0.1
threig=0.005 reseig=0.005 thrbig=3.0 scale=1.00 damping=0.0

\$forceinit on
diag=default

\$energy file=energy
\$grad file=grad
\$forceapprox file=force
\$lock off
\$last step define

\$end

File coord

\$coord
-0.000000000000000 0.000000000000000 0.54561506935122 n
-.8780623311566 1.52084856970468 -.18187168978374 h
-.8780623311566 -1.52084856970468 -.18187168978374 h
1.75612466223131 .000000000000000 -.18187168978374 h

\$intdef
# definitions of internal coordinates
1 k 1.000000000000000 stre 4 1 val= 1.90084
2 k 1.000000000000000 bend 4 3 1 val= 106.27756
1.000000000000000 bend 3 2 1
1.0000000000000000 bend 2 4 1

\$end
File basis

$basis

* 
 n def-SVP

# n (7s4p1d) / [3s2p1d] {511/31/1}

* 
 5 s
 1712.8415853 -.53934125305E-02
 257.64812677 -.40221581118E-01
 58.458245853 -.17931144990
 16.198367905 -.46376317823
 5.0052600809 -.44171422662
 1 s
  .5873185671 1.0000000000
 1 s
  .18764592253 1.0000000000
 3 p
 13.571470233 -.40072398852E-01
 2.9257372874 -.21807045028
  .79927750754 -.51294466049
 1 p
  .21954348034 1.0000000000
 1 d
 1.0000000000 1.0000000000

* 
 h def-SVP

# h (7s) / [3s] {511}

* 
 3 s
 13.010701000 .19682158000E-01
 1.9622572000 .13796524000
  .44453796000 .47831935000
 1 s
  .12194962000 1.0000000000
 1 p
  .80000000000 1.0000000000

* 
 $end

File mos

$scfmo expanded format(4d20.14)
22.2. NH$_3$ INPUT FOR A RHF CALCULATION

1 a1  eigenvalue=-.15633041862301D+02 nsaos=10
   .98699003163455D+00-.47221435341751D-01 .55873125006179D-02-.48016374887169D-02
   .26746008768233D-02 .20823779196149D-03 .14270460008808D-01 .90849517503597D-02
   .58676121352806D-03 .29091871198884D-03

2 a1  eigenvalue=-.99896275238736D+00 nsaos=10
   .26412162337482D+00 .51846472345768D+00 .37623729061179D+00-.77139882704089D-02
   -.47252329287316D-02-.21494050853221D-02 .11795673774658D+00 .83316086019184D-01
   -.11292039334888D-01-.27038186251429D-02

3 a1  eigenvalue=-.57101279949392D+00 nsaos=10
   -.35584199011701D-01-.96938258881594D-01-.70254605702716D-01 .65569041318341D+00
   -.44746149963029D+00 .40094287741992D-03 .51691151834284D-01 .47722350097160D-01
   .19189122068531D-02 .56638497851180D-03

1 e  eigenvalue=-.64374209294851D+00 nsaos=9
   -.49313475446075D+00 .33757893447603D+00-.76142296567409D-04-.74524664248740D-04
   -.26407572210452D+00-.22619038902975D+00-.50035170531670D-05-.12199166245418D-03
   .63021657663245D-04
$end
22.3 NO$_2$ input for an unrestricted DFT calculation

Main File control

$title
NO2 c2v UKS SVP
$operating system unix
$symmetry c2v
$coord file=coord
$intdef file=coord
$atoms
n 1
  basis =n def-SVP
o 2-3
  basis =o def-SVP
$popple AO
$basis file=basis
$rundimensions
  dim(fock,dens)=1098
  natoms=3
  nshell=18
  nbf(CAO)=45
  nbf(AO)=42
  dim(trafo[SAO<-->AO/CAO])=85
  rhfshells=2
$uhfmo_alpha none file=alpha
$uhfmo_beta none file=beta
# none : hamilton core guess will be made
# files alpha and beta will be generated by the program
$uhf
$alpha shells
  a1  1-6  (1)
  a2  1   (1)
  b1  1-4  (1)
  b2  1   (1)
$beta shells
  a1  1-5  (1)
  a2  1   (1)
  b1  1-4  (1)
  b2  1   (1)
$scfiterlimit 30
$scfconv  7
$thize .10000000E-04
22.3. $NO_2$ INPUT FOR AN UNRESTRICTED DFT CALCULATION

```plaintext
$thime 5
$scfdamp start=1.500 step= .050 min= .100
$scfdump
$scfintunit
unit=30 size=2 file=/work/user/twoint
$scfdiis start=0.5
$scforbitalshift closedshell=.3
$drvopt
cartesian on
basis off
global off
hessian on
dipole on
nuclear polarizability
$interconversion off
qconv=1.d-10
maxiter=25
$optimize
internal on
cartesian off
global off
basis off logarithm
$coordinateupdate
dqmax=0.3
interpolate on
statistics 5
$forceupdate
ahlrichs numgeo=0 mingeo=3 maxgeo=4 modus=<g|dq> dynamic fail=0.1
threig=0.005 reseig=0.005 thrbig=3.0 scale=1.00 damping=0.0
$forceinit on
diag=default
$energy file=energy
$grad file=grad
$forceapprox file=force
$lock off
$dft
functional b-p
gridsize m3
$last step define
$end
```
File coord

$coord

  .00000000000000  .00000000000000  -1.00494155217173  n
  1.85766051386774  .00000000000000  .50247077608587  o
  -1.85766051386774  .00000000000000  .50247077608587  o

$intdef

# definitions of internal coordinates

  1  k  1.00000000000000  stre  2  1  val=  2.39232
  2  d  1.00000000000000  stre  3  1  val=  2.39232
  3  k  1.00000000000000  bend  2  3  1  val= 101.88429

$end

File basis

$basis

*  
n def-SVP  

# n  (7s4p1d) / [3s2p1d]  {511/31/1}  

*  

  5  s  
  1712.8415853  -.53934125305E-02
  257.64812677  -.40221581118E-01
  58.458245853  -.17931144990
  16.198367905  -.46376317823
  5.0052600809  -.44171422662
  1  s
  .58731856571  1.0000000000
  1  s
  .18764592253  1.0000000000
  3  p
  13.571470233  -.40072398852E-01
  2.9257372874  -.21807045028
  .79927750754  -.51294466049
  1  p
  .21954348034  1.0000000000
  1  d
  1.0000000000  1.0000000000

*  
o def-SVP  

# o  (7s4p1d) / [3s2p1d]  {511/31/1}  

*  

  5  s  

22.3. NO₂ INPUT FOR AN UNRESTRICTED DFT CALCULATION

2266.1767785  -.53431809926E-02
340.87010191  -.39890039230E-01
77.363135167  -.17853911985
21.479644940  -.46427684959
6.6589433124  -.44309745172

1 s
.80975975668  1.0000000000

1 s
.25530772234  1.0000000000

3 p
17.721504317  .43394573193E-01
3.8635505440  .23094120765
1.0480920883  .51375311064

1 p
.27641544411  1.0000000000

1 d
1.2000000000  1.0000000000

* 
$end
22.4 TaCl$_5$ Input for an RI-DFT Calculation with ECPs

Main File control

$title
$operating system unix
$symmetry d3h
$coord file=coord
$intdef file=coord
$atoms
  ta 1
    jbas=ta def-SVP
    basis =ta def-SVP
    ecp  =ta def-ecp
  cl 2-6
    jbas=cl def-SVP
    basis =cl def-SVP
$pople AO
$basis file=basis
$ecp file=basis
$rundimensions
  dim(fock,dens)=7662
  natoms=6
  nshell=51
  nbf(CAO)=122
  nbf(AO)=115
  dim(trafo[SAO<-->AO/CAO])=346
$scfmo none file=mos
  # none : hamilton core guess will be made
  # file mos will be generated by the program
$scfiterlimit 30
$scfconv 6
$thize .10000000E-04
$thime 5
$scfdamp start= .900 step= .050 min= .100
$scfdump
$scfintunit
  unit=30 size=0 file=twoint
$scfdiis start=0.5
$drvopt
  cartesian on
  basis    off
  global   off
hessian on
dipole on
nuclear polarizability
$interconversion off
qconv=1.d-10
maxiter=25
$optimize
internal on
cartesian off
global off
basis off logarithm
$coordinateupdate
dqmax=0.3
interpolate on
statistics 5
$forceupdate
ahlrichs numgeo=0 mingeo=3 maxgeo=4 modus=<g|dq> dynamic fail=0.1
threig=0.005 reseig=0.005 thrbig=3.0 scale=1.00 damping=0.0
$forceinit on
diag=default
$energy file=energy
$grad file=grad
$forceapprox file=force
$lock off
$dft
functional b-p
gridsize m3
$last step define
$ricore 20
$ridft
$jbas file=auxbasis
$closed shells
a1' 1-11 ( 2 )
\hfill \text{a1'} 1-11 ( 2 )
a2' 1-2 ( 2 )
\hfill \text{a2'} 1-2 ( 2 )
e' 1-10 ( 2 )
\hfill \text{e'} 1-10 ( 2 )
a2'' 1-8 ( 2 )
\hfill \text{a2''} 1-8 ( 2 )
e'' 1-4 ( 2 )
\hfill \text{e''} 1-4 ( 2 )
$end

File coord
$coord
\begin{verbatim}
.00000000000000 .00000000000000 .00000000000000 ta
\end{verbatim}
2.19392179448315 -3.79998401587749 .00000000000000 cl
2.19392179448315  3.79998401587749 .00000000000000 cl
-4.38784358896629  .00000000000000 .00000000000000 cl
   .00000000000000  .00000000000000  4.46615918865523 cl
   .00000000000000  .00000000000000 -4.46615918865523 cl

$intdef
# definitions of internal coordinates
   1 k 1.00000000000000 stre 1 2       val= 4.38784
   2 k 1.00000000000000 stre 1 5       val= 4.46616
$end

File basis

$basis
 *
 ta def-SVP
# ta  (7s6p5d) / [6s3p2d]  {21111/411/41}
 *
   2 s
 14.400000000 .99343296745
 12.000000000 -1.6510077975
  1 s
  5.0701477302  1.0000000000
  1 s
  .86033356487  1.0000000000
  1 s
  .37158938894  1.0000000000
  1 s
  .1074536254  1.0000000000
  1 s
  .3914277656E-01  1.0000000000
  4 p
  7.4188720000 .26979695152
  5.6984100000 -.46968874449
  1.1777211960 .50905100155
  .54478533555 .52298161137
  1 p
  .22309270117  1.0000000000
  1 p
  .43100000000E-01  1.0000000000
  4 d
  3.9738796278  -.52799310714E-01
  1.4528884813  .18558319471
.61042908544 0.42959071631
.24216276510 0.43497228232
1 d
.87909318337E-01 1.0000000000
*
cl def-SVP
# cl (7s5p) / [6s2p] {211111/41}
*
5 s
10449.827566 0.19708362484E-02
1571.7365221 0.14754727977E-01
357.12065523 0.66679112875E-01
100.25185935 0.17228924084
30.812727554 0.15883786100
3 s
51.923789434 -0.10009298909
5.7045760975 0.60841752753
2.3508376809 0.54352153355
1 s
.44605124672 1.0000000000
1 s
.16848856190 1.0000000000
5 p
307.66790569 -0.87801484118E-02
72.102015515 -0.6356335471E-01
22.532680262 -0.24016428276
7.8991765444 -0.47798866557
2.8767268321 -0.38515850005
1 p
.77459363955 1.0000000000
1 p
.21037699698 1.0000000000
1 d
.65000000000 1.0000000000
*
$ecp
*
ta def-ecp
*
ncore = 60 lmax = 3
# coefficient r^n exponent
f
12.0179609 2 2.0178811
s-f
1345.8806470 2 14.5464077
36.7668062 2 7.2732038
-12.0179609 2 2.0178811

p-f
378.4253015 2 9.9355653
22.2930909 2 4.9677824
-12.0179609 2 2.0178811

d-f
104.8839557 2 6.3473769
8.7558481 2 3.1736885
-12.0179609 2 2.0178811

* $end

File auxbasis

$jbas
*

$ ta def-SVP
*

3 s
15.521335  .493702989D+00
7.555743  .259256574D+01
3.699576  -.523168657D+01
1 s
1.820141  .262393615D+01
1 s
0.898838  .157711902D+01
1 s
0.445062  .200789711D+00
1 s
0.220729  .185974307D+00
1 s
0.109530  .765184411D-01
1 p
1.5024958  1.0
1 p
0.5629855  1.0
1 p
0.2281880  1.0
1 p
0.09507835  1.0
2 d
 1.337006  .190072032D-01
 0.599535  -.155214344D-01
1 d
 0.280427  -.138946250D-01
1 d
 0.133078  -.895263676D-02
1 f
 1.1428211  1.0
1 f
 0.4395465  1.0
1 f
 0.1758186  1.0
3 g
 1.630421  .100251139D+00
 0.747093  .737448223D-01
 0.349040  .276219913D-01
1 g
 0.164143  .546316580D-02
*
*
c1 def-SVP
*
8 s
 4097.080409  .198054511D+01
 1203.083193  .530973450D+01
 386.280948  .132352655D+02
 135.337690  .107149960D+02
 51.567046  -.132565114D+01
 21.261034  .271180364D+01
 9.420135  .754640511D+01
 4.445228  .173603618D+01
1 s
 2.209399  -.140197496D+01
1 s
 1.141575  .982719736D+00
1 s
 0.604182  .464178589D+00
1 s
 0.322378  .369336889D+00
4 p
 51.8499902611  .359335506D-01
 17.5847835188  -.869599318D-01
6.49227239618          .721211200D-01
2.55889114714          -.634201864D-01
1  p
1.05118767781          .264152293D-01
1  p
.437994865757           -.197670692D-01
4  d
34.705550              -.548703710D-01
10.704427              -.619019402D-02
3.568067               .337450480D-01
1.249848               -.905232209D-01
1  d
0.445360               .418680075D-01
1  f
1.1872146118            1.0000000
1  g
1.30000000             1.0000000
*
$end
22.5 Basisset optimization for Nitrogen

Main File control

$title
Basisset-optimization for nitrogen SV(P)
$operating system unix
$symmetry oh

#--- uncomment following line to clean the basis-file after optimization ----
#$dump basis set
$coord file=coord
$user-defined bonds file=coord
$pople AO
$basis file=basis
$rundimensions
  dim(fock,dens)=141
  natoms=1
  nshell=6
  nbf(CAO)=15
  nbf(AO)=14
  dim(trafo[SAO<-->AO/CAO])=17
  rhfshells=2
$scfmo none file=mos
$roothaan 1
  a = 1  b = 2
$scfiterlimit 60
$scfconv 10
$thize 0.10000000E-04
$thime 5
$scfdamp start=1.500 step=0.050 min=0.100
$scfdump
$scfintunit
  unit=30  size=90  file=twoint
$scfdiis start=0.5
$scforbitalshift closedshell=.4
$drvopt
  cartesian off
#---- optimize basis! --> basis on ----
  basis on
  global off
  hessian on
  dipole on
  nuclear polarizability
$interconversion off
qconv=1.d-7
maxiter=25
$optimize
  internal off
cartesian off
global off
#---- optimize basis! -> basis on logarithm ----
basis on logarithm
$coordinateupdate
dqmax=0.3
interpolate on
statistics 5
$forceupdate
  ahlrichs numgeo=0 mingeo=3 maxgeo=4 modus=<g|dq> dynamic fail=0.6
  threig=0.005 reseig=0.005 thrbig=3.0 scale=1.00 damping=0.0
$forceinit on
diag=default
$energy file=energy
$grad file=gradient
#---- optimize basis! -> $egrad file=egradient ----
$egrad file=egradient
$forceapprox file=forceapprox
$lock off
$atoms
n 1
  basis =n def-SV(P)
$closed shells
alg 1-2 ( 2 )
$open shells type=1
tlu 1 ( 1 )
$end

File coord

$coord
  0.0000000000000 0.0000000000000 0.0000000000000  n
$user-defined bonds
$end

File basis

*
22.5. BASISSET OPTIMIZATION FOR NITROGEN

n def-SV(P)
# n (7s4p1d) / [3s2p1d] {511/31/1}
# use expopt to optimize exponents and contopt to optimize contractions
*
  5 s expopt contopt
  1712.8415853  0.53934125305E-02
  257.64812677  0.40221581118E-01
  58.458245853  0.17931144990
  16.198367905  0.46376317823
  5.0052600809  0.44171422662
  1 s expopt
  0.58731856571  1.0000000000
  1 s expopt
  0.18764592253  1.0000000000
  3 p expopt contopt
  13.571470233  0.40072398852E-01
  2.9257372874  0.21807045028
  0.79927750754  0.5129466049
  1 p expopt
  0.21954348034  1.0000000000
#  1 d
#  1.0000000000  1.0000000000
*

File mos

$scfmo scfconv=10 format(4d20.14)
# SCF energy is  -54.3329250250 a.u. (virial theorem =  2.000000001)
#
  1 alg  eigenvalue=-.15623888057347D+02 nsaos=3
  -.99166890864040D+00-.28420294406651D-01 .91519592317893D-02
  2 alg  eigenvalue=-.92524548524703D+00 nsaos=3
  0.30506869715453D+00-.65051761026701D+00-.44610487551870D+00
  3 alg  eigenvalue=0.748812299845801D+00 nsaos=3
  0.3075932935434D+00-.16295969601691D+01 0.16126161147521D+01
  1 t1u  eigenvalue=-.56865046629517D+00 nsaos=2
  0.67926397018841D+00.46005039868410D+00
  2 t1u  eigenvalue=0.96169069264790D+00 nsaos=2
 -.95675659621171D+00.10794148212163D+01
$end
22.6 ROHF of Two Open Shells

Extracts from control for O$_2$ in $D_{3d}$ Symmetry

# HF-SCF/SVP

# Reference: triplet-sigma in D3d
# This is a Roothaan case (as is D-infinity-h).
#
$coord
  0.0  0.0  1.08597397921317  o
  0.0  0.0  -1.08597397921317  o
$symmetry d3d
$closed shells
  a1g  1-3  (2)
  a2u  1-2  (2)
  eu   1   (2)
$open shells type=1
  eg   1   (1)
$roothaan  1
    a = 1  b = 2
$energy  SCF  SCFKIN  SCFPOT
         1  -149.4774402753  149.4799190239  -298.9573592992

# Reference: singlet-delta in D3d
# This is a Roothaan case (as is D-infinity-h).
#
$coord
  0.0  0.0  1.08597397921317  o
  0.0  0.0  -1.08597397921317  o
$symmetry d3d
$closed shells
  a1g  1-3  (2)
  a2u  1-2  (2)
  eu   1   (2)
$open shells type=1
  eg   1   (1)
$roothaan  1
    a = 1/2  b = 0
$energy  SCF  SCFKIN  SCFPOT
         1  -149.4297623470  149.4298692899  -298.8596316369
Extracts from control for O$_2$ in $D_{2h}$ symmetry

# HF-SCF/SVP

# Triplet-sigma in $D_{2h}$
#
$coord
0.0  0.0  1.08597397921317  o
0.0  0.0  -1.08597397921317  o

$symmetry  d2h$
$closed shells$
$ag  1-3   ( 2 )$
b1u  1-2   ( 2 )
b2u  1    ( 2 )
b3u  1    ( 2 )$

$open shells type=1$
b2g  1    ( 1 )
b3g  1    ( 1 )

$roothaan  1
a = 1    b = 2$

$energy  SCF   SCFKIN   SCFPOT$
  1  -149.4774402750  149.4798706643  -298.9573109393

# Singlet-delta in $D_{2h}$ : xx-yy component
# where x = b2g and y = b3g. In D-infinity-h, b2g and b3g combine to eg.
#
$coord
0.0  0.0  1.08597397921317  o
0.0  0.0  -1.08597397921317  o

$symmetry  d2h$
$closed shells$
$ag  1-3   ( 2 )$
b1u  1-2   ( 2 )
b2u  1    ( 2 )
b3u  1    ( 2 )$

$open shells type=1$
b2g  1    ( 1 )
b3g  1    ( 1 )

$roothaan  2$
$rohf$
1b2g-1b3g  a = 0    b = 2
1b2g-1b2g \ a = 1 \ b = 0
1b3g-1b3g \ a = 1 \ b = 0

$energy \ \ \ SCF \ \ \ SCFKIN \ \ \ SCFPOT
1 \ -149.4297623516 \ 149.4298351805 \ -298.8595975321

# Singlet-delta in D2h : xy+yx component
# (an example of the general type: [xy]-singlet)
# where in D2h x = b2g and y = b3g are of different symmetry.
# In D-infinity-h, b2g and b3g combine to eg; see the reference
# calculation in D3d above.
#
$coord
0.0 \ 0.0 \ 1.08597397921317 \ o
0.0 \ 0.0 \ -1.08597397921317 \ o

$symmetry d2h
$closed shells
ag \ 1-3 \ \ \ \ \ \ (2)
b1u \ 1-2 \ \ \ \ \ \ (2)
b2u \ 1 \ \ \ \ \ \ (2)
b3u \ 1 \ \ \ \ \ \ (2)

$open shells type=1
b2g \ 1 \ \ \ \ \ \ (1)
b3g \ 1 \ \ \ \ \ \ (1)

$roothaan 2
$rohf
1b2g-1b3g \ a = 1 \ b = -2
1b2g-1b2g \ a = 0 \ b = 0
1b3g-1b3g \ a = 0 \ b = 0

$energy \ \ \ SCF \ \ \ SCFKIN \ \ \ SCFPOT
1 \ -149.4297623501 \ 149.4298391833 \ -298.8596015334
Chapter 23

The Perl-based Test Suite
Structure

23.1 General

Testing the TURBOMOLE modules for correctness and speed is the first task once the coding is completed. It is subject to automatization and thus requires a structure which is as simple and flexible as possible. In the Perl-based test suite this is implemented by a Perl script TTEST which performs all the testing and benchmarking tasks and resides in the central scripts directory of the TURBOMOLE installation. The test examples are located in subdirectories of the TURBOTEST directory, grouped according to the modules modules to be tested and a rough short/long classification. The benchmark suite shows the same directory structure and is rooted in the TURBOBENCH directory.

The central idea of the Perl-based test suite is that only the specific information about an individual test example is included in its local directory along with the input and reference files. This information is stored in the criteria file CRIT which contains the program calls, test criteria, and specific reference timings. Running the test script creates a new test subdirectory, usually called like TESTDIR.i786-pc-linux-gnu, where the TURBOMOLE programs are run and the results are summarized in the protocol file TESTPROTOKOLL.

23.2 Running the tests

Starting a single test example is simple. Change to the test example of your choice and call the TTEST script without arguments. The test is started in a subdirectory named TESTDIR.sysname, where sysname is the current platform name as returned by the Sysname
script. The tested executable, a short description, and the test summary are output to the screen. Detailed information about the performed commands and results of all test criteria are found in the TESTPROTOKOLL file in the test subdirectory.

The default location for the binaries and scripts used for testing is the $TURBODIR directory. If you like to test some other, e.g., your local version of the TURBOMOLE binaries or scripts, you can specify the loading paths by the -l or -ls options for the binaries and scripts, respectively,

```
TTEST -l /usr/local/TURBOMOLE/bin/i786-pc-linux-gnu \
   -ls /usr/local/TURBOMOLE/scripts.
```

A specific executable can be chosen by the -x option,

```
TTEST -x /usr/local/TURBOMOLE/bin/i786-pc-linux-gnu/dscf.
```

If a test output is already present, e.g., in the TESTDIR directory, you may wish to check the results. This is accomplished by calling TTEST in check mode,

```
TTEST --check TESTDIR,
```

which compares the results in TESTDIR with the reference and writes the results to the CHECKPROTOKOLL file in the test directory.

Testing parts of the TURBOTEST directory structure or the entire test suite at once is performed by calling the TTEST script from the appropriate place. The test script works recursively, executing all test examples underneath its starting directory. This requires that the test examples be arranged in a TURBOTEST-like directory structure,

```
progname/short|long/example (e.g., dscf/short/H2O.SCF.E1),
```

and the TURBOTEST directory contain a DEFCRIT file with general test suite settings. If TTEST is started in the central TURBOTEST without any options, all available test examples are executed. By giving the list of module names (for full list, check TTEST -help) as argument to the script, the test can be restricted to these modules. The -short and -long options allow the user to select only the short or long test examples, respectively. Some examples of usage are given in the following table:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTEST dscf</td>
<td>called in the TURBOTEST directory, performs only the tests for Dscf module.</td>
</tr>
<tr>
<td>TTEST</td>
<td>called in the TURBOTEST/dscf directory, does the same.</td>
</tr>
<tr>
<td>TTEST -long</td>
<td>executes long examples for all modules.</td>
</tr>
<tr>
<td>TTEST ridft -short</td>
<td>performs all short examples from the ridft directory.</td>
</tr>
</tbody>
</table>
Recursive testing creates some additional files in the central TURBOTEST directory. The global protocol file TESTPROTOKOLL.sysname contains short result messages for all test and a list of errors occurred. The list of failed tests is also written to the PROBLEMS.sysname file and can be rerun by calling the test script with the -r option,

```
TTEST -r PROBLEMS.i786-pc-linux-gnu.
```

The -r may also be useful to create any user-defined selection of test examples. The full list of available examples is obtained by the TTEST --list command.

Once you are done with testing, you may wish to clean up afterwards. To do it, use the -clean and -realclean options of the TTEST script. The difference between these two is that TTEST -clean deletes only the test directories and protocols that were created for the current computer architecture as returned by Sysname. In contrast, the TTEST -realclean wipes out all test directories and protocols that get in its way.

### 23.3 Taking the timings and benchmarking

Benchmarking differs from testing only in that program timings are computed and compared with reference timings. Calling the script as

```
TTEST --timings
```

performs the test, calculates the CPU and wall clock timings, and writes the raw results to the TESTTIMINGS.sysname.nodename file. Auxiliary scripts Tbtim and Tblist help to convert this data to a more readable form and produce summaries as \texttt{\LaTeX} tables. The Tbtim script creates a summary of benchmark results for a given computer platform from the original timings file. Tblist produces benchmark comparisons of different platforms. The corresponding timings files must be provided as arguments to the Tblist script. For more details and options, see TBTIM -help and TBLIST -help.

### 23.4 Modes and options of the TTEST script

The TTEST script knows several operation modes: "run", "check", "list", "clean", "realclean", and "validate", controlled by its options. The "run" mode is default and means that the test calculations are performed and the results are written to the TESTPROTOKOLL file. The "check" mode differs only in that the programs are not executed, but the existing program output is checked against the reference. The results of the check are written to the CHECKPROTOKOLL file. Calling the test script in the "list" mode simply lists the test examples that are currently available. This allows the user to save the full list to file, edit, and re-use it with the -r option. The "clean" and "realclean" options are for cleaning up the test directories and protocols. Finally, the "validate" mode is mainly of use for writing the CRIT files. It helps to verify the match patterns provided in the test criteria and shows
if it extracts the expected data for comparison with the reference. For every output file used for testing, the "validate" option produces a copy with an additional .val extension. The match strings evaluated for test criteria are highlighted in the output by «< and >>» marks.

There is a lot of options controlling the behavior of TTEST. Testing specific versions of TURBOMOLE modules is provided by loading path options, -l for binaries, -ls for scripts, and -x for a specific executable. For benchmarking, you need the -timings option to produce the timing summaries, and the -newref option to save the current program timings as the new reference. The module specifications and -short, -long, and -r options can be used for selecting the test examples. The more specialized options are summarized in the following table. Note that most of these options can also be set in the DEFCRIT file (see below).
23.4. Modes and Options of the TTest Script

Operation modes

- `--help`  
  Prints out the help message and exits.
- `--h`  
- `--?`  
- `--list`  
  Lists the available test examples.
- `--clean`  
  Deletes the test directories and summary files for the current architecture (given by `sysname`, see Chapter 1.5).
- `--realclean`  
  Deletes all test directories and protocols.
- `--check dir`  
  Checks the correctness of an existing program test in the directory `dir` (default: `TESTDIR.sysname`). Useful if new criteria or new references are established.
- `--validate dir`  
  Examines the output files in the directory `dir` (default: `TESTDIR.sysname`) and highlights the positions of the retrieved matches.

Loading path and naming options

- `--loaddir dir`  
  Loading path for the Turbomole binaries (default: `$TURBODIR/bin/sysname`).
- `--scriptdir dir`  
  Loading path for the Turbomole scripts (default: `$TURBODIR/scripts`).
- `--testprog prog`  
  Tests the given executable `prog`.
- `--x prog`  
- `--dir dir`  
  Name for the local test directory (default: `TESTDIR.sysname`).
- `--critfile file`  
  Name for the local criteria file (default: `CRIT`).
- `--defcritfile file`  
  Name for the test suite settings file (default: `DEFCRIT`).
- `--protfile file`  
  Name for the local protocol file (default: `TESTPROTOKOLL`).
- `--output file`  
  Name for the global protocol file (default: `TESTPROTOKOLL.sysname`).
- `--gprotfile file`  
  Name for the check protocol file (default: `CHECKPROTOKOLL`).
- `--checkfile file`  
  Name for the failed tests list (default: `PROBLEMS.sysname`).
- `--errfile file`  
  Name for the local error output file (default: `output.err`).
- `--probfile file`  
  Name for the failed tests list (default: `PROBLEMS.sysname`).
- `--timfile file`  
  Name for the timings file (default: `TIMINGS.sysname`).
- `--valfile file`  
  Name for the validation file for ‘run’ criteria (default: `RUNCRIT.val`).
Bibliography


[26] C. van Wüllen. Shared-memory parallelization of the TURBOMOLE programs AO-
FORCE, ESCF, and EGRAD: How to quickly parallelize legacy code. J. Comp. Chem.,

[27] M. von Arnim; R. Ahlrichs. Geometry optimization in generalized natural internal

[28] P. Pulay; G. Fogarasi; F. Pang; J. E. Boggs. Systematic ab initio gradient calculation of

[29] M. Dolg; U. Wedig; H. Stoll; H. Preuš. Energy-adjusted ab initio pseudopotentials for


electron densities using multipole accelerated resolution of identity approximation. J.

[33] F. Weigend. A fully direct RI-HF algorithm: Implementation, optimised auxiliary


[36] F. Jensen. Locating transition structures by mode following: A comparison of six
methods on the Ar8 Lennard-Jones potential. J. Chem. Phys., 102(17), 6706–6718,

[37] P. Császár; P. Pulay. Geometry optimization by direct inversion in the iterative sub-


[40] H. B. Schlegel. Estimating the hessian for gradient-type geometry optimizations.


