Contents

1 Preface 11
   1.1 Contributions and Acknowledgements ............................................ 11
   1.2 Features of TURBOMOLE ............................................................... 12
   1.3 How to Quote Usage of TURBOMOLE ............................................... 12
   1.4 Modules and Their Functionality .................................................... 17
   1.5 Tools .............................................................................................. 19
   1.6 Installation of TURBOMOLE ............................................................. 22
   1.7 How to Run TURBOMOLE: A ‘Quick and Dirty’ Tutorial ....................... 23
      1.7.1 Single Point Calculations: Running TURBOMOLE Modules .............. 24
      1.7.2 Energy and Gradient Calculations .............................................. 25
      1.7.3 Calculation of Molecular Properties .......................................... 26
      1.7.4 Modules and Data Flow ............................................................. 26
   1.8 Parallel Runs ..................................................................................... 26
      1.8.1 Running Parallel Jobs ............................................................... 26
   1.9 Running TURBOMOLE using the script TMOLE ................................... 30
      1.9.1 Implementation ........................................................................ 30
      1.9.2 The file turbo.in ...................................................................... 31

2 Preparing your input file with DEFINE 37
   2.0.1 Universally Available Display Commands in DEFINE ...................... 38
   2.0.2 Specifying Atomic Sets .................................................................. 38
   2.0.3 control as Input and Output File ................................................... 39
   2.0.4 Be Prepared ................................................................................ 39
   2.1 The Geometry Main Menu ................................................................. 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1 Description of commands</td>
<td>42</td>
</tr>
<tr>
<td>2.1.2 Internal Coordinate Menu</td>
<td>45</td>
</tr>
<tr>
<td>2.1.3 Manipulating the Geometry</td>
<td>50</td>
</tr>
<tr>
<td>2.2 The Atomic Attributes Menu</td>
<td>51</td>
</tr>
<tr>
<td>2.2.1 Description of the commands</td>
<td>54</td>
</tr>
<tr>
<td>2.3 Generating MO Start Vectors</td>
<td>56</td>
</tr>
<tr>
<td>2.3.1 The MO Start Vectors Menu</td>
<td>56</td>
</tr>
<tr>
<td>2.3.2 Assignment of Occupation Numbers</td>
<td>59</td>
</tr>
<tr>
<td>2.3.3 Orbital Specification Menu</td>
<td>61</td>
</tr>
<tr>
<td>2.3.4 Roothaan Parameters</td>
<td>62</td>
</tr>
<tr>
<td>2.4 The General Options Menu</td>
<td>63</td>
</tr>
<tr>
<td>2.4.1 Important commands</td>
<td>64</td>
</tr>
<tr>
<td>2.4.2 Special adjustments</td>
<td>70</td>
</tr>
<tr>
<td>2.4.3 Relax Options</td>
<td>72</td>
</tr>
<tr>
<td>2.4.4 Definition of External Electrostatic Fields</td>
<td>77</td>
</tr>
<tr>
<td>2.4.5 Properties</td>
<td>78</td>
</tr>
<tr>
<td>3 Calculation of Molecular Structure and <em>Ab Initio</em> Molecular Dynamics</td>
<td>87</td>
</tr>
<tr>
<td>3.1 Structure Optimizations using the JOBEX Script</td>
<td>87</td>
</tr>
<tr>
<td>3.1.1 Options</td>
<td>87</td>
</tr>
<tr>
<td>3.1.2 Output</td>
<td>89</td>
</tr>
<tr>
<td>3.2 Program STATPT</td>
<td>89</td>
</tr>
<tr>
<td>3.2.1 General Information</td>
<td>89</td>
</tr>
<tr>
<td>3.2.2 Hessian matrix</td>
<td>91</td>
</tr>
<tr>
<td>3.2.3 Finding Minima</td>
<td>91</td>
</tr>
<tr>
<td>3.2.4 Finding transition states</td>
<td>92</td>
</tr>
<tr>
<td>3.3 Program Relax</td>
<td>93</td>
</tr>
<tr>
<td>3.3.1 Purpose</td>
<td>93</td>
</tr>
<tr>
<td>3.3.2 Optimization of General Coordinates</td>
<td>93</td>
</tr>
<tr>
<td>3.3.3 Force Constant Update Algorithms</td>
<td>95</td>
</tr>
<tr>
<td>3.3.4 Definition of Internal Coordinates</td>
<td>97</td>
</tr>
</tbody>
</table>
## CONTENTS

3.3.5 Structure Optimizations Using Internal Coordinates ........................................ 97
3.3.6 Structure Optimization in Cartesian Coordinates ........................................... 98
3.3.7 Optimization of Basis Sets (SCF only) .......................................................... 99
3.3.8 Simultaneous Optimization of Basis Set and Structure .................................... 99
3.3.9 Optimization of Structure and a Global Scaling Factor .................................... 100
3.3.10 Conversion from Internal to Cartesian Coordinates ....................................... 100
3.3.11 Conversion of Cartesian Coordinates, Gradients and Force Constants to Internals .................................................................................................................. 100
3.3.12 The m-Matrix .................................................................................................. 101
3.3.13 Initialization of Force Constant Matrices ....................................................... 101
3.3.14 Look at Results .............................................................................................. 102
3.4 Force Field Calculations ...................................................................................... 103
  3.4.1 Purpose ........................................................................................................... 103
  3.4.2 How to Perform a UFF Calculation ............................................................... 103
  3.4.3 The UFF implementation ................................................................................ 103
3.5 Molecular Dynamics Calculations ....................................................................... 106
  3.5.1 Background Theory ...................................................................................... 109
  4.2 Exchange-Correlation Functionals Available .................................................... 110
  4.3 Restricted Open-Shell Hartree–Fock .................................................................. 112
    4.3.1 Brief Description ......................................................................................... 112
    4.3.2 One Open Shell .......................................................................................... 112
    4.3.3 More Than One Open Shell ........................................................................ 115
    4.3.4 Miscellaneous ............................................................................................ 117
5 Second-order Möller–Plesset Perturbation Theory .................................................. 119
  5.1 Functionalities of MPGRAD and RIMP2 ......................................................... 119
  5.2 Some Theory ...................................................................................................... 120
  5.3 How to Prepare and Perform MP2 Calculations ................................................ 120
  5.4 General Comments on MP2 Calculations, Practical Hints ............................... 122
6 Hartree–Fock and DFT Response Calculations: Stability, Dynamic Response Properties, and Excited States ........................................................... 125
## Contents

6.1 Functionalities of ESCF and EGRAD ................................................. 125  
6.2 Theoretical Background ................................................................. 126  
6.3 Implementation ............................................................................. 128  
6.4 How to Perform ............................................................................. 129  
6.4.1 Preliminaries ............................................................................. 129  
6.4.2 Polarizabilities and Optical Rotations ............................................ 130  
6.4.3 Stability Analysis ..................................................................... 131  
6.4.4 Vertical Excitation and CD Spectra ............................................... 131  
6.4.5 Excited State Geometry Optimizations ......................................... 133  
6.4.6 Excited State Force Constant Calculations .................................... 134  

7 Second-Order Approximate Coupled-Cluster (CC2) Calculations 135  
7.1 CC2 Ground-State Energy Calculations ............................................ 138  
7.2 Calculation of Excitation Energies .................................................... 141  
7.3 First-Order Properties and Gradients ................................................. 144  
7.3.1 Ground State Properties, Gradients and Geometries ....................... 144  
7.3.2 Excited State Properties, Gradients and Geometries ....................... 147  
7.3.3 Visualization of densities ............................................................. 149  
7.4 Transition Moments ..................................................................... 150  
7.5 Parallel RI-MP2 and RI-CC2 Calculations ......................................... 151  

8 Calculation of Vibrational Frequencies and Infrared Spectra 153  
8.1 Analysis of Normal Modes in Terms of Internal Coordinates ............... 154  

9 Calculation of NMR Shieldings ........................................................... 156  
9.1 Prerequisites ................................................................................ 156  
9.2 How to Perform a SCF of DFT Calculation ....................................... 157  
9.3 How to Perform a MP2 calculation ................................................... 157  
9.4 Chemical Shifts ........................................................................... 158  
9.5 Other Features and Known Limitations ............................................. 158  

10 Molecular Properties, Wavefunction Analysis, and Interfaces to Vi-  
    sualization Tools ........................................................................... 159  
10.1 Wavefunction analysis and Molecular Properties ............................. 159
## CONTENTS

10.2 Interfaces to Visualization Tools ........................................... 160

11 Treatment of Solvation Effects with COSMO ................................. 164

12 Keywords in the control file ....................................................... 169
   12.1 Introduction ........................................................................ 169
   12.2 Format of Keywords and Comments ..................................... 169
      12.2.1 General Keywords .................................................... 169
      12.2.2 Keywords for System Specification ............................... 171
      12.2.3 Keywords for redundant internal coordinates in $\text{redund.inp}$ .... 173
      12.2.4 Keywords for Module UFF ......................................... 175
      12.2.5 Keywords for Modules DSCF and RIDFT ......................... 180
      12.2.6 Keywords for COSMO ............................................... 197
      12.2.7 Keywords for Modules GRAD and RDGRAD .................... 199
      12.2.8 Keywords for Module AOFORCE ................................ 200
      12.2.9 Keywords for Module ESCF ....................................... 203
      12.2.10 Keywords for Module EGAD ...................................... 206
      12.2.11 Keywords for Modules MPGRAD and RIMP2 ................... 206
      12.2.12 Keywords for Module RICC2 .................................... 209
      12.2.13 Keywords for Module RELAX .................................... 216
      12.2.14 Keywords for Module STATPT .................................. 225
      12.2.15 Keywords for Module MOLOCH ................................ 226
      12.2.16 Keywords for wave function analysis and generation of plotting data ........................................................................ 231
      12.2.17 Keywords for Module Frog ........................................ 236
      12.2.18 Keywords for Module MPSHIFT .................................. 241
      12.2.19 Keywords for Parallel Runs ........................................ 243

13 Sample control files .................................................................. 247
   13.1 Introduction ........................................................................ 247
   13.2 NH$_3$ Input for a RHF Calculation ...................................... 248
      13.2.1 Main File control ...................................................... 248
      13.2.2 File coord .............................................................. 249
Chapter 1

Preface

1.1 Contributions and Acknowledgements

TURBOMOLE has been designed by the Quantum Chemistry Group, University of Karlsruhe, Germany, since 1988. The following members of the group have made contributions:

Reinhart Ahlrichs, Michael Bär, Hans–Peter Baron, Rüdiger Bauernschmitt, Stephan Böcker, Nathan Crawford, Peter Deglmann, Michael Ehrig, Karin Eichkorn, Simon Elliott, Filipp Furche, Frank Haase, Marco Häser, Hans Horn, Christof Hättig, Christian Huber, Uwe Huniar, Marco Kattannek, Andreas Köhn, Christoph Kölmel, Markus Kollwitz, Klaus May, Paola Nava, Christian Ochsenfeld, Holger Öhm, Holger Patzelt, Dmitrij Rappoport, Oliver Rubner, Ansgar Schäfer, Uwe Schneider, Marek Sierka, Oliver Treutler, Barbara Unterreiner, Malte von Arnim, Florian Weigend, Patrick Weis, Horst Weiss

Contact address:

Lehrstuhl für Theoretische Chemie
Institut für Physikalische Chemie
Universität Karlsruhe
Kaiserstr. 12
D-76128 Karlsruhe
E-mail: reinhart.ahlrichs@chemie.uni-karlsruhe.de
or: helpdesk@turbomole.com
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 Berlin
- Stefan Brode, BASF AG, Ludwigshafen
- Heinz Schiffer, HOECHST AG, Frankfurt

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

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

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 \[7\]).

Additionally (but not alternatively), the version employed should be indicated, e.g. TURBOMOLE V5.6.

- general program structure and features: [4]
1.3. **HOW TO QUOTE USAGE OF TURBOMOLE**

- HF-SCF [Dscf]: [II]
- DFT (quadrature) [Dscf, Ridft, Escf, Aoforce]: [IV] XXXI (m grids)
- RI-DFT [Ridft, Aoforce, Escf]: XXXI [XXXI] [XXIII] (marij), [VII] (Escf), [XXIV] (Aoforce)
- MP2 [Mpggrad]: [II]
- RI-MP2 [Rimp2]: [VIII] XXXII
- stability analysis [Escf]: [V]
- electronic excitations by CIS, RPA, TD-DFT [Escf]: [VI] [VII] [XVII] [XXVII]
- excited state structures and properties with CIS, RPA, TD-DFT [Egrad]: [XIX] XXVI [XXVII]
- RI-CC2 [Ricc2]: XII [XIII] (triplet excitations) XIV (properties for triplet states) XV (transition moments and properties of excited states) XXI (ground state geometry optimizations)
- analytical second derivatives (force fields) [Aoforce]: XVI XVII
- basis sets: SV, SV(P), SVP, DZ [XXVIII], TZV, TZVP, TZVPP [XXIX], TZVPP(Rb-Hg) [XXXII], QZV [XXXV]
- new consistent basis sets (with smaller ECPs): [XXXVI]
- all-electron basis sets for Rb to Xe: [XXXIII]
- auxiliary basis sets for RI-DFT: [XXX] XXXI
- auxiliary basis sets for RI-MP2: [XXXII] XXXI (for Dunning basis sets)
- RI-JK [Ridft]: [XX]
- NMR chemical shifts [Mpsight]: [IX] (MP2)
- parallel DFT [Ridft]: [X]
- geometry optimization in redundant internal coordinates [Relax]: [XI]
- RI integral evaluation: [XXV]

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


IV. Efficient Molecular Numerical Integration Schemes.

V. Stability Analysis for Solutions of the Closed Shell Kohn–Sham Equation.

VI. Treatment of Electronic Excitations within the Adiabatic Approximation of Time Dependent Density Functional Theory.


XI. Geometry optimization in generalized natural internal Coordinates.


Basis sets


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


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 [1,2,3,4,5] 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) [6] 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 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

**RIDGE**

perform DFT calculations—as **DSCF** and **GRAD**—within the RI-J approximation, i.e. the total density is approximated by a sum of atom centered s, p, d... functions—the auxiliary (or fitting) basis. This allows for a very efficient treatment of Coulomb interactions. The functionals supported are specified in **DEFINE**.

**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.

**RIMP2**

calculates MP2 energies and gradients for RHF and UHF wavefunctions, significantly more efficient than **MPGRAD** by using the RI technique [7,8].

**RICC2**

calculates electronic excitation energies, transition moments and properties of excited states at the CIS, CIS(D), ADC(2) and CC2 level using
either a closed-shell RHF or a UHF SCF reference function. Employs the RI technique to approximate two-electron integrals \[9,10,11,12\].

**Relax** requires a gradient run—by Grad, Rdgrad, Rimp2 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 polarizibility)
- 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) \[13\],
- stability analysis of single-determinant closed-shell wave functions (second derivative of energy with respect to orbital rotations) \[14\].
1.5. **TOOLS**

**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 3.1), and in finite difference force constant calculations (using Numforce). Details see [15].

**Mpsift** requires a converged SCF or DFT run for closed shells. Mpsift 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 Mpsift [16].

**Moloch** computes a variety of first-order properties and analyses of the wavefunction as can be seen from the keywords. Also atomic point charges can be fitted to the electrostatic potential of a molecule.

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`

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

CONVGREP greps lines for convergence check out of control file.

COSMOPREP sets up control file for a COSMO run (see Chapter 11).

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

EIGER displays orbital eigenvalues obtained from data group $scfmo$.

FINIT initialises the force constant matrix for the next RELAX step.

FREEH calculates thermodynamic functions from molecular data in a control file; an AOFORCE run is a necessary prerequisite.

HCORE prepares the control file for a Hamilton core guess (RHF only).

HOLUMO displays the highest occupied and the lowest unoccupied orbital.

JOBEX usage: see Section 3.1
   is the TURBOMOLE driver for all kinds of optimizations.

KD G example: kdg scfdiis
   kills a data group (here $scfdiis$) in the control file.

KONTO interface between MOLOCH grid output and gle graphics; perl is required, please adjust the path at the top of the script.

LHFPREP prepares lhf calculations by adjusting parameters of the control file.

LOG2X converts the file logging an MD trajectory into coordinates in frames appropriate for XMOL animation program.

LOG2EGY extracts the energy data (KE, total energy, PE) 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).

MOLOCH2 population analysis for UHF input.

MP2PREP prepares MP2 calculations interactively by adjusting parameters of the control file according to your system resources.

NUMFORCE calculates numerically force constants, vibrational frequencies, IR and Raman cross sections (the latter only for closed-shell molecules). Open-shell molecules have to be calculated in point group $C_1$. (Note that the name of the kornshell script is NumForce with capital F.)
1.5. **TOOLS**

**OUTP** displays out-of-plan angles.

**RIMP2PREP** interactive tool for preparing the control file for RIMP2 calculations by adjusting the required parameters according to your system resources and by specifying auxiliary basis sets and frozen core shells. This can also be done in Define.

**SCREWER** distorts a molecule along a vibrational mode.

**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.

**STATTI** prepares the control file for a statistics run.

**T2S** converts TURBOMOLE coordinates to SCHAKAL format.

**T2X** converts TURBOMOLE coordinates to xyz format.

**TM2MOLDEN** creates a molden format input file for the Molden program. 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](http://www.cmbi.ru.nl/molden/molden.html)).

**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.

**TURBO_START** is a script to set up and start parallel TURBOMOLE runs on IBM SP2: for detailed information see Section 1.8.1.

**TBTIM** is used to convert timings output files from TURBOBENCH calculations to \LaTeX tables (for options please type TBLIST --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 2.3.1.

**X2T** converts standard xyz files into TURBOMOLE coordinates.
1.6 **Installation of TURBOMOLE**

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

```bash
gunzip turbomole.tar.gz
```

and unpacked

```bash
tar -xvf turbomole.tar
```

to produce the whole directory structure. Executable modules are in the `bin/[arch]` directory (for example, IBM modules are in `bin/ibm`). Tools (including JOBEX) are in `scripts` and (auxiliary) basis sets are kept in the directories `basen`, `jbasen`, `jkbasen` and `cbasen`. Coordinates for some common chemical fragments are supplied in `structures`.

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

```bash
TURBODIR=/my_disk/my_name/my_turbo
```

Check that the `sysname` tool works on your computer:

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

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

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

```bash
PATH=$PATH:$TURBODIR/bin/[arch]:$TURBODIR/scripts
```

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

```bash
$TURBODIR/bin/dscf > dscf.out
```

or if the path is OK, simply

```bash
dscf > dscf.out
```

In addition, some sample calculations are supplied in `TURBOTEST` so that the modules can be tested. Just run `TEST` from this directory to get help on how this works.
1.7 How to Run Turbomole: A ‘Quick and Dirty’ Tutorial

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 12). Define step by step provides 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 13.2.2) as input for Define.

The main problem in using Turbomole appears to be the definition of the molecule: atoms, coordinates, etc. The easiest way around 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:
  \[ x2t \text{ xyzinputfile} > \text{ coord} \]
- 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 \[ b \text{ all def-TZVP} \] or \[ b \text{ all def2-TZVP} \] 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 \text{dft} in the last menu and then enter on
for non-hybrid functionals you best choose the efficient RI approximation by entering \( ri \) and providing roughly \( \frac{3}{4} \) of the memory (with \( m \) number; \( m \) number in MB) your computer has available. Auxillary basis sets are provided automatically (in the printout of an RIDFT run you can check how much is really needed; a \texttt{top} statement will tell you if you overplayed your cards.).

- By the way: we strongly recommend B-P86 (with RI) or B3-LYP (as non-hybrid and hybrid functionals).

- for an SCF or hybrid-functional DFT run, you simply enter:
  
  \[
  \texttt{[nohup] dscf > dscf.out \&}
  \]

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

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

  or
  
  \[
  \texttt{[nohup] rdgrad > rdgrad.out \&}
  \]

- for a geometry optimization simply call \texttt{JOBEX}:
  
  for a standard SCF input:
  
  \[
  \texttt{[nohup] jobex \&}
  \]

  for a standard RI-DFT input:
  
  \[
  \texttt{[nohup] jobex -ri \&}
  \]

- many features, such as NMR chemical shifts on SCF and DFT level, do not require further modifications of the input, just call e.g. \texttt{MPSHIFT} after the appropriate energy calculation (mpshift runs with SCF or DFT using a hybrid-functional need a filesize of the semi-direct file \texttt{twoint} that is non-zero).

- other features, such as MP2 need further action on the input, using tools like \texttt{MP2PREP} or \texttt{RIMP2PREP}. Please refer to the following pages of this documentation.

1.7.1 Single Point Calculations: Running TURBOMOLE Modules

All calculations are carried out in a similar way. First you have to run \texttt{DEFINE} to obtain the \texttt{control} file or to add/change the keywords you need for your purpose. This can also be done manually with an editor. Given a kornshell and a path to \$\texttt{TURBODIR/bin/\[arch]} (see installation, Section \[1.6]\) you call the appropriate module in the following way (e.g. module DSCF):

\[
\texttt{nohup dscf > dscf.out \&}
\]
1.7. HOW TO RUN TURBOMOLE: A ‘QUICK AND DIRTY’ TUTORIAL

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 [12]. The features of TURBOMOLE will be described in the following section.

1.7.2 Energy and Gradient Calculations

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

Hartree–Fock–SCF

use modules DSCF and GRAD to obtain the energy and gradient. The energy can be calculated after a DEFINE run without any further keywords or previous runs. The gradient calculation however requires a converged DSCF 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 all RI flags and will try to do a normal calculation. 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.

Excited states

Single point excited state energies for CIS, TDHF and TDDFT methods can be calculated using Escf. Excited state energies, gradients, and other first order properties are provided by Egrad. Both modules require well converged ground state orbitals.

MP2

the module MPGRAD calculates the MP2 energy as well as the energy gradient. If only the energy is desired use the keyword $mp2energy. MP2 calculations need well converged SCF runs (the SCF run has to be done with at least the density convergence $denconv 1.d-7, and $scfconv 6 as described in Section [12]. For all further preparations run the tool Mp2prep. For MP2 calculations in the RI approximation use the Ricc2 module. The input can be prepared with the cc2 menu in DEFINE. (Alternatively, the older Rimp2 module and for preparation of its input the tool Rimp2prep maybe used).

CC2

the module Ricc2 calculates MP2 and CC2 ground state energies and CIS/CCS,
CIS(D) or CC2 excitation energies using the resolution-of-the-identity (RI) approximation. Gradients are available for ground states the MP2 and CC2 and for excited states at the CC2 level. In addition transition moments and first-order properties (for ground and excited states) are available for some of the methods. For more details see Section 7. The R1cc2 module requires are well converged SCF molecular orbitals. The input can be prepared using the cc2 menu of Define.

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

\begin{verbatim}
  stati dscf
  nohup dscf > dscf.stat &
\end{verbatim}

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 12.2.5). The requirement of other combinations will be computed as well and be written to the output file dscf.stat.

1.7.3 Calculation of Molecular Properties

See Section 1.4 for the functionality and Section 12 for the required keywords of the modules Aoforce, Escf, Mpsift, and Moloch.

1.7.4 Modules and Data Flow

See Figure 1.1 above.

1.8 Parallel Runs

The additional keywords neccessary for parallel runs are described in Chapter 12.

1.8.1 Running Parallel Jobs

The parallel version of TURBOMOLE runs on all supported systems:

- workstation cluster with Ethernet (or other) connection
- SMP systems: DEC, IBM, HP, SGI, SUN,…
- or combinations of SMP and cluster (like IBM SP3)
Figure 1.1: The modules of **TURBOMOLE** and the main data flow between them.
CHAPTER 1. PREFACE

Setting up the parallel environment

In addition to the installation steps described in Section 1.6 (see page 22) you just have to set the variable PARA_ARCH to MPI, i.e. in sh/bash/ksh syntax:

```
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 or Mpgrad from your command line without their explicit path, expand your $PATH environment variable to:

```
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 SP3) automatically. The number of CPUs that shall be used can be chosen by setting the environment variable PARNODES:

```
export PARNODES=8
```

On all systems TURBOMOLE is using the MPI library that has been shipped with your operating system. On Linux the freely available, portable implementation of MPI, MPICH (http://www-unix.mcs.anl.gov/mpi/mpich/) is used. If several different MPI implementations have been installed on your system, please make sure that the right mpirun is called.

**Note:** TURBOMOLE needs 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 does not need any CPU time. So if you are setting PARNODES to N, N+1 tasks will be started. If a queueing system is used, check that N+1 processes will be provided.

Starting parallel jobs

After setting up the parallel environment as described in Chapter 1.8.1 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.

This is not true for the program Mpgrad. To prepare the parallel input for this program, call turbo_preproc in the directory where your serial input lies and follow the instructions.

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

\[
\text{unit}=30 \ \text{size}=????? \ \text{file}=/\text{local}\_\text{scratchdir}/\text{twoint}
\]

For the additional mandatory or optional input for parallel runs with the RICC2 program see Section 7.5.

Testing the parallel binaries

The binaries Ridft, Rdgrad, Dscf and Grad can be tested by the usual testsuite: go to $\text{TURBODIR/TURBOTEST}$ and call TEST progname, where progname is one of ridft, rdgrad, dscf, grad.
1.9 Running **TURBOMOLE using the script TMOLE**

The Perl script TMOLE drives the required TURBOMOLE modules on the basis of a GAUSSIAN style input file *turbo.in*. This facilitates the use of TURBOMOLE for users familiar with GAUSSIAN, which we assumed to be the case. TMOLE allows e.g. to calculate the potential curve for stretch, bending and dihedral modes, a feature not automatically available in TURBOMOLE. TMOLE does not support yet the whole functionality of TURBOMOLE and GAUSSIAN.

To give an idea, here a simple example for using TMOLE. If you want to perform a geometry optimization of water at DFT-level with the B-P86 correlation-exchange functional and a basis set of SVP quality, you have to create the following file *turbo.in*:

```plaintext
%title
geometry optimization for water
%method
GEOMY :: b-p/SVP
%charge
0
%coord
0.00000000000000 0.00000000000000 -0.69098999073900 o
-1.46580510295113 0.00000000000000 0.34549499536950 h
1.46580510295113 0.00000000000000 0.34549499536950 h
%end
```

Then start TMOLE to perform the calculation. A successful completion is indicated by ‘tmole ended normally’ at the end of output. The output is the same as a JOBEX output. Additional examples for *turbo.in* are given in Chapter 14.

1.9.1 Implementation

TMOLE first generates from *turbo.in* an input file for DEFINE, the general input generator for TURBOMOLE (see Section 2). Then DEFINE is executed to generate the input file control, specifying type of calculation, basis set etc. TMOLE finally executes the required modules of TURBOMOLE. The output of each program will be written to a file with suffix ‘.out’. So the output of RIDFT for example is in the file *ridft.out*. If one wants to perform a geometry optimization TMOLE starts the script JOBEX (for a description see Section 3.1).
1.9.2 The file turbo.in

The file turbo.in is the input file for a TURBOMOLE calculation with TMOLE. This file consists of the following sections: \%method, \%coord and the optional ones \%charge, \%title, \%add_control_commands, \%scan. The file has to end with \%end.

Section \%method

description:

defines the properties to calculate, the level of calculation, the basis set used and further options.

general syntax:

\%method

PROPERTY :: level of calculation / basis set [run options]

Note: the '::' after PROPERTY and the brackets [run options].
If you want to continue on the next line, type '& at the end of the line, e.g.

ENRGY :: b-p/SVP [gen_stat=1, scf_msil=99, &
     scf_grid=m4]

Available Properties

GEOMY optimization of all structure parameters for ground states (default: geo_nrgc=20).

ENRGY single point energy calculation (default: gen_spca=1).

GRADI calculation of the gradient (default: gen_spca=1).

FORCE calculation of the vibrational spectrum. First the energy will be calculated.

Possible levels of calculation

UFF universal force field (see Section 3.4).

HF Hartree–Fock (see Chapter 1).

DFT switch to choose the exchange-correlational functional, e.g.

ENRGY :: s-vwn/SVP

The functionals available and their abbreviations are listed in Menu 2.4.1 and are described in Section 4.2.

MP2 second order Møller-Plesset Pertubation Theory (see Chapter 5).
RI-DFT and RI-MP2

to use the RI approximation, type ri- before the description of the level. This is possible for all non-hybrid functionals (see Section 4.2) and for MP2, e.g.

\texttt{ENERGY :: ri-s-vwn/SVP}

UHF and UKS

the molecule will be calculated in unrestricted formalism, if the first letter of the level ist an ‘u’, e.g.

\texttt{ENERGY :: uhf/TZVP}

Basis set choice

the available basis sets are the standard basis sets of \textsc{Turbomole} (see Section 2.2). Default basis set is def-SV(P). If the level of calculation is UFF, there is no need to specify the basis set.

Available general run options

\texttt{gen\_crds=\textit{options}}

choose coordinate system (see \texttt{\$optimize}):

- \texttt{ired} redundant, internal coordinates (default)
- \texttt{intern} internal coordinates
- \texttt{cart} cartesian coordinates

\texttt{gen\_symm=\textit{options}}

assign symmetry of the molecule (Schönflies symbol):

- \texttt{auto} DEFINE assigns point group (default)
- \texttt{any} any Schönflies symbol, e.g. \texttt{gen\_symm=c2v}.

\texttt{gen\_sthr=\textit{real}}

threshold for symmetry determination (default: 1d – 3).

\texttt{gen\_prep=\textit{options}}

switch for a preparation run:

- \texttt{gen\_prep=0} a calculation is done (default)
- \texttt{gen\_prep=1} only input files such as \texttt{control}, etc. are generated

\texttt{gen\_stpt=\textit{options}}

switch for using \texttt{Relax} or \texttt{Statpt}
1.9. **RUNNING TURBOMOLE USING THE SCRIPT TMOLE**

- `gen_stpt=0` using RELAX (default)
- `gen_stpt=1` using STATPT

**gen_spca=** *options*

- switch for single point calculation:
  - `gen_spca=0` structure optimization (default)
  - `gen_spca=1` single point calculation

**gen_stat=** *options*

- switch for statistics run (see `$statistics`):
  - `gen_stat=0` no statistics run (default)
  - `gen_stat=1` statistics run will be performed

**gen_blow=** *integer*

- add dummy orbitals per irrep (default=0). Needed for non-default occupation (if one changes the occupation with `%add_control_commands`).

**gen_basl=** `<path>`

- path for basis sets (default: `$TURBODIR/basen`).

**gen_jbas=** `<path>`

- path for auxiliar basis sets (default: `$TURBODIR/jbasen`).

**gen_scrd=** `<path>`

- path for scripts (default: `$TURBODIR/scripts`).

**gen_bind=** `<path>`

- path for binaries (default: `$TURBODIR/bin/’sysname’`).

**gen_mult=** *integer*

- multiplicity of molecule (default: 1).

**gen_ncpu=** *integer*

- number of CPUs, only necessary for parallel runs (default: 1).

**gen_mpil=** `<path>`

- path for MPI, only necessary for parallel runs (default: `/usr/app/lib/mpich/bin`).

**Available SCF run options**

**scf_grid=** *gridsize*

- definition of the gridsize, necessary for DFT. Possible values are 1–5 and m1–m5 (default: m3, see `$dft`).
\texttt{scf\_mrij=\textit{options}}

switch for MARI-J (details see \$\texttt{marij}):

\begin{itemize}
\item \texttt{scf\_mrij=0} no MARI-J (default)
\item \texttt{scf\_mrij=1} MARI-J is enabled
\end{itemize}

\texttt{scf\_msil}

maximum numbers of SCF cycles (default: 30, see \$\texttt{scf\_iterlimit})

\texttt{scf\_conv=\textit{integer}}

SCF convergency criterion will be $10^{-\textit{integer}}$ for the energy (default: for SCF: 7, for DFT: 6, see \$\texttt{scf\_conv}).

\texttt{scf\_rico=\textit{integer}}

memory core for RI calculation in MB (default: 200 MB, see \$\texttt{ricore}).

\texttt{scf\_dsta=\textit{real}}

start value for SCF damping (default: 1.000, see \$\texttt{scf\_damp}).

\texttt{scf\_dink=\textit{real}}

increment for SCF damping (default: 0.050, see \$\texttt{scf\_damp}).

\texttt{scf\_dste=\textit{real}}

minimum for SCF damping (default: 0.050, see \$\texttt{scf\_damp}).

\texttt{scf\_ferm=\textit{options}}

switch for fractional occupation (FON) numbers (see \$\texttt{fermi}):

\begin{itemize}
\item \texttt{scf\_ferm=0} no fractional occupation numbers (default)
\item \texttt{scf\_ferm=1} fractional occupation numbers enabled
\end{itemize}

\texttt{scf\_fets=\textit{real}}

starting temperature for FON (default: 300 K).

\texttt{scf\_fete=\textit{real}}

dend temperature for FON (default: 300 K).

\texttt{scf\_fetf=\textit{real}}

temperature factor for FON (default: 1.0).

\texttt{scf\_fehl=\textit{real}}

hlcrt parameter for FON (default: 0.1)

\texttt{scf\_fest=\textit{real}}

energy convergence parameter for FON (default: $1d - 3$)
Available run options for structure optimizations

**geo_nasp=option**

starting program:

- **geo_nasp=en**  energy step (default)
- **geo_nasp=gd**  gradient step
- **geo_nasp=rx**  relax or statpt step

**geo_nrgc=integer**

number of optimization cycles (default: 20).

**geo_suff=option**

switch for UFF start Hessian:

- **geo_suff=0**  no UFF Hessian is used
- **geo_suff=1**  UFF Hessian is used (default)

**geo_dqmax=real**

maximum allowed atom displacement in a.u. (default: 0.3, see $coordinateupdate$).

**geo_ecoc=integer**

SCF convergence critertia will be $10^{-integer}$ a.u. for the energy (default: integer = 6).

**geo_gcoc=integer**

gradient convergence criteria in $10^{-integer}$ a.u (default: integer = 3).

Miscellaneous run options

**for_maxc=integer**

memory flag in MB ($maxcor$ in case of AOFORCE calculations, default: 200 MB).

**for_nfre=option**

switch for frequency calculation:

- **for_nfre=0**  calculation of analytical frequencies (default)
- **for_nfre=1**  calculation of frequencies by numerical differentiation of gradients

Section %coord

This section defines the molecular structure. If the coord file does not exist, TMOLE will read in the cartesian coordinates from turbo.in and will write them to the newly generated coord file.
syntax: \%coord options
coordinates

Available options:

tmxyz       TURBOMOLE format in a.u (default).
xyz         xyz format in Ångström.
gauzmat     a Z-matrix as in GAUSSIAN is used (distances in Ångström and angles in degree). You can generate the Z-matrix with Molden. For more information about Molden see:
(http://www.cmbi.ru.nl/molden/molden.html).

Optional Sections

\%charge     specifies the charge of the molecule in a.u.
\%title      title of the calculation
\%add_control_commands
specifies additional commands which will be added to the generated control-file. So a TURBOMOLE expert may start only with this section.
Example:
\%add_control_commands
$marij
$scfiterlimit 300
:
ADD END

\%scan       specifies a path along which a potential curve is calculated. Coords of the starting, final and intermediate geometries have to be defined in the Z-matrix format (see option gauzmat in Section \%coord). If remaining coordinates are to be optimized at every scan point, one needs—for the present implementation—an additional system of internal coordinates, which contains the mode in question as a separate internal.
syntax: \%scan
<input internal coordinate> <starting point> <increment> <end point>
see Sample inputs in Chapter 14
Chapter 2

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.
2.0.1 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.

2.0.2 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` or `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.
2.0.3 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.

2.0.4 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 13 “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 (XMol 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 2.1.2.

*redundant internals*
defined as linearly independent combinations of *internals* (see ref. [17]), provided automatically by the command *ired* in the ‘geometry main menu’ in Section 2.1 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.

### 2.1 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
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
```
2.1. THE GEOMETRY MAIN MENU

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 com-
mand 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. Therefore the Universal Force Field (UFF) developed from Rappé et al.
in 1992 [6] is implemented (see also Section 3.4). Beyond this one can calculate
a Cartesian analytical Hessian. If one does so, the start Hessian for the \textit{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 \textit{redundant internals}:

- a coord read coord
- desy determine symmetry, if you expect a higher symmetry, repeat with in-
  creased tolerance desy 0.1 , you may go up to desy 1..
- ired get redundant internals
- * quit main geometry menu

To define \textit{internals}:

- a coord read coord
- desy determine symmetry
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

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

2.1.1 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$_3$ as example). The commands in this menu will now be described briefly:

sy

   Definition of the Schönflies symbol of the molecular point group symmetry. If you enter only sy, DEFINE will ask you to enter the symbol, but you may also directly enter sy c3v. DEFINE will symmetrize the geometry according to the new Schönflies symbol and will create new nuclei if necessary. You therefore have to take care that you enter the correct symbol and that your molecule is properly oriented. All TURBOMOLE programs require the molecule to be in a standard orientation depending on its point group. For the groups $C_n$, $C_{nv}$, $C_{nh}$, $D_n$, $D_{nh}$ and $D_{nd}$ the z-axis has to be the main rotational axis, secondary (twofold) rotational axis is always the x-axis, $\sigma_v$ is always the xz-plane and $\sigma_h$ the xy-plane. $O_h$ is oriented as $D_{4h}$. For $T_d$, the threefold rotational axis points in direction (1,1,1) and the z-axis is one of the twofold axes bisecting one vertex of the tetrahedron.

desy  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: \(T_d \rightarrow D_{2d} \rightarrow C_{2v} \rightarrow C_s\).

\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 \texttt{*}. If you enter \texttt&, 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*, \texttt{DEFINE} asks for the next atom type. Entering \texttt& here will allow you to re-define the last atom type and \texttt* to leave this mode and return to the geometry main menu. Enter \texttt{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.

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

The Cartesian coordinates and the definitions of the internal coordinates are read in free format; you only have to care for the keywords \texttt{$coord$} and (optionally) \texttt{$intdef$} and (important!) for the \texttt{$end$} 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 \cite{2.1.2}. Entering ‘!’ as first character of \texttt{file} will tell \texttt{DEFINE} to take \texttt{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 \cite{2.1}).

\texttt{aa file} same as \texttt{a}, but assumes the atomic coordinates to be in Å rather than a.u.

\texttt{sub} This command allows you to replace one atom in your molecule by another 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 2.1). Definitions of internal coordinates will be adjusted after substitution, but no new internal coordinates are created.

This command offers a submenu which contains everything related to internal coordinates. It is further described in Section 2.1.2.

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 2.1.3.

The command `w file` 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` 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.

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.

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 2.1.2).
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 2.2

2.1.2 Internal Coordinate Menu

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

imet <a> : PROVIDE B-MATRIX FOR ACTIVE INTERNAL COORDINATES
(CHECK COMPLETENESS AND NUMERICAL QUALITY
AND CHANGE REDUNDANT INTERNALS TO display)
idef : SUB-MENU FOR INTERACTIVE DEFINITION OF INTERNAL COORDINATES
ideg <a> : OUTPUT NUMBER OF TOT. SYMMETRIC INTERNAL DEGREES OF FREEDOM
iaut : TRY AUTOMATIC DEFINITION OF INTERNAL COORDINATES
iman <a> : MANIPULATE GEOMETRY BY CHANGING INTERNAL COORDINATE VALUES
imanat <i> : AS iman BUT STARTING AT INTERNAL COORD. NUMBER i
ic <i> <x>: CHANGE STATUS OF INTERNAL COORDINATE <i> TO <x>
e.g. ic 5 d TO MAKE 5TH COORD. display OR ic k d
irem <i> : REMOVE INTERNAL COORDINATE <i>,
e.g. irem d TO REMOVE ALL display COORDS
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

\texttt{dis} : ANY DISPLAY COMMAND e.g. disi OR disc
\texttt{disiat} \texttt{i} : AS disi BUT STARTING AT INTERNAL COORD. NUMBER \texttt{i}

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

ADDSING A QUESTION MARK TO ANY COMMAND MAY PROVIDE EXPLANATIONS

ENTER COMMAND OR HIT \texttt{>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 \textit{active} internal coordinates specified up to now. Only these coordinates are optimized during a geometry optimization.

\#f is the number of \textit{fixed} internal coordinates specified. These coordinates will be included in the \textbf{B}-matrix (see command \texttt{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 \texttt{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 \texttt{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 \texttt{idef} or \texttt{iaut}, see below); if you have more coordinates than degrees of freedom, you will have to throw away some of them (commands \texttt{irem} or \texttt{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.

\textbf{Description of commands}

\texttt{imet \textit{a}} This command computes the so-called \textbf{B}-matrix, which is the matrix of the derivatives of the (\textit{active} and \textit{fixed}) internal coordinates with respect to Cartesian coordinates. This matrix is used in program RELAX
2.1. THE GEOMETRY MAIN MENU

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 ired in the preceding menu.

Efficiency 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.

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 an 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
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

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 a.u. = 52.9 pm).

**ic i x**  
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
2.1. **THE GEOMETRY MAIN MENU**

\[
\text{<type> = stre invr bend outp tors linc linp}
\]

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 45) 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** The *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 \(b - c\).

- **tors** Dihedral angle: *tors abcd* is the angle between the planes \(a - b - c\) and \(b - c - d\).

- **linc** 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.

- **linp** 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\textsubscript{2}CCO, 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 \texttt{linc} and \texttt{linp}. The two coordinates \texttt{linc 1 3 2 4} and \texttt{linp 1 3 2 4} (where 1=oxyen, 2=carbon, 3=carbon, 4=hydrogen) would solve the problem.

The type \texttt{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. [18]). Command \texttt{iaut} uses linear combinations in most cases.

After you entered \texttt{k comp n} where \texttt{n} is the number of primitive internal coordinates to be combined, you will be asked to enter the type of the coordinate (\texttt{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 \texttt{k ring n} where \texttt{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 \texttt{<return>} will bring you back to the internal coordinate menu where you can see the new number of internal coordinates in the headline.

2.1.3 Manipulating the Geometry

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

\texttt{CARTESIAN COORDINATE MANIPULATION MENU :}
2.2. THE ATOMIC ATTRIBUTES MENU

move : 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.

2.2 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:
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.

- **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).
- **QZV1f** for basis set limit SCF or DFT; quadruple zeta + 3d1f or 4d1f for atoms beyond Ne, 3p1d for H.
- **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-\( \zeta \) (TZV) basis sets types with ECPs also for Rb–Rn, except lanthanides.

For calculations with the programs RIMP2 and RICC2 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, de-
noted def2-XYZ are available for atoms H–Rn [5]. 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 polarisation 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

<table>
<thead>
<tr>
<th>Exploratory</th>
<th>MP2: SVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlmostQuantitative</td>
<td>DFT: SV(P), HF: SVP, MP2: TZVPP; properties (HF and DFT): TZVPP</td>
</tr>
<tr>
<td>Quantitative</td>
<td>DFT: TZVP, HF: TZVPP, MP2: QZVPP</td>
</tr>
<tr>
<td>Basis Set Limit</td>
<td>DFT: QZVP, HF: QZVP</td>
</tr>
</tbody>
</table>

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

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).

**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 \texttt{b 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:

- \texttt{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.
  (\texttt{cc-pV6Z} only for H, He, B–Ne, Al–Ar; for Al–Ar also the recommended newer \texttt{cc-pV(X+d)Z} sets are available)

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

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

For calculations with the programs \texttt{RIMP2} and \texttt{RICC2} optimized auxiliary basis sets are available for the basis set series \texttt{cc-pVXZ}, \texttt{cc-pV(X+d)Z}, \texttt{cc-pwCVXZ}, \texttt{cc-pwCV(X+d)Z}, \texttt{aug-cc-pVXZ}, \texttt{aug-cc-pV(X+d)Z}, \texttt{aug-cc-pwCVXZ}, and \texttt{aug-cc-pwCV(X+d)Z} with $X = D, T, Q, 5$, but not for $X = 6$.

2.2.1 Description of the commands

\texttt{b} \quad With \texttt{b} you can specify basis sets for all atoms in your molecule. After entering \texttt{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 2.0.2), including \texttt{all} and \texttt{none}. Then you will be asked to enter the \textit{nickname} of the basis set to be assigned. There are two principal ways to do this:

1) If you are in the \textit{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 cannot 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 \cite{19} and references therein.

ecpl  \texttt{ecpl} gives you a list of all pseudopotentials assigned so far.

ecprm  ecprm allows to remove a pseudopotential assignment from the list. This command is useful if you want to perform an all electron calculation after an ECP treatment.

c  Command \texttt{c} assigns a special nuclear charge to an atom. This is useful to define dummy centers for counterpoise calculations where you set the nuclear charge to zero.

m  This command allows you to assign non-default atomic masses to an atom. Use this if you want to analyze isotopic shifts of calculated harmonic frequencies. The standard masses are those of the natural isotope mix.

dat  \texttt{dat} gives you a list of all data already specified.

*  This is again the usual command to leave a menu and write all data to file \texttt{control} (or any other output file). It is not possible to leave this menu unless basis sets have been specified for all atoms in your molecule. If you do not want to use a basis set for one or more atoms, use the basis set nickname \texttt{none}. On leaving this menu, the data groups $\texttt{atoms}$ and $\texttt{basis}$ will be written to the output file.

After you finished this menu, you will enter the third main menu of \texttt{DEFINE} which deals with start vectors and occupation numbers.

\section{Generating MO Start Vectors}

\subsection{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 \texttt{use}), by Hamilton core guess (\texttt{hcore}), or by an extended Hückel calculation which can be performed automatically (\texttt{eht}). An occupation of the start orbitals will be proposed and can be modified if desired.
2.3. GENERATING MO START VECTORS

\begin{itemize}
\item \texttt{infsao} : OUTPUT SAO INFORMATION
\item \texttt{eht} : PROVIDE MOS \\& OCCUPATION NUMBERS FROM EXTENDED HUECKEL GUESS
\item \texttt{use <file>} : SUPPLY MO INFORMATION USING DATA FROM <file>
\item \texttt{man} : MANUAL SPECIFICATION OF OCCUPATION NUMBERS
\item \texttt{hcore} : HAMILTON CORE GUESS FOR MOS
\item \& : MOVE BACK TO THE ATOMIC ATTRIBUTES MENU
\end{itemize}

THE COMMANDS \texttt{use} OR \texttt{eht} OR \* OR \texttt{q(uit)} TERMINATE THIS MENU !!!

FOR EXPLANATIONS APPEND A QUESTION MARK (?) TO ANY COMMAND

Recommendation

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

Based on the EHT orbital energies \texttt{DEFINE} proposes an occupation. If you accept you are done, if not you get the “occupation number assignment menu” explained in \ref{2.3.2}

Description of Commands

\texttt{infsao} \quad Command \texttt{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 \texttt{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 (\textbf{S}ymmetry \textbf{A}dapted \textbf{O}rbital). 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). \texttt{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 \texttt{DSCF} is written in terms of these SAOs.

\texttt{eht} \quad \texttt{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 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 unoccu-
pied 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 2.3.2. Note that the
occupation based on the Hückel calculation may be unreliable if the dif-
ference 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 cal-
culation, of which all data groups related to occupation numbers and
vectors will be read. As the new generated data will overwrite the ex-
isting 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 di-
rectory 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 calcula-
tion and the SCF vector from the old calculation will be projected onto
the space which is spanned by your present basis set. These start vec-
tors 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 2.3.3.
The same procedure applies to the open-shell occupation numbers after
you finished the closed-shell occupations.
2.3. GENERATING MO START VECTORS

**hcore**  
hcore tells programs DSCF and RIDFT to run without a start vector (it writes the data group $\text{scfmo none}$ 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).

*  
This command (as well as use and eht) terminates this menu, but without providing a start vector. If the keyword $\text{scfmo}$ exists in your input file, it will be kept unchanged (i.e. the old vector will be taken), otherwise $\text{scfmo none}$ 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 $\text{closed shells}$, $\text{open shells}$ (optionally) and $\text{scfmo}$ will be written to file. You will then reach the last of the four main menus (the General Menu) which is described in Section 2.4.

2.3.2 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
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

<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 eht 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 NH₃ in C₃ᵥ and TZVP you get, e.g.:

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

\textbf{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.

\textbf{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.

\textbf{&} 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).

\textbf{*} * 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 2.4. 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.

\textbf{e} Calculates and displays the extended Hückel total energy of your molecule.

\textbf{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.

### 2.3.3 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 \textbf{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 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.

2.3.4 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 (\( ^3P, ^1D, ^1S \)) which have different values of a and b. For the definition of these parameters and their use refer to Roothaan’s original paper [20]. 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 \( d^0 \) 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 \( d^0 \) 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 4.3.
2.4 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**

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

**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/cc2` : OPTIONS AND DATA GROUPS FOR MP2, CC2, ETC.
- `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
- `e` : DEFINE EXTERNAL ELECTROSTATIC FIELD
- `dft` : DFT Parameters
ri : RI Parameters
rijk : RI-JK-HF Parameters
trunc : USE TRUNCATED AUBASIS DURING ITERATIONS
marij : MULTIPOLE ACCELERATED RI-J
dis : DISPLAY MOLECULAR GEOMETRY
list : LIST OF CONTROL FILE
& : GO BACK TO OCCUPATION/ORBITAL ASSIGNMENT MENU
* or q : END OF DEFINE SESSION

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.

2.4.1 Important commands

DFT calculations

Command dft leads you to the menu:

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.

To activate DFT input on and then specify the grid for the quadrature of exchange-correlation terms. TURBOMOLE offers grids 1 (coarse) to 7 (finest), and the multiple grids m3 to m5. 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 m3, for clusters with more than 50 atoms use m4.

The functionals supported are obtained with the command func:
### 2.4. The General Options Menu

#### 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>pulda</td>
<td>LDA</td>
<td>S</td>
<td>PW</td>
<td>1,2,4</td>
</tr>
<tr>
<td>beck-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>HGGA</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>
</tbody>
</table>

Default is b-p, i.e. B-P86, which is probably best for the whole of Chemistry [21]. For main group compounds we recommend b3-lyp; note that GAUSSIAN uses partly different implementations [21].

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
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

m: CHANGE MEMORY FOR RI
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 R1DFT 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 2.2). Where available, the program will select by default the auxiliary basis sets optimized for the orbital basis used.

The RI-J option is only supported by programs R1DFT 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.

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)

**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 [23], 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 2.4.1. 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>thre</td>
<td>1.000000E-06</td>
<td>threshold for ENERGY CHANGE</td>
</tr>
<tr>
<td>thrd</td>
<td>1.000000E-03</td>
<td>threshold for MAX. DISPL. ELEMENT</td>
</tr>
<tr>
<td>thrg</td>
<td>1.000000E-03</td>
<td>threshold for MAX. GRAD. ELEMENT</td>
</tr>
<tr>
<td>rmsd</td>
<td>5.000000E-04</td>
<td>threshold for RMS OF DISPL.</td>
</tr>
<tr>
<td>rmsg</td>
<td>5.000000E-04</td>
<td>threshold for RMS OF GRAD.</td>
</tr>
</tbody>
</table>

defl : set default values.

OPTIMIZATION refers to

<table>
<thead>
<tr>
<th>Option</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>off</td>
<td>int: INTERNAL coordinates</td>
</tr>
<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

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>itvc</td>
<td>0</td>
<td>itvc : change INDEX OF TRANSITION VECTOR</td>
</tr>
<tr>
<td>updte</td>
<td>bfgs</td>
<td>updte: change method of HESSIAN UPDATE</td>
</tr>
<tr>
<td>hsfrq</td>
<td>0</td>
<td>hsfrq: frequency of HESSIAN CALCULATION</td>
</tr>
<tr>
<td>kptm</td>
<td>0</td>
<td>kptm : FREEZING transition vector INDEX</td>
</tr>
<tr>
<td>hdiag</td>
<td>5.000000E-01</td>
<td>hdiag: change DIAGONAL HESSIAN ELEMENTS</td>
</tr>
<tr>
<td>rmax</td>
<td>3.000000E-01</td>
<td>rmax : change MAX. TRUST RADIUS</td>
</tr>
<tr>
<td>rmin</td>
<td>1.000000E-04</td>
<td>rmin : change MIN. TRUST RADIUS</td>
</tr>
<tr>
<td>trad</td>
<td>3.000000E-01</td>
<td>trad : change TRUST RADIUS</td>
</tr>
</tbody>
</table>

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 $\text{scfconv}=7$, see Section 2.4.2.

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>
</table>
2.4. THE GENERAL OPTIONS MENU

<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. rpas, and quit this menu, you will get another menu:

SELECT IRREP AND NUMBER OF STATES
ENTER ? FOR HELP, * OR Q TO QUIT, & TO GO BACK

This should be self-evident.

**MP2 and RI-MP2**

We recommend to use MP2 together with the RI technique: program Rimp2. This is more efficient and supports the frozen core option in the gradient calculation.

The entry 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. It covers all keywords required for Rimp2 calculations, Mandatory for Rimp2 runs is the specification of the auxiliary basis set using the menu entry cbas. (Alternatively, the rimp2prep tool can be used to set the keywords needed for Rimp2 calculations.)

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

**CC2 calculations**

The entry cc2 leads to a submenu which allows to set a number of keywords essential for calculations with the program Ricc2. In particular it allows the assignment of auxiliary basis sets (mandatory for Ricc2!), the specification of frozen orbitals, and the definition of a scratch directory and of the maximum core memory usage.
2nd analytical derivatives

The program 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 d-functions, and ROHF as well as broken occupation numbers are not allowed. For better efficiency, in case of larger systems, use the keyword $maxcor as described in Chapter 8 to reduce computational cost. RI will be used if the RI option for DFT has been specified.

2.4.2 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 12. 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

conv : ACCURACY OF SCF-ENERGY $scfconv
thi : INTEGRAL STORAGE CRITERIA $thize $thime
ints : INTEGRAL STORAGE ALLOCATION $scfintunit
iter : MAXIMUM NUMBER OF ITERATIONS $scfiterlimit
diis : DIIS CONVERGENCE ACCELERATION $scfdiis
damp : OPTIONS FOR DAMPING $scfdamp
shift: SHIFTING OF ORBITALS $scforbitalshift
order: ORDERING OF ORBITALS $scforbitalorder
fermi: THERMAL SMEARING OF OCC. NUMBERS $fermi

By the command $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 '$drvopt, $drvtol'
---

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.4. **THE GENERAL OPTIONS MENU**

<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 BASIS SET exponents/scaling factors/contraction coefficients</td>
</tr>
<tr>
<td>glb</td>
<td>F</td>
<td>energy derivative with respect to 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>
<tr>
<td>fa</td>
<td>F</td>
<td>SPECTROSCOPIC ANALYSIS only</td>
</tr>
<tr>
<td>tol</td>
<td>0.1000-06</td>
<td>derivative integral cutoff</td>
</tr>
</tbody>
</table>

---

use <opt> for enabling, -<opt> for disabling of logical switches
&lt;&gt; 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:

```
------------------------------------------------------------------------
derivative debug options '$drvdebug'
------------------------------------------------------------------------
| option | status | description |
------------------------------------------------------------------------
| disp1e | F | display 1e contributions to desired derivatives |
| only1e | F | calculate 1e contributions to desired derivatives only |
| debug1e | F | display 1e shell contributions to desired derivatives |
| | | (WARNING : this produces large outputs!) |
| debug2e | F | display 2e shell contributions to desired derivatives |
| | | (WARNING : this produces VERY large outputs!) |
| debugvib | F | debug switch for vibrational analysis (force only) |
| notrans | F | disable transfer relations (gradient only!) |
| novirial | F | disable virial scaling invariance in basis set |
| | | (WARNING : this produces VERY large outputs!) |
| | | optimizations (gradient only) |
------------------------------------------------------------------------
```

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.

2.4.3 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 3.3). 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:
2.4. THE GENERAL OPTIONS MENU

optimization options for RELAX

<table>
<thead>
<tr>
<th>option</th>
<th>status</th>
<th>description : optimization refers to</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:

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:

---

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:
2.4. THE GENERAL OPTIONS MENU

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 (STEPEST DESCENT)</td>
</tr>
<tr>
<td>bfgs</td>
<td>F</td>
<td>BROYDEN-FLETCHER-GOLDFARB-SHANNON 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>schlegel</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 convergence in most cases.

General Boundary Conditions for Update

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

DEFINE BOUNDARY CONDITIONS FOR UPDATE

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

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

Special Boundary Conditions for Ahlrichs and Pulay Updates

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

DEFINE BOUNDARY CONDITIONS FOR AHLRICHS OR PULAY UPDATE

<table>
<thead>
<tr>
<th>option</th>
<th>description</th>
</tr>
</thead>
</table>
| modus <i>  | DEFINE MODUS FOR GDIIS PROCEDURE : MINIMIZE
|            | <dq|dq> IF <i> = 0                                  |
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

| <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 |

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

For detailed description consult Section 3.3.

OPTIONS FOR MANIPULATING THE HESSIAN

<table>
<thead>
<tr>
<th>option</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagonal</td>
<td>RESTRICT UPDATE TO DIAGONAL-ELEMENTS IF METHOD IS BFGS, DFP OR MS. DEFAULT=n</td>
</tr>
<tr>
<td>offreset</td>
<td>DISCARD OFF-DIAGONAL ELEMENTS. DEFAULT=n</td>
</tr>
<tr>
<td>offdamp &lt;r&gt;</td>
<td>DAMP OFF-DIAGONAL ELEMENTS BY 1/(1+&lt;r&gt;) DEFAULT= 1.000</td>
</tr>
<tr>
<td>damp &lt;real&gt;</td>
<td>DAMP UPDATE BY 1/(1+&lt;real&gt;), DEFAULT=.0000E+00</td>
</tr>
<tr>
<td>scale &lt;real&gt;</td>
<td>SCALE INPUT HESSIAN BY &lt;real&gt;, DEFAULT= 1.000</td>
</tr>
<tr>
<td>allow &lt;real&gt;</td>
<td>SCALE INPUT HESSIAN BY &lt;real&gt;/</td>
</tr>
<tr>
<td>min &lt;real&gt;</td>
<td>DO NOT ALLOW EIGENVALUES OF HESSIAN TO DROP BELOW &lt;real&gt;. DEFAULT= .1000E-02</td>
</tr>
<tr>
<td>reset &lt;real&gt;</td>
<td>USE &lt;real&gt; AS A RESET VALUE FOR TOO SMALL EIGENVALUES (CP. min). DEFAULT=.1000E-02</td>
</tr>
<tr>
<td>max &lt;real&gt;</td>
<td>DO NOT ALLOW EIGENVALUES OF HESSIAN TO BECOME LARGER THAN &lt;real&gt;. DEFAULT= 1000.</td>
</tr>
</tbody>
</table>

WITH THE EXCEPTION OF min, reset AND max, ALL OPTIONS MAY BE DISABLED BY ENTERING -<opt>
<RETURN> OR * OR q(uilt) 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:
2.4. THE GENERAL OPTIONS MENU

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 2nd derivatives calculation. DEFAULT(n)</td>
</tr>
<tr>
<td>diag</td>
<td>use diagonal matrix with diagonal elements set individually within data groups $intdef or $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 ). 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 calculation 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).

2.4.4 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:

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.

2.4.5 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 $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 $properties shall be written:

data group $properties 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:

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
2.4. THE GENERAL OPTIONS MENU

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>add/change options for data group $moments</th>
<th>option</th>
<th>status</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- point <x> <y> <z> | T | reference point = (x,y,z)
- atom <i> | F | reference point = atom no. <i>
- 0th | T | compute 0th moment
- 1st | F | compute 1st moment
- 2nd | F | compute 2nd moment
- 3rd | F | compute 3rd moment

* or q(uit) : terminate input

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.
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

Option potential

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

list of 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
----------------------------------------------------------------
shells to be localized
----------------------------------------------------------------
a1  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>

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

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.
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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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):
-------------------------------------------------------------------
MAOs are employed if 'atomic' density eigenvalues exceed a threshold of .1000
-------------------------------------------------------------------

specify the appropriate option if you want to use another global criterion for selecting 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>'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:
CHAPTER 2. PREPARING YOUR INPUT FILE WITH DEFINE

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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3c &lt;r&gt;</td>
<td>T</td>
<td>compute 3-center SEN and print if</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4c &lt;r&gt;</td>
<td>T</td>
<td>compute 4-center SEN and print if</td>
</tr>
<tr>
<td></td>
<td></td>
<td></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 10. 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:
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 3

Calculation of Molecular Structure and *Ab Initio* Molecular Dynamics

3.1 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.

3.1.1 Options

Given a Korn shell the usage is:

```
nohup jobex &
```

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

```
nohup jobex -statpt &
```

`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
    $10^\left(-<\text{integer}>\right)$ Hartree (default: 6)

-gcart integer converge maximum norm of cartesian gradient up to
    $10^\left(-<\text{integer}>\right)$ atomic units (default: 3)

-c integer perform up to integer cycles (default: 20)

-dscf begin with a direct SCF step

-grad begin with a gradient step

-relax begin with a force relaxation step

-statpt use the STATPT program for force relaxation

-trans perform transition state search using program STATPT (implies -statpt)

-level level define the optimization level, level=scf, mp2, cc2, or uff
    (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

-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: mdmaster).

-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
3.1.2 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.

3.2 Program STATPT

3.2.1 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 [24]. 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) [25].

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 [26].

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 threchange (default: $10^{-6}$ a.u.),
- the maximum displacement element drops below the value given by thrmax\-displ (default: $10^{-3}$ a.u.),
- the maximum gradient element drops below the value given by thrmaxgrad (default: $10^{-3}$ a.u.),
- the root mean square of the displacement elements drops below the value given by thrrmsdispl (default: $5 \cdot 10^{-4}$ a.u.),
- the root mean square of the gradient elements drops below the value given by thrmsgrad (default: $5 \cdot 10^{-4}$ a.u.).

The default values for the convergence criteria can be changed using the stp menu of DEFINE. The necessary keywords are described in Section [12.2.14] below.

For structure optimization of minima with STATPT as relaxation program use
3.2. PROGRAM STATPT

jobex -statpt &

TS optimizations are performed by the JOBEX invokation as

jobex -trans &

which implies use of STATPT for force relaxation.

3.2.2 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 STATPT. For minimization, a diagonal matrix or approximate Hessian matrix from a forcefield calculation using UFF (see Section 3.4) 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 8). 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.

STATPT automatically takes the best choice of the Hessian from the 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 STATPT, in addition to the $les keyword require the following keywords to be added manually in the control file:

$h0hessian
$nomw

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

3.2.3 Finding Minima

Simply specify the $statpt keyword in the control file and run JOBEX as explained above. You can very often speedup the optimization by calculating the initial Hessian matrix using UFF.
3.2.4 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 build 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 STATPT uses a different ordering of eigenvalues as compared to the AOFORCE output—six (five) zero eigenvalues are shifted to the end.

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 necessarily 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.
3.3 Program Relax

3.3.1 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 \textit{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.

3.3.2 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).
CHAPTER 3. STRUCTURE OPTIMIZATIONS

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**

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

- **dqmax=real** Maximum total coordinate change (default: 0.3).
- **interpolate on/off** Calculate coordinate update by inter/extrapolation using coordinates and gradients of the last two optimization cycles (default: interpolate on) if possible.
- **statistics integer/off** Display optimization statistics for the integer previous optimization cycles. Without integer all available information will be displayed. off suppresses optimization statistics.
3.3. PROGRAM RELAX

The following data blocks are used by program RELAX:

1. Input data from gradient programs 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 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 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 2.0.4. Normally internal coordinates are not used for input or output by the electronic structure programs (DSCF, MPGRAD, etc.). Instead the coordinates, gradients, etc. are automatically converted to internal coordinates by RELAX on input and the updated positions of the nuclei are written in cartesian coordinates to the data group \$\text{coord}$. Details are explained in the following sections.

3.3.3 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):

   $$F^k = F^{k-1} + \frac{Z^{k-1}(Z^{k-1})^\dagger}{(Z^{k-1})^\dagger dG^{k-1}}$$
CHAPTER 3. STRUCTURE OPTIMIZATIONS

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):

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

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:

- \( F^k = (H^k)^{-1} \) 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

\[ H_{ii}^k = \sqrt{H_{ii}^{k-1}(h_i + d_i)} \]

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

\[ h_i = \frac{\sum_k dG^k_i dq^k_i}{\sum_k (dq^k_i)^2} \]

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

\[ d_i = \sqrt{\frac{\sum_k (dg^k_i)^2}{\sum_k (dq^k_i)^2} - h_i^2} \]

Another alternative is to use DIIS-like methods: structure optimization by direct inversion in the iterative subspace. (See ref. [27] 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 \(n = 4\):

\[ n_{\text{cycles}} < n: \] geometry update by inter/extrapolation using the last 2 geometries.

\[ n_{\text{cycles}} \geq n: \] 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: \([28,24,29,27,30,31]\)

### 3.3.4 Definition of Internal Coordinates

If structure optimizations are to be performed in the space of internal coordinates \((\text{\$optimize internal}, \text{is the default setting})\), appropriate internal coordinate definitions have to be provided on data block \$intdef. The types available and their definitions are described in Section 2.1.2. For recommendations about the choice of internal coordinates consult ref. [18]. Nevertheless the structure of \$intdef will shortly be described. The syntax is (in free format):

\[
1\ k\ 1.00000000\ \text{bend}\ 1\ 2\ 3\ \text{val}=1.9500\ \text{fdiag}=.6666
\]

The first items have been explained in Chapter 2.

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

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

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

### 3.3.5 Structure Optimizations Using Internal Coordinates

This is the default task of RELAX \((\text{\texttt{\$optimize internal on}} \text{does not need to be specified!})\) You need as input the data groups :
$\text{grad}$ cartesian coordinates and gradients as provided and accumulated in subsequent optimization cycles by the programs GRAD, or RDGRAD etc.

$\text{intdef}$ definitions of internal coordinates.

$\text{redundant}$ definitions of redundant coordinates.

Output will be the updated coordinates on $\text{coord}$ and the updated force constant matrix on $\text{forceapprox}$. If any non-default force constant update option has been chosen, RELAX increments its counting variables $\text<numgeo>, \text<numpul>$ within command keyword $\text{forceupdate}$. If the approximate force constant has been initialized ($\text{forceinit on}$) RELAX switches the initialization flag to $\text{forceinit off}$. 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 3.3.11) or within a DEFINE session.

2. Have a look at the eigenvectors of the $\text{Bmb}^\dagger$-matrix. Set some ‘?’ behind keyword $\text{intdef}$, 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)!

### 3.3.6 Structure Optimization in Cartesian Coordinates

For this task you have to specify:

$\text{optimize}$

```
  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 $\text{grad}$ as provided by on of the gradient programs. For the first coordinate update an approximate force constant matrix is needed in data group $\text{forceapprox}$. Output will be the updated coordinates on $\text{coord}$, and the updated force constant matrix on $\text{forceapprox}$.

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:
$coord$

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

3.3.7 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 2.2

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

For an example, see Section 13.5.1

3.3.8 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).

3.3.9 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 [32].

To optimize the structure and a global scaling factor specify:

$optimize$

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).

3.3.10 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.

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

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

$interconversion on$

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.
3.3. **PROGRAM RELAX**

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)

### 3.3.12 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 $BmB^{-1}$-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:

$$m-matrix$$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### 3.3.13 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. [30]. For optimization of basis set parameters less information is available. When neither data block $forceapprox is available nor $forceinit on is set,
CHAPTER 3. STRUCTURE OPTIMIZATIONS

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:

- 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.

3.3.14 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. H2O.

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

This should be self-evident. To see the current—or, if the optimization is converged, the final—atomic distances use the tool dist. Bond angles, torsional angles 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>.

3.4 Force Field Calculations

3.4.1 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.

3.4.2 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, inverison and non-bonded terms (force field terms) based on the connectivity specified in the topology file.

3.4.3 The UFF implementation

The UFF implementation follows the paper by Rappé [6]. The energy expression in UFF is as follows:
The Fourier coefficients $C_A^0, C_A^1, C_A^2$ of the general angle terms are evaluated as a function of the natural angle $\theta_0$:

\[
C_A^2 = \frac{1}{4 \sin^2 \theta_0} \quad (3.2)
\]
\[
C_A^1 = -4 \cdot C_A^3 \cos \theta_0 \quad (3.3)
\]
\[
C_A^0 = C_A^2 \left( 2 \cos^2 \theta_0 + 1 \right) \quad (3.4)
\]
3.4. FORCE FIELD CALCULATIONS

\( \omega \) \hspace{1cm} inversion- or out-of-plane-angle at atom \( I \).

\( V_\omega \) \hspace{1cm} height of the inversion barrier.

\( C_0^I, C_1^I, C_2^I \) \hspace{1cm} Fourier coefficients of the inversions terms.

\( x, x_{IJ} \) \hspace{1cm} distance and natural distance of two non-bonded atoms \( I \) and \( J \).

\( D_{IJ} \) \hspace{1cm} depth of the Lennard–Jones potential.

\( q_I, \epsilon \) \hspace{1cm} 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 \( \text{sp}^3 \) 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 [33]. 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 [34,35,28,36]. Pulay’s DIIS procedure is implemented for big molecule to accelerate the optimization [37,27].

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
  1.31008893646566  3.07002878668872  1.68840815751978  h
  1.31008893646566  3.07002878668872  -1.68840815751978  h
  4.12184425921830  2.06288409251899  0.00000000000000  h
$end
```
3.5 Molecular Dynamics Calculations

*Ab initio* molecular dynamics (MD) can be carried out on the ground state Born–Oppenheimer potential hypersurface. 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 program Mdprep and generating the command file `mdmaster`. If this is successful, the MD run itself may be started: `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. DSCF and GRAD). The MD program FROG uses the Leapfrog Verlet algorithm\(^{38}\) to turn the gradients into new atomic positions and velocities. The atoms thus undergo classical Newtonian dynamics on the *ab initio* potential hypersurface. Trajectory information is recorded in a log file (`mdlog`). It is possible to instruct FROG to heat or cool the system, conserve total energy or read in new positions or velocities: the appropriate keywords are described in Section\(^{12.2.17}\) below.
Chapter 4

Hartree–Fock and DFT Calculations

DSCF and GRAD are modules for energy and gradient calculations at the Hartree–Fock (HF) and density functional theory (DFT) levels 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. All functionalities are implemented for closed-shell RHF and open-shell UHF reference wavefunctions.

The most important special features of the DSCF and GRAD modules are:

- Selective storage of the most time consuming and frequently used integrals. The integral storage is controlled by two threshold parameters, $\text{thize}$ and $\text{thime}$, 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 DFT level using pure functionals, i.e. without the HF exchange term. 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 a tenfold speedup compared to the conventional method based on four-center ERI’s (for example the
DSCF module).

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 1000–2000 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 with automatic generation of optimal start vectors by solving the HF instability equations 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 $ridft$ is found in the control file but not the keyword $dft$. 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.

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
4.1. BACKGROUND THEORY

requires a separate DSCF statistics run to estimate the disk space needed for integral
storage. The statistics run requires the keyword $statistics dcf 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 3.1 for additional options and parameters for geometry optimiza-
tions and ab initio molecular dynamics calculations.

4.1 Background Theory

In Hartree–Fock theory, the energy has the form,

\[ E_{HF} = h + J - K + V_{nuc}, \]  \hspace{1cm} (4.1)

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 determi-
nant is replaced by a more general expression, the exchange-correlation functional,
which can include terms accounting for both exchange energy and the electron cor-
relation which is omitted from Hartree–Fock theory. The DFT energy is expressed
as a functional of the molecular electron density \( \rho(r) \),

\[ E_{DFT}[\rho] = T[\rho] + V_{ne}[\rho] + J[\rho] + E_x[\rho] + E_c[\rho] + V_{nuc}, \]  \hspace{1cm} (4.2)
where $T[\rho]$ is the kinetic energy, $V_{ne}[\rho]$ 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, DSCF and GRAD modules support hybrid functionals in which the exchange functional includes the Hartree–Fock exchange, e.g. B3-LYP.

### 4.2 Exchange-Correlation Functionals Available

The following exchange-correlation functionals are available for all four modules (DSCF, GRAD, RIDFT, and RDGRAD):

- The Slater–Dirac exchange functional only (S) \cite{41,42}.
- The 1980 correlation functional (functional V in the paper) of Vosko, Wilk, and Nusair only (VWN) \cite{43}.
- A combination of the Slater–Dirac exchange and Vosko, Wilk, and Nusair 1980 (functional V) correlation functionals (S-VWN) \cite{41,42,43}.
- The S-VWN functional with VWN functional III in the paper. This is the same functional form as available in the Gaussian program \cite{41,42,43}.
- A combination of the Slater–Dirac exchange and Perdew-Wang (1992) correlation functionals \cite{41,42,44}.
- A combination of the Slater–Dirac exchange and Becke’s 1988 exchange functionals (B88) \cite{41,42,45}.
- Lee, Yang, and Parr’s correlation functional (LYP) \cite{46}.
- The B-LYP exchange-correlation functional (B88 exchange and LYP correlation functionals) \cite{41,42,45,46}.
- The B-VWN exchange-correlation functional (B88 exchange and VWN (V) correlation functionals) \cite{41,42,45,43}.
- The B-P86 exchange-correlation functional (B88 exchange, VWN(V) and Perdew’s 1986 correlation functionals) \cite{41,42,45,43,47}.
- The Perdew, Burke, and Ernzerhof (PBE) exchange-correlation functional \cite{41,42,44,48}.
- The Tao, Perdew, Staroverov, and Scuseria functional (Slater–Dirac, TPSS exchange and Perdew-Wang (1992) and TPSS correlation functionals) \cite{41,42,44,49}.
4.2. EXCHANGE-CORRELATION FUNCTIONALS AVAILABLE

Additionally, for the Dscf and Grad modules 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) \[41,42,45,46,50\].

- The B3-LYP exchange-correlation functional (Becke’s three-parameter functional) with the form,

\[
0.8S + 0.72B88 + 0.2HF + 0.19VWN(V) + 0.81LYP \tag{4.3}
\]

where HF denotes the Hartree-Fock exchange \[41,42,45,46,51\].

- 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,

\[
0.75(S + PBE(X)) + 0.25HF + PW + PBE(C) \tag{4.4}
\]

where PBE(X) and PBE(C) are the Perdew–Burke–Ernzerhof exchange and correlation functionals and PW is the Perdew–Wang correlation functional \[41,42,44,48,52\].

- The TPSSH exchange-correlation functional of Staroverov, Scuseria, Tao and Perdew with the form,

\[
0.(S + TPSS(X)) + 0.1HF + PW + TPSS(C) \tag{4.5}
\]

where HF denotes the Hartree–Fock exchange \[41,42,44,49,53\].

- The localized Hartree–Fock method (LHF) to obtain an effective exact exchange Kohn–Sham potential \[54,55\].
4.3 Restricted Open-Shell Hartree–Fock

4.3.1 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 $e_g^2$) are:

\begin{verbatim}
$open shells type=1
   eg 1
$roothaan 1
   a=1  b=2
\end{verbatim}

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 13.6 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 ($k, m$):

\[
E = 2 \sum_k h_{kk} + \sum_{k,l} (2J_{kl} - K_{kl})
+ f [2 \sum_m h_{mm} + f \sum_{m,n} (2aJ_{mn} - bK_{mn})
+ 2 \sum_{k,m} (2J_{km} - K_{km})]
\]

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.

4.3.2 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 4.1: Roothaan-coefficients $a$ and $b$ for cases with degenerate orbitals.

$n_{ir}=2$: $e$ (div. groups), $\pi$, $\delta$ ($C_{\infty v}, D_{\infty h}$)

<table>
<thead>
<tr>
<th>$n$</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>$^3\Sigma$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$^1E^*$</td>
<td>$^1\Delta$</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>$^2E$</td>
<td>$^2\Pi$</td>
<td>$^2\Delta$</td>
<td>8/9</td>
<td>8/9</td>
</tr>
</tbody>
</table>

$n_{ir}=3$: $p$ ($O(3)$), $t$ ($T$, $O$, $I$)†

<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></td>
<td></td>
<td>$^1S$</td>
<td>0</td>
<td>-3</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>

### (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>††</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>$^5A$</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>††</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 4.1: Roothaan-coefficients $a$ and $b$ for cases with degenerate orbitals (continued).

<table>
<thead>
<tr>
<th>$d(O3)$, $h(I)$</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>($mainly$ $high$-spin $cases$ $work$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>$f$</td>
<td>$d^n$</td>
</tr>
<tr>
<td>1</td>
<td>1/10</td>
<td>$2^D$</td>
</tr>
<tr>
<td>2</td>
<td>1/5</td>
<td>$^3F + ^3P_{\dagger\dagger}$</td>
</tr>
<tr>
<td>3</td>
<td>3/10</td>
<td>$^4F + ^4P_{\dagger\dagger}$</td>
</tr>
<tr>
<td>4</td>
<td>2/5</td>
<td>$^5D, ^5H$</td>
</tr>
<tr>
<td>5</td>
<td>1/2</td>
<td>$^6S, ^6A$</td>
</tr>
<tr>
<td>6</td>
<td>3/5</td>
<td>$^5D, ^5H$</td>
</tr>
<tr>
<td>7</td>
<td>7/10</td>
<td>$^4F + ^4P_{\dagger\dagger}$</td>
</tr>
<tr>
<td>8</td>
<td>4/5</td>
<td>$^3F + ^3P_{\dagger\dagger}$</td>
</tr>
<tr>
<td>9</td>
<td>9/10</td>
<td>$^2D, ^2H$</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 $E+T$

$\dagger\dagger$ $(a,b)$ describes weighted average of high spin states, not a CSF.

Example

The $4d^95s^2\ 2D$ state of Ag, in symmetry $I$

$\$closed shells
  a  1-5       (2)
  t1 1-3       (2)
  h  1          (2)
$\$open shells type=1
  h  2           (9/5)
$\$roothaan
  1
  a = 80/81      b = 80/81
4.3.3 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 \(4d^55s^1\) \(7S\) state of Mo, treated in symmetry I

\[ \text{$roothaan} 1 \]
\[ a = 1 \quad b = 2 \]
\[ \text{$closed shells} \]
\[ a \quad 1-4 \quad (2) \]
\[ t1 \quad 1-3 \quad (2) \]
\[ h \quad 1 \quad (2) \]
\[ \text{$open shells type=1} \]
\[ a \quad 5 \quad (1) \]
\[ h \quad 2 \quad (1) \]

Two-electron singlet coupling

The two MOs must have different symmetries (not required for triplet coupling, see example [4.3.3]). 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.

\[ \text{$closed shells} \]
\[ a1 \quad 1-2 \quad (2) \]
\[ b1 \quad 1 \quad (2) \]
\[ \text{$open shells type=1} \]
\[ a1 \quad 3 \quad (1) \]
\[ b2 \quad 1 \quad (1) \]
\[ \text{$roothaan} 1 \]
\[ \text{$rohf} \]
\[ 3a1-3a1 \quad a = 0 \quad b = 0 \]
\[ 1b2-1b2 \quad a = 0 \quad b = 0 \]
\[ 3a1-1b2 \quad a = 1 \quad b = -2 \]

Two open shells

This becomes tricky in general and we give only the most important case:
shell 1 is a Roothaan case, see 4.3.2

shell 2 is one electron in an a (s) MO \((n_{ir} = 1)\)

with parallel spin coupling of shells.

This covers e.g. the \(p^5 s^1\) \(^3\)P states, or the \(d^4 s^1\) \(^6\)D states of atoms. The coupling information is given following the keyword $rohf$. The \((a, b)\) within a shell are taken from above [4.3.2], the cross term (shell 1)–(shell 2) is in this case:

\[
a = 1 \quad \text{always} \\
b = 2 \quad \text{if } n \leq n_{ir} \quad b = \frac{2n_{ir}}{n} \quad \text{if } n > n_{ir}
\]

where \(n_{ir}\) and \(n\) refer to shell 1.

**Example 1:** The \(4d^4 5s^1\) \(^6\)D 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 \(4d^6 5s^1\) \(^7\)S state of Mo, symmetry I (see Section 4.3.3) 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
  a 1-4 (2) \\
t1 1-3 (2) \\
h 1 (2) 
```
4.3. RESTRICTED OPEN-SHELL HARTREE–FOCK

\$open shells type=1
\[
\begin{array}{ll}
a & 5 \\
\text{h} & 2
\end{array}
\] (1)

The shells 5s and 4d have now been made \textit{inequivalent}. Result is identical to 4.3.3 which is also more efficient.

**Example 3:** The \(4d^9 \, 5s^1 \, 3\text{D}\) state of Ni, symmetry I

\$closed shells
\[
\begin{array}{ll}
a & 1-3 \\
\text{t1} & 1-2
\end{array}
\] (2)

\$open shells type=1
\[
\begin{array}{ll}
a & 4 \\
\text{h} & 1
\end{array}
\] (1)

\$roothaan 1
\$rohf

4a-4a a = 0 b = 0
1h-1h a = 80/81 b = 80/81
4a-1h a =1 b = 10/9

(see basis set catalogue, basis \texttt{SV.3D} requires this input and gives the energy you must get)

4.3.4 Miscellaneous

**Valence states**

Valence states are defined as the weighted average of \textit{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.
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 \(1^S\) states of \(p^2\), \(p^4\), \(d^2\), \(d^6\), 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.
Chapter 5

Second-order Møller–Plesset Perturbation Theory

5.1 Functionalities of Mpgrad and Rimp2

TURBOMOLE offers two possibilities for the calculation of MP2 data. A "conventional" implementation [56], Mpgrad, based on the calculation of four-center integrals (not further developed for several years), and a treatment within the resolution-of-the-identity (RI) approximation [7], Rimp2.

Functionality 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.

Functionality of Rimp2:

- Calculation of MP2 energies and/or gradients for RHF and UHF wave functions within the efficient RI-approximation.
- The frozen core approximation is implemented for both RI-MP2 energies and gradients.
- Rimp2 needs optimised auxiliary basis sets, which are available for all TURBOMOLE standard basis sets (SVP, TZVP, TZVPP, QZVPP) as well as for the (aug-)cc-p(wC)VXZ (X = D, T, Q, 5) basis sets series (for Al–Ar also for the (aug-)cc-p(wC)V(X+d)Z series).
• Exploitation of symmetry of all point groups.

5.2 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:

\[ E_{MP2} = \frac{1}{4} \sum_{iajb} t_{ij}^{ab} \langle ij||ab \rangle, \]  

(5.1)

with the t-amplitudes

\[ t_{ij}^{ab} = \frac{\langle ij||ab \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}, \]  

(5.2)

\( i \) 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 RIMP2 (compared to MPGRAD) by a factor of ca. 5 (small basis sets like SVP) to ca. 10 (large basis sets like TZVPP) or more (for cc-pVQZ basis sets). The errors (differences to MPGRAD) of RIMP2 in connection with optimised auxiliary basis sets are small and well documented [8, 57]. The use of the MPGRAD module is recommended rather for reference calculations or if suitable auxiliary basis sets are not available.

5.3 How to Prepare and Perform MP2 Calculations

Prerequisites

Calculations with MPGRAD or RIMP2 require
5.3. **HOW TO PREPARE AND PERFORM MP2 CALCULATIONS**

- a converged SCF calculation with the one-electron density convergence threshold set to \$\text{denconv}\ 1.d-7\text{ or less}

- the maximum core memory the program is allowed to allocate should be defined in the data group \$\text{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 \$\text{freeze}

- the calculation of MP2 gradients is omitted by adding the flag \$\text{mp2energy}\ to\ the\ control\ file;\ in\ this\ case\ only\ MP2\ energy\ is\ calculated.

Calculations with RIMP2 moreover require

- an auxiliary basis defined in the data group \$\text{cbas}

this is not needed for Mpgrad, but here one needs

- a specification for scratch files and their size in data group \$\text{mointunit}\ (see\ Section\ 12.2.11)

- and the number of passes for integral evaluations and transformations in data group \$\text{traloop}

**Rimp2 Calculations**

1. RI-MP2 calculations require the specification of auxiliary basis sets (\$\text{cbas}) and a converged SCF calculation with the one-electron density convergence threshold set to \$\text{denconv}\ 1.d-7\ or less. In addition, the options \$\text{freeze}\ (frozen core approximation) and \$\text{maxcor}\ (maximum core memory usage) should be set. All these settings can be done during the input generation with the program Define under the entry mp2 of last main menu.

2. Alternatively, the interactive program RIMP2PREP can be used: This program sets default values for auxiliary basis sets (data group \$\text{cbas}), for frozen core orbitals (data group \$\text{freeze}, all orbitals with energies below -3 a.u. are suggested to be frozen) and for the amount of memory to be allocated (\$\text{maxcor}). These defaults can be confirmed with return or modified, if desired. Note: the amount of memory to be allocated determines the number of multiple passes and thus the efficiency of RIMP2.

   It is also possible to run RIMP2PREP directly after Define.

3. Start a single RIMP2 calculation with the command rimp2.

4. For optimisation of structure parameters at the RI-MP2 level use the command jobex -ri -level mp2.
Mpgrad Calculations

1. Add $\text{denconv} \; 1.\; d^{-7}$ to the control file and perform a DSCF run.

2. If any orbitals are decided to be excluded from MP2 treatment, add data group $\text{freeze}$ manually to the control file, see also Section 12.2.11.

3. For preparation of an MPGRAD run use the script $\text{Mp2prep}$:
   
   ```
   mp2prep -e/g -m memory -p discspace [scratch file directory]
   ```

   As an example, with the command
   
   ```
   mp2prep -e -m 100 -p 1000 /work
   ```

   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 control file ($straloop$) as well as the specification of scratch files ($mointunit$, see Section 12.2.11). Note: less disc space means more passes and thus lower efficiency of MPGRAD. Settings obtained by $\text{Mp2prep}$ may be changed manually. You may change the number of passes in $straloop$ by editing the control file (e.g. if the originally intended disc space is not available). To adapt the size of scratch files add $\text{statistics}$ mpgrad to control file and start an MPGRAD statistics run with the command $\text{mpgrad}$.

4. Start a single MPGRAD calculation with the command $\text{mpgrad}$.

5. For optimisation of structure parameters at the (non-RI-) MP2 level use the command $\text{jobex}$ -level mp2. Note, that the frozen core approximation is ignored in this case.

5.4 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 comparable) is recommended, when using SV(P) basis sets a qualitative trend can be expected at the most. Basis sets much larger than TZVPP (or cc-pVTZ) usually do not significantly improve results, moreover in this case the errors of the method and those of the basis sets are no longer balanced.

- It is recommended to exclude all non-valence orbitals from MP2 calculations, as neither TURBOMOLE standard basis sets nor cc-pVXZ (X=T,Q,5) 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 RIMP2PREP (orbitals below -3 a.u. are frozen) provides a reasonable guess.

- RIMP2: We strongly recommend the use of auxiliary basis sets optimized for the corresponding (MO) basis sets.

Comments on Output

- Most important output for RIMP2 and MPGRAD are of course MP2(+HF) energies (written standard output and additionally to file energy) and MP2(+HF) gradients (written to file gradient).

- In case of MP2 gradient calculations both modules also calculate the MP2 dipole moment from the MP2 density matrix (note, that in case of MPGRAD 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, but comparison of values for several similar calculations will yield some information. 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 ($d^8s^2$, ”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 re-
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 $\text{mp2occ}$ 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.
Chapter 6

Hartree–Fock and DFT
Response Calculations:
Stability, Dynamic Response
Properties, and Excited States

6.1 Functionalities of Escf and Egrad

Escf and Egrad are designed as efficient tools for response and excited state calculations on large molecules. 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 ⇒ 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 and GGA 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).

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

•Exited state densities ⇒ Charge moments, population analysis

•Excited state force constants by numerical differentiation of gradients (using the script Numforce)

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 [58].

6.2 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. [59,15,58,60,61] and refs. therein.

The first-order frequency dependent response of the density matrix can be expanded as

\[ \gamma(x, x') = \sum_{ai} \left\{ X_{ai} \phi_i(x) \phi_a^*(x') + Y_{ai} \phi_a(x) \phi_i^*(x') \right\} . \]  

(6.1)

The (real) expansion coefficients \( X_{ai} \) and \( Y_{ai} \) are conveniently gathered in a “super-vector”

\[ |X, Y\rangle = \begin{pmatrix} X \\ Y \end{pmatrix} \]  

(6.2)

on \( L \), the linear space of products of occupied and virtual ground state MOs \( \phi_i(x) \phi_a^*(x') \) 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”

\[ \Lambda = \begin{pmatrix} A & B \\ B & A \end{pmatrix}, \quad \Delta = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \]  

(6.3)

where the four-index quantities \( A \) and \( B \) are the so-called “orbital rotation Hessians”. Explicit expressions for \( A \) and \( B \) can be found, e.g., in ref. [15]. The vector \( |X, Y\rangle \) is determined as the solution of the TDHF/TDDFT response problem,

\[ (\Lambda - \omega \Delta)|X, Y\rangle = -|P, Q\rangle. \]  

(6.4)
6.2. THEORETICAL BACKGROUND

If $|X_\alpha, Y_\alpha\rangle$ arises from an electric dipole perturbation $|\mu_\alpha\rangle$, the electronic dipole polarizability at frequency $\omega$ is

$$\alpha_{\alpha\beta}(\omega) = -\langle X_\alpha, Y_\alpha|\mu_\beta \rangle,$$

(6.5)

$\alpha, \beta \in \{x, y, z\}$. Similarly, if $|m_\alpha\rangle$ is a component of the magnetic dipole moment operator, the optical rotation is

$$\delta_{\alpha\beta}(\omega) = -\frac{c}{\omega} \text{Im}\langle X_\alpha, Y_\alpha|m_\beta \rangle,$$

(6.6)

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. (6.4),

$$(\Lambda - \Omega_n \Delta)|X_n, Y_n\rangle = 0.$$  

(6.7)

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

$$\langle X_n, Y_n|\Delta|X_n, Y_n\rangle = 1.$$  

(6.8)

Transition moments are evaluated by taking the trace with one-particle operators, e.g.,

$$\mu^{0n} = \langle X_n, Y_n|\mu \rangle$$  

(6.9)

for the electric and

$$m^{0n} = \langle X_n, Y_n|m \rangle$$  

(6.10)

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 [59], but it is somewhat less affected by stability problems (see below).

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 DFT, while $A + B$ generally is not. See refs. [63,14] for further details.
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 [15],

\[
L[X, Y, \Omega, C, Z, W] = E_{\text{GS}} + \langle X, Y | \Lambda | X, Y \rangle - \Omega(\langle X, Y | \Delta | X, Y \rangle - 1) + \sum_{ia} Z_{ia} F_{ia} - \sum_{pq} W_{pq}(S_{pq} - \delta_{pq}).
\]  

(6.11)

Here \(E_{\text{GS}}\) 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/TDKS response equation \((6.4)\), 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...4.

### 6.3 Implementation

Without giving details, we discuss features of the implementation in ESCF and EGRAD that matter for applications. The interested reader is referred to the refs. given in the program headers as well as ref. \[64\].

**Simultaneous vector iteration.** The solutions of Eqs. \((6.4)\) and \((6.7)\) are expanded in a subspace of \(L\) which is iteratively expanded (Davidson method \[65\]). 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 \$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 vfile_<IR>, wfile_<IR>, and/or rhs_<IR>, where IR denotes an IR-REP of the point group (see below). Before the programs terminate, the converged vectors are written onto formatted files (type)<IR>, where type is an abbreviation for the type of response calculation performed (cf. \$scfinstab). Given these files in the working directory, ESCF and EGRAD calculations can be restarted or continued, e.g., with a larger number of roots.
6.4. HOW TO PERFORM

Integral direct algorithm. In the iterative method outlined above, the supermatrices $A$ 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 $\$thime$, $\$thize$, $\$scfintunit$, see Chapter 4. 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, → $\$symmetry$). The response and eigenvalue problems (6.4) and (6.7) 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 ($\$soes$, see below). For response calculations, the perturbation is automatically subduced 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.

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 $\$freeze$).

6.4 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 2 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 (TURBOSTEST directory).

6.4.1 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

\$$\texttt{scfconv 7}$$

\$$\texttt{denconv 1d-7}$$

in control, perform a DSCF statistics run, if semi-direct integral processing is to be used (see Chapter 1.7), 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.

### 6.4.2 Polarizabilities and Optical Rotations

The calculation of dynamic polarizabilities is controlled by the keyword

\$$\texttt{scfinstab dynpol \textit{unit}}$$

\textit{unit} specifies the unit of the following frequencies and may be \textit{ev}, \textit{nm}, \textit{1/cm}, or \textit{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{scfinstab dynpol nm 590 400 i}$$

and run ESCF,

```
escf > escf.out &
```

The resulting polarizabilities and rotatory dispersions are given in a.u. in the program output (\textit{escf.out} in the above example). For conversion of the optical rotation in a.u. to the specific rotation in deg\cdot[dm-(g/cc)]^{-1} \text{ see Eq. (15) of ref. [62].}

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{scfinstab polly}$$

before starting ESCF.
6.4. HOW TO PERFORM

6.4.3 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 eg and t2g 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 (→ $uhfmo_alpha, $uhfmo_beta) as well as occupation numbers (→ $alpha shells, $beta shells) for a spin-unrestricted calculation with equal numbers of α and β electrons (pseudo-singlet occupation).

6.4.4 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

\$scfinstab\ urpa
    for excitations out of spin-unrestricted reference states.

If it is intended to use the TDA instead, specify

\$scfinstab\ ciss
    for closed-shell singlet excitations,

\$scfinstab\ cist
    for closed-shell triplet excitations, and

\$scfinstab\ ucis
    for excitations out of spin-unrestricted reference states.

Next, the IRREPs of the excitations need to be defined, which is again accomplished using \$soes. For example, to calculate the 17 lowest excitations in IRREP b1g, the 23 lowest excitations in IRREP eu, and all excitations in IRREP t2g, use

\$soes
  b1g 17
  eu 23
  t2g all

and run \texttt{Escf}.

Note that \texttt{$soes$} specifies the IRREP of the \textit{excitation vector} which is not necessarily identical to the IRREP of the \textit{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

\$soes
  b1 1

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 \texttt{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 \texttt{Escf} 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

$$\texttt{rpacor} \ m$$

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 ($\texttt{ridft}$), it is recommended to set $\texttt{ricore}$ to a small value and $\texttt{rpacor}$ to a large value if the number of states is large, and vice versa if it is small.

By specifying

$$\texttt{spectrum} \ \texttt{unit} \quad \text{and/or}$$

$$\texttt{cdspectrum} \ \texttt{unit}$$

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.

### 6.4.5 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 $\texttt{soes}$ (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

$$\texttt{exopt} \ n$$

which explicitly enforces treatment of the $n$-th state; $n$ must be less or equal the number of states specified in $\texttt{soes}$. 
CHAPTER 6. HF AND DFT RESPONSE CALCULATIONS

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 1.7).

### 6.4.6 Excited State Force Constant Calculations

Excited state vibrational frequencies can be calculated by numerical differentiation of analytic gradients using NUMFORCE (see Chapter 8). 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 and 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.
Chapter 7

Second-Order Approximate Coupled-Cluster (CC2) Calculations

RICC2 is a module for the calculation of excitation energies and response properties at a correlated *ab initio* level, in particular the second-order approximate coupled-cluster model CC2 [66]. All calculations employ the resolution-of-the-identity (RI) approximation for the electron repulsion integrals needed for the correlation treatment and and the description of excitation processes. At present the following functionalities are implemented:

**ground state energies** for MP2 and CC2; 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∞), ADC(2), and CC2

**transition moments** for ground state—excited state transition and the models CCS and CC2

**first-order properties** for the ground state (SCF/CCS, MP2, and CC2) and excited states (CCS and CC2)

**geometric gradients** for the electronic ground state at the MP2 and the CC2 level; for electronically excited states at the CC2 level

**gradients for auxiliary basis sets** for RI-MP2, -CC2, etc. calculations based on the RI-MP2 error functional.

All functionalities are implemented for closed-shell RHF and open-shell UHF reference wavefunctions.
Prerequisites

Calculations with 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 $\text{denconv} 1.d-7$ or less
2. an auxiliary basis defined in the data group $\text{cbas}$
3. if orbitals should be excluded from the correlation treatment (and excitation processes) 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 75–85% of the available (physical) core memory.
5. depending on the type of calculations that should be carried out in addition the data groups $\text{ricc2}$, $\text{excitations}$, and $\text{response}$ have to be set (see below and Section 12.2.12).

For calculations with the RICC2 program it is recommended to use the cc2 submenu of the DEFINE program to set the data groups $\text{denconv}$, $\text{freeze}$, $\text{cbas}$, and $\text{maxcor}$.

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 abelian point groups ($C_1$, $C_s$, $C_2$, $C_i$, $C_{2h}$, $C_{2v}$, $D_2$, $D_{2h}$).

How To Perform a Calculation

Single point calculations:

Call the ricc2 program after a converged SCF calculation.

Geometry optimizations and molecular dynamics:

Invoke jobex with the -level CC2 option; see Section 3.1 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 on the gradients. Invoke for this NumForce with the -level CC2 option; see Chapter 8 for details about NUMFORCE.
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 interfaces, is specified in the option geo\texttt{opt} (see Section 7.3.1) in datagroup $\texttt{ricc2}$ (see Section 12.2.12). For calculations on excited states this state has in addition to be included in the input for excitation energies in datagroup $\texttt{excitations}$.

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):

Implementation:

- please, include always are reference to the publication reporting the implemementation of the core part of the Ricc2 program:
- for transition moments and excited state first order properties:
- for triplet excited states include:
- for geometry optimizations include:
• for geometry optimizations for excited states include:
• for calculations with RI-ADC(2), RI-CIS(D), RI-CIS(D$_\infty$) include:
• if the parallel version of Ricc2 is used include a reference to:
  C. Hättig, A. Hellweg, A. Köhn, to be published.

**Appropriate 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:
• 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 gradients implemented in the Ricc2 program for the optimization of
  your own auxiliary basis set.

### 7.1 CC2 Ground-State Energy Calculations

The CC2 ground-state energy is—similar as other coupled-cluster energies—obtained from the expression

\[
E_{CC} = \langle \text{HF}|H|\text{CC}\rangle = \langle \text{HF}|H \exp(T)|\text{HF}\rangle, \\
= E_{SCF} + \sum_{iajb} \left[ t_{ab}^{ia} + t_{a}^{ib} t_{b}^{ia} \right] \left[ 2(ia|jb) - (ja|ib) \right],
\]

(7.1)

(7.2)

where the cluster operator $T$ is expanded as $T = T_1 + T_2$ with

\[
T_1 = \sum_{ai} t_{ai} \tau_{ai} \\
T_2 = \frac{1}{2} \sum_{ai bj} t_{ai bj} \tau_{ai bj}
\]

(7.3)

(7.4)

(for a closed-shell case; in an open-shell case an additional spin summation has to be included). The cluster amplitudes $t_{ai}$ and $t_{ai bj}$ are obtained as solution of the
CC2 cluster equations \cite{66}:
\begin{align}
\Omega_{\mu_1} &= \langle \mu_1 | \hat{H} + [\hat{H}, T_2] | HF \rangle = 0 , \\
\Omega_{\mu_2} &= \langle \mu_2 | \hat{H} + [F, T_2] | HF \rangle = 0 ,
\end{align}
with
\[ \hat{H} = \exp(-T_1) H \exp(T_1). \]

The residual of the cluster equations \( \Omega(t_{ai}, t_{aibj}) \) is the so-called vector function. The recommended reference for the CC2 model is ref. \cite{66}, the implementation with the resolution-of-the-identity approximation, RI-CC2, was first described in ref. \cite{9}.

**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 \( O(O^2 V^2 N_x) \), 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 \eqref{7.5} and \eqref{7.6}, they are about 10–20 times more expensive than MP2 calculations. The disk space requirements are approximately \( O(2V + N)N_x + N_x^2 \) double precision words. The details of the algorithms see ref. \cite{9}, for the error introduced by the RI approximation see refs. \cite{57,67}.

**Required input data:** In addition to the above mentioned prerequisites ground-state energy calculations with the Ricc2 module require only the data group \$ricc2\) (see Section 12.2.12), 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
\begin{verbatim}
$ricc2
  mp2
  cc2
  conv=6
\end{verbatim}
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. For CCS/CIS the double-substitution amplitudes are excluded from the cluster expansion and in this case the single-substitution amplitudes for the ground state become zero and the energy is identical to the SCF energy. For the Methods CIS(D), CIS(D\(_\infty\)) and ADC(2) the ground state is identified with the MP2 ground state.
Fast RI-MP2 calculations with the Ricc2 program: The Ricc2 program
includes as a subset also the functionalities of the RI-MP2 program. Because of
refined batching algorithms, screening and symmetry treatment the Ricc2 program
is usually somewhat faster than the Rimp2 program. This is in particular the cases
in the following situations:

- when the molecular point group is $D_{2h}$ or one of its subgroups and a significant
  number of atoms is positioned on symmetry elements (e.g. planar molecules)
- when because of memory restrictions the Rimp2 program needs many passes
  of the integral evaluation

All what is needed for a RI-MP2 gradient calculation with the Ricc2 program is:

```
$ricc2
g_eoopt model=mp2
```

If you want only the RI-MP2 energy for a single point use as input:

```
$ricc2
  mp2 energy only
```

The supplement `energy only` disables the calculation of intermediates for the re-
siduum or vector function which are not needed to evaluate only the energy. But
note that it will also disable the calculation of the $D_1$ diagnostic (see below).

Diagnostics: Together with the MP2 and/or CC2 ground state energy the pro-
gramm evaluates the $D_1$ diagnostic proposed by Janssen and Nielsen [68], which is
defined as:

$$
D_1 = \sqrt{ \max \left( \lambda_{\text{max}} \left[ \sum_i t_{ai} t_{bi} \right], \lambda_{\text{max}} \left[ \sum_a t_{ai} t_{aj} \right] \right) } \quad (7.7)
$$

where $\lambda_{\text{max}}[M]$ is the largest eigenvalue of a positive definite matrix $M$. 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, respec-
tively, $D_1(\text{MP2}) \leq 0.015$ and $D_1(\text{CC2}) \leq 0.030$ [68][12]. 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!
7.2 Calculation of Excitation Energies

With the RICC2 program excitation energies can at present be calculated with the RI variants of the methods CCS/CIS, CIS(D), CIS(D∞), 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.

\[
A^{CC2}_{\mu\nu} = \frac{d\Omega_\mu}{d\nu} = \begin{pmatrix}
\langle \mu_1 | [\hat{H}, \tau_{\nu_1}] | HF \rangle & \langle \mu_1 | [\hat{H}, \tau_{\nu_2}] | HF \rangle \\
\langle \mu_2 | [\hat{H}, \tau_{\nu_1}] | HF \rangle & \langle \mu_2 | [F, \tau_{\nu_2}] | HF \rangle
\end{pmatrix}
\]  

(7.8)

Since the CC2 Jacobian is a non-symmetric matrix, left and right eigenvectors are different and the right (left) eigenvectors \( E_\nu \) (\( \bar{E}_\nu \)) are not orthogonal among themselves, but form a biorthonormal basis (if properly normalized):

\[
E^i E^j = \bar{E}_\mu^i \bar{E}_\nu^j + E_\mu^i \bar{E}_\nu^j = \delta_{ij} .
\]  

(7.9)

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. [69,70] or other reviews.

The 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:

\[
A^{eff}_{\mu_1 \nu_1}(t, \omega) = A^{CC2}_{\mu_1 \nu_1}(t) - A^{CC2}_{\mu_1 \gamma_1}(t)(A_{\gamma_2 \gamma_2} - \omega)A^{CC2}_{\gamma_2 \nu_1}(t)
\]

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. [9,10], about the RI error see ref. [67].

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 $O(V + N)N_e$ 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,

$$A_{\mu\nu}^{\text{CCS}} = \frac{d\Omega_\mu}{dt_\nu} = \langle \mu_1 | [H, \tau_\nu] | \text{HF} \rangle,$$

(7.10)
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 $O(O^2 N_e)$ per iteration and transformed trial vector.

The second-order perturbative correction CIS(D) to the CIS excitation energies is calculated from the expression

$$\omega^{\text{CIS(D)}} = \omega^{\text{CIS}} + \omega^{(D)} = E^{\text{CIS}} A^{\text{eff}}(t^{\text{MP1}}, \omega^{\text{CIS}}) E^{\text{CIS}}$$

(7.11)

(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 12.2.12). With the following input the 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 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.

```
$ricc2
  cis
  cis(d)
  cc2
$excitations
  irrep=a1 nexc=2
  irrep=b1 nexc=2
```

*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.
7.2. **CALCULATION OF EXCITATION ENERGIES**

The single-substitution parts of the right eigenvectors are stored in files named \textit{CCRE0-s-m-xxx}, where \textit{s} is the number of the symmetry class (irreducible representation), \textit{m} is the multiplicity, and \textit{xxx} the number of the excitation within the symmetry class. For the left eigenvectors the single-substitution parts are stored in files named \textit{CCLE0-s-m-xxx}. These files can be kept for later restarts.

**Trouble shooting:** For the iterative second-order methods \textit{CIS(D_{\infty})}, \textit{ADC(2)}, and \textit{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 $< \texttt{preopt}$
2. pre-optimization of the eigenvectors by a robust modified Davidson algorithm (see ref. [9]) using the \texttt{LINEAR CC RESPONSE SOLVER} until the norm of all residuals are below \texttt{preopt}, combined with a DIIS extrapolation for roots assumed to be converged below the threshold \texttt{thrdiis}.
3. solution of the nonlinear eigenvalue problem with a DIIS algorithm using the \texttt{DIIS CC RESPONSE SOLVER} until the norm of the residuals are below the required threshold \texttt{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 in the neighbourhood of 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 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 than \texttt{thrdiiis}.

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\(_\infty\)) 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 RIcc2 program is not well suited to deal with such situations.

Large contributions from double excitations can not be monitored in the output of the (quasy-) linear solver. But it is possible to do in advance a CIS(D) calculation. The CIS(D) results for the \(||T_2||\) diagnostic correlate usually well with the CC2 results for this diagnostic. Else the DIIS solver will print the \(||T_2||\) 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 double excited states will anyway be poor. It is strongly recommended to use in such situations a higher-level method.

### 7.3 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.

#### 7.3.1 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. \textit{Orbital-unrelaxed} CC2 properties are calculated as first derivatives of the real part of the unrelaxed
7.3. FIRST-ORDER PROPERTIES AND GRADIENTS

Lagrangian \[66\]

\[ L^{\text{ur CC2}}(t, \bar{t}, \beta) = \langle HF|H|CC \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{H} + [\hat{H}, T_2]|HF \rangle \]

\[ + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | \hat{H} + [F_0 + \beta \hat{V}, T_2]|HF \rangle \]

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:

\[ \langle V \rangle^{\text{ur CC2}} = \Re \left( \frac{\partial L^{\text{ur CC2}}(t, \bar{t}, \beta)}{\partial \beta} \right)_0 = \sum_{pq} D^{\text{ur}}_{pq} V_{pq} , \]

\[ = \Re \left( \langle HF|\hat{V}|HF \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{V} + [V, T_2]|HF \rangle \right) \]

\[ + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | [\hat{V}, T_2]|HF \rangle \],

where \( \Re \) indicates that the real part is taken. Relaxed CC2 properties (and gradients) are calculated from the full variational density including the contributions from the orbital response to the external perturbation, which are derived from the Lagrangian \[70,12\]

\[ L^{\text{rel CC2}}(t, \bar{t}) = \langle HF|H|CC \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{H} + [\hat{H}, T_2]|HF \rangle \]

\[ + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | \hat{H} + [F, T_2]|HF \rangle + \sum_{\mu_0} \bar{\kappa}_{\mu_0} F_{\mu_0} , \]

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:

\[ \langle V \rangle^{\text{rel CC2}} = \Re \left( \langle HF|\hat{V}|HF \rangle + \sum_{\mu_1} \bar{t}_{\mu_1} \langle \mu_1 | \hat{V} + [V, T_2]|HF \rangle \right) \]

\[ + \sum_{\mu_2} \bar{t}_{\mu_2} \langle \mu_2 | [\hat{V}, T_2]|HF \rangle + \sum_{\mu_0} \bar{\kappa}_{\mu_0} V_{\mu_0} \],

\[ = \sum_{pq} D^{\text{rel}}_{pq} V_{pq} . \]

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.
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. [12]. The following is in example input for a MP2 and CC2 single point calculation of first-order properties and gradients:

```plaintext
$ricc2
    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 $ricc2 by the keyword geoopt:

```plaintext
$ricc2
    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 then in CC response theory and are not implemented). For MP2, only the CPHF-like Z-vector equations for $\kappa_{\mu 0}$ need to be solved, no equations have to be solved for the Lagrangian multipliers $\tilde{t}_\mu$. CPU time and disk space requirements are does somewhat smaller than for CC2 properties or gradients.

For SCF/CIS/CCS it is recommended to use the modules Grad and Moloch for the calculation of, respectively, ground state gradients and first-order properties.
7.3. FIRST-ORDER PROPERTIES AND GRADIENTS

7.3.2 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.

The unrelaxed first-order properties are calculated from the variational excited states Lagrangian \[71\], which for the calculation of unrelaxed properties is decomposed into a ground state contribution and Lagrange functional for the excitation energy which leads to expressions for difference densities (or changes of the density matrix upon excitations):

\[
L_{ur,ex}^{\bar{N}, E, \bar{t}, t, \beta} = L_{ur,gs}^{\bar{t}, t, \beta} + L_{ur,diff}^{\bar{N}, \bar{E}, E, \bar{t}, t, \beta}
\]

(7.18)

\[
L_{ur,diff}^{\bar{N}, E, \bar{t}, t, \beta} = \sum_{\mu\nu} \bar{E}_\mu A_{\mu\nu}(t, \beta) E_\nu + \sum_{\mu^2} \bar{N}_{\mu^2}\langle \mu^2 | \hat{H} + [F_0 + \beta \hat{V}, T_2]|HF \rangle ,
\]

(7.19)

\[
(V)^{ur,ex} = \Re \left( \frac{\partial L_{ur,ex}(\bar{E}, E, \bar{t}, t, \beta)}{\partial \beta} \right)\bigg|_0
\]

(7.20)

\[
= \sum_{pq} D_{pq}^{ur,ex} V_{pq} = \sum_{pq} \left( D_{pq}^{ur} + \Delta D_{pq}^{ur,ex} \right) V_{pq} ,
\]

(7.21)

with \( H = H_0 + \beta \hat{V} \) and \( \Re \) indicating that only the real part is taken, \( D_{pq}^{ur} \) is the unrelaxed ground-state density and \( \Delta D_{pq}^{ur,ex} \) the difference density matrix. The unrelaxed excited-state properties obtained thereby are equivalent to those identified from the second residues of the quadratic response function and 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. For a detailed description of the theory see refs. \[71\][70]; the algorithms for the RI-CC2 implementation are described in refs. \[67\][11]. ref. \[67\] also contains a discussion of the basis set effects and the errors introduced by the RI approximation.

In the present implementation, the ground-state and the difference density matrices are evaluated separately. The calculation of excited-state first-order properties thus requires also the calculation of the ground-state density matrix. In addition, the left (\( \bar{E}_\mu \)) and right (\( E_\mu \)) eigenvectors and the Lagrangian multipliers \( \bar{N}_\mu \) need to be determined for each excited state. The disk space and CPU requirements for solving the equations for \( \bar{E}_\mu \) and \( \bar{N}_\mu \) 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 \( O(n_{root}N^2) \) size are written, where \( n_{root} \) is the number of excited states.
The single-substitution parts of the excited-states Lagrangian multipliers $\bar{N}_\mu$ are saved in files named CCNE0-s--m-xxx.

For the calculation of first-order properties for excited states, the keyword `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
$excitations
  irrep=a1 nexc=2
  exprop states=all operators=diplen,qudlen
```

Because for calculation of excited-states first-order properties also the (unrelaxed) ground-state density is evaluated, it is recommended to specify also ground-state first-order properties in the input, since they are obtained without extra costs.

To obtain orbital-relaxed first-order properties or analytic derivatives (gradients) the Lagrange functional for the excited state in Eq. (7.18) is—analogously to the treatment of ground states—augmented by the equations for the SCF orbitals and external perturbations are (formally) included in the SCF step, i.e. also in the Fock operator. Since relaxed densities or often computed in connection with geometry optimizations for individual states (rather than simultaneously for many states) a Lagrangian for the total energy of the excited state is used. This has the advantage the only one equation for Lagrangian multipliers for cluster amplitudes ($\bar{N}$) needs to be evaluated instead of two (one for the ground state and one for the energy difference):

$$L^{rel,ex}(\bar{E}, E, t) = \langle HF | H | CC \rangle + \sum_{\mu\nu} \bar{E}_\mu A_{\mu\nu}(t) E_\nu$$

$$+ \sum_{\mu_2} \bar{N}_{\mu_2} (\mu_2 | \hat{H} + [F, T_2] | HF) + \sum_{\mu_0} \bar{\kappa}_{\mu_0} F_{\mu_0} .$$

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 (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. [72]. There also some information about the performance of CC2 for structures and vibrational frequencies of excited states can found.

The following is an example for the CC2 single point calculation for an an excited
state gradient (not that in the present implementation it is not possible to compute gradients for several excited states at the same time):

\[
\text{\$ricc2}
\begin{align*}
\text{cc2} \\
\text{\$excitations} \\
\quad \text{irrep}=\text{a1} \ \text{nexc}=2 \\
\quad \text{exprop states=all} \ \text{operators=diplen,qudlen} \\
\quad \text{xgrad states=(a1 2)}
\end{align*}
\]

A different input is again required for geometry optimizations: in this case the model and excited state for which the geometry should be optimized have to be specified in the data group \$ricc2 with the keyword \texttt{geoopt}:

\[
\text{\$ricc2} \\
\begin{align*}
\text{geoopt model}=\text{cc2} \ \text{state=(a1 2)} \\
\text{\$excitations} \\
\quad \text{irrep}=\text{a1} \ \text{nexc}=2 \\
\quad \text{exprop states=all} \ \text{operators=diplen,qudlen}
\end{align*}
\]

### 7.3.3 Visualization of densities

By default \texttt{Ricc2} saves \textit{relaxed} densities generated during a calculation in files named \texttt{cc1td-<type>-<mult><irrep>-<number>}, where \texttt{cc1td} stands for “coupled-cluster one-electron total density”. \texttt{<type>} is one of \texttt{mp2-gs} (MP2 ground state), \texttt{cc2-gs} (CC2 ground state), \texttt{ccs-xs} (CCS excited state), \texttt{cc2-xs} (CC2 excited state), or \texttt{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

\[
\begin{align*}
\text{cc1td-cc2-gs-1a1-001} \\
\text{cc1td-cc2-xs-3a2-001}
\end{align*}
\]

In case of open shell molecules, additional files with names \texttt{cc1sd...} (for one-electron spin-densities) will be generated.

These files are (currently) in a binary format, similar as the files \texttt{dens, mdens} and \texttt{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 [10.2]). This can be done by calling \texttt{Ricc2} with the option \texttt{--fanal} which bypasses the usual wavefunction calculation and triggers the program
into an analysis mode for densities. In this mode the program interpretes \$anadens and the keywords described in Section 10.2. 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

\$anadens
calc my_favourite_diffden from
1d0 cc1td-cc2-xs-3a2-001
-1d0 cc1td-cc2-gs-1a1-001
$pointval
and invoke

> ricc2 -fanal

This will generate the files my_favourite_diffden and my_favourite_diffden.map. The latter can be converted into gOpenMol format as described in Section 10.2.

7.4 Transition Moments

Transition moments are presently only implemented for excitations out of the ground state and only for the coupled cluster models CCS and CC2. Note, that for transition moments (as excited-state first-order properties) CCS is not equivalent to CCS and CIS transition moments are not implemented in the Ricc2 program.

In response theory, transition strengths (and moments) are identified from the first residues of the response functions. Due to the non-variational structure of the coupled cluster models different expressions are obtained for “left” and “right” transition moments $M^{V\rightarrow 0}$ and $M^{0\rightarrow V}$ and the transition strengths $S^{0\rightarrow f}_{V_{1}V_{2}}$ are obtained as a symmetrized combinations of both:

$$S^{0\rightarrow f}_{V_{1}V_{2}} = \frac{1}{2} \left\{ M^{V_{1}\rightarrow f}_{0\rightarrow 0} \right. + \left. \left( M^{V_{2}\rightarrow f}_{0\rightarrow 0} M^{V_{1}\rightarrow f}_{0\rightarrow 0} \right)^{*} \right\}$$

(7.23)

Note, that only the transition strengths $S^{0\rightarrow f}_{V_{1}V_{2}}$ are a well-defined observables but not the transition moments $M^{V\rightarrow 0}_{0\rightarrow f}$ and $M^{0\rightarrow V}_{f\rightarrow 0}$. For a review of the theory see refs. [70,73].

The transition strengths calculated by coupled-cluster response theory according to Eq. (7.23) have the same symmetry with respect to 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. [67,12]. 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 CCME0-$s$--$m$--xxx.

To obtain the transition strengths for excitations out of the ground state the keyword spectrum must be added with appropriate options (see Section 12.2.12) to the data group $\$excitations$; else the input is same as for the calculation of excitation energies and first-order properties:

```
$ricc2
  cc2
$excitations
     irrep=a1 nexc=2
     spectrum states=all operators=diplen,qudlen
```

### 7.5 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.

While in general the parallel execution of Ricc2 works similar to that other parallelized Turbomole modules (as e.g. Dscf and Grad), there are some important difference 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 writeable) 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 12.2.12) which should point to a directory in a file system with a good performance. The parallel version of the Ricc2 program can presently account for the following two situations:

**Clusters with single processor nodes and local disks:** Specify in $\$tmpdir$ a directory in the file system on the local disk. All large files will be places on the nodes in these file systems. (The local file system must have the same name on all nodes)
Clusters with multiple (e.g. dual) processor nodes and local disks Set in addition to $tmpdir$ the keyword $sharedtmpdir$ to indicate that several processes might share the same local disk. The program will then create in $tmpdir$ subdirectories with node-specific names.

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
```
Chapter 8

Calculation of Vibrational Frequencies and Infrared 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 RDFT-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, that have an (analytical) gradient program, i.e. the main use of this script is the prediction of vibrational spectra on the MP2 level as well as 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.).

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. To select the correct option(s), use the explanations you get by calling NumForce -h.

For a review of theory and implementation see refs. [74,75].

8.1 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 2 for further details. If the molecule’s 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 $drvopts$ 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).

Note that for large molecules or complicated topologies the B-matrix (that is used to transform from Cartesian coordinates into internal coordinates and vice versa) may become singular. In this case only the normal modes in the internal coordinate basis can be listed.
Chapter 9

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 \cite{76} similar to DSCF.

- **DFT**: using either non-hybrid functionals where no iterations are needed \cite{77} or hybrid functionals where the same algorithm as at the HF-SCF level is used.

- **MP2**: semi-direct method, see ref. \cite{16}.

### 9.1 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

3. it is not possible to run the program in the fully direct mode when doing an SCF, MP2 or a DFT (using hybrid functionals) run, so you will have to perform a statistics run of DSCF before calling Mpshift, or just set the size of the $twoint$ file to a non-zero value

4. to perform an MP2 calculation of the NMR shieldings you have to prepare the input with mp2prep -c
9.2 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`.

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.

9.3 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=dtodb#1 unit=76 size=0 file=dtodb#1
  type=dtodb#2 unit=77 size=0 file=dtodb#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
CHAPTER 9. SHIELDINGS

mp2prep.

9.4 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 [12.2.18]

9.5 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
- molecular point groups that contain reducible e representations are not supported ($C_n$, $C_{nh}$ 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 10

Molecular Properties, Wavefunction Analysis, and Interfaces to Visualization Tools

10.1 Wavefunction analysis and Molecular Properties

Molecular properties (electrostatic moments, relativistic corrections, population analyses for densities and MOs, construction of localized MOs, etc.) can be calculated with the module Moloch. Note that this program does not support unrestricted open-shell input (a script called moloch2 can currently be used as a work-around; type moloch2 -help for further information). Moreover, analyses of densities apart from those calculated from molecular orbitals (e.g. MP2 densities, densities of excited states) are not possible. For the current version of moloch we refer to the keywords listed in Section 12.2.15 which partly can also be set by DEFINE (see also Chapter 2).

Note: Moloch is no longer supported, but most functionalities of Moloch now are integrated in programs that generate MOs or densities and can be done directly within the modules DSCF, RIFDFT, RIMP2, MPGRAD, RICC2 and EGRAD. If (some of) following keywords are set, corresponding operations will be performed in the end of these programs. If one desires to skip the MO- or density generating step, in case of programs DSCF, RIFDFT, RIMP2 and MPGRAD it is possible to directly jump to the routine performing analyses by typing "<program> -proper". Currently, the respective keywords have to be inserted in the control file by hand (not by DEFINE).

Here we briefly present the functionalities (i.e. the default use of keywords), non-default suboptions are described in detail in Section 12.2.16.
Electrostatic moments: up to quadrupole moments are calculated by default for the above modules.

Relativistic corrections: $\texttt{mvd}$ 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 $\langle p^2 \rangle$, $\langle p^4 \rangle$ and the Darwin term $\left( \sum 1/Z_A \ast \rho(R_A) \right)$. 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 when using of ECPs these quantities are not too reasonable (a respective warning is written to the output).

Population analyses: $\texttt{pop}$ enforces Mulliken population analyses 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 DSCF or RIdFT, SCF+MP2 densities in RIMP2 or MPGRAD, excited state densities in EGRAD. Suboptions (see Section 12.2.16) also allow for calculation of Mulliken contributions of selectable atoms to selectable MOs including provision of data for graphical output (simulated density of states).

Generation of localized MOs: $\texttt{localize}$ enables 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 lmos or 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.

Fit of charges due to the electrostatic potential: $\texttt{esp\_fit}$ fits point charges at the positions of nuclei to electrostatic potential arising from electric charge distribution (for UHF cases also for spin density). 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.

10.2 Interfaces to Visualization Tools

Visualization of Molecular Geometry

The tool $\texttt{T2X}$ can be used to convert the atomic coordinates stored in the $\texttt{grad}$ and $\texttt{coord}$ data groups into the xyz-format, which is supported by most viewers, e.g.
10.2. INTERFACES TO VISUALIZATION TOOLS

jmol (http://jmol.sourceforge.net/). Typing

t2x > opt.xyz

in a directory containing the control file generates a series of frames using the information of $grad$. Note t2x writes to standard output which here is redirected to a file. If you are only interested in the most recent structure, type

t2x -c > str.xyz

which only extracts the information on $coord$.

Visualization of Densities, MOs, Electrostatic Potentials and Fields

There are several possibilities to visualize molecular orbitals or densities. tm2molden simply converts MO and geometry information to molden format. The conversion program is interactive and self-explanatory. The generated file can be visualized using either molden (http://www.cmbi.ru.nl/molden/molden.html) or molekel (http://www.cscs.ch/molekel/). For larger systems this may become very time-consuming, as plotting data (values on grids) are calculated by the respective programs (molden, molekel). It is more efficient to calculate the data for plots (MO amplitudes, densities, etc.) by TURBOMOLE modules 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 http://www.csc.fi/gopenmol/) is driven by the keyword $pointval$. This keyword is evaluated by all density matrix generating TURBOMOLE modules, i.e. by DSCF, RIFDT, RIMP2 MPGRAD, RICC2 (see Section 7.3.3) and EGRAD. Note, that all of the following quantities may be calculated simultaneously, and that for programs DSCF, RIFDT, RIMP2 and MPGRAD the density matrix generating steps may be skipped by typing "<program> -proper".

Electron densities For the above mentioned programs setting of keyword $pointval$ dens

or simply $pointval$

yields calculation of densities

$$\rho(\vec{R}_P) = \sum_{\nu\mu} D_{\nu\mu}(\vec{R}_P)\phi_{\nu}(\vec{R}_P)\phi_{\mu}(\vec{R}_P) \quad (10.1)$$
on an orthogonal grid \((R_P)\), the size of which is automatically adjusted to the size of the molecule and the resolution is adjusted to yield acceptable gOpenMol plots (for specification of non-default grid types (planes, lines) and non-default output formats see Section \ref{12.2.16}).

Names of output files are:

- \texttt{td.plt} total density (UHF: \(\alpha\) density plus \(\beta\) density)
- \texttt{sd.plt} spin density (\(\alpha\) density minus \(\beta\) density)
- \texttt{mp2d.plt} MP2 density
- \texttt{mp2sd.plt} MP2 spin density
- \texttt{ed.plt} differential density for excited state
- \texttt{esd.plt} differential spin density for excited state
- \texttt{<myname>.plt} general density passed e.g. by the Ricc2 program.

The .plt files may directly be visualized by gOpenMol; the file \texttt{coord.xyz}, which is also necessary for gOpenMol, is generated by the above programs, if \texttt{$pointval$} is set in the control-file.

**Electrostatic potentials** In an analogous way electrostatic potentials can be calculated on grids.

\texttt{$pointval pot$}

leads to calculation of the electrostatic potential of electrons and nuclei (and external constant electric fields and point charges \(Q\) if present).

\[
V(R_P) = - \int \frac{\rho(\vec{r})}{r_{PR}} d^3r + \sum_A \frac{Z_A}{R_{PA}} \left( \vec{R}_P \vec{E} + \sum_Q \frac{Q}{R_{PQ}} \right)
\]  \hspace{1cm} (10.2)

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 \texttt{tp.plt}, \texttt{sp.plt}, etc.

**Electric fields** (as derivatives of potentials) are calculated by

\texttt{$pointval fld$}

The absolute values of electric fields are written to files \texttt{tf.plt}, \texttt{sf.plt}, etc. For non-default grid types and outputs that allow also for displaying of components of electric fields see Section \ref{12.2.16}.
10.2. INTERFACES TO VISUALIZATION TOOLS

Molecular orbitals. Visualization of molecular orbitals, i.e. generation of .plt-files containing amplitudes of MOs $i$,

$$A_i(\vec{R}_P) = \sum_\nu c_{i\nu} \phi_\nu(\vec{R}_P)$$

is achieved e.g. by

$\text{pointval mo 10-12,15}$

This yields amplitudes for MOs 10-12 and 15 on the default grid. The numbering of MOs refers to that you get from the first column of the output of the tool Eiger. 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. 2alig_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 $t_{2g}$.

If one has generated localized molecular orbitals (LMOs, see above) they can also be visualized.

$\text{pointval lmo 3-6,8}$

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 is due to the output from the localization section. For an UHF case this means: If you included in the localization procedure e.g. 5 $\alpha$-type orbitals and 3 $\beta$-type orbitals, then, if you are interested in plotting the $\beta$-type LMOs only, you have to type

$\text{pointval lmo 6-8}$

Non-default grids are described in detail in Sections 12.2.16. Calculation of the above quantities at single points is needed quite often, thus an example is given here.

$\text{pointval geo=point}$

7 5 3
0 0 7
1 2 3

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. [78]).
Chapter 11

Treatment of Solvation Effects with Cosmo

The Conductor-like Screening Model (CSM) 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 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 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.

$$f(\varepsilon) = \frac{\varepsilon - 1}{\varepsilon + \frac{1}{2}}$$ 

$$q^* = f(\varepsilon)q$$
The deviations of this COSMO approximation from the exact solution are rather small. For strong dielectrics like water they are less than 1%, while for non-polar solvents with $\varepsilon \approx 2$ they 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.

$$E_{\text{diesel}} = \frac{1}{2} f(\varepsilon) \mathbf{q}^\dagger \Phi^\text{sol}$$

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.

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, which then also allows the evaluation of analytic gradients.

**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.

**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 $A_{kk} \approx 3.8\sqrt{S_i}$, if no associated basis grid points exist.

**Outlying charge correction:** Because the electron density is not zero outside the cavity one makes a mistake, which should be corrected 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, which is constructed by an outward projection of the spherical part of the surface onto the radius $R_i + ROUTF \times RSOLV$, for the estimation of the energy and charge correction $[80]$. 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 $[81]$ the dielectric energy for the disturbed state can be written as

$$E_{d\text{dier}} = \frac{1}{2} f(\epsilon) q(P^0) \Phi(P^0) + \frac{1}{2} f(n^2) q(P^\Delta) \Phi(P^\Delta) + f(\epsilon) q(P^0) \Phi(P^\Delta),$$

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 $[82]$. 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 $[83][84]$. 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.
Chapter 12

Keywords in the control file

12.1 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.

12.2 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 13. An alphabetical list of all keywords is given in the index.

12.2.1 General Keywords

$operating system unix
$path
$lock off
$suspend off

The four keywords above are set by Define, but are not necessary.
$\text{statistics dscf}$

or

$\text{statistics mpgrad}$

Only a statistics run will be performed to determine file space requirements as specified for DSCF or MPGRAD. On return the statistics option will be changed to $\text{statistics off}$.

$\text{actual step dscf}$

means \textit{current step}. Keyword and data group (as e.g. dscf) is set by every program and removed on successful completion.

$\text{last step relax}$

Keyword and data group (as e.g. relax) set by every program on successful completion.

General file cross-references:

$\text{coord} \quad \text{file=coord}$

$\text{intdef} \quad \text{file=coord}$

$\text{user-defined bonds} \quad \text{file=coord}$

$\text{basis} \quad \text{file=basis}$

$\text{ecp} \quad \text{file=basis}$

$\text{scfmo} \quad \text{file=auxbasis}$

$\text{uhfmo_alpha} \quad \text{file=alpha}$

$\text{uhfmo_beta} \quad \text{file=beta}$

$\text{natural orbitals} \quad \text{file=natural}$

$\text{natural orbital occupation} \quad \text{file=natural}$

$\text{energy} \quad \text{file=energy}$

$\text{grad} \quad \text{file=gradient}$

$\text{forceapprox} \quad \text{file=forceapprox}$

It is convenient not to include all input in the \textit{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.

$\text{coord} \quad \text{(and \texttt{intdef} \text{and} \texttt{userdefined bonds})}$

contains atom specification—type and location—and the bonds and internal coordinates convenient for geometry optimizations.

$\text{basis}$

specification of basis sets.

$\text{ecp}$ specification of effective core potentials.
12.2. FORMAT OF KEYWORDS AND COMMENTS

$jbas
auxiliary (fitting) basis for the Coulomb terms in RIDFT.

$scfmo, $uhfmo_alpha, $uhfmo_beta
MO vectors of SCF or DFT calculations for RHF or UHF runs.

$natural orbitals, $natural orbital occupation
keywords and data groups set by unrestricted DSCF or RIDFT runs. Contain natural MO vector and orbital occupation.

$energy, $grad
energies and gradients of all runs, e.g. for documentation in a geometry optimizations.

$forceapprox
approximate force constant for geometry optimizations.

The control file must end with this keyword:
$end

12.2.2 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.

$title
give title of run or project here.

$symmetry d4h
Schönflies symbol of the point group. All point groups are supported with the exception of NMR shielding and force constant calculations etc. which do not work for groups with complex irreps ($C_3$, $C_3h$, $T$, etc). Use a lower symmetry group in this case.

$atoms
Example:
$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

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.

$poole 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).

$newecp char
allows use of ECPs with g-projectors for energy and gradient calculations. Second derivatives are not supported yet. This keyword must be added manually to the control file; it will become default in the next release. The usage is demonstrated in the control file of the example $TURBODIR/TURBOTEST/dscf/long/Au4.SCF.ECPl4.

RHF

$closed shells
Specification of MO occupation for RHF, e.g.

a1g 1-4  (2)
a2g 1    (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:
12.2. FORMAT OF KEYWORDS AND COMMENTS

$roothaan

Roothaan parameters for the open shell, here a triplet case. DEFINE recognises most cases and suggests good Roothaan parameters.

For further information on ROHF calculations, see the sample input in Section \[13.6\] and the tables of Roothaan parameters in Section \[4.3\].

UHF

$suhf$ directs the program to carry out a UHF run, e.g.

$alpha shells
a1g 1-4 (1)
a2g 1 (1)

$beta shells
a1g 1-4 (1)
a2g 1 (1)

The specification of MO occupation for UHF, $suhf$ overwrites closed-shell occupation specification.

12.2.3 Keywords for redundant internal coordinates in $redund.inp$

With the parameters in $redund.inp$ the generation of redundant internal coordinates can be modified. All entries have to be made in the control file before invoking the $ired$ option. Important options are:

\textbf{iprint} \(n\)

print parameter for debug output: The larger \(n\) is, the more output is printed
\(n \geq 0, n \leq 5\) (default: 0)

\textbf{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 \textbf{metric} option:
n = 1  “Delocalized Coordinates”
The $BmB^t$ matrix is diagonalized for the complete set of redundant internal coordinates, matrix $m$ is a unit matrix.

n = -3  Delocalized Coordinates obtained with a modified matrix $m$, the values of $m$ can be defined by user input (see below).

n = -1  “Hybrid Coordinates”
Natural internal coordinates are defined as in the old iaut option. If a cage remains, delocalized coordinates (as for n=1) are defined for the cage.

n = -2  Very similar to the $n = 1$ option, but for the remaining cage delocalized coordinates with modified matrix $m$ are defined as for $n = -3$.

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 $BB^t$ matrix is diagonalized for each block separately after the columns of $B$ were orthogonalized against the columns of $B$ of the the preceding blocks.

n = 3  “Generalized natural coordinates”
Natural internal coordinates are defined first, for the remaining cage decoupled coordinates are defined.

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 $m$ in $BmB^t$.

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)
12.2. FORMAT OF KEYWORDS AND COMMENTS

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>linp</td>
<td>bending of the chain a–b–c perpendicular to the plane of b–c–d (default: 0.2)</td>
</tr>
<tr>
<td>wstr</td>
<td>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)</td>
</tr>
<tr>
<td>winv</td>
<td>inverse stretch of a weak bond (default: 0.05)</td>
</tr>
<tr>
<td>wbnd</td>
<td>bond angle involving at least one weak bond (default: 0.02)</td>
</tr>
<tr>
<td>wout</td>
<td>Out of plane angle for weak bonds (default: 0.02)</td>
</tr>
<tr>
<td>wtor</td>
<td>dihedral angle for weak bonds (default: 0.02)</td>
</tr>
<tr>
<td>wlnp</td>
<td>linc coordinate for weak bonds (default: 0.02)</td>
</tr>
<tr>
<td>wlnp</td>
<td>linp coordinate for weak bonds (default: 0.02)</td>
</tr>
</tbody>
</table>

12.2.4 Keywords for Module Uff

One has to specify only the cartesian coordinates (datagroup $coord$) 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,nqeq
111111 ! 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
25 0.10 0.00 ! mxls,dhls,ahls
1.00 0.00 0.00 ! alpha,beta,gamma
F F F ! transform,lnumhess,lmd
```

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 linesearch step will be done after the Newton atep.

dqmax
max. displacement in a.u. for a coordinate in a relax step.

mxls, dhls, ahls
parameters of linesearch:

ahls start value
dhls increment
mxls number of energy calculations

alpha, beta, gamma
modification parameter for the eigenvalues of the hessian (see below): \( f(x) = x \cdot (\alpha + \beta \cdot \exp(-\gamma \cdot x)) \).
transform
   a switch for the transformation in the principal axis system.

lnumhess
   a switch for the numerical hessian.

lmd   a switch for a MD calculation.

**Input Data Blocks Needed by UFF**

$coord
   cartesian coordinates of the atoms (default: $coord file=coord)

$ufftopology
   contains a list of the next neighbours of each atom (see Section 12.2.4). Sometimes it is useful to enter the connectivity (in the input block nxtnei12 in the file ufftopology) by hand (not always necessary; default: $ufftopology file=ufftopology).

Beyond this UFF reads the force field parameters for the atoms from the file parms.in. If this file exists in the directory from which one starts an UFF calculation the program will use this file, if not the program reads the data from the file $TURBODIR/uff/parms.in. If one wants own atom types, one has to add these atom types in the file parms.in. For each new atomtype one has to specify the natural bond distance, the natural bond angle, the natural non-bond distance, the well depth of the Lennard-Jones potential, the scaling factor $\zeta$, the effective charge, torsional barriers involving 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.

**UFF Output Data Blocks**

$coord
   contains the (updated) cartesian coordinates of the atoms (default: $coord file=coord).

$ufftopology
   contains the full information of the topology of the molecule and the whole force field terms (see below; default: $ufftopology file=ufftopology).

$uffgradient
   contains the accumulated cartesian analytical gradients (default: $uffgradient file=uffgradient).
CHAPTER 12. KEYWORDS IN THE CONTROL FILE

\$uffhessian
contains the cartesian analytical hessian;
(default: \$uffhessian file=uffhessian0-0).

The file \texttt{ufftopology}

The topology file \texttt{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:

\[
J \quad I \quad K \quad \text{wtyp} \quad \theta \quad nr_{JI} \quad nr_{IK}.
\]

Here are $J$, $I$ und $K$ the atoms number, where atom $I$ is in the apex. “wtyp” is the angle type and has the following values:

\[
\begin{align*}
\text{wtyp} & = 1 \quad \text{linear case} \\
\text{wtyp} & = 2 \quad \text{trigonal planar case} \\
\text{wtyp} & = 3 \quad \text{quadratic planar case} \\
\text{wtyp} & = 6 \quad \text{ocathedral case} \\
\text{wtyp} & = 9 \quad \text{all other cases}.
\end{align*}
\]

$\theta$ 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:

\[
I \quad J \quad K \quad L \quad nr_{JK} \quad \text{ttyp} \quad \phi \quad \theta_{IJ} \quad \theta_{JKL}.
\]

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:
12.2. FORMAT OF KEYWORDS AND COMMENTS

\[ ttyp = 1 \quad J \text{(sp}^3 \text{)} - K \text{(sp}^3 \text{)} \]
\[ ttyp = 11 \quad \text{like } ttyp=1, \text{ but one or both atoms are in Group 16} \]
\[ ttyp = 2 \quad J \text{(sp}^2 \text{)} - K \text{(sp}^3 \text{)} \text{ or vice versa} \]
\[ ttyp = 21 \quad \text{like } ttyp=2, \text{ but one or both atoms are in Group 16} \]
\[ ttyp = 22 \quad \text{like } ttyp=2, \text{ but } J \text{ or } K \text{ is next a sp}^2 \text{ atom} \]
\[ ttyp = 3 \quad J \text{(sp}^2 \text{)} - K \text{(sp}^2 \text{)} \]
\[ ttyp = 9 \quad \text{all other cases.} \]

\( \phi \) is the value of the torsion angle in degree. \( \theta_{IJK} \) is the angle value of \((I - J - K)\) and \( \theta_{JKL} \) is the cowne 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 \( \text{NH}_3 \)). 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. \( \text{ityp1}, \text{ityp2}, \text{ityp3} \) are the types of the inversions:

\[ \text{ityp} = 10 \quad \text{atom } I \text{ is C and atom } L \text{ is O} \]
\[ \text{ityp} = 11 \quad \text{like } \text{ityp}=10, \text{ but } L \text{ is any atom} \]
\[ \text{ityp} = 2 \quad I \text{ is P} \]
\[ \text{ityp} = 3 \quad I \text{ is As} \]
\[ \text{ityp} = 4 \quad I \text{ is Sb} \]
\[ \text{ityp} = 5 \quad I \text{ is Bi} \]
\[ \text{ityp} = 9 \quad \text{all other cases.} \]

\( \omega_1, \omega_2 \) and \( \omega_3 \) are the values of the inversion angles in degree.

The nonbond terms follow starting with the number of the nonbonded 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 $\lambda_i$ 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.

### 12.2.5 Keywords for Modules DSCF and RIDFT

**$denconv 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.

**$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>**

Requests calculation of occupation numbers at a finite temperature $T$. For an orbital with the energy $\varepsilon_i$ the occupation number $n_i \in [0, 1]$ is calculated as

$$n_i = \frac{1}{2} \mathrm{erfc} \left( \frac{\varepsilon_i - \mu}{fT} \right),$$

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 temperature given by \texttt{tmstrt} (default: 300 K) is reduced at each SCF cycle by the factor \texttt{tmfac} (default: 1.0) until it reaches the value \texttt{tmend} (default: 300 K). Note that the default values lead to occupation numbers calculated at a constant $T = 300$ K. Current occupation numbers are frozen if the energy change drops below the value given by \texttt{stop} (default: $1 \cdot 10^{-3}$). 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 \texttt{scfdamp} and \texttt{scforbitshift} 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 \texttt{tmend} should be set to a 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.

\texttt{$\$firstorder$}

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 [10]).

\texttt{$\$fldopt options$}

Specification of options related with external electrostatic fields. The following options are available:

\textbf{1st derivative on/off}

Calculate numerical 1st derivative of SCF energy with respect to electrostatic field (default: off), increment for numerical differentiation is \texttt{edelt} (see below).

\textbf{2nd derivative on/off}

Calculate numerical 2nd derivative of SCF energy with respect to electrostatic field (default: off), increment for numerical differentiation is \texttt{edelt}.

\texttt{edelt= real}

Increment for numerical differentiation (default: 0.005).

\textbf{fields on/off}

Calculate SCF energy for non-zero external electrostatic fields defined in \texttt{$\$electrostatic field$}.

\textbf{geofield on/off}

Calculate SCF energy for one external field definition and dump disturbed MOs onto \texttt{$\$scfmo$}. 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!!

$\text{incore \hspace{1em} integer}$

By using this option the two-electron integrals are kept in RAM; \textit{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:

$\text{scfdenapprox} \hspace{1em} 1$

and rhfshells 1 or 2. It is advisable to set $\text{thize}$ as small as possible (e.g. $\text{thize} \hspace{1em} 0.1d-08$) and to remove the keyword $\text{scfdump}$.

**Note:** this keyword does not work for parallel runs.

$\text{intsdebug \hspace{1em} cao}$

Output of one-electron matrices expressed in cao basis. This works only properly if the molecule is in $C_1$ symmetry.

Note that the output gives one triangle of the one-electron matrices. Thus the entries are:

$$
(11) \hspace{1em} (21) \hspace{1em} (22) \\
(31) \hspace{1em} (32) \hspace{1em} (33) \\
(41) \hspace{1em} (42) \hspace{1em} (43) \\
\ldots
$$

The order of the basis functions is such that all s-orbitals are given first, then all \textit{p}-orbitals, all \textit{d}-orbitals and so on. So we have:

1. atom 1s,2s,3s\ldots
2. atom 1s,2s,3s\ldots
\ldots
1. atom 1px,1py,1pz,2px,2py,2pz
2. atom 1px,1py,1pz,2px,2py,2pz
\ldots

$\text{mo-diagram \hspace{1em} only nirreps=integer}$

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 \textit{only} has been set only the start MOs are dumped and the program quits. \textit{nirreps} will hold the total number of displayed orbitals after the successful run.
12.2. **FORMAT OF KEYWORDS AND COMMENTS**

$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.

$mo output format format

If this line is present, the dscf program is forced to output the MOs using the new FORTRAN format format regardless of the format-option in data group $scfmo. Otherwise the input format will be used.

Example: $mo output format(3(2x,d15.8))

$natural orbitals

This data group will be written after an UHF calculation (together with $natural orbital occupation) and contains the natural space orbitals (same syntax as $scfmo).

$natural orbital occupation

This data group will be written after an UHF calculation (together with $natural orbitals) and contains the occupation of natural orbitals (syntax as any data group related with orbital occupation information, e.g. $closed shells), e.g.

```
a 1-5 ( 2.00000000000000 )
a 6 ( 1.99949836999366 )
a 7 ( 1.99687490286069 )
a 8 ( 1.00000000000000 )
a 9 ( .00312509713931 )
a 10 ( .00050163000634 )
```

$point_charges

Specification of location and size 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
  2. 2. 2. 5.
  5. 0. 0. 2.5
```

$prediag

concerns the first SCF iteration cycle if start MOs from an EHT guess are used.
CHAPTER 12. KEYWORDS IN THE CONTROL FILE

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 twoint \ is appended, the program will read the two-electron integrals from the files specified in \$scfintunit, so there will be almost no loss of cpu-time.

All this information is normally provided by the previous Dscf run if the keyword \$scfdump \ (see there) was given.

\$restartd \ data

Data provided by a previous dscf run that has been interrupted. This keyword is created when \$scfdump \ was given.

\$rundimensions \ data

is set by 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:

\[
\text{dim(fock,dens)=6072} \\
\text{natoms=6} \\
\text{nshell=34} \\
\text{nbf(CAO)=108} \\
\text{nbf(AO)=98} \\
\text{dim(trafo[SAO<-->AO/CAO])=256} \\
\text{rhfshells=1}
\]

\$scfconv \ integer

SCF convergency criterion will be \( 10^{-\text{integer}} \) for the energy. Gradients will only be evaluated if \text{integer} > 6.

\$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 \text{start}; if convergence is good, this weight is then reduced by the \text{step} 0.05 in each successive iteration until the \text{minimum} of 0.1 is reached. (These are the default
settings of 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.

$\texttt{scfdebug} \ \texttt{options}$

Flags for debugging purposes. Following options are available:

**vectors integer**

Output level concerning molecular orbitals. integer=0 (default) means minimal output, >1 will output all start MOs and all MOs in each iteration.

**density integer**

Output level concerning difference density matrices.

**debug integer**

integer > 0 will dump a lot of information—be careful!

$\texttt{scfdenapproxl} \ \texttt{integer}$

This keyword can be used to reduce the time needed to update the fock matrix in each SCF iteration by exploiting information on previously computed densities. The differential density will be minimized using a linear combination of up to integer previous density matrices. If this keyword is absent the default value is 20.

$\texttt{scfdiis} \ \texttt{options}$

Control block for convergence acceleration via Pulay’s DIIS. Options are:

**errvec=char**

specifies the kind of error vector to be used (two different kind of DIIS algorithms)

- **char=’FDS’ or ’SDF’ or ’FDS-SDF’**
  
  use the commutator $[FDS, SDF]$ as error vector.

- **char= none**
  
  no DIIS

- **char= sFDS**
  
  use $S^{-1/2} FDS^{1/2} -$ transposed

- **char=dF**
  
  not supported anymore

Further suboptions:

**maxiter=integer**

maximum number of iterations used for extrapolation.

debug=integer
   debug level (default: 0)
   
   integer=1   print applied DIIS coefficients
   integer=2   print DIIS matrix and eigenvalues, too

qscal=real
   scaling factor in DIIS procedure: \( q_{scal} > 1 \) implies some damping,
   \( q_{scal} = 1.0 \): straight DIIS.

thrd=real
   directs the reduction of \( q_{scal} \) to \( q_{scal} = 1.0 \) (no damping in DIIS),
   done if \( ||errvec|| < thrd \).

Defaults for $prediag$ (see above) and $scfdiis$

errvec=FDS-SDF, maxiter=5, qscal=1.2, thrd=0.0, this implies DIIS damping
in all iterations, prediag is switched off.

Recommended:

errvec=sFDs leads to the following defaults:
   qscal=1.2, for SCF runs: maxiter=6 and thrd=0.3, prediag is off; for DFT
   runs: maxiter=5 and thrd=0.1 prediag is on. If you want to switch off prediag
   put $prediag$ none.

$scfdump
   Dump SCF restart information onto data group $restartd$ and dump SCF
   MOs in each iteration onto $scfmo$ (sefdump = iter). Additionally, a data
   block $scfiterinfo$ 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
   datagroup $restartd$ (see also $restart$).

$scfintunit options
   Disc space specification for two-electron integrals. The following suboptions
   are available (and necessary):

unit=integer
   Fortran unit number for this file. Unit numbers 30,31,\ldots are recommended.

size=integer
   Filespace in megabytes for this file. size=0 leads to a fully direct run.
   size is set by a statistics run, see $statistics$. DSCF switches to
direct mode if the file space is exhausted.

file=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 $scfintunit may look like this:

\begin{verbatim}
$scfintunit
  unit=30  size=35  file=twoint1
  unit=31  size=35  file=/users/work/twoint2
\end{verbatim}

Maximal 30 files may be specified in this way.

\$scfiterlimit \textit{integer}

Maximum number of SCF iterations (default: 30).

\$scfmo \textit{none file=char}

Input/output data group for SCF MOs. You can specify

\textit{none}

To perform a calculation without a startvector (i.e. use a core hamiltonian guess).

\textit{file=char}

The file where the MOs are written on output (default: mos).

These two options can also be used for $uhfmo_alpha and $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:

\textit{expanded}

These MOs were obtained by projection form another basis set. They should not be used for wavefunction analysis.

\textit{scfconv=integer}

The MOs are converged SCF MOs, the convergence criterion applied was $10^{-integer}$.

\textit{scfdump=integer}

The MOs are unconverged SCF MOs which were written on this data group after iteration \textit{integer}. The latter three options are mutually exclusive.

\textit{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:

- noautomatic
  - Automatic virtual shell shift switched off.
- automatic
  - Automatic virtual shell shift switched on; the energies of virtual orbitals will be shifted if the HOMO-LUMO gap drops below real such that a gap of real is sustained. This is the default setting if the keyword is missing with real=0.1.
- closedshell=real
  - Option for open-shell cases. Closed shells are shifted to lower energies by real. The default shift value is closedshell=0.4.
  - 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:
    a1  1,2,4-6  (-.34)
    b1  8   (+.3)

$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 \( real = \frac{10^{-(scf.conv+1)}}{3\cdot#bf} \) (#bf = number of basisfunctions).

\$scratch\$ files
The scratch files allocated by Dscf 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:

\$scratch\$ files
\begin{verbatim}
dscf   dens   path1/file1
dscf   fock   path2/file2
dscf   dfock  path3/file3
dscf   ddens  path4/file4
dscf   xsv    path5/file5
dscf   pulay  path6/file6
dscf   statistics path7/file7
dscf   errvec path8/file8
dscf   oldfock path9/file9
dscf   oneint path10/file10
\end{verbatim}

The first column specifies the program (Dscf in this case), 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

- off: Do not perform integrals statistics
- dscf: Perform integrals statistics for Dscf
- kora: see KORA
- mpgrad: see MPGRAD
- polly: see POLLY
- dscf parallel: see PARALLEL PROCESSING

Options kora, dscf parallel, grad, mpgrad, polly 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.

\[
\begin{align*}
a_{1g} & : 1-4 \quad (2) \\
a_{2g} & : 1 \\
\end{align*}
\]

$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:

\[
\begin{align*}
b_{2g} & : 1 \quad (1) \\
b_{3g} & : 1 \quad (1) \\
\end{align*}
\]

$rohf
This data group is necessary for ROHF calculations with more than one open shell. Example:

\[
$rohf 1 \\
a -a \quad a=0 \quad b=0 \\
h -h \quad a=1 \quad b=2 \\
a -h \quad a=1 \quad b=2 \\
\]

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 4.3.

$roothaan
For ROHF-calculations with only one open shell the Roothaan parameters $a$ and $b$ have to be specified within this data group (see also $rohf$). Example:

\[
12.2. FORMAT OF KEYWORDS AND COMMENTS

$\text{roothaan}$

\[
\begin{align*}
a &= 3/4 & b &= 3/2
\end{align*}
\]

This example is for the 3P ground state of carbon ($2p^2$) in symmetry group I. 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 13.6 and the tables of Roothaan parameters in Section 4.3.

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**

$\text{alpha shells}$ and $\text{beta shells}$

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. $\text{closed shells}$)

Example:

$\text{alpha shells}$

\[
\begin{align*}
a &\quad 1-8 \quad (1) \\
b &\quad 1-2 \quad (1)
\end{align*}
\]

$\text{beta shells}$

\[
\begin{align*}
a &\quad 1-7 \quad (1) \\
b &\quad 1-3 \quad (1)
\end{align*}
\]

$\text{uhf}$

directs the program to carry out a UHF run. $\text{uhf}$ overwrites closed-shell occupation specification.

$\text{uhfmo}_\alpha$ and $\text{uhfmo}_\beta$

These two data groups contain the UHF MO vectors for alpha and beta spin respectively (same syntax as $\text{scfmo}$).

$\text{uhfmo}_\beta$

see $\text{uhfmo}_\alpha$

**DFT**

$\text{dft}$

functional b-p

gridsize m3
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
      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
\end{verbatim}

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\textunderscore Gaussian and s-vwn\textunderscore Gaussian are made available only for comparibility with Gaussian. The functional VWNIII is much less well founded than VWN5 and the TURBOMOLE team does not recomend the use of VWNIII.

**RI**

DSCF ignores RI flags: you must call the RI modules \texttt{Ridft} and \texttt{Rdgrad} for energy and gradient calculations.

**\texttt{$ridft}$**

Enforces a ridft calculation if module \texttt{Ridft} is used

**\texttt{$ricore \ integer$}**

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)

**\texttt{$jbas file=auxbasis$}**

Cross reference for the file specifying the auxiliary basis as referenced in \texttt{$atoms$}. We strongly recommend using auxbasis sets optimized for the respective MO basis sets, e.g. use SVP (or TZVP) for the basis and the corresponding auxbasis as provided by \texttt{DEFINE} (default: \texttt{file=auxbasis}).
$ripop
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 $ridft$ is found in the control file but not the keyword $dft$, 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 $ricore$):

$jkbas file=auxbasis
Cross reference for the file specifying the JK-auxiliary basis as referenced in $atoms. This group is created by the rijk menu in 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 RIDFT module and requires setting of the $ridft$ keyword.

$marij
precision 1.0D-06
lmaxmom 10
nbimax 8
wsindex 0.0
extmax 20.0
thrmom 1.0D-18

The following options are available:

- **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 $1 \cdot 10^{-8}$–$1 \cdot 10^{-9}$.
- **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.

\textbf{thrmom} \quad \text{Threshold for moment summation. For highly accurate calculations it should be set to } 1 \cdot 10^{-24}.

\textbf{nbinmax} \quad \text{number of bins per atom for partitioning of electron densities. Default value is 8 and hardly ever needs to be changed.}

\textbf{wsindex} \quad \text{minimum separation between bins. Only bins separated more than the sum of their extents plus } \textbf{wsindex} \text{ are considered as far-field. Default is 0.0 and should be changed only by the experts.}

\textbf{extmax} \quad \text{maximum extent for charge distributions of partitioned densities. Extents with values larger then this are set to } \textbf{extmax}. Hardly ever needs to be changed.

\section*{LHF}

Use the Localized Hartree–Fock (LHF) method to obtain an effective Exact-Exchange Kohn–Sham potential (module $\text{Dscf}$). The LHF method is a serial implementation for spin-restricted closed-shell and spin-unrestricted ground states.

\begin{verbatim}
$dft
  functional lhf
  gridsize 6
  test-integ
\end{verbatim}

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.

\textbf{gridsize 4} is the most diffuse with special radial scaling; \textbf{gridsize 5} is for very good Rydberg orbitals; \textbf{gridsize 6} (default in \text{Lhfprep}) is the least diffuse, only for the first Rydberg orbitals.

Only gridsize 3–5 can be used, no modified grid.

\textbf{test-integ} will check if the selected grid is accurate enough for the employed basis-set by performing a numerical integration of the norm of all orbitals.

\section*{How to do LHF runs}

1) Do a Hartree–Fock calculation using $\text{Dscf}$.

2) Use the script \text{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 \text{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)

```
$lhf
    off-diag on
    numerical-slater off
    pot-file save
    asymptotic dynamic=1.d-3
    homo 1b1u
    homob 1b1u # 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
    corrct-region 10.0 0.5
    slater-b-region 7.0 0.5 10.0 0.5 # ONLY UNRESTRICTED
    corrct-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.

    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.

    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 corresponding files are slaterA.pot, slaterB.pot, corctA.pot and corctB.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 expert users
Options for the conjugate-gradient algorithm for the computation of the correction potential: 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
12.2. FORMAT OF KEYWORDS AND COMMENTS

\[ R'_F + \Delta'_F \ldots + \infty : \text{asymptotic Slater} \]

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.

12.2.6 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 [11]. The model is currently implemented for SCF energy and gradient calculations (DSCF/RIDFT and GRAD/RDGRAD) and MP2 energy calculations (RIMP2 and MPGRAD).

Please note: due to improvements in the \( A \) matrix and cavity setup the Cosmo energies and gradients may differ from older versions. The use_old_amat option can be used to calculate energies (not gradients) using the old cavity algorithm.

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
# the following options are not used by default
  allocate_nps= 140
  use_old_amat
```

ee[nion=real
        defines a finite permittivity used for scaling the screening charges.

allocate_nps=integer
        skips the Cosmo segment statistics run and allocates memory for the
given number of segments.

All other parameters affect the generation of the surface and the construction of the \( A \) matrix:
nppa= integer  
    number of basis grid points per atom  
    (allowed values: \( k = 10 \times 3^i \times n^2 + 2 = 12, 32, 42, 92... \))

nspa= integer  
    number of segments per atom  
    (allowed values: \( k = 10 \times 3^i \times n^2 + 2 = 12, 32, 42, 92... \))

disex= real  
    distance threshold for A matrix elements (Ångstrom)

rsolv= real  
    distance to outer solvent sphere for cavity construction (Ångstrom)

routf= real  
    factor for outer cavity construction in the outlying charge correction

cavity closed  
    fill in seams between atoms with points

cavity open  
    leave untidy seams between atoms

ampran= real  
    amplitude of the cavity de-symmetrization

phsran= real  
    phase of the cavity de-symmetrization

use_old_amat  
    uses A matrix setup of TURBOMOLE 5.7

If the \$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, which must be provided in Ångstrom units in the data group \$cosmo_atoms, e.g. for a water molecule:

\$cosmo\_atoms
# radii in Angstrom units
  o 1
     radius= 1.7200
  h 2-3
     radius= 1.3000

\$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:
SCREENING CHARGE:

- cosmo : -0.003925
- correction : 0.003644
- total : -0.000282

ENERGIES [a.u.]:

- Total energy = -76.0296831863
- Total energy + OC corr. = -76.0297567835
- Dielectric energy = -0.0118029468
- Diel. energy + OC corr. = -0.0118765440
- The following value is included for downward compatibility
- Total energy corrected = -76.0297199849

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.

**Cosmo in MP2 Calculations:** The iterative Cosmo PTED scheme (see chapter 11) can be used with the mp2cosmo script. Options are explained in the help message (mp2cosmo -h). Both MP2 modules RIMP2 and MPRGRAD can be utilized. The control file can be prepared by a normal COSMO SCF input followed by a RIMP2 or MPRGRAD input.

**Cosmo in Numerical Frequency Calculations:** NUMFORCE can handle two types of COSMO frequency calculations. The first used the normal relaxed COSMO energy and gradient. It can be performed with a standard DSCF or RIDFT 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 11, 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).

### 12.2.7 Keywords for Modules Grad and Rdgrad

Many of the DSCF and RIDFT keywords are also used by Grad and Rdgrad.

**$drvopt**

This keyword and corresponding options are required in gradient calculations only in special circumstances. Just $drvopt is fine, no options needed to compute deriva-
tives 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. 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 derivative option, add

\[ \text{weight derivatives} \]

in \$dft. 

The option

\[ \text{point charges} \]

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.

12.2.8 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] intcoord [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 8. The effect of the keyword only is the same as described above.

$\text{maxcor} \ 50$
 fixes the RAM memory to be used by the run (here 50 MB), about 70% of available memory should be fine, because $\text{maxcor}$ 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 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.

\$isosub
3 2.001
5 13

The interpolation then takes place between the mass(es) specified in \$atoms (or the default mass(es), if none specified) and the mass(es) in \$isosub. Take care of symmetry equivalent atoms, otherwise symmetry analysis will fail. This feature can not be used in a lowest eigenvalue search (keyword \$les).

\$isopts 6
Sets the number of points for interpolation between the two isotopes compared by the \$isosub option to six. Default value is 21.

Keywords for the treatment of only selected nuclear displacement vectors:

\$ironly
CPHF-iteration is done only for distortions, that are IR active.

\$ramanonly
CPHF-iteration is done only for distortions, that are Raman active.

\$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). \$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.

\$les
a1 3
a2 all
b2 1

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 \$forceconv as explained above.

The maximum number of macro-iterations is specified by \$lesiterlimit x
with the default $x=25$. The maximum number of iterations for each solution of the CPHF-equations is again determined by $\text{$\textdollar$forceiterlimit}$ 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.

$\text{$\textdollar$hesscond 8.0d-5}$

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.

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 $\text{$\textdollar$dft}$.

**12.2.9 Keywords for Module Escf**

**ESCF 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 $\text{$\textdollar$dft}$ or $\text{$\textdollar$ridft}$ keywords, specified above. It is recommended to use well-converged orbitals, specifying $\text{$\textdollar$scfconv 7}$ and $\text{$\textdollar$denconv 1d-7}$ for the ground-state calculation. The input for an Escf calculation can be conveniently generated using the ex menu in DEFINE, see Section 2.

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))

   $\text{$\textdollar$scfinstab rpas}$   or

   $\text{$\textdollar$scfinstab rpat}$   or

   $\text{$\textdollar$scfinstab urpa}$

2. CI singles singlet or triplet or spin-unrestricted excitation energies (HF)
$scfinstab$ ciss or
$scfinstab$ cist or
$scfinstab$ ucis

3. Eigenvalues of singlet or triplet or nonreal stability matrices (HF+RI(DFT), RHF)

$scfinstab$ singlet or
$scfinstab$ triplet or
$scfinstab$ non-real

4. Static polarizability and rotatory dispersion tensors (HF+(RI)DFT, RHF+UHF)

$scfinstab$ polly

5. Dynamic polarizability and rotatory dispersion tensors (HF+(RI)DFT, RHF+UHF)

$scfinstab$ dynpol unit
list of frequencies

where 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):

$scfinstab$ dynpol nm
  590
  400 i

The number and symmetry labels of the excited states to be calculated is controlled by the data group $soes$. Example:

$soes$
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 $soes$ all textitn; to calculate the n first excitations in all IRREPS. If n is not specified, all excitations in all IRREPS will be obtained.
12.2. FORMAT OF KEYWORDS AND COMMENTS

During an 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 $rpaconv$). An Escf run can be interrupted by typing “touch stop” in the working directory.

**general keywords**

$rpacor n$

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 \( n = 25 \).

12.2.10 Keywords for Module EGRAD

EGRAD uses the same general keywords as Escf and Grad, see Sections 12.2.7 and 12.2.9.

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 $soes$.

12.2.11 Keywords for Modules Mpgrad and Rimp2

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 \ $\text{denconv} \ 1.d-7$
2) energy convergence \ $\text{scfconv} \ 6$

Keywords Valid for Both Mpgrad and Rimp2

$maxcor \ n$

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.

$freeze$

\begin{align*}
\text{a1g} & \ 1-2 \\
\text{t1u} & \ 1
\end{align*}
The data group $\texttt{freeze}$ specifies frozen orbitals, in the above syntax by irreducible representations. The symmetry-independent and for standard-applications recommended syntax is

\begin{verbatim}
$\texttt{freeze}
  \texttt{implicit core=5 virt=2}
\end{verbatim}

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. In case of MPGRAD frozen orbitals have to be specified manually, for RIMP2 the preparation tool \texttt{rimp2prep} may be used to specify frozen core orbitals, frozen virtuals have to be specified manually. Note: In case of gradient calculations frozen core orbitals are regarded only by RIMP2, but not by MPGRAD, moreover freezing of virtual orbitals is generally not supported by MPGRAD.

\section*{MPGRAD: Essential Keywords}

All essential data groups for MPGRAD may be generated by the preparation tool \texttt{mp2prep}, apart from $\texttt{maxcor}$ (see above) these are the following:

\begin{verbatim}
$\texttt{traloop} n
  \texttt{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.}
\end{verbatim}

\begin{verbatim}
$\texttt{mointunit}
  \texttt{type=intermed unit=61 size=0 file=halfint}
  \texttt{type=1111 unit=62 size=0 file=moint#0}
  \texttt{type=1112 unit=63 size=0 file=moint#1}
  \texttt{type=1122 unit=64 size=0 file=moint#j}
  \texttt{type=1212 unit=65 size=0 file=moint#k}
  \texttt{type=1212a unit=70 size=0 file=moint#a}
  \texttt{type=gamma#1 unit=71 size=0 file=gamma#1}
  \texttt{type=gamma#2 unit=72 size=0 file=gamma#2}
  \texttt{type=1212u unit=73 size=0 file=moint#u}
  \texttt{type=1112u unit=74 size=0 file=moint#v}
  \texttt{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.

$\text{statistics mpgrad}$

statistics run (estimation of disc space needed) for the adjustment of the file sizes will be performed.

MPGRAD: **Optional Keywords**

$\text{mp2pair}$

calculation of MP2 pair correlation energies.

RIMP2: **Essential Keywords**

Apart from keywords $\text{maxcor}$, $\text{mp2energy}$ and $\text{freeze}$ (see above) RIMP2 also needs

$\text{cbas file=auxbasis}$

cross reference for the file specifying the auxiliary basis as referenced in $\text{atoms}$. We strongly recommend using auxbasis sets optimized for the corresponding MO basis sets.

Reasonable settings for these keywords may be generated by the tool RIMP2PREP. Moreover you may specify by hand:

$\text{tmpdirworkthisjob}$

specification of directory for scratch files; by default files are written to the working directory; works also with capital letters (for consistency with ricc2).

$\text{c1algorithm}$

avoids symmetry gymnastics in case of $C_1$-symmetry, rather for debugging

$\text{cbasopt}$

enforces calculation of

$$
\frac{-|\langle ij|ab \rangle_{(exact)} - \langle ij|ab \rangle_{(RI)}|^2}{(\epsilon(i) + \epsilon(j) - \epsilon(a) - \epsilon(b))},
$$

necessary for characterisation of auxiliary basis set quality and for auxiliary basis optimizations; works only for $C_{1}$-symmetry.

**Note**: all integrals are kept in memory, so this is for atoms and small molecules only.
12.2. FORMAT OF KEYWORDS AND COMMENTS

$tplot
Enforces plotting of five largest t-amplitudes as well as five largest norms of
t-amplitudes for fixed pair of occupied orbitals \( ij \). By additional integer this
number may be changed.

$mp2occ
Enforces plotting of all eigenvalues of the MP2 density matrix.

12.2.12 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.
$denconv, $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. $thime, $thize, $scfintunit). For the explanation
of these keywords see Section 12.2.5.

$cbas file=auxbasis
Auxiliary basis set for RI approximation. For details Section 12.2.11.

$freeze
Freeze orbitals in the calculation of correlation and excitation energies. For
details see Section 12.2.11.

$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 Ricc2 program is started.

$maxcor 20
The data group $maxcor adjusts the maximum size of core memory in MB
which will be allocated during the RI-CC2 run. This keyword can be set with
the RIMP2PREP tool, the default is 20 MB.
$maxcor has a large influence on computation times for RI-CC2 runs! It is
recommended to set $maxcor to ca. 75–85% of the available (physical) core
memory.

$ricc2
ccs
cis
CHAPTER 12. KEYWORDS IN THE CONTROL FILE

mp2     energy only

**cis(d)** energy only
**cis(dinf)**
**adc(2)**
**cc2**
**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)**

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 Rimp2 module. **cis(dinf)** denotes the iterative CIS(D) variant CIS(D∞).

**mp2 energy only**

If the **energy only** flag is given after the **mp2** keyword, it is assumed that only a MP2 ground state energy is requested. This switches on some shortcuts to avoid the computation of intermediates needed e.g. for the calculation of properties, gradients, the $D_1$ diagnostic, etc. or for CIS(D) and CC2 calculations.

**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.

**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 CC2 ground state energy 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 CC2 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 $\text{printlevel}$ 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.

\$excitations

\texttt{irrep=au \ multiplicity=1 \ nexc=4 \ npre=6 \ nstart=8}
\texttt{irrep=bg \ multiplicity=3 \ nexc=2 \ npre=4 \ nstart=5}
\texttt{spectrum \ states=all \ operators=diplen,dipvel}
\texttt{exprop \ states=all \ operators=qudlen}
\texttt{xgrad \ states=(ag{3} 1)}
\texttt{conv = 6}
\texttt{thrdiiis = 2}
\texttt{preopt = 3}
\texttt{leftopt}
\texttt{bothsides}

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 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 (Note that only a single state should be specified; simultaneous calculation of gradients for several states is in the present version not 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 $\max(10^{-\text{conv}}, 10^{-o\text{conv}}, 10^{-6})$, where conv refers to the values given in the data group $\$ricc2$.

**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).

**$response**

fop unrelaxed_only operators=diplen
gradient
conv = 6
zconv = 6
semicano
nosemicano
thrsemi = 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 $ricc2$).

**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 $10^{-z\text{conv}}$.

**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 thrsemi below).

**nosemicano**
use non-canonical formulation for the calculation of (transition) one-electron densities. Default is to use the semi-canonical formulation.
12.2. FORMAT OF KEYWORDS AND COMMENTS

\texttt{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.

Common options for keywords in the data groups \$\text{ricc2}$, \$\text{response}$, and \$\text{excitations}$:

\texttt{operators=diplen,dipvel}

input of operator labels for first-order properties, transition moments, etc.

Currently implemented operators/labels are

\texttt{overlap} overlap (charge) operator: the integrals evaluated in the AO basis are \(\langle \mu | \nu \rangle \)

\texttt{diplen} 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 \texttt{x\text{diplen}}, \texttt{y\text{diplen}}, \texttt{z\text{diplen}}).

\texttt{dipvel} dipole operator in velocity gauge: \(\langle \mu | \nabla_{i} | \nu \rangle\)

(all three components, individual components can be specified with the labels \texttt{x\text{dipvel}}, \texttt{y\text{dipvel}}, \texttt{z\text{dipvel}}).

\texttt{qudlen} quadrupole operator \(\langle \mu | r_{i}^{O} r_{j}^{O} | \nu \rangle\)

(all six components, individual components can be specified with the labels \texttt{xx\text{qudlen}}, \texttt{xy\text{qudlen}}, \texttt{xz\text{qudlen}}, \texttt{yy\text{qudlen}}, \texttt{yz\text{qudlen}}, \texttt{zz\text{qudlen}}).

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

\[
\Theta_{ij} = \frac{1}{2} \langle 3r_{i}r_{j} - r_{i}^2 \delta_{ij} \rangle
\]

(including nuclear contributions) is given.
angmom angular momentum $\langle \mu | L_i^O | \nu \rangle$
(all three components, individual components can be specified with the labels xangmom, yangmom, zangmom).

nef electronic force on nuclei $\langle \mu | Z_I r_I | \nu \rangle$, where $Z_I$ is the charge of the nucleus $I$ and $r_I$ is the position vector of the electron relative to the nucleus (all three components for all nuclei: the labels are xnef001, ynef001, znef001, xnef002, 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 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. states=(ag{3} 1,3-5; b1u{1} 1-3; b2u 4). This will select the triplet $a_g$ states no. 1, 3, 4, 5 and the singlet $b_{1u}$ states no. 1, 2, 3 and the singlet (which is default if no {} is found) $b_{2u}$ state no. 4.

$\$cgrad 1000
Calculate the error functional $\delta_{RI}$ for the RI approximation of $\langle ai | bj \rangle$ integrals

\[
\delta_{RI} = \frac{1}{4} \frac{\langle ab | ij \rangle_{\text{exact}} - \langle ab | ij \rangle_{RI}}{\epsilon_a - \epsilon_i + \epsilon_b - \epsilon_j}
\]

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).

12.2.13 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 2.1 (default: on).

redundant on/off
optimize molecular structures in redundant internal coordinates using
definitions of redundant internal coordinates given in $\text{redundant}$. For an optimization in redundant internal coordinates option \text{internal} has to be switched \text{on} too, and option \text{cartesian} has to be switched \text{off} (default: \text{on}).

\text{cartesian on/off}

optimize molecular structures in the space of (symmetry-distinct) cartesian coordinates (default: \text{off}).

\text{basis on/off suboptions}

optimize basis set exponents (default=\text{off}).

Available suboptions are:

\text{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

\text{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

\text{global on/off}

optimize a global scaling factor for all basis set exponents (default: \text{off}).

\begin{center}
\begin{tabular}{|l|}
\hline
\text{NOTES:} & \begin{itemize}
\item basis and global have to be used exclusively!
\item if \$\text{optimize} \ has \ been \ specified \ but \ \$\text{forceapprox} \ is \ absent,
the \ option \ \$\text{forceinit} \ \text{on} \ is \ switched \ on \ by \ default.
\item specification of the option \$\text{interconversion} \ \text{on} \ will \ over-
ride \ \$\text{optimize}!
\end{itemize}
\end{tabular}
\hline
\end{center}

\$\text{coordinateupdate} \ \text{options}

define some variables controlling the update of coordinates.

Available options are:

\text{dqmax real}

maximum allowed total change for update of coordinates. The maximum change of individual coordinate will be limited to $dq_{\text{max}}/2$ and the collective change $dq$ will be damped by $dq_{\text{max}}/(dq\cdot dq)$ if $(dq\cdot dq) > dq_{\text{max}}q$

(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).

$gdiishistory 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.

$interconversion options default=off

special input related to the transformation of atomic coordinates between cartesian and internal coordinate spaces (default: off).

Available options are:

maxiter=n

maximum number of iterations for the iterative conversion procedure internal → cartesian coordinates (default: 25).

qconv

convergence criterion for the coordinate conversion (default: 1.d-10).

on/off options

this switch activates special tasks: transform coordinates/gradients/hessians between spaces of internal/cartesian coordinates using the definitions of internal coordinates given in $intdef:

available suboptions are:

cartesian --> internal coordinate gradient hessian
cartesian <-- internal coordinate the direction of the transformation is indicated by the direction of the arrow

Note: specification of $interconversion on will override $optimize!

$forceupdate 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 method:
12.2. Format of Keywords and Comments

none no update (steepest descent)

ms suboptions Murtagh–Sargent update

dfp suboptions Davidon–Fletcher–Powell update

bfgs suboptions Broyden–Fletcher–Goldfarb–Shanno update

dfp-bfgs suboptions combined (bfgs+dfp) update

schlegel suboptions Schlegel update

ahlrichs suboptions Ahlrichs update (macro option)

suboptions if method=ms, dfp, bfgs, schlegel, ahlrichs

numgeo=integer number of structures used

maxgeo=integer maximum number of geometries (= rank of the update procedure, for ahlrichs only)

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|dg> dynamic

NOTES: if the macro option ahlrichs has been chosen and n=numgeo, ncycl=’number of geometries available’

• if ncycl < n: geometry update by inter/extrapolation using the last two geometries

• if ncycl ≥ n: diagonal update for the hessian by least mean squares fit; pulay update for the geometry (using specified modus, fmode (see pulay below))

• if (ncycl ≥ max(5, n + 3) and max(|g|) < 0.01 and g < 0.001) or H_{ij} ≠ 0 ∀ i ≠ 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 \textit{modus} (see below).

Available suboptions are:

\texttt{numpul}=integer
\begin{itemize}
  \item number of geometries to be utilized
\end{itemize}

\texttt{maxpul}=integer
\begin{itemize}
  \item maximum number of geometries
\end{itemize}

\texttt{minpul}=integer
\begin{itemize}
  \item minimum number of geometries needed to start update
\end{itemize}

\texttt{modus}=char \texttt{fmode}
\begin{itemize}
  \item \texttt{char} = \texttt{g|g} or \texttt{g|dq} or \texttt{dq|dq} defines the quantity to be minimized (\texttt{dq} = internal coordinate change).
  \item \texttt{fmode} specifies the force constants to be used (only if \texttt{char}=\texttt{g|dq} or \texttt{dq|dq}!)
\end{itemize}

\texttt{fmode}=\texttt{static}: use static force constants
\texttt{fmode}=\texttt{dynamic}: use updated force constants

\texttt{fail}=real
\begin{itemize}
  \item \texttt{real} defines the threshold for the quantity $g \cdot dq / |g| \cdot |dq|$ which defines the angle between gradient vector and coordinate change (default: 0.1). If \texttt{pulay} is used in connection with a multidimensional BFGS update for the hessian than the default is \texttt{real}=0.0. If $\frac{g \cdot dq}{|g| \cdot |dq|} > -\texttt{real}$ the pulay update for the geometry is expected to fail and will be ignored. For example:
  \begin{verbatim}
  pulay numpul=4 maxpul=4 minpul=3 modus=<dq|dq> static fail=0.2
  \end{verbatim}
\end{itemize}

\textbf{options for $\$forceupdate$}

\textbf{diagonal}
\begin{itemize}
  \item update only the diagonal force constants (update for off-diagonals will be suppressed) (only active if \texttt{method}=\texttt{ms, dfp, bfgs})
\end{itemize}

\textbf{offdamp real}
\begin{itemize}
  \item this allows to damp off-diagonal force constants by $1/\texttt{real}$ (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 automatically by RELAX. This is useful in difficult cases, where the non-diagonal update has lead to too large non-diagonal elements of the hessian.
\end{itemize}

\textbf{offreset}
\begin{itemize}
  \item 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 = \text{real}

$last MP2 energy change = \text{real}

These keywords depend on the optimization task to be processed and are updated by the corresponding program (i. g. 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 follows:

$scratch files
relax ftmp path/file

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**

**$intdef** or **$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}=. . .).

**$grad**
Cartesian coordinates and gradients calculated in subsequent optimization cycles. Entries are accumulated by one of the gradient programs (\texttt{Grad}, \texttt{Mpggrad}, \texttt{Rimp2}, \texttt{Ricc2}, \texttt{Egrad}, etc.).

**$egrad**
Basis set exponents scale factors and their gradients as calculated in subsequent optimization cycles. Entries are accumulated by one of the gradient programs.

**$globgrad**
Global scale factors and gradients as calculated in subsequent optimization cycles. Entries are accumulated by the \texttt{Grad} or \texttt{Aoforce} program.

**$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.

**$corrgrad**

<table>
<thead>
<tr>
<th>#</th>
<th>coordinate</th>
<th>increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0600</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.0850</td>
<td></td>
</tr>
</tbody>
</table>

**$forceapprox** options
Approximate force constant matrix (as needed for geometry optimization tasks). The storage format may be specified by the available options:

`format=format`  
The default format is \texttt{format=(8f10.5)}, but other 10-digit \texttt{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:

**$forceapprox format=(8f10.4)**

| 0.9124 | 0.3347 |
| -0.0108 | 0.2101 | 0.0076 |
| 0.0299 | 1.3347 | 0.1088 |
| 0.6515 | 0.0778 |
$\text{hessian (projected)}$

this data block contains the analytical cartesian force constant matrix (with translational and rotational combinations projected out) as output by the AOFORCE program and may be used to supply a high quality force constant matrix $\text{forceapprox}$ for geometry optimizations (specifying $\text{forceinit}$ on carthess, or $\text{interconversion cartesian} \rightarrow \text{internal hessian}$).

\textbf{RELAX Output Data Groups}

$\text{coord}$

either updated cartesian coordinates if a successful coordinate update has been performed, or cartesian coordinates for input internal coordinates if only a conversion from internal to cartesian coordinates has been performed.

$\text{basis}$

updated basis set exponents, basis sets contraction coefficients or scaling factors, if $\text{optimize basis on}$ has been specified.

$\text{global}$

updated global scaling factor for all basis set exponents, if $\text{optimize global on}$ has been specified.

$\text{forceapprox}$

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 ($\text{forceupdate}$) has been chosen. See \ref{3.3.13} and \ref{12.2.13}.

$\text{forcestatic}$

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: $\text{forceupdate pulay ...modus=<dq|dq> static}$.

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 \texttt{JOBEX}.

$\text{maximum norm of cartesian gradient} = \text{real}$

$\text{maximum norm of internal gradient} = \text{real}$

$\text{maximum norm of basis set gradient} = \text{real}$

\textit{real} is the absolute value of the maximum component of the corresponding gradient.
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.

12.2.14 Keywords for Module Statpt

\$statpt

itrvec 0
update bfgs
hssfreq 0
keepmode
hssidiag 0.5
radmax 0.3
radmin 1.0d-4
tradius 0.3
threchange 1.0d-5
thrmadispl 1.0d-3
thrmaxgrad 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.

hessfreq
Frequency of hessian calculation.

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-5).

thrmxdisp threshold for maximal displacement element (default: 1.0d-3).

thrmxgrad threshold for maximal gradient element (default: 1.0d-3).

thrmxdispl threshold for RMS of displacement (RMS = root mean square) (default = 5.0d-4)

thrmxgrad threshold for RMS of gradient (default: 5.0d-4).

All values are in atomic units.

12.2.15 Keywords for Module Moloch

$properties specifies the global tasks for program Moloch by virtue of the fol-
lowing options
12.2. FORMAT OF KEYWORDS AND COMMENTS

$properties

- trace off
- moments active
- potential off
- 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

$loewdin

$pa boon
momao  maodump  maofile=mao  all

$loewdin
to perform a L"owdin population analysis (options are invalid here). A L"owdin population analysis is based on decomposing $\sqrt{SD}\sqrt{S}$ instead of $DS$ in case of a Mulliken PA.

or

$mulliken

$mloewdin
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

$paboon
momao  maodump  maofile=mao  all

$mloewdin
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: $\text{shared electron numbers}$

$\text{orbitals}$

2-center threshold = $\text{real}$

3-center threshold = $\text{real}$

4-center threshold = $\text{real}$

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 $\text{mao selection}$ has to be specified:

$\text{mao selection threshold} = \text{real}$;

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 $\text{mao selection}$. 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.

$$\text{atom symb list nmao=i method=meth threshold=r}$$

The parameters in this definition have the following meaning:

$symb$ atom symbol

$list$ list of all atoms for which this definition should apply. The syntax for this list is as usual in TURBOMOLE, e.g. $2,3,8-10,12$

$nmao=i$ means number of MAOs to be included

$method=meth$ means selection criterion for MAOs. $meth$ can be $\text{occ}$ (default), $\text{eig}$, or

$man string$, where $\text{occ}$ denotes selection of MAOs by occupation numbers, $\text{eig}$ selection by eigenvalues and $\text{man}$ allows manual selection. In the latter case the string (max. 8 characters) appended to $\text{man}$ 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 $\text{mao selection}$ and is followed by the indices of the modified atomic orbitals which are to be selected.

$threshold=r$ means the threshold to be applied for the selection criteria $\text{occ}$ or $\text{eig}$ (default: 0.1).
Example:

```plaintext
$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
```

plot

option plot is out of fashion; to plot quantities on a grid, rather use $pointval
in connection with DSCF, RIEF, RIMP2 or EGRAD, as described below. If
nevertheless plot is active you need

```plaintext
$grid  #1
mo  4a1g
  origin  .000000  .000000  .000000
  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.

fit

if fit is active you need

```plaintext
$vdw_fit
shell  number_of_gridpoints  radial_factor
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.

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.
12.2. FORMAT OF KEYWORDS AND COMMENTS

$vdw\_radii
number\_of\_atom \quad \text{van\_d\_waals\_radius}$

One line per atom has to be specified, it contains the number of the atom and the van der Waals radius in [Bohr].

12.2.16 Keywords for wave function analysis and generation of plotting data

Properties of RHF and UHF wave functions as well as those of SCF+MP2 densities or such from excited state DFT-calculations can be directly analysed within the respective programs (DSCF, RIFDT, MPRGRAD, 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, $D(\text{SCF})$, in case of DSCF and RIFDT, the MP2-corrected density, $D(\text{SCF})+D(\text{MP2})$, for MPRGRAD and RIMP2 and the entire density of the excited state in case of EGRAD. For modules DSCF and RIFDT the analysis of properties may be directly started by calling dscf -proper (or ridft -proper). In case of MPRGRAD and RIMP2 this is possible only, if the MP2 density has already been generated, i.e. after a complete run of MPRGRAD or RIMP2.

Functionalities of analyses are driven by the following keywords.

$\text{mvd}$

leads to calculation of relativistic corrections for the SCF total density in case of DSCF and RIFDT, for the SCF+MP2 density in case of RIMP2 and MPRGRAD 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 $(\sum 1/Z_A \ast \rho(R_A))$.

$\text{moments}$

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:

$\text{moments} \ <i>
\begin{align*}
x_1 & \ y_1 & \ z_1 \\
x_2 & \ y_2 & \ z_2 \\
. & \ . & \ .
\end{align*}$
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 \( (D^\alpha + D^\beta) \) leading to Mulliken (brutto) charges and, in case of spin-unrestricted calculations also for the spin density \( (D^\alpha - D^\beta) \) leading to Mulliken (brutto) numbers for unpaired electrons. Besides total numbers also contributions from \( s-, p-, \ldots \) functions are listed separately.

The following modifications and extensions are supported, if the respective commands are written in the same line as $pop$:

- \texttt{lall} Additional information about \( p_x, p_y, p_z \) (and analogous for \( d \) and \( f \) functions) is displayed (lengthy output).
- \texttt{atoms list of atoms} Contributions are plotted only if arising from atoms selected by list.
- \texttt{thrpl=real} Contributions smaller than \texttt{thrpl} are not displayed (default: 0.01).
- \texttt{overlap} Mulliken atomic overlap matrix is displayed.
- \texttt{netto} Mulliken netto populations (diagonal elements of Mulliken overlap matrix) are calculated.
- \texttt{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 \texttt{eiger}. Note that occupancy of MOs is ignored, i.e. all orbitals are treated as occupied.
- \texttt{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
- \texttt{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 (\texttt{dos} in case of RHF wave functions, and \texttt{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.

$localize$

enables the generation of localized molecular orbitals (LMOs) using Boys localization. By default, all occupied orbitals are included, localised orbitals are written (by default in the AO-basis) to file(s) lmo in case of RHF and laxp and lbet 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 $pointval$).

The keyword allows for following options (to be written in the same line):

mo normalfont list of MOs

Include only selected MOs (e.g. valence MOs) in localization procedure (numbering as available from EIGER).

sweeps=integer

maximum nuber of orbital rotations to get LMOs; default value is 10000 (sometimes not enough, in particular for highly delocalised systems).

thrcont=real

lower threshold for displaying MO and Mulliken contributions (default: 0.1).

CAO

LMOs are written to file in the CAO basis (instead of AO)

$esp fit$

fits point charges at the positions of nuclei to electrostatic potential arising from electric charge distribution (for UHF cases also for spin density). 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
CHAPTER 12. KEYWORDS IN THE CONTROL FILE

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:

$esp_fit
shell i1 s1
shell i2 s2
:

integer numbers $i$ define the number of points for the respective shell, real numbers $s$ the scaling of radii (default corresponds to one shell with $s=1.0$).

$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 ($D(\text{SCF} + \text{MP}2)$) on td.plt and the MP2 contribution on mp2d.plt and in the UHF case one obtains the total density ($D^\alpha(\text{SCF} + \text{MP}2) + D^\beta(\text{SCF} + \text{MP}2)$) on td.plt, the spin density ($D^\alpha(\text{SCF} + \text{MP}2) - D^\beta(\text{SCF} + \text{MP}2)$) 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

$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 $pointval$):

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 eonly" 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).

lmo list of LMO numbers
calculation of amplitudes of LMOs (previously generated by $localize$) ordered by the corresponding diagonal element of the Fock matrix in the LMO basis.

dens has to be set, if additionally to one of the above quantities also the density is to be computed.

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.

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.
CHAPTER 12. KEYWORDS IN THE CONTROL FILE

For 3D grids non-default boundarys, 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
origin 1 1 1

Grid vectors (automatically normalised) 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.

Grids of lower dimensionality may be specified (in the same line as $pointval) by typing either geo=plane or geo=line or geo=point. The way to use is best explained by some examples:

$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

Values are calculated at a plane spanned by vectors (0,1,0) and (0,0,1) centered at (1,1,1).

$pointval geo=line
grid1 vector 0 1 0 range -2,2 points 50
origin 0 0 1

Values are calculated at a line in direction (0,1,0) centered at (0,0,1). Output format as above.

$pointval geo=point
7 5 3
0 0 7

Values are calculated at the two points (7.0,5.0,3.0) and (0.0,0.0,7.0).

12.2.17 Keywords for Module Frog

The \emph{ab initio} molecular dynamics (MD) program FROG needs a command file named \texttt{mdmaster}. The interactive MD\textsc{prep} program manages the generation of \texttt{mdmaster}.
and associated files. It is always a good idea to let MDPREP check over mdmaster before starting an MD run. MDPREP 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 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 timestep chosen in MDPREP. 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 MDPREP 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 Frog:

\$nsteps \ 75

Number of MD timesteps to be carried out. \$nsteps is decreased by 1 every time Frog is run and JOBEX -md stops when \$nsteps reaches 0.

\$natoms \ 9

Number of atoms in system.

\$current \ file=mdlog.aa

The file containing the current position, velocity, time and timestep, that is, the input configuration. During an MD run the \$current information is generally kept at the end of the \$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 \texttt{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.

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
  thickness 1.0
  temperature 300.0

\$barrier specifies a virtual cavity for simulating condensed phases. The optional flag, \texttt{angstroms}, can be used to indicate that data will be entered in Ångströms rather than Bohr.

\texttt{type}
  can be one of \texttt{orth}, \texttt{elps}, or \texttt{none}, for orthorhombic, ellipsoidal, or no barrier (the default) respectively.

\texttt{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 2.0H/Bohr/Ångström, a bastard combination of units.

thickness
  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  0  0.9  1.2
  type  F  C  0.0  1.7
  type  H  C  -1.0  1.2
  2  1  0.0
  3  1  1.54
  4  1  -1.0

$constraints
  specifies and/or automatically generates atomic distance constraints. The optional flag, angstroms, can be used to indicate that data will be entered in Ångströms rather than Bohr.

tolerance
  is the convergence criterion for application of constraints. All distances must be within +/- tolerance of the specified constraint. Additionally, the RMS deviation of all constrained distances must be below 2/3 of tolerance.

adjpercyc
  is the fraction of the total distance correction to be applied on each constraint iteration.

type X A normalfont const rmax
  commands FROG to find the closest A atom to each atom X that is closer
than \( \text{rmax} \) and apply \( \text{const} \). The first \textbf{type} line above examines each \( \text{H} \) atom and looks for any \( \text{O} \) atoms within 1.2 Å. The shortest distance, if any, is then fixed at 0.9 Å. Similarly, the second \textbf{type} line binds each \( \text{F} \) to the closest \( \text{C} \) within 1.7 Å, but since \( \text{const}=0.0 \), that distance is fixed at the current value. The third \textbf{type} line attaches \( \text{H} \) atoms to the appropriate nearby \( \text{C} \), but at the current average \( \text{H-C} \) distance multiplied by the absolute value of \( \text{const} \).

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 \( t=\text{real number} \). 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:

\begin{verbatim}
$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 \texttt{$log\_history$, $ke\_control$}).

$md\_action
  set temperature at t=<real> to x=<real> K
  set total energy at t=<real> to x=<real> H
\end{verbatim}
set kinetic energy at \(t=<\text{real}>\) to \(x=<\text{real}>\) H
set position file=\(<\text{filename}>\) at \(t=<\text{real}>\)
set velocity file=\(<\text{filename}>\) at \(t=<\text{real}>\)
set velocity at \(t=<\text{real}>\) random
set velocity at \(t=<\text{real}>\) zero

At some time during the \textit{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 \(x=\text{real number}\). Vectors must be read in FROG format from \texttt{file=file}.

\$\text{md\_action}\$
\begin{verbatim}
anneal from \(t=<\text{real}>\)
anneal from \(t=<\text{real}>\) \(x=<\text{real}>\)
quench from \(t=<\text{real}>\)
quench from \(t=<\text{real}>\) \(x=<\text{real}>\) \texttt{file=file}
relax at \(t=<\text{real}>\)
\end{verbatim}

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 (\texttt{anneal}; default factor 0.905 over 100 steps) or lowered rapidly by reversing all uphill motion (\texttt{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 optimisation can be carried out in a subdirectory (\texttt{relax}).

\$\text{md\_action}\$
\begin{verbatim}
free from \(t=<\text{real}>\)
\end{verbatim}

Finally, this instruction turns off any previous action and resumes free dynamics. This is the default status of an MD run.

\subsection{12.2.18 Keywords for Module \texttt{MPSHIFT}}

In order to control the program execution, you can use the following keywords within the control file:

\$\text{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 \texttt{mp2prep}.

\$\text{straloop }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 \texttt{mp2prep}. 
$\textit{mointunit}$

Scratch file settings for an MP2 calculation. Please refer to Section 12.2.11 for a description of the syntax. This flag will be set by the script $\textit{mp2prep}$.

$\textit{csconv} \, \text{real}$

Sets the convergence threshold for the shielding constant of succeeding CPHF iterations. The unit is ppm and the default value is 0.01.

$\textit{csconvatom} \, \text{integer}$

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).

$\textit{thime}, \textit{thize}, \textit{scftol}, \textit{scfintunit}, \textit{scfmo}$

have the same meaning as in DSCF (see Section 12.2.5); Since mpshift works 'semi-direct' it uses the same integral storage.

$\textit{scratch files}$

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:

$\textit{scratch files}$

```
mpshift csssmat path1/file1
mpshift cshsmat path2/file2
mpshift csdgsmat path3/file3
mpshift csusmat path4/file4
mpshift dens path5/file5
mpshift fock path6/file6
mpshift dfock path7/file7
mpshift idvds1 path8/file8
mpshift idvds2 path9/file9
mpshift idvds3 path10/file10
mpshift jdvds1 path11/file11
mpshift jdvds2 path12/file12
mpshift jdvds3 path13/file13
mpshift cshmmat path14/file14
```

$\textit{trast}, \textit{trand} \, \text{tralooop-number}$

stands for tralooop start and tralooop end. Each loop or pass in MP2 chemical shift calculations can be done individually by providing the keywords $\textit{trast}$ and $\textit{trand}$. This can be used to do a simple parallelization of the run:
Create separate inputs for each tralooop. Add

$\textit{trast} \, \langle\text{number}\rangle$

$\textit{trand} \, \langle\text{number}\rangle$
12.2. FORMAT OF KEYWORDS AND COMMENTS

in the control files, number goes from 1 to the number of $traloops. Each calculation will create a restart file called restart.mpshift. To collect all steps and to do the remaining work, copy all restart files to one directory and rename them to restart.mpshift.number, add $trast -1 and $trand number_of_traloops to the control file and start MPshift.

12.2.19 Keywords for Parallel Runs

On all systems the parallel input preparation is done automatically. Details for the parallel installation are given in Section I.8.1. The following keywords are necessary for all parallel runs:

$parallel_platform architecture
$numprocs number CPUs

Currently the following parallel platforms are supported:

SMP for systems with very fast communication; all CPUs are used for the linear algebra part. Synonyms for SMP are:
HP V-Class, SP3-SMP and HP S/X-Class

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:
SP3 and linuxcluster

cluster for systems with slow communication, the linear algebra part will be done on one single node. Synonyms for cluster are:
HP Cluster and every platform that is not known by TURBOMOLE

SGI 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.

$numprocs is the number of slaves, i.e. the number of nodes doing the parallel work. If you want to run MPRGRAD, $traloop has to be equal to or a multiple of $numprocs.

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 RIFDT/RIDGRAD) $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
12.2. FORMAT OF KEYWORDS AND COMMENTS

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 behaviour 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 13

Sample control files

13.1 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 12.
13.2 NH₃ Input for a RHF Calculation

13.2.1 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
$pople 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
13.2. \textit{NH}_3 \textit{INPUT FOR A RHF CALCULATION}

\begin{verbatim}
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

13.2.2 File coord

$coord
   0.00000000000000 0.00000000000000 0.54561506935122 n
   -0.87806233111566 1.52084856970468 -0.18187168978374 h
   -0.87806233111566 -1.52084856970468 -0.18187168978374 h
   1.75612466223131 0.00000000000000 -0.18187168978374 h
$intdef
   # definitions of internal coordinates
   1 k 1.00000000000000 stre 4 1 val= 1.90084
   2 k 1.00000000000000 bend 4 3 1 val= 106.27756
   1.00000000000000 bend 3 2 1
   1.00000000000000 bend 2 4 1
$end

13.2.3 File basis

$basis
   *
n def-SVP
   # n (7s4p1d) / [3s2p1d] \{511/31/1\}
   *
   5 s
   1712.8415853  -53934125305E-02
\end{verbatim}
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
*
 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

13.2.4 File mos

$scfmo expanded format(4d20.14)

1 a1 eigenvalue= -1.5633041862301D+02 nsaos=10
 .9869903163455D+00 - .47221435341751D-01 -.55873125006179D-02-.48016374887169D-02
 .26746008768233D-02 - .20823779196149D-03 - .14270460008808D-01-.90849517503597D-02
 .5876121352806D-03 -.29091871198884D-03
 2 a1 eigenvalue= -.99896275238736D+00 nsaos=10
 .26412162337482D+00 .51846472345768D+00 .37623729061179D+00-.77139882704089D-02
 -.4725232928716D-02 -.21494050853221D-02 - .11796673774658D+00 -.83316086019840D-01
 -.11229203934388D-01 - .27038186251429D-02
 3 a1 eigenvalue= -.57101279949392D+00 nsaos=10
 -.35584190011701D-01 -.9639258881594D-01 -.70254605702716D-01 -.65569041318341D+00
 -.4474614963029D+00 -.40094278774992D-03 -.51691158134284D-01 -.47722350097160D-01
 .19189122068531D-02 .56638497851180D-03
 1 e eigenvalue= -6.4374209294851D+00 nsaos=9
 -.49313475446075D+00 -.33757893444760D+00 -.76142296576409D-04 -.74524664248740D-04
13.2. $NH_3$ INPUT FOR A RHF CALCULATION

$-.26407572210452D+00-.22619038902975D+00-.50035170531670D-05-.12199166245418D-03$
$.63021657663245D-04$
$end$
13.3 NO$_2$ input for an unrestricted DFT calculation

13.3.1 Main File control

$title
NO$_2$ 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
$pople AO
$basis file=basis
$rundimensions
dim(fock,dens)=1098
natoms=3
nshell=18
nbf(CAO)=45
nbf(AO)=42
dim(trafos[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 .1000000E-04
$thime 5
$scfdamp start=1.500 step=.050 min=.100
$scfdump
$scfintunit
13.3. NO$_2$ INPUT FOR AN UNRESTRICTED DFT CALCULATION

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

13.3.2 File coord

$coord
0.00000000000000  0.00000000000000  -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
13.3.3 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

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
13.3. NO₂ INPUT FOR AN UNRESTRICTED DFT CALCULATION

.27641544411  1.0000000000
  1  d
  1.2000000000  1.0000000000

* $end
13.4 TaCl₅ Input for an RI-DFT Calculation with ECPs

13.4.1 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
$people 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
$interconversion off
13.4. TACL$_5^5$ INPUT FOR AN RI-DFT CALCULATION WITH ECPS

```
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
  ahrlrichs numgeo=0 mingeo=3 maxgeo=4 modulus=<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)
a2' 1-2 (2)
e' 1-10 (2)
a2" 1-8 (2)
e" 1-4 (2)
$end
```

13.4.2 File coord

```
$coord
  .00000000000000 .00000000000000 .00000000000000 ta
  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
```
13.4.3 File basis

$basis
* ta def-SVP
# ta (7s6p5d) / [6s3p2d] {211111/411/41} *
  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
  .10745336254 1.0000000000
  1 s
  .39142776556E-01 1.0000000000
  4 p
  7.4188720000 .26979695152
  5.6984100000 -.4696874449
  1.1777211960 .50905100155
  .54478533555 .52298161137
  1 p
  .22309270117 1.0000000000
  1 p
  .43100000000E-01 1.0000000000
  4 d
  3.9738796278 -.52799310714E-01
  1.4528884813 .1855819471
  .61042908544 .42959071631
  .24216276510 .43497228232
  1 d
  .87909318337E-01 1.0000000000
*
cl def-SVP
# cl (7s5p) / [6s2p] {211111/41} *
  5 s
  10449.827566 .19708362484E-02
  1571.7365221 .14754727977E-01
  357.12065523 .66679112875E-01
13.4. TACL\textsubscript{5} INPUT FOR AN RI-DFT CALCULATION WITH ECPS

\begin{verbatim}
 100.25185935  .17228924084
 30.812727554  .15883786100
  3 s
  51.923789434  -.10009298909
  5.7045760975  .60841752753
  2.3508376809  .54352153355
  1 s
  .44605124672  1.0000000000
  1 s
  .16848856190  1.0000000000
  5 p
  307.66790569  -.87801484118E-02
  72.102015515  -.63563355471E-01
  22.532680262  -.24016428276
  7.8991765444  -.47798866557
  2.8767268321  -.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
\end{verbatim}
13.4.4 File auxbasis

```plaintext
$jabas
  *  
  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   .765184111D-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   -.13896250D-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
```
13.4. TACL_5 INPUT FOR AN RI-DFT CALCULATION WITH ECPS

* 
* 
cl 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 
1.3000000 1.0000000 
* 
$end
13.5 Basisset optimization for Nitrogen

13.5.1 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
  a1g 1-2 ( 2 )
$open shells type=1
t1u 1 ( 1 )
$end

13.5.2 File coord

$coord
  0.00000000000000  0.00000000000000  0.00000000000000
$n
$user-defined bonds
$end

13.5.3 File basis

sis
*
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 0.51294466049
1 p expopt
0.21954348034 1.0000000000
# 1 d
# 1.0000000000 1.0000000000
*

13.5.4 File mos

$scfmo scfconv=10 format(4d20.14)
# SCF energy is -54.3329250250 a.u. (virial theorem = 2.000000001)
#
  1 a1g eigenvalue=-.15623888057347D+02 nsaos=3
  -.99166890864040D+00-.28420294406651D-010.915199592317893D-02
  2 a1g eigenvalue=-.92524548524703D+00 nsaos=3
  0.30506869715453D+00-.65051761026701D+00-.44610487551870D+00
  3 a1g eigenvalue=0.74881229854801D+00 nsaos=3
  0.307593029354343D+00-.16295969601691D+00.16126161147521D+01
  1 t1u eigenvalue=-.56865046629517D+00 nsaos=2
  0.67926397018841D+00.46005039886841D+00
  2 t1u eigenvalue=0.96169069264790D+00 nsaos=2
  -.95675659621171D+00.10794148212163D+01
$end
13.6 ROHF of Two Open Shells

13.6.1 Extracts from control for \( \text{O}_2 \) 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
  a2u  1-2
  eu   1
$open shells type=1
  eg   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
  a2u  1-2
  eu   1
$open shells type=1
  eg   1
$roothaan 1
    a = 1/2    b = 0
$energy    SCF     SCFKIN     SCFPOT
  1  -149.4297623470  149.4298692899  -298.8596316369
```
13.6.2 Extracts from control for $O_2$ in $D_{2h}$ Symmetry

# HF-SCF/SVP

# Triplet-sigma in D2h
#
$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)
$rooothaan 1
    a = 1 b = 2
$energy SCF SCFKIN SCFPOT
  1 -149.4774402750 149.4798706643 -298.9573109393

# Singlet-delta in D2h : 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)
$rooothaan 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 14

Samples for turbo.in files

14.1 Introduction

turbo.in is a GAUSSIAN-type input file from which the script TMOLE generates the TURBOMOLE input and executes the necessary TURBOMOLE modules and/or tools. The following sample inputs demonstrate typical usage of TMOLE. The syntax of the file turbo.in is explained in Section 1.9

14.2 RI-MP2 calculation of Phenyl

Geometry optimization at MP2 level using the RI approximation. Number of max. SCF iterations is set to 99 (scf.msil=99). The number of geometry cycles is set to 40 (geo.nrgc=40). The program will determine the symmetry of the molecule (gen_symm=auto, default). The coordinates are in TURBOMOLE format, there is no need for specifying a coordinate format.
14.3 Vibrational Spectrum of Phenyl

Calculation of the vibrational spectrum of Phenyl at MP2 level. Analytical second derivatives are not implemented in Turbomole, so they are calculated numerically (for_nfre=1). Symmetry is set explicitly to $C_2v$ (gen_symm=c2v).

%method
FORCE :: ri-mp2/TZVP [for_nfre=1, gen_symm=c2v]
%charge
0
%coord

%end
14.4 DFT calculation of Benzene

Energy calculation of Benzene at DFT(B-P86) level using SVP basis set. Integration grid is set to m4 (scf_grid=m4). A title is specified (%title). A statistics run is performed before the energy calculation (gen_stat=1). The options are continued in the next line (& at the end of the line).

%title
DFT calculation of Benzol
%method
ENERGY :: b-p/SVP [gen_stat=1,scf_msil=99,&
    scf_grid=m4]
%charge
0
%coord
    .00000000000000 -2.68300008773804 .00000000000000 c
    .00000000000000 -4.74300003051758 .00000000000000 h
    .00000000000000 -1.34150004386902 -2.32354623433702 c
    .00000000000000 -2.37150001525879 -4.10755851657859 h
    .00000000000000 -1.34150004386902 2.32354623433702 c
    .00000000000000 -2.37150001525879 4.10755851657859 h
    .00000000000000 1.34150004386902 -2.32354623433702 c
    .00000000000000 2.37150001525879 -4.10755851657859 h
    .00000000000000 1.34150004386902 2.32354623433702 c
    .00000000000000 2.37150001525879 4.10755851657859 h
    .00000000000000 2.68300008773804 .00000000000000 c
    .00000000000000 4.74300003051758 .00000000000000 h
%end

14.5 AOFORCE calculation of Benzene

Analytical (for nfre=0, default) calculation of the vibrational spectrum at DFT/B-3LYP level using the RI approximation. Number of SCF cycles ist set to 99 (scf_msil=99).
14.6 UFF calculation of Water

Geometry optimization with max. 99 cycles (geo_nrgc=99) of water at UFF level. The coordinates are in the general xyz format (%coord xyz). The symmetry is determined automatically (gen_symm=auto)

%method
GEOMY :: uff/ [geo_nrgc=99,gen_symm=auto]
%charge
0
%coord xyz
3
Energy = -76.46516801323
0  0.0000000  0.0000000  -0.0668805
H -0.7658756  0.0000000  0.5307937
H  0.7658756  0.0000000  0.5307937
%end
14.7 Potential curve for the O–H bond in H$_2$O

Calculation of the potential curve for stretching one O–H bond in H$_2$O. The bond will be stretched from 0.95 Å to 1.35 Å in steps of 0.10 Å. The geometry is specified in Z-matric format (see Section %coord gauzmat).

```
%method
GEOMY :: ri-b-p/SVP [gen_symm=c1]
%charge
0
%coord gauzmat
   o
      h 1 b1
      h 1 b2 2 a1

b1 0.95
b2 0.95
a1 109
%scan
   b1 0.95 0.1 1.35
%end
```

14.8 Bending potential for Ag$_3$

Calculation of the potential curve of Ag$_3$ for bending in the range from 62° to 142° in 4° steps. For each step an optimization of remaining geometry parameters (here Ag–Ag distance) will be done. The symmetry (C$_{2v}$) is preserved during the calculation. The MO-occupation will be overwritten with Section %add_control_commands.
%method
GEOMY :: b3-lyp/SVP [geo_nrgc=99,gen_stpt=0,gen_crd=ired,gen_symm=auto,gen_blow=1,&
scf_msi=99]

%charge
0
%coord gauzmat
  ag
  ag 1 b1
  ag 1 b2 2 a1

b1 2.70
b2 2.70
a1 62
%scan
  a1 62 4 140
%add_control_commands
$alpha shells
  a1 1-11 ( 1 )
  a2 1-4 ( 1 )
  b1 1-9 ( 1 )
  b2 1-5 ( 1 )
$beta shells
  a1 1-11 ( 1 )
  a2 1-4 ( 1 )
  b1 1-8 ( 1 )
  b2 1-5 ( 1 )
ADD END
%end
Chapter 15

The Perl-based Test Suite

Structure

15.1 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.

15.2 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,

```
prognamed/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.

### 15.3 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 \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.

### 15.4 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 \texttt{-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 \texttt{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 \texttt{.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 \texttt{Ttest}. Testing specific versions of \texttt{TURBOMOLE} modules is provided by loading path options, \texttt{-l} for binaries, \texttt{-ls} for scripts, and \texttt{-x} for a specific executable. For benchmarking, you need the \texttt{--timings} option to produce the timing summaries, and the \texttt{--newref} option to save the current program timings as the new reference. The module specifications and \texttt{--short}, \texttt{--long}, and \texttt{-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 \texttt{DEFCRIT} file (see below).

### 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 \texttt{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 \texttt{dir} (default: \texttt{TESTDIR.sysname}). Useful if new criteria or new references are established.
- **--validate dir**
  - Examines the output files in the directory \texttt{dir} (default: \texttt{TESTDIR.sysname}) and highlights the positions of the retrieved matches.

### Loading path and naming options

- **--loaddir dir**
  - Loading path for the \texttt{TURBOMOLE} binaries (default: \texttt{$TURBODIR/bin/sysname}).
- **-l dir**
- **--scriptdir dir**
  - Loading path for the \texttt{TURBOMOLE} scripts (default: \texttt{$TURBODIR/scripts}).
- **-ls dir**
- **--testprog prog**
  - Tests the given executable \texttt{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
--output file  (default: TESTPROTOKOLL).
--gprotfile file  Name for the global protocol file
    (default: TESTPROTOKOLL.sysname).
--checkfile file  Name for the check protocol file
    (default: CHECKPROTOKOLL).
--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: RUNCRT.val).

Execution options
--short  Only short / long subdirectories of the
    test suite will be tested (default: --short
    --long).
--restart file  The list of test examples for execution will
    be read in from file
    (default: PROBLEMS.sysname).
--newref string  Produces new reference timings and writes them
    to the CRIT file. A short description of the reference
    platform is provided by string.
--fileref  Produces new reference files.
--batchmode  Running in batch mode, no screen output.
--errorstop  Stops / Does not stop after the first error.
--noerrorstop  (default: --noerrorstop).
--timings  Writes / Does not write the timings on file for
    further processing. (default: --notimings).
--runopts  Sets the conditions under which the test is run
    (default: "sequential, parallel").
Bibliography


Index

(non)-append mode, 55
*, 37
-fanal, 149
-statpt, 87
.map, 150, 235
.sys.data, 45
$2e-ints`_shell_statistics, 244
$2e-ints_shell_statistics, 244
$TURBODIR/uff/parms.in, 177
.actual step, 158
.alpha shells, 131, 173, 191
.anadens, 150
.atoms, 56, 171, 172, 192, 193, 202, 208
.barrier, 238
.basis, 56, 95, 99, 100, 170, 224
.beta shells, 131, 173, 191
.boys, 227
.cialgorithm, 208
.cbas, 121, 136, 208, 209, 216
.cbasopt, 208
.cdspectrum, 133, 205
.cgrad, 216
.closed shells, 58, 59, 172, 183, 190, 191
.constraints, 239
.coord, 43, 45, 95, 98, 100, 160, 170, 175, 177, 224
.coordinateupdate, 94, 217
dqmax, 217
.interpolate, 218
.statistics, 218
.corrgad, 223
.cosmo, 197–199
.allocate_nps, 197
.ampran, 197
cavity, 197
closed, 197
.open, 197
disex, 197
epsilon, 197
.nppa, 197
.nspa, 197
.phsran, 197
.routf, 197
.rsolv, 197
.use_old_amat, 197
.cosmo_atoms, 197, 198
.csconv, 242
.csconvatom, 242
.csmp2, 157, 241
.current, 237
.denconv, 25, 121, 122, 130, 136, 180, 203, 209
.functional, 192
.gridsize, 192
.test-integ, 194
.drvopt, 71, 199, 200
.basis on, 99
.drvopts, 155
.drvtol, 71
.ecp, 129, 170
.egrad, 95, 99, 100, 216, 223, 225
.electrostatic field, 129, 180, 181
.end, 43, 171
.energy, 134, 171, 212, 214
.escfiterlimit, 206
<table>
<thead>
<tr>
<th>Command</th>
<th>95th</th>
<th>100th</th>
<th>102th</th>
<th>200th</th>
<th>201th</th>
<th>201th</th>
</tr>
</thead>
<tbody>
<tr>
<td>$esp_fit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$excitations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bothsides</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conv</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exprop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>irrep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>leftopt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>preopt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spectrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>thrdiis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xgrad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$exopt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$fermi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hlcrt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmfac</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tmstrt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$firstorder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$fldopt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st derivative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd derivative</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edelt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>geofield</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$forceapprox</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97–101, 171, 217, 221, 223, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$forceconv</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41, 221, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>carthess</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>default</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>individual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>off</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>carthess, 41, 102, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diag</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$forceiterlimit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201, 203</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$forcestatic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$forceupdate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98, 218, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ahlrichs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indgeo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxgeo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>numgeo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>allow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bfgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>damping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dfp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dfp-bfgs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagonal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>offdamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>offreset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pulay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxpul</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>minpul</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>modulus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>numpul</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reseig</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scale</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schlegel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>thrbig</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>threig</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$freeze</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121, 122, 129, 136, 207–209</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$gdiishistory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$global</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95, 100, 222, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$globgrad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95, 100, 223</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$grad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95, 98, 100, 160, 171, 212, 214, 223, 225</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$grad_send dens</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>245</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85, 230</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h0hessian</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$hessian</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71, 95, 101, 102, 200, 201, 222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$hessian (projected)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71, 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$incore</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$intdef</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43, 45, 97, 98, 155, 170, 216, 218, 222, 223</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$interconversion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93, 218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxiter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97, 217, 218</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
qconv, 218
$intsdebug, 182
$ironly, 202
$isopts, 202
$issub, 201, 202
$jbas, 109, 171, 192
$jbasis, 109, 193
$ke_control, 238, 240
$last MP2 energy change, 222
$last SCF energy change, 222
$last excitation energy change, 134
$last step
  relax, 170
$les, 91, 154, 202
  all, 202
$lesiterlimit, 202
$hf, 195
$localize, 160, 233, 235
  mo, 233
  sweeps, 233
  thrcont, 233
$lock off, 169
$loewdin, 228
$log, 237
$log_history, 238, 240
$m-matrix, 101, 222
$m-ao, 229
$marij, 109, 193
  extmax, 193
  lmaxmom, 193
  nbinmax, 193
  precision, 193
  thrmom, 193
  wsindex, 193
$maxcor, 70, 121, 136, 154, 201, 206–209
$maximum norm of
  basis set gradient, 224
  cartesian gradient, 224
  internal gradient, 224
$md_action, 240, 241
$md_status, 237, 240
$mo output format, 183, 187
$mo-diagram, 182
$mo-intensity, 121, 122, 157, 207, 242
$moments, 227, 231
$moprint, 183
$mp2energy, 25, 121, 206, 208
$mp2occ, 124, 209
$mp2pair, 208
$mulliken, 228
$mrdist, 160, 231
$natoms, 237
$natural orbital
  occupation, 171
$natural orbitals, 171, 183
  occupation, 183
$newcoord, 153
$newecp, 172
$nmr, 157
  dft, 157
  mp2, 157
  rhf, 157
  shielding constants, 157
$nomw, 91, 201
$nopproj, 201
$nosalc, 201
$nrhess, 201
$nrpmodes, 201
$nrpspec, 201
$nsteps, 237
$numprocs, 243
$oldgrad, 225
$open shells, 58, 59, 172, 190
$operating system, 169
$optimize, 93, 94, 155, 216–218
  basis, 94, 217
    logarithm, 217
    scale, 217
  cartesian, 94, 217
  global, 94, 217
  internal, 94, 97, 216, 222
  redundant, 94, 216
$paboon, 228
$parallel\_parameters, 244
$parallel\_platform, 243
$pardft, 244
$path, 169
$point\_charges, 129, 183
$points, 80, 85, 227
$pointval, 150, 161, 162, 234
dens, 235
fld, 162, 235
fmt, 235
map, 236
plt, 236
vec, 236
xyz, 236
geo, 163, 236
line, 236
plane, 236
point, 236
integrate, 234
lmo, 163, 235
mo, 163, 235
pot, 162, 235
$pop, 160, 232, 233
atoms, 233
dos, 233
lall, 233
mo, 233
netto, 233
overlap, 233
thrlpl, 233
$pople, 172
$prediag, 183, 186
$printlevel, 209, 211
$properties, 78, 227
$ramanonly, 202
$redund\_inp, 173
$redundant, 98, 155, 217, 223
$response, 214
conv, 214
fop, 214
gradient, 214
nosemicano, 214

semicano, 214
thrssemi, 214
zconv, 214
$response, 136, 146, 213, 215
$restart, 186
$restartd, 184, 186
$ricc2, 136, 137, 139, 142, 146, 209, 210, 214, 215
adc(2), 210
cc2, 210
ccs, 210
cis, 210
cis(d), 210
cisdinf, 210
conv, 210
fmtprop, 210
geoopt, 146, 210
hard\_restart, 210
iprint, 210
lindep, 210
maxiter, 210
maxred, 210
mp2, 210
mxdios, 210
nohard\_restart, 210
norestart, 210
oconv, 210
restart, 210
$ricore, 66, 108, 109, 129, 133, 154, 192, 193, 245
$ricore\_slave, 245
$ridft, 108, 129, 133, 192, 193
$ripop, 193
$rohf, 190
$roothaan, 173, 190
$rpacconv, 128, 205
$rpacor, 133, 205
$rundimensions, 184
$scfconv, 25, 68, 130, 184, 189, 203, 209
settings for

AOFORCE, 154
INDEX

**Numforce**, 154

$scfdenapproxl, 182, 185
$scfdiis, 184, 186
$scfdump, 182, 184, 186
$scfinstab, 108, 128, 205
ciss, 204
cist, 204
dynpol, 204
non-real, 204
polly, 204
rpas, 203
rpat, 203
singlet, 204
triplet, 204
ucis, 204
urpa, 203
$scfintunit, 129, 184, 186, 187, 189, 209, 242
file, 187
size, 187
unit, 187
$scfiterinfo, 186
$scfiterlimit, 187
$scfmo, 59, 171, 172, 183, 184, 186–188, 191, 242
expanded, 188
file, 187
format, 188
none, 59, 187
scfconv, 188
scfdump, 188
$scfmo none, 59
$scforbitalorder, 188
$scforbitalshift, 188
automatic, 188
closedshell, 188
individual, 188
noautomatic, 188
$scftol, 188, 189, 209, 242
$scratch
$scratch
files, 242, 244
$scratch files, 189, 222, 243
$seed, 238
$sharedtmpdir, 152
$soes, 129, 132, 133, 204–206
$spectrum, 133, 205
$start vector
generation, 131, 205
$statistics, 186, 189, 190
dscf, 109, 170, 189
dscf parallel, 189, 244
grad parallel, 244
kora, 189
mpgrad, 122, 170, 189, 208
mpshift, 157
off, 170, 189
polly, 189
$statpt, 90, 91, 226
bfgs, 91
hessfreq, 226
hssidiag, 226
itrvec, 90, 226
keepmode, 226
powell, 91
radmax, 226
radmin, 226
threchange, 90
thrmaxedisp, 90
thrmaggrad, 90
thrrmsdisp, 90
thrrmsgrad, 90
tradius, 90, 226
update, 226
$sum
rules, 205
$suspend off, 169
$symmetry, 129, 171
$thime, 107, 108, 129, 190, 209, 242
$thize, 107, 108, 129, 157, 182, 190, 209, 242
$title, 171, 238
$tmpdir, 151, 152, 208, 209
$tplot, 123, 209
$traloop, 121, 122, 157, 207, 241, 243
$trand, 242
$trast, 242
$turbomole, 237
$twoint, 156
$suff, 103, 175
    maxcycle, 103
$suffgradient, 175, 177
$suffhessian, 175, 178
$sufftopology, 175, 177
$suhf, 173, 191
$suhfmo_alpha, 131, 171, 187, 191
$suhfmo_beta, 131, 171, 187, 191
$userdefined bonds, 170
$vdw_fit, 230
$velocity gauge, 205
%add_control_commands, 36
%charge, 36
%coord, 35
%method, 31
    basis set choice, 32
    ENRGY, 31
    FORCE, 31
    GEOMY, 31
    GRADI, 31
    level of calculation, 31
    properties, 31
    run options, 32
        general, 32
        SCF, 33
    structure optimization, 35
%scan, 36
%title, 36
&plt, 234

Actual, 19
actual step
    dscf, 170
ADC(2)
    RI-, 209
analysis of normal modes

$traloop, 121, 122, 157, 207, 241, 243
$trand, 242
$trast, 242
$turbomole, 237
$twoint, 156
$suff, 103, 175
    maxcycle, 103
$suffgradient, 175, 177
$suffhessian, 175, 178
$sufftopology, 175, 177
$suhf, 173, 191
$suhfmo_alpha, 131, 171, 187, 191
$suhfmo_beta, 131, 171, 187, 191
$userdefined bonds, 170
$vdw_fit, 230
$velocity gauge, 205
%add_control_commands, 36
%charge, 36
%coord, 35
%method, 31
    basis set choice, 32
    ENRGY, 31
    FORCE, 31
    GEOMY, 31
    GRADI, 31
    level of calculation, 31
    properties, 31
    run options, 32
        general, 32
        SCF, 33
    structure optimization, 35
%scan, 36
%title, 36
&plt, 234

Actual, 19
actual step
    dscf, 170
ADC(2)
    RI-, 209
analysis of normal modes

$traloop, 121, 122, 157, 207, 241, 243
$trand, 242
$trast, 242
$turbomole, 237
$twoint, 156
$suff, 103, 175
    maxcycle, 103
$suffgradient, 175, 177
$suffhessian, 175, 178
$sufftopology, 175, 177
$suhf, 173, 191
$suhfmo_alpha, 131, 171, 187, 191
$suhfmo_beta, 131, 171, 187, 191
$userdefined bonds, 170
$vdw_fit, 230
$velocity gauge, 205
%add_control_commands, 36
%charge, 36
%coord, 35
%method, 31
    basis set choice, 32
    ENRGY, 31
    FORCE, 31
    GEOMY, 31
    GRADI, 31
    level of calculation, 31
    properties, 31
    run options, 32
        general, 32
        SCF, 33
    structure optimization, 35
%scan, 36
%title, 36
&plt, 234

Actual, 19
actual step
    dscf, 170
ADC(2)
    RI-, 209
analysis of normal modes

internal coordinate, 154
AOFORCE, 13, 18, 20, 26, 27, 35, 70,
    71, 77, 92, 95, 101, 153–155,
    200, 223, 224
keys, 200

B-matrix, 46
BABEL, 23
BEND, 19
Boys localization, 233

Cbasopt, 19
cc1sd, 149
cc1td, 149
cc1td-cc2-gs-1a1-001, 149
cc1td-cc2-xs-3a2-001, 149
CC2, 18
    RI-, 209
CCL0--m-ss-xxx, 146
CCLE0--m-xxx, 143
CCME0--m-xxx, 151
CCNE0--m-xxx, 147
CCRE0--m-xxx, 143
CCS
    RI-, 209
Cgnce, 20
CIS, 18
    RI-, 209
CIS(D), 18
    RI-, 209
conjugate gradients, 93
control, 17, 23–25, 37, 39, 55, 58, 63,
    90, 124, 151, 152, 159
converged, 89
Convgr, 20
coord, 23
COSMO, 20, 164–166, 197–199
    keywords, 197
COSMOPREP, 20, 198
counterpoise calculation, 55
CPHF, 156

DEFINE, 17, 21, 23–27, 30, 32, 37–39,
<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>degrees of freedom</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>dens</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>DIIS</td>
<td>93, 218</td>
<td></td>
</tr>
<tr>
<td>Dist</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>dos_a+b</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>dos_a-b</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>dos_alpha</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>dos_beta</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>edens</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>EGRAV</td>
<td>13, 19, 25, 27, 88, 95, 125, 128–130, 133, 134, 159–161, 206, 223, 230, 231</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>energy</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>ESCE</td>
<td>13, 18, 25–27, 68, 125, 128–134, 203, 205, 206</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finit</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>FREEH</td>
<td>20, 154</td>
<td></td>
</tr>
<tr>
<td>FROG</td>
<td>18, 27, 88, 106, 236, 237, 239–241</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>geometry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>manipulation of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAD</td>
<td>17, 18, 25, 27–29, 65, 72, 88, 95, 98, 99, 106–111, 134, 146, 151, 197, 199, 206, 223, 225, 244, 245</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gradients</td>
<td></td>
<td></td>
</tr>
<tr>
<td>excited states</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ground state</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hcore</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>HOLUMO</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Infrared Spectra</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>internal coordinates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>linear combination of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>manual definition of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>types of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intersections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>job.&lt;cycle&gt;</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>job.last</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>job.start</td>
<td>89</td>
<td></td>
</tr>
<tr>
<td>JOBEX</td>
<td>19, 20, 22, 24, 30, 40, 66, 69, 87, 89, 91, 103, 126, 134, 136, 137, 212, 224, 245</td>
<td></td>
</tr>
<tr>
<td>JOBEX</td>
<td>109</td>
<td></td>
</tr>
<tr>
<td>jobex</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-c</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-dscf</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-energy</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-ex</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-gcart</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-grad</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-keep</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-l</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-level</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-ls</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-md</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-mdfile</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>-mdmaster</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>
-relax, 87
-ri, 87
-statpt, 87
-trans, 87
KDG, 20
kinetic energy, 240
KONTO, 20
lalp, 233
lbet, 233
Leapfrog Verlet algorithm, 106, 237
LHFPREP, 20, 194
LHFPREP, 194
lmo, 233
LOG2EGY, 20
LOG2X, 20
mdens, 149
mdlog, 106
mdmaster, 236
mdmaster, 106
MDPREP, 20, 106, 236, 237
mdprep, 236
menu
  atomic attributes, 51, 54
  general, 63, 64
  geometry main, 40
  geometry menu, 42
  internal coordinate, 45, 46
  occupation number assignment, 59, 60
  start vectors, 56, 57
molecular dynamics, 106, 236
MLOCH, 19, 20, 26, 27, 78, 81, 82, 84, 85, 146, 159, 226
  keywords, 226
MLOCH2, 20
MP2
  RI-, 209
MP2PREP, 20, 24, 25, 122
MPGRAD, 13, 17, 18, 25, 27, 28, 69, 88, 95, 99, 119–123, 158–161,
  170, 180, 189, 197, 199, 207, 223, 231, 243, 245
  turbo.preproc, 28
  keywords, 206
MPSHIFT, 13, 19, 24, 26, 27, 156–158, 243
  keywords, 241
no weight derivatives, 203
not.converged, 89
NUMFORCE, 199
NUMFORCE, 18–20, 126, 134, 136, 137, 153, 154, 199
OUTP, 21
parms.in, 177
plotting data
  keywords, 231
population analysis, 233
  excited states, 147
  ground state, 144
q, 37
quasi–Newton, 93
Rdgrad, 17, 18, 25, 27–29, 66, 88, 95, 98, 107–110, 134, 192, 197, 199, 244, 245
  keywords, 199
RELAX, 13, 18–20, 27, 32, 33, 41, 46, 72–74, 87, 88, 93–95, 97, 98, 100–102, 216, 218, 220–222, 224, 225
  keywords, 216
restart.cc, 211
RI-ADC(2), 209
RI-CC2, 18, 209
  keywords, 209
RI-CCS, 209
RI-CIS, 18, 209
RI-CIS(D), 18, 209
RI-MP2, 209
INDEX

Ricc2, 12, 13, 17, 19, 25–27, 29, 53, 54, 69, 135–142, 144, 147, 149–152, 159, 161, 162, 209, 223
keywords, 209
keywords, 180
keywords, 206
Rimp2prep, 21, 24, 25, 121, 123, 208, 209
Roothaan parameters, 62

Schakal, 21
Screwer, 21, 92
Sdg, 21
Simulated Annealing, 241
Stati, 21, 109
Statpt, 18, 27, 32, 33, 67, 87–92
keywords, 225
steepest descent, 93
STOP, 89
stop, 89
structure library, 43
structure optimization, 87
substitution, 43
Sysname, 21, 22, 274, 276

T2s, 21
T2x, 21, 160, 161
Tblist, 21, 276
Tbtim, 21, 276
temperature, 240
time, 106, 237
timestep, 237
tm2molden, 21, 161
TMOLE, 30, 31, 35, 268
input, 30

TORS, 21
Ttest, 274–277
turbo.in, 30, 35
turbo_start, 21
Turbomole
installation, 22
modules, 17
quotations of, 12
tools, 19
Turbotest, 22
twoint, 24, 29

UFF, 17, 35, 91, 103, 175, 177, 271
keywords, 175
uffgradient, 103
uffhessian0-0, 103
ufftopology, 103, 177, 178, 180
nxtn12, 177
UHFUSE, 21

vector
function, 139
velocity, 240
vibration, 153
Vibrational Frequencies, 153

wave function analysis
keywords, 231

x2t, 21, 23, 39
XMOL, 20