ORCA

- An ab *initio*, DFT and semiempirical SCF-MO package -

Version 3.0

Design and Scientific Directorship:
Frank Neese
Technical Directorship:
Frank Wennmohs

Max-Planck-Institute for Chemical EnergyConversion
Stiftstr. 34-36, 45470 Mülheim a. d. Ruhr, Germany

tccec@mpi-mail.mpg.de

With contributions from:
Ute Becker, Dmytro Bykov, Dmitry Ganyushin, Andreas Hansen, Róbert Izsák, Dimitrios G. Liakos, Christian Kollmar, Simone Kossmann, Dimitrios A. Pantazis, Taras Petrenko, Christoph Reimann, Christoph Riplinger, Michael Roemelt, Barbara Sandhöfer, Igor Schapiro, Kantharuban Sivalingam, Boris Wezisla
And contributions from our collaborators:
Mihály Kállay, Stefan Grimme, Edward Valeev, Garnet Chan

Input Description

Contents

List of Figures

List of Tables

Foreword 2013 (Version 3.0)

The ORCA project now looks back onto a history of almost 15 years of development. Indeed, there have been so many additions and major changes since the last released version (2.9.1), that we decided to give the new version the label 3.0. We feel that with this release the program has reached the next level of its development. The development team has worked very hard during the last 18 months to make this release happen, and we are proud to pass the results of our efforts on to the fast growing ORCA user community. In fact, the estimated number of ORCA users worldwide has now surpassed 10,000. The ORCA forum, which was initiated two years ago, after the development team moved to the Max Planck Institute for Chemical Energy Conversion, is highly active and we enjoy seeing so much activity from users worldwide. And of course, we also enjoy witnessing the growing number of scientific publications that make use of ORCA. Presently an estimated 1500 publications report results from ORCA calculations and this number is growing very quickly. We therefore feel that one of the missions of the project, to deeply anchor the ORCA code in academic research, has been achieved. We also look forward to welcoming many more future users.

Since its shy beginnings in 1999, the code has seen dramatic changes and improvements, as is witnessed by about 1.5 million lines of source code. We have recently surpassed the number of 4000 submissions into our source code repository. While it is evident that every project of this size is difficult to maintain, we put major efforts into consolidating the code and creating an infrastructure that allows for efficient modifications and extensions in the future. It is pleasing, in this respect, that the rules that we laid upon ourselves at the start of the project still stand unaltered today. I am particularly grateful to the development team that the rules are observed with a great deal of discipline. We therefore look with considerable excitement and optimism to the further development of ORCA. Now that the code is hosted at the MPI-CEC, the environment is ideally set up for a fruitful long-term effort.

Users of ORCA 3.0 will (hopefully) notice considerable performance improvements over previous versions. There are several reasons for this. We are foremost indebted to our collaborator, Ed Valeev from Virginia Tech University, for continuously improving the libint integral package that is now used by default everywhere in ORCA. The present libint version exploits savings arising from basis set contraction. Furthermore, the interface between libint and ORCA has been greatly streamlined, such that overhead arising from redundant arithmetic or storage operations has been eliminated. The parallelization of ORCA is now complete and has been continuously improved. Most noticeably, we make use of parallel linear algebra to realize significant speedups. This is perhaps most noticeable in large-scale RI-DFT calculations that tended to be diagonlization bound in the past. Truly giant calculations with more than 20000 basis functions are now a routine exercise.

A major new functionality in ORCA is the development of a large-scale, almost linear scaling local coupled cluster method (DLPNO-CCSD(T)). We are very proud of this development and have managed to report CCSD(T) level calculations on entire proteins with more than 650 atoms and almost 10000 basis functions. The new opportunities that arise from this development are enormous. Hence, we look forward to the creative use of this method by the ORCA community. So far, the method is limited to closed shell energies but the

development team is working hard on extensions. Another major feature in ORCA 3.0 is the implementation of the ROCIS/DFT method, which we consider a breakthrough technology in the calculation of transition metal L-edge spectra. Furthermore, ORCA 3.0 features explicitly correlated MP2 and CCSD(T) methods – implemented, again, in close collaboration with Ed Valeev. Thanks to Garnet Chan and his co-workers, ORCA is now able to perform DMRG calculations and also to use DMRG as a substitute for configuration interaction in the CASSCF module. Stefan Grimme and his team have made many important contributions to the density functional capabilities of ORCA 3.0, including nonlocal, self-consistent van der Waals interactions, counterpoise corrected minimal basis set Hartree-Fock methods and many modern density functionals, to name only a few. Last but not least – ORCA 3.0 features analytic second derivatives for Hartree-Fock, DFT and MP2 methods. It took us a considerable amount of time to arrive there, but the final result was well worth the effort. The analytic Hessian module executes very efficiently and fully exploits the RI and COSX approximations. Again, we are indebted to Ed Valeev for implementing the indispensible second derivative two-electron integrals into libint.

We are highly grateful to all of our collaborators who have contributed their expertise and enthusiasm to the ORCA project. In addition to the groups already mentioned, we are highly grateful to Marcel Nooijen who spent a sabbatical with us in 2012/2013. We have greatly profited from the many discussions with him and the outcome of our collaborative efforts will likely be seen in the forthcoming ORCA release. Likewise Jiri Pittner and his co-workers and Nicolas Ferré have initiated coding projects within ORCA that we are excited about and which we hope to implement in the forthcoming ORCA release.

No acknowledgement could be complete without expressing my heartfelt thanks to the in-house ORCA development team: Frank Wennmohs, Simone Kossmann, Kantharuban Sivalingam and Ute Becker all have invested countless hours in code debugging, streamlining, testing, improving, advising students, answering questions in the forum, solving hardware issues ... All developers and users live off the hard work of this team, who have paved the way for future developments and have made sure that ORCA remains in a healthy state. We are all grateful to the Ph.D. student and postdocs in Mülheim, who have devoted their skills, energy, enthusiasm and time (... together with a lot of coffee ;) to push the further development of ORCA. The results of all these efforts are documented in the publications of our research group and we hope that they will find the attention that, we feel, they deserve.

Even in version 3.0, ORCA is still a program that is free of charge for the academic user community. If you, as an academic user, make use of ORCA and appreciate it, then **PLEASE** "pay us back" by referencing the relevant original literature from our research group. This ensures that we can document the importance of our developments and raise the funds that we need to continue the development.

We sincerely hope that with the release of ORCA 3.0 you will be able to push your research to the next level. We look very forward to read about your results in the literature and discuss your insights together with other users in our forum. If you are an ambitious method developer and feel that you want to join the team, we look forward to your application!

Frank Neese, on behalf of the ORCA development team, August 24th 2013

A final word to the users of this program package:
ORCA is a strictly non-commercial computer program. You can count on the fact that it will remain free of charge in the future. The best way to show your appreciation of our efforts is to cite our
original research and development papers in your own work. This will provide us with the possibility of raising sufficient funding for continuing to do what we enjoy most - develop quantum chemistry to our mutual benefit!

Frank Neese, 2013

1 General Information

1.1 Program Components

The program system ORCA consists of several separate programs that call each other during a run. The following basic modules are included in this release:

orca	:	Main input+driver program.
orca_anoint	:	Integral generation over ANOs
orca_ciprep	:	Preparation of data for MRCI calculations (frozen core matrices and the like).
orca_cis	:	Excited states via CIS and TD-DFT.
orca_cpscf	:	Solution of the coupled-perturbed SCF equations.
orca_casscf	:	Main program for CASSCF driver
orca_eprnmr	:	SCF approximation to EPR and NMR parameters.
orca_fci	:	Full-CI program
orca_gtoint	:	Calculation of gaussian integrals.
orca_gstep	:	Relaxation of the geometry based on energies and gradients.
orca_loc	:	Calculation of localized molecular orbitals.
orca_md	:	Molecular dynamics program.
orca_mdci	:	Matrix driven correlation program. CI,CEPA,CPF,QCISD,CCSD(T)
orca_mp2	:	MP2 program (conventional, direct and RI).
orca_mrci	:	MRCI and MRPT calculations (individually selecting).
orca_ndoint	:	Calculates semiempirical integrals and gradients.
orca_numfreq	:	Numerical hessian computation.
orca_pc	:	Addition of point charge terms to the one-electron matrix.
orca_plot	:	Generation of orbital and density plots.
orca_pop	:	External program for population analysis on a given density.
orca_rel	:	(Quasi) Relativistic corrections.
orca_rocis	:	Excited states via the ROCIS method
orca_scf	:	Self consistent field program (conventional and direct).
orca_scfgrad	:	Analytic derivatives of SCF energies (HF and DFT).
orca_scfhess	:	Analytical hessian calculation for SCF
orca_soc	:	Calculation of spin-orbit coupling matrices.
orca_vpot	:	Calculation of the electrostatic potential on a given molecular surface.

Utility programs:

orca_asa	:	Calculation of absorption, fluorescence and resonance Raman spectra.
orca_vib	:	Calculation of vibrational frequencies from a completed frequency run (also used for isotope shift calculations).
orca_pltvib	:	Produces files for the animation of vibrations.
orca_2mkl	:	Produces an ASCII file to be read by molekel, molden or other visualization programs.
orca_2aim	:	Produces WFN files suitable for AIM analysis.
orca_fitpes	:	Simple program to fit potential energy curves of diatomics.
orca_mapspc	:	Produces files for transfer into plotting programs.
orca_euler	:	Calculate Euler angles from .prop file
orca_chelpg	:	Electrostatic potential derived charges.
otool_gcp	:	Geometrical Counterpose Correction

Friends of ORCA:

gennbo	:	The NBO analysis package of Weinhold (must be purchased separately from the university of Wisconsin; older versions available for free on the internet may also work).
otool_cosmo	:	This module performs different steps of the COSMO solvation procedure. It has been developed jointly between COSMOlogic (special thanks to Drs Andreas Klamt and Michael Diedenhofen) and FN.
Molekel	:	Molecular visualization program (see 7.20.2.3).
gOpenMol	:	Molecular visualization program (see 7.20.2.2).

In principle every individual module can also be called 'standalone'. However, it is most convenient to do everything via the main module.

There is no real installation procedure. Just copy the executables wherever you want them to be and make sure that your path variable contains a reference to the directory where you copied the files. This is important to make sure that the programs can call each other (but you can also tell the main program the explicit position of the other programs in the input file as described below).

1.2 Units and Conversion Factors

Internally the program uses atomic units. This means that the unit of energy is the Hartree (E_h) and the unit of length is the Bohr radius (a_0). The following conversion factors to other units are used:

$$
\begin{array}{lll}
1\ E_h & = & 27.2113834\ \text{eV} \\
1\ \text{eV} & = & 8065.54477\ \text{cm}^{-1} \quad = 23.0605\ \frac{kcal}{mol} \\
1\ \text{cm}^{-1} & = & 29979.2458\ \text{MHz} \\
1\ a_0 & = & 0.5291772083\ \text{Å} \\
1\ \text{a.t.u.} & = & 2.4188843\ 10^{-17}\ \text{s}
\end{array}
$$

2 Calling the Program (Serial and Parallel)

2.1 Calling the Programm

Under Windows the program is called from the command prompt! (Make sure that the PATH variable is set such that the orca executables are visible)

```
orca  MyMol.inp > MyMol.out
```

Under UNIX based operating systems the following call is convenient[1] (here also: make sure that the PATH variable is set to the directory where the orca executables reside):

```
nohup orca  MyMol.inp >& MyMol.out &
```

The 'nohup' command lets the program run even if the user is logged out. The program writes to *stdout* and *stderr*. Therefore the output must be redirected to the file *MyMol.out* in this example. *MyMol.inp* is a free format ASCII file that contains the input description. The program will produce a number of files *MyMol.x.tmp* and the file *MyMol.gbw*. The '*.gbw' file contains a binary summary of the calculation. GBW stands for 'Geometry-Basis-Wavefunction'. Basically this together with the calculation flags is what is stored in this file. You need this file for restarting SCF calculations or starting other calculations with the orbitals from this calculation as input. The '*.tmp' files are temporary files that contain integrals, density matrices, fock matrices etc. that are used as intermediates in the calculation. If the program exits normally all of these files are deleted. If it happens to crash you have to remove the files manually (`rm MyMol*.tmp` under Unix or `del MyMol*.tmp` under Windows). In case you want to monitor the output file while it is written, you can use the command (under Unix):

```
tail -f MyMol.out
```

to follow (option -f) the progress of the calculation. Under Windows you have to either open another command shell and use:

```
type  MyMol.out
type  MyMol.out |more
```

or you have to copy the output file to another file and then use any text editor to look at it.

---

[1]Many people (including myself) will prefer to write a small shellscript that, for example, creates a run directory, copies the input there, runs the program, deletes possibly left over temporary files and then copies the output back to the original directory.

```
    copy  MyMol.out temp.out
    edit temp.out
```

you cannot use edit `MyMol.out` because this would result in a sharing violation.

There is a parallel version for Linux computers (thanks to the work of Ms Ute Becker) which make use of OpenMPI (open-source MPI-2 implementation). Assuming that OpenMPI libraries are installed properly on your computer, it is fairly easy to run the parallel version of ORCA. You simply have to give it the number of processes, like:

```
  ! PAL4
    # everything from PAL2 to PAL8 is recognized
  or


  %pal nprocs 4 # any number (positive integer)
        end
```

The following modules are presently parallelized:

- SCF

- SCFGRAD

- SCFHESS

- CASSCF / NEVPT2

- MDCI (Canonical-, PNO-, DLPNO-Methods)

- CPSCF

- CIS/TDDFT

- MP2 and RI-MP2 (including gradient)

- EPRNMR

- SOC

- ROCIS

- PC

- MRCI

- Numerical Gradients and Frequencies

Thus, all major modules are parallelized in the present version. The efficiency is such that for RI-DFT perhaps up to 16 processors are a good idea while for hybrid DFT and Hartree-Fock a few more processors are appropriate. Above this, the overhead becomes significant and the parallelization loses efficiency. Coupled cluster calculations usually scale well up to at least 8 processors but probably it is also worthwhile to try 16. For Numerical Frequencies or Gradient runs it makes sense to use as many processors as 3*Number of Atoms. If you run a queuing system you have to make sure that it works together with ORCA in a reasonable way. NOTE:

- Parallelization is a difficult undertaking and there are many different protocols that work differently for different machines. Please understand that we can not provide a 1:1 support for each platform. We are trying our best to make the parallelization transparent and provide executables for various platforms but we can not possibly guarantee that they always work on every system. Please see the download information for details of the version.

2.2 Hints on the Use of Parallel ORCA

Many questions that are asked in the discussion forum deal with the parallel version of ORCA. Please understand that we cannot possibly provide one-on-one support for every parallel computer in the world. So please, make every effort to solve the problems locally together with your system administrator. Here are some of the most common problems and how to deal with them.

1. Parallel ORCA can be used with OpenMPI only. Please see the download information for details of the relevant OpenMPI-version for your platform.

2. The OpenMPI version is configurable in a large variety of ways, which cannot be coverd here. For a more detailed explanation of all available options, cf. http://www.open-mpi.org

Otherwise, the usage of the OpenMPI version is like the older MPICH version, especially with regard to the provision of a nodefile (<inputfile>.nodes). If you run the OpenMPI version on only one computer, you do not need to provide a machinefile, and neither have to enable an rsh/ssh access, as in this case the processes will simply be forked!

Please note that the OpenMPI version is dynamically linked, that is, it needs at runtime the OpenMPI libraries (and several other standard libraries)!

(Remember to set the LD_LIBRARY_PATH)

3. Many problems arise, because parallel ORCA doesn't find its executables. To avoid this, it's crucial to provide ORCA with its complete pathname. The easiest and safest way to do so is to include the directory with the orca-executables in your $PATH. Then start the calculation:

- interactively: start orca with full path: /mypath_orca_executables/orca MyMol.inp

- batch : export your path: export PATH=$PATH:/mypath_orca_executables (for bash) then start orca with full path: $PATH/orca $jobname.inp

This seems redundant, but it really is important if you want to start a parallel calculation to run ORCA with full path! Otherwise it will not be able to find the parallel executables.

4. It is recommended to run orca in local (not nfs-mounted) scratch-directories, (for example /tmp1, /usr/local, ...) and to renew these directories for each run to avoid confusion with left-overs of a previous run.

5. It has proven convenient to use 'wrapper' scripts. These scripts should

- set the path

- create local scratch directories

- copy input files to the scratch directory

- start orca

- save your results

- remove the scratch directory

A basic example of such a submission script for the parallel ORCA version is shown below (this is for the Torque/PBS queuing system, running on Apple Mac OS X):

```zsh
#!/bin/zsh


setopt EXTENDED_GLOB
setopt NULL_GLOB

#export MKL_NUM_THREADS=1

b=${1:r}

#get number of procs.... close your eyes... (it really works!)
if [[ ${$(grep -e '^!' $1):u} == !*(#b)PAL(<0-9>##)* ]]; then
  nprocs=$match
  let "nodes=nprocs"
elif [[ ${(j: :)$(grep -v '^#' $1):u} == *%(#b)PAL*NPROCS' '#(<0-9>##)* ]]; then
  nprocs=$match
  let "nodes=nprocs"
fi

cat > ${b}.job <<EOF
#!/bin/zsh
#PBS -l nodes=1:ppn=${nodes:=1}
#PBS -S /bin/zsh
#PBS -l walltime=8760:00:00


setopt EXTENDED_GLOB
setopt NULL_GLOB

export PATH=$PBS_O_PATH

logfile=$PBS_O_WORKDIR/$b.log
tdir=$(mktemp -d /Volumes/scratch/$USER/$b__XXXXXX)


trap '
echo "Job terminated from outer space!" >> $logfile
rm -rf $tdir
exit
' TERM
```

```
    cp $PBS_O_WORKDIR/$1 $tdir
    foreach f ($PBS_O_WORKDIR/*.gbw $PBS_O_WORKDIR/*.pot)  cp $f $tdir
    cd $tdir

    echo "Job started from $PBS_O_HOST, running on $(hostname) in $tdir using
    $(which orca)" > $log
    file
    =orca $1 1>>$logfile 2>&1

    cp ^(*.(inp|tmp*))  $PBS_O_WORKDIR/

    rm -rf $tdir


    EOF

    qsub -j oe -o $b.job.out $b.job
```

6. Parallel ORCA distinguishes 3 cases of disk availability:

   - each process works on its own (private) scratch directory (the data on this directory cannot be seen from any other process)

   - all processes work in a common scratch directory (all processes can see all file-data)

   - there are at least 2 groups of processes on different scratch directories, one of the groups consisting of more than 1 process

Parallel ORCA will find out, which case exists and handle its I/O respectively. If ORCA states disk availability differently from what you would expect, check the number of available nodes and/or the distribution pattern (fill_up/round_robin)

7. If Parallel ORCA finds a file named 'MyMol.nodes' in the directory where it's running, it will use the nodes listed in this file to start the processes on, provided your input file was 'MyMol.inp'. You can use this file as your machinefile specifying your nodes, using the usual OpenMPI machinefile notation.

8. For multi-process numerical calculations (frequencies, gradient, hybrid hessian): The processes that execute these calculations do not work in parallel, but independently, often in a totally asynchronous manner. The numerical calculations will start as many processes, as you dedicated for the parallel parts before and they will run on the same nodes. If your calculation runs on multiple nodes, you have to set the environment variable RSH_COMMAND to either "rsh" or "ssh". You may specify special flags, like "ssh -x". If RSH_COMMAND is not set, ORCA will start all processes of a multi-process run on localhost. (Take care not to exceed your localhost's ressources!) There is no gain in taking more processes than 3-times the number of atoms to be displaced. For restart (available for numerical frequencies, hybrid hessian and partial hessian calculations) make sure you have all local Hessian files (input.proc%d.hess). ORCA will check these files to determine which displacements are left to be done.

3 General Structure of the Input File

Starting from version 2.1 revision 70 of ORCA, there are two distinct input syntaxes that can be freely mixed. This section describes both of them. The 'new' syntax is provided to make things simpler and is therefore recommended.

In general, the input file is a free format ASCII file. Comments in the file start by a '#'. For example:

```
# This is a comment. Goes until the end of the line
```

Comments can also be closed by a second '#'. For example:

```
TolE=1e-5;    #Energy conv.#  TolMaxP=1e-6; #Density conv.#
```

In this example `TolE` and `TolMaxP` are two variables that can be user specified. The input consists of several blocks. Blocks consist of logically related data that can be user controlled. The program tries to choose sensible default values for all of these variables. However, it is impossible to give defaults that are equally sensible for all systems. In general the defaults are slightly on the conservative side and more aggressive cutoffs etc. can be chosen by the user and may help to speed things up for actual systems or give higher accuracy if desired.

3.1 Input Blocks

The following blocks exist:

BASIS	:	Basis sets are specified
CASSCF	:	Control of CASSCF/NEVPT2 and DMRG calculations
CIS	:	Control of CIS and TD-DFT calculations (synonym is TDDFT)
COORDS	:	Input of atomic coordinates
COSMO	:	Control of the conductor like screening model
ELPROP	:	Control of electric property calculations
EPRNMR	:	Control of SCF level EPR and NMR calculations
FREQ	:	Control of frequency calculations
GEOM	:	Control of geometry optimization
MD	:	Control of molecular dynamics simulation
LOC	:	Localization of orbitals
MDCI	:	Controls single reference correlation methods
METHOD	:	Here a computation method is specified
MP2	:	Controls the details of the MP2 calculation

MRCI	:	Control of MRCI calculations
OUTPUT	:	Control of output
PAL	:	Control of parallel jobs
PARAS	:	Input of semiempirial parameters
PLOTS	:	Control of plot generation
REL	:	Control of relativistic options
RR	:	Control of resonance Raman and absorption/fluorescence bandshape calculations
SCF	:	Control of the SCF procedure

In the old syntax, blocks start with the block name enclosed in '[ ]' and end with an '[END]'. In the new syntax blocks start with '%' and end with 'end'. For example:

```
# In the "old" syntax:
[METHOD]
  Method = HF;
[END]


# In the "new" syntax:
%method  method hf
         end
# i.e. the new syntax has "%" instead of "[]" and does not require "=" or ";"
#
# Input is NOT case sensitive
```

No blocks *need* to be present in an input file but they *can* be present if detailed control over the programs behavior is desired. Otherwise all normal jobs can be defined via the keywords described in the next section. Variable assignments have the following general structure:

```
# in the old syntax:
VariableName = Value;

# in the new syntax
VariableName Value
```

Some variables are actually arrays. In this case several possible assignments are useful:

```
# in the old syntax:
Array[1] = Value1;
Array[1] = Value1,Value2,Value3;
Array = Value1,Value2;


# in the new syntax:
Array[1]  Value1
Array[1]  Value1,Value2,Value3
Array  Value1,Value2
```

Note: Arrays always start with index 0 in ORCA (this is because ORCA is a C++ program). The first line in the example gives the value 'Value1' to Array[1], which is the *second* member of this array. The second line assigns Value1 to Array[1], Value2 to Array[2] and Value3 to Array[3]. The third line assigns Value1 to Array[0] and Value2 to Array[1]. Strings (for examples filenames) must be enclosed in quotes. For example:

```
# in the old syntax:
MOInp = "Myfile.gbw";

# in the new syntax:
MOInp  "Myfile.gbw"
```

In general the input is not case sensitive. However, inside strings the input is case sensitive. This is because on unix systems MYFILE.GBW and MyFile.gbw are different files. Under Windows the file names are not case sensitive.

3.2 Keyword Lines

In the new syntax it is possible to give a line of keywords that assign certain variables that normally belong to different input blocks. The syntax for the 'simple input' is line-oriented. Simple input starts with a '!' sign.

```
! Keywords
```

ATTENTION:

- **Starting from version 2.5.15, the keyword line is checked for correct syntax! Keywords that appear twice will produce an error!**

- **Some keywords have immediate actions that are dependent on each other. Thus, separating keywords in several lines may have 'bad' effects (for example, for scalar relativistic calculations the keywords ZORA and def2-TZVP should appear in the same line!). The ECP basis and the canonical basis should be placed in a single line for the same reason.**

3.2.1 Main Methods and Options

Table 3.1 provides a list of keywords that can be used within the 'simple input' keyword line to request specific methods and/or algorithmic options. Most of them are self-explanatory. The others are explained in detail in the section of the manual that deals with the indicated input block. Note especially that there exist compound keywords (see page 20), which are designed to make the input more compact.

Table 3.1: Main keywords that can be used in the simple input of ORCA.

Keyword	Input block	Variable	Comment
HF	METHOD	METHOD	Selects the Hartree–Fock method
DFT			Selects the DFT method (see table 3.2 on page 18 for a list of functionals)
Runtypes			

ENERGY or SP	METHOD	RUNTYP	Selects a single point calculation
OPT			Selects a geometry optimization calculation (using internal redundant coordinates)
COPT			Optimization in Cartesian coordinates (if you are desperate)
ZOPT			Optimization in Z-matrix coordinates (dangerous)
GDIIS-COPT			COPT using GDIIS
GDIIS-ZOPT			ZOPT using GDIIS
GDIIS-OPT			Normal optimization using GDIIS
ENGRAD			Selects an energy and gradient calculation
NUMGRAD			Numerical gradient (has explicitly to be asked for, if analytic gradient is not available)
FREQ			Frequencies (analytic, if available, otherwise switching to NUMFREQ)
NUMFREQ			Numerical frequencies
MD			Molecular dynamic simulation
Symmetry handling			
UseSym			Turns on the use of molecular symmetry (see section 3.3). **THIS IS VERY RUDIMENTARY!**
NoUseSym			Turns symmetry off
Second order many body perturbation theory			
MP2			Selects Method=HF and DoMP2=true
MP2RI or RI-MP2			Select the MP2-RI method
SCS-MP2			Spin-component scaled MP2
RI-SCS-MP2			Spin-component scaled RI-MP2 (synonym is SCS-RI-MP2)
OO-RI-MP2			Orbital optimized RI-MP2
OO-RI-SCS-MP2			Orbital optimized and spin-component scaled RI-MP2
MP2-F12			MP2 with F12 correction (synonym is F12-MP2)
MP2-F12-RI			MP2-RI with RI-F12 correction (synonyms are F12-RI-MP2, RI-MP2-F12)
High-level single reference methods. These are implemented in the MDCI module. They can be run in a number of technical variants.			
CCSD	MDCI	CITYPE	Coupled cluster singles and doubles
CCSD(T)			Same with perturbative triples correction
CCSD-F12			CCSD with F12 correction
CCSD(T)-F12			CCSD(T) with F12 correction
CCSD-F12/RI			CCSD with RI-F12 correction
CCSD(T)-F12/RI			CCSD(T) with RI-F12 correction
QCISD			Quadratic Configuration interaction
QCISD(T)			Same with perturbative triples correction
QCISD-F12			QCISD with F12 correction
QCISD(T)-F12			QCISD(T) with F12 correction
QCISD-F12/RI			QCISD with RI-F12 correction

QCISD(T)-F12/RI			QCISD(T) with RI-F12 correction
CPF/1			Coupled pair functional
NCPF/1			A 'new' modified coupled pair functional
CEPA/1			Coupled electron pair approximation
NCEPA/1			The CEPA analogue of NCPF/1
RI-CEPA/1-F12			RI-CEPA with F12 correction
MP3			MP3 energies
SCS-MP3			Grimme's refined version of MP3

Other coupled pair methods are available and are documented later in the manual in detail (section 7.8) In general you can augment the method with RI-METHOD in order to make the density fitting approximation operative; RI34-METHOD does the same but only for the 3- and 4-external integrals). MO-METHOD performs a full four index transformation and AO-METHOD computes the 3- and 4-external contributions on the fly. With AOX-METHOD this is is done from stored AO integrals.

Local correlation methods. These are local, pair natural orbital based correlation methods. They must be used together with auxiliary correlation fitting basis sets. Open-shell variants are available for some of the methods, for full list please see section 6.1.3. We recommend $n = 1$ for the CEPA methods.

LPNO-CEPA/n	MDCI	Various	Local pair natural orbital CEPA methods
LPNO-CPF/n			Same for coupled pair functionals
LPNO-NCEPA/n			Same for modified versions
LPNO-NCPF/n			Same for modified versions
LPNO-QCISD			Same for quadratic CI with singles and doubles
LPNO-CCSD			Same for coupled cluster theory with single and double excitations
DLPNO-CCSD			Domain based local pair natural orbital coupled cluster method with single and double excitations (closed shell only)
DLPNO-CCSD(T)			DLPNO-CCSD with perturbative triple excitations (closed shell only)

Automatic basis set extrapolation			
Extrapolate (n/m, bas)			Extrapolation of the basis set family 'bas' (bas=cc,aug-cc, ano, saug-ano, aug-ano; if omitted 'cc-pVnZ is used) for cardinal numbers n,m (n<m=2,3,4,5), e.g. Extrapolate(2/3,cc) extrapolates the SCF, MP2 and MDCI energies to the basis set limit
Extrapolate (n, bas)			Calculate the first n-energies for member of the basis set family basis, e.g. Extrapolate(3) is doing calculations with cc-pVDZ, cc-pVTZ and cc-pVQZ.
ExtrapolateEC (n/m, bas)			Similar: performs SCF, MP2 and MDCI calculations. The higher basis set is only done at MP2 and then used to extrapolate the MDCI calculation to the basis set limit.

CASSCF related options			
DMRG			Sets DMRG as 'CIStep' in CASSCF
NEVPT2			N-Electron Valence State Perturbation Theory

RI-NEVPT2			NEVPT2 with the RI approximation

Multireference methods: If specified in a single keyword all information about reference spaces, number of roots etc. is taken from the CASSCF module that is assumed to be run in advance. In general, these calculations are of the individually selecting type and are very time consuming. Very many flags can be set and modified for these methods and in general using these methods requires expert users! In general see the variables Tsel, Tpre and Tnat that define the individual selection process. All of these methods can be used with RI integrals by using RI-MRCI etc. However, then the calculations become even more time consuming since integrals are made one- by one on the fly. Non-RI calculations will be pretty much limited to about 200-300 orbitals that are included in the CI

MRCI	MRCI	CIType	Initiates a multireference configuration interaction calculation with single and double excitations
MRCI+Q			Same with multireference Davidson correction for unlinked quadruples
MRACPF			Average coupled pair functional
MRAQCC			Average quadratic coupled cluster
MRDDCI1			Difference dedicated CI with one degree of freedom
MRDDCI2			Same with two degrees of freedom
MRDDCI3			Same with three degrees of freedom
MRDDCIn+Q			MRDDCI with Davidson correction
SORCI			Spectroscopy oriented CI

Frozen core features. NOTE: this deviates from previous versions of ORCA! We are now counting core electrons rather than using an energy window. If you do want to use an orbital energy window use %method FrozenCore FC_EWIN end. Otherwise the EWin commands will be ignored! (alternatives are FC_ELECTRONS (default) and FC_NONE).

FROZENCORE	METHOD	FrozenCore	Use a frozen core. By default this is done by counting the number of chemical core electrons
NOFROZENCORE			Do not use a frozen core

Semiempirical methods

ZINDO/S			Selects the ZINDO/S method
ZINDO/1			Selects the ZINDO/1 method
ZINDO/2			Selects the ZINDO/2 method
NDDO/1			Selects the NDDO/1 method
NDDO/2			Selects the NDDO/2 method
MNDO			Selects the MNDO method
AM1			Selects the AM1 method
PM3			Selects the PM3 method

Algorithmic variations, options, add-ons, modifiers,...

RHF or RKS	SCF	HFTYP	Selects closed shell SCF
UHF or UKS			Selects spin unrestricted SCF
ROHF or ROKS			Selects open shell spin restricted SCF
AllowRHF	METHOD	ALLOWRHF	Allow a RHF calculation even if the system is open-shell (Mult>1). Default is to switch to UHF then
RI	METHOD	RI	Sets RI=true to use the RI approximation in DFT calculations. Default to Split-RI-J
NORI			Sets RI=false

RIJCOSX	METHOD/ SCF	RI, KMatrix	Sets the flag for the efficient RIJCOSX algorithm (treat the Coulomb term via RI and the Exchange term via seminumerical integration)
RI-JK	METHOD/ SCF	RI, KMatrix	Sets the flag for the efficient RI algorithm for Coulomb and Exchange. Works for SCF (HF/DFT) energies (not gradients!). Works direct or conventional.
SPLITJ	SCF	JMATRIX	Select the efficient Split-J procedure for the calculation of the Coulomb matrix in non-hybrid DFT (rarely used)
SPLIT-RI-J	SCF	JMATRIX,RI	Select the efficient Split-RI-J procedure for the improved evaluation of the RI-approximation to the Coulomb-matrix
NoSplit-RI-J	SCF	JMATRIX,RI	Turns the Split-RI-J feature off (but does not set the RI flag to false!)
RI-J-XC	SCF	JMATRIX, KMATRIX,RI	Turn on RI for the Coulomb term *and* the XC terms. This saves time when the XC integration is significant but introduces another basis set incompleteness error. (rarely used)
DIRECT SEMIDIRECT CONV	SCF	SCFMODE	Selects an integral direct calculation Selects an integral semidirect calculation (rarely used nowadays) Selects an integral conventional calculation
NOITER	SCF	MAXITER	Sets the number of SCF iterations to 0. This works together with MOREAD and means that the program will work with the provided starting orbitals.
Initial guess options: In most cases the default PMODEL guess will be adequate. In some special situations you may want to switch to a different choice			
PATOM PMODEL HUECKEL HCORE MOREAD AUTOSTART NOAUTOSTART	SCF	GUESS AUTOSTART	Selects the polarized atoms guess Select the model potential guess Selects the extended Hückel guess Selects the one-electron matrix guess Selects input from previous MOs[1] Try to start from the existing GBW file of the same name as the present one Don't try to do that
Relativistic options: There are several variants of scalar relativistic Hamiltonians to use in all electron calculations			
DKHn ($n = 1$–5)	REL	METHOD/ORDER	Selects the scalar relativistic Douglas–Kroll–Hess Hamiltonian of order n
ZORA	REL	METHOD	Selects the scalar relativistic ZORA Hamiltonian
IORA	REL	METHOD	Selects the scalar relativistic IORA Hamiltonian
ZORA/RI	REL	METHOD	Selects the scalar relativistic ZORA Hamiltonian in RI approximation

IORA/RI	REL	METHOD	Selects the scalar relativistic IORA Hamiltonian in RI approximation
IORAmm/RI	REL	METHOD	Selects the scalar relativistic IORA mm (modified metric) Hamiltonian in RI approximation
Grid options			
GRIDn ($n = 0$–7)	METHOD	GRID	Selects the DFT integration grid no n
FINALGRIDn			Selects the DFT final integration grid no n
NOFINALGRID			Turns the final grid feature off
GRIDXn ($n = 1$–9)	METHOD	GRIDX	Grids for the COSX approximation. A sequence of three grids is used. Higher accuracy at higher cost is offered by the higher grids.
NOFINALGRIDX			Turn off the final grid in COSX (not recommended)
Convergence thresholds: These keywords control how tightly the SCF and geometry optimizations will be converged. The program makes an effort to set the convergence thresholds for correlation modules consistently with that of the SCF.			
NORMALSCF	SCF	CONVERGENCE	Selects normal SCF convergence
LOOSESCF			Selects loose SCF convergence
SLOPPYSCF			Selects sloppy SCF convergence
STRONGSCF			Selects strong SCF convergence
TIGHTSCF			Selects tight SCF convergence
VERYTIGHTSCF			Selects very tight SCF convergence
EXTREMESCF			Selects "extreme" convergence. All thresholds are pratically reduced to numerical precision of the computer. Only for benchmarking (very expensive).
SCFCONVn			Selects energy convergence check and ETol $= 10^{-n}$ ($n = 6$–10). Also selects appropriate *thresh, tcut, bfcut* values.
TIGHTOPT	GEOM	TolE,TolRMSG	Selects tight optimization convergence
NORMALOPT		TolMaxG	Selects default optimization convergence
LOOSEOPT		TolRMSD,TolMaxD	Selects loose optimization convergence
Convergence acceleration: the default is DIIS which is robust. For most closed-shell organic molecules SOSCF converges somewhat better and might be a good idea to use. For "trailing convergence", KDIIS or the true second-order procedures NRSCF and AHSCF might be good choices.			
DIIS	SCF	DIIS	Turns DIIS on
NODIIS			Turns DIIS off
KDIIS	SCF	KDIIS	Turns Kollmar's DIIS on
NRSCF	SCF	NR	Turns Newton-Raphson SCF on
AHSCF			Same but with augmented Hessian step
NONRSCF			Turns Newton-Raphson SCF off
SOSCF	SCF	SOSCF	Turns SOSCF on
NOSOSCF			Turns SOSCF off
DAMP	SCF	CNVDAMP	Turns damping on
NODAMP			Turns damping off
LSHIFT	SCF	CNVSHIFT	Turns level shifting on
NOLSHIFT			Turns level shifting off

Convergence strategies (does not modify the convergence criteria)			
EasyConv			Assumes no convergence problems.
NormalConv			Normal convergence criteria.
SlowConv			Selects appropriate SCF converger criteria for difficult cases. Most transition metal complexes fall into this category.
VerySlowConv			Selects appropriate SCF converger criteria for very difficult cases.
Solvation			
COSMO(solvent) COSMO	COSMO		Invoke the conductor-like screening model with a standard solvent (see section 6.19 for a list of solvents). If no solvent is given, infinity (a conductor) is assumed.
Spin-orbit coupling			
SOMF(1X)	REL	SOCType, SOCFlags	Invokes the RI-SOMF(1X) treatment of the spin-orbit coupling operator
Miscellaneous options			
ANGS	COORDS	UNITS	Select angstrom units
BOHRS			Select input coordinates in atomic units
FRACOCC	SCF	FRACOCC	Turns the fractional occupation option on
SMEAR	SCF	SMEARTEMP	Temperature for occupation number smearing on (default is 5000 K)
NOSMEAR			Turn occupation number smearing off
KEEPINTS	SCF	KEEPINTS	Keep two electron integrals on disk
NOKEEPINTS			Do not keep two electron integrals
KEEPDENS	SCF	KEEPDENS	Keep the density matrix on disk
NOKEEPDENS			Do not keep the density matrix
READINTS	SCF	READINTS	Reading of two electron integrals on
NOREADINTS			Reading of two electron integrals off
CHEAPINTS	SCF	USECHEAPINTS	Use the cheap integral feature in direct SCF calculations
NOCHEAPINTS			Turn that feature off
FLOAT	SCF	VALFORMAT	Set storage format for numbers to single precision (SCF, RI-MP2, CIS, CIS(D), MDCI)
DOUBLE	SCF	VALFORMAT	Set storage format for numbers to double precision (default)
UCFLOAT	SCF	VALFORMAT COMPRESSION	Use float-storage in the matrix containers without data compression
CFLOAT	SCF	VALFORMAT COMPRESSION	Use float storage in the matrix containers with data compression
UCDOUBLE	SCF	VALFORMAT COMPRESSION	Use double-storage in the matrix containers without data compression
CDOUBLE	SCF	VALFORMAT COMPRESSION	Use double storage in the matrix containers with data compression
Output control			
NORMALPRINT	OUTPUT	PRINTLEVEL	Selects the normal output
MINIPRINT			Selects the minimal output
SMALLPRINT			Selects the small output
LARGEPRINT			Selects the large output

NOMOPRINT	OUTPUT		Suppress printing of MOs
PRINTBASIS	OUTPUT	Print[p_basis]	Print the basis set in input format
PRINTGAP	OUTPUT	Print[p _homolumogap]	Prints the HOMO/LUMO gap in each SCF iteration. This may help to detect convergence problems
ALLPOP	OUTPUT	Print[...]	Turns on all population analysis
NOPOP			Turns off all populaton analysis
MULLIKEN			Turns on the Mulliken analysis
NOMULLIKEN			Turns off the Mulliken analysis
LOEWDIN			Turns on the Loewdin analysis
NOLOEWDIN			Turns off the Loewdin analysis
MAYER			Turns on the Mayer analysis
NOMAYER			Turns off the Mayer analysis
NPA			Turns on interface for the NPA analysis using the GENNBO program
NBO			Turns on the interface for the NPA plus NBO analysis with the GENNBO program
NONPA			Turns off NPA analysis
NONBO			Turns of NBO analysis
PRINTMOS			Prints Loewdin reduced orb.pop per MO
NOPRINTMOS			Turns this feature off
NMOPRINT			(identical)
UNO	SCF	UNO	Produce UHF natural orbitals
AIM			Produce a WFN file
XYZFILE	OUTPUT	XYZFILE	Produce an XYZ coordinate file
PDBFILE		PDBFILE	Produce a PDB file

[1] give `%moinp "myorbitals.gbw"` in a separate line to specify where the MOs are read from.

Compression and storage. The data compression and storage options deserve some comment: in a number of modules including RI-MP2, MDCI, CIS, (D) correction to CIS, etc. the program uses so called 'Matrix Containers'. This means that the data to be processed is stored in terms of matrices in files and is accessed by a double label. A typical example is the exchange operator $\mathbf{K^{ij}}$ with matrix elements $K^{ij}(a,b) = (ia|jb)$. Here the indices i and j refer to occupied orbitals of the reference state and a and b are empty orbitals of the reference state. Data of this kind may become quite large (formally N^4 scaling). To store the numbers in single precision cuts down the memory requirements by a factor of two with (usually very) slight loss in precision. For larger systems one may also gain advantages by also compressing the data (e.g. use a 'packed' storage format on disk). This option leads to additional packing/unpacking work and adds some overhead. For small molecules UCDOUBLE is probably the best option, while for larger molecules UCFLOAT or particularly CFLOAT may be the best choice. Compression does not necessarily slow the calculation down for larger systems since the total I/O load may may be substantially reduced and thus (since CPU is much faster than disk) the work of packing and unpacking takes less time than to read much larger files (the packing may reduce disk requirements for larger systems by approximately a factor of 4 but it has not been extensively tested so far). There are many factors contributing to the overall wall clock time in such cases including the total system load. It may thus require some experimentation to find out with which set of options the program runs fastest with.

<div style="border:1px solid">

! CAUTION !

- In some isolated examples we have observed that FLOAT leads to unacceptable errors. Thus it is not recommended to use this option if MP2 or RI-MP2 gradients or relaxed densities are computed.

- For this reason the default is DOUBLE.

- If you have convinced yourself that FLOAT is OK, it may save you a factor of two in both – storage and CPU.

</div>

Global memory use. Some ORCA modules (in particular those that perform some kind of wavefunction based correlation calculations) require large scratch arrays. Each module has an independent variable to control the size of these dominant scratch arrays. However, since these modules are never running simultaneously, we provide a global variable `MaxCore` that assigns a certain amount of scratch memory to all of these modules. Thus:

```
%MaxCore 4000
```

sets 4000 MB (= 4 GB) as the limit for these scratch arrays. **This limit applies per processing core**. Do not be surprised if the program takes more than that – this size only refers to the dominant work areas. Thus, you are well advised to provide a number that is significantly less than your physical memory. Note also that the memory use of the SCF program cannot be controlled: it dynamically allocates all memory that it needs and if it runs out of physical memory you are out of luck. This, however, rarely happens unless you run on a really small memory computer or you are running a gigantic job.

3.2.2 Density Functional Methods

For density functional calculations a number of standard functionals can be selected via the 'simple input' feature. Since any of these keywords will select a DFT method, the keyword 'DFT' is not needed in the input. Further functionals are available via the `%method` block.

Table 3.2: Density functionals available in ORCA.

Local and gradient corrected functionals	
HFS	Hartree–Fock–Slater Exchange only functional
LDA or LSD	Local density approximation (defaults to VWN5)
VWN or VWN5	Vosko–Wilk–Nusair local density approx. parameter set 'V'
VWN3	Vosko–Wilk–Nusair local density approx. parameter set 'III'
PWLDA	Perdew-Wang parameterization of LDA
BP86 or BP	Becke '88 exchange and Perdew '86 correlation
BLYP	Becke '88 exchange and Lee-Yang-Parr correlation
OLYP	Handy's 'optimal' exchange and Lee-Yang-Parr correlation
GLYP	Gill's '96 exchange and Lee-Yang-Parr correlation
XLYP	The Xu and Goddard exchange and Lee-Yang-Parr correlation
PW91	Perdew-Wang '91 GGA functional
mPWPW	Modified PW exchange and PW correlation
mPWLYP	Modified PW exchange and LYP correlation

PBE	Perdew-Burke-Erzerhoff GGA functional
RPBE	'Modified' PBE
REVPBE	'Revised' PBE
PWP	Perdew-Wang '91 exchange and Perdew '86 correlation
Hybrid functionals	
B1LYP	The one-parameter hybrid functional with Becke '88 exchange and Lee-Yang-Parr correlation (25% HF exchange)
B3LYP and B3LYP/G	The popular B3LYP functional (20% HF exchange) as defined in the TurboMole program system and the Gaussian program system, respectively
O3LYP	The Handy hybrid functional
X3LYP	The Xu and Goddard hybrid functional
B1P	The one-parameter hybrid version of BP86
B3P	The three-parameter hybrid version of BP86
B3PW	The three-parameter hybrid version of PW91
PW1PW	One-parameter hybrid version of PW91
mPW1PW	One-parameter hybrid version of mPWPW
mPW1LYP	One-parameter hybrid version of mPWLYP
PBE0	One-parameter hybrid version of PBE
PW6B95	Hybrid functional by Truhlar
BHANDHLYP	Half-and-half hybrid functional by Becke
Meta-GGA and hybrid meta-GGA functionals	
TPSS	The TPSS meta-GGA functional
TPSSh	The hybrid version of TPSS (10% HF exchange)
TPSS0	A 25% exchange version of TPSSh that yields improved energetics compared to TPSSh but is otherwise not well tested
M06L	The Minnesota M06-L meta-GGA functional
M06	The M06 hybrid meta-GGA (27% HF exchange)
M062X	The M06-2X version with 54% HF exchange
Perturbatively corrected double hybrid functionals	
B2PLYP	The new mixture of MP2 and DFT from Grimme
RI-B2PLYP	B2PLYP with RI applied to the MP2 part
B2PLYP-D	B2PLYP with Grimme's empirical dispersion correction from 2006 (D2) [1]
B2PLYP-D3	B2PLYP with Grimme's atom-pairwise dispersion correction from 2010 [2] and Becke-Johnson damping (D3BJ)
RI-B2PLYP RIJONX	The same but with RI also applied in the SCF part
mPW2PLYP	mPW exchange instead of B88 (also with RI and RIJONX as above for B2PLYP). mPW is supposed to improve on weak interactions.
mPW2PLYP-D	mPW2PLYP with Grimme's empirical dispersion correction from 2006 (D2)
mPW2PLYP-D3	mPW2PLYP with Grimme's atom-pairwise dispersion correction from 2010 and Becke-Johnson damping (D3BJ)
B2GP-PLYP	Gershom Martin's 'general purpose' reparameterization
B2K-PLYP	Gershom Martin's 'kinetic' reparameterization
B2T-PLYP	Gershom Martin's 'thermochemistry' reparameterization
PWPB95	Hybrid functional with MP2 mixture from Grimme
RI-PWPB95	PWPB95 with RI for the MP2 part
Dispersion corrections (see 5.1.4.6 and 6.2.2.10 for details)	

D3BJ	Atom-pairwise dispersion correction to the DFT energy with Becke-Johnson damping (recommended)
D3ZERO	Atom-pairwise dispersion correction with zero damping
D2	Empirical dispersion correction from 2006 (not recommended)

3.2.3 Predefined Procedures

There are several keywords that are designed to make your life much easier because they take multiple actions. Note that the following keywords assume DFT; no post-HF procedures are standardized yet.

Table 3.3: Keywords for automated procedures.

Predefined procedures	
Quick-DFT	Defines a calculation with a small basis set (DefBas-1) and the BP86 functional (used for survey calculations)
Quick-Opt	Defines a 'quick' optimization run with a small basis set, the RI approximation and the BP86 functional (DefBas-1; no polarization functions in the basis and pretty much SV quality)
Medium-Opt	The same with a somewhat larger basis (DefBas-2; including polarization functions)
Good-Opt	The same with an even larger basis (DefBas-3)
Acc-Opt	The same with a large basis (DefBas-4)
DFT-Energy	Good quality single-point calculation at the B3LYP level with a large basis set (DefBas-4)
DFT-Energy+	The same with an even larger basis set (DefBas-5)
Default basis sets	*(note: includes an appropriate Coulomb fitting basis)*
DefBas-n	Selects default-basis set n with an appropriate auxiliary basis set (Synonyms are DefBas1, DefBas/1, Def1, Def-1 and Def/1). Presently $n = 1$ to $n = 6$ are defined and provide basis sets of increasing quality (see section 6.2.2.9)

3.2.4 Basis Sets

There are standard basis sets that can be specified via the 'simple input' feature in the keyword line. See the BASIS input block for a full description of each basis set (section 6.3). See also the ORCA default basis sets at the end of this section (fully described in section 6.2.2.9) that simultaneously define reasonable basis sets together with auxiliary Coulomb fitting bases.

You can request decontraction of the basis with the keyword `Decontract`. This will decontract *both* the atomic basis set and the auxiliary fitting basis (if it exists). Decontraction *always* gives more accurate results but may also lead to *much* more expensive calculations since the size of the matrices to be handled becomes fairly large. Note that if your basis set arises from general contraction it will contain duplicate primitives in several contractions and these will be removed such that only unique primitives remain and there is no problem with redundancy.

Table 3.4: Basis sets and related keywords.

ECP{...}	Effective core potentials and their use are described in section 3.2.5

Decontract	Decontract the basis set
Uncontract	Identical
NoDecontract	Prevent decontraction
NoUncontract	Identical
DecontractAux	Decontract the auxiliary basis set
UnContractAux	Identical
NoDecontractAux	Prevent decontraction of the auxiliary basis
NoUncontractAux	Identical
Pople-style basis sets	
3-21G	Pople 3-21G
3-21GSP	Buenker 3-21GSP
4-22GSP	Buenker 4-22GSP
6-31G	Pople 6-31G and its modifications
6-311G	Pople 6-311G and its modifications
Polarization and diffuse functions for the Pople basis sets	
* and ** =(d) and (d,p)	One set of first polarization functions on all atoms except hydrogen or on all atoms including hydrogen
(2d) and (2d,2p)	Two sets of polarization functions on all atoms except hydrogen or on all atoms including hydrogen
(2df) and (2df,2pd)	Three sets of polarization functions on all atoms except hydrogen or on all atoms including hydrogen
(3df) and (3df,3pd)	Four sets of polarization functions on all atoms except hydrogen or on all atoms including hydrogen
+ and ++ before "G"	Put diffuse functions on all atoms but hydrogen or on all atoms including hydrogens (e.g. 6-31+G)
Dunning basis sets	
cc-pVDZ	Dunning correlation concisistent polarized double zeta
cc-(p)VDZ	Same but no polarization on hydrogens
aug-cc-pVDZ	Same but including diffuse functions
cc-pVTZ	Dunning correlation concisistent polarized triple zeta
cc-(p)VTZ	Same but no polarization on hydrogen
aug-cc-pVTZ	Same but including diffuse functions. Note: due to software limitations higher than g-functions had unfortunately to be deleted in the present version!
cc-pVQZ	Dunning correlation concisistent polarized quadruple zeta
aug-cc-pVQZ	. . . with diffuse functions
cc-pV5Z	Dunning correlation concisistent polarized quintuple zeta
aug-cc-pV5Z	. . . with diffuse functions
cc-pV6Z	Dunning correlation concisistent polarized sextuple zeta
aug-cc-pV6Z	. . . with diffuse functions
cc-pCVDZ	Core-polarized double-zeta correlation consistent basis set
cc-pCVTZ	Same for triple zeta
cc-pCVQZ	Same for quadruple zeta
cc-pCV5Z	Same for quintuple zeta
cc-pV6Z	Same for sextuple zeta
aug-pCVDZ	Same double zeta with diffuse functions augmented
aug-pCVTZ	Same for triple zeta
aug-pCVQZ	Same for quadruple zeta
aug-pCV5Z	Same for quintuple zeta

aug-cc-pV6Z	Same for sextuple zeta
DUNNING-DZP	Dunning's original double zeta basis set
Ahlrichs basis sets	
DZ	Ahlrichs double-zeta basis set
VTZ	Ahlrichs valence triple-zeta basis set
SV or VDZ	Ahlrichs split valence basis set
TZV	Ahlrichs triple-zeta valence basis set. NOT identical to VTZ. Note that TZV rather than VTZ is used in the TurboMole program system.
QZVP	Ahlrichs quadruple-zeta basis set. Makes no sense with other polarization functions. "P" is already extensively polarized.
QZVPP(-g,-f)	Slightly stripped down version of QZVPP with higest angular momentum polarization functions deleted (identical to def2-QZVPP(-g,-f))
Polarization functions for Ahlrichs basis sets	
P and (P)	One set of first polarization functions on all atoms including or not including hydrogens
(d),(d,p)	(Identical meaning)
(2D) and (2D,2P)	Two sets of first polarization functions on all atoms including or not including hydrogens
(2d), (2d,2p)	(Identical meaning)
PP and (PP)	Three sets of first polarization functions on all atoms including or not including hydrogens.
(2df), (2df,2pd)	(Identical meaning)
PPP	Extended set of poloization functions. For transition metals this amounts to two p and one f function. For main group elements to much larger poarlization set.
Diffuse functions for Ahlrichs basis sets	
+	In combination with SV(P) or TZV(P) or TZV(2D) this adds Pople diffuse functions on all non-hydrogens (e.g. SV(P)+).
++	In combination with SVP, TZVP and TZV(2D) this adds Pople diffuse functionss on all atoms (e.g. SVP++).
aug	In combination with SVP,SV(P),TZVP ancd TZV(P) this adds the diffuse functions from the Dunning aug-cc-pVDZ basis (e.g. aug-TZVP).
	In combination with TZV(2D),TZV(2D,2P), TZVPP, TZV(PP), TZV(PPP) and TZVPPP this adds the diffuse functions from the Dunning aug-cc-pVTZ basis (e.g. aug-TZVPP).
The def2 basis sets of the Ahlrichs group (2005/2006)	
Def2-SV(P)	SV basis set with "new" polarization functions.
Def2-SVP	
Def2-TZVP	TZVP basis set with "new" polarization functions. Note that this is quite similar to TZVPP for the main group elements and TZVP for hydrogen.
Def2-TZVP(-f)	Delete the f-polarization functions from def2-TZVP for the main group elements. Pretty much like the older TZV(2d,p).
Def2-TZVP(-df)	Also delete the double d-function and replace it by the older single d-function. This is pretty much like the old TZVP = TZV(d,p) basis set.
Def2-TZVPP	TZVPP basis set with "new" polarization functions.

Def2-QZVPP	Very accurate quadruple-zeta basis. Even more polarized than QZVP.
Def2-QZVPP(-g,-f)	Slightly stripped down version of QZVPP with higest angular momentum polarization functions deleted.
Diffuse def2 Ahlrichs basis sets	
Minimally augmented def2 basis sets	Augmented def2 basis sets by diffuse s and p functions according to Truhlar [3]. Combine with ECP{def2-SD} for non-relativistic calculations on heavy elements. Use ma-ZORA-def2-XVP for all-electron ZORA calculations or ma-DKH-def2-XVP for DKH calculations.
ma-def2-SVP	Minimally augmented def2-SVP basis set.
ma-def2-TZVP	Minimally augmented def2-TZVP basis set.
ma-def2-TZVPP	Minimally augmented def2-TZVPP basis set.
ma-def2-QZVPP	Minimally augmented def2-QZVPP basis set.
Rappoport property-optimized diffuse def2 basis sets	Augmented def2 basis sets by diffuse functions according to Rappoport et al. [4]
def2-SVPD	Diffuse def2-SVP basis set for property calculations
def2-TZVPD	Diffuse def2-TZVP basis set for property calculations
def2-TZVPPD	Diffuse def2-TZVPP basis set for property calculations
def2-QZVPD	Diffuse def2-QZVP basis set for property calculations
def2-QZVPPD	Diffuse def2-QZVPP basis set for property calculations
Def2-aug-TZVPP	def2-TZVPP basis set with diffuse functions from aug-cc-pVTZ.
Jensen basis bets	
PC-1,2,3,4	Polarization consistent basis sets 1, 2, 3, 4 (H-Ar) of double, triple, quadruple and quintuple zeta type. These basis sets are optimized for SCF (DFT) calculations.
Aug-PC-1,2,3,4	Polarization consistent basis sets 1, 2, 3, 4 (H-Ar) augmented by diffuse functions.
Atomic Natural Orbital basis sets	
ano-pVnZ	(n = D, T, Q, 5). Our newly contracted ANO basis sets on the basis of the cc-pV6Z (or pc-4 where missing) primitives. These are very accurate basis sets that are significantly better than the cc-pVnZ counterparts for the same number of basis functions (but much larger number of primitives of course).
saug-ano-pVnZ	(n = D, T, Q) augmentation with a single set of sp functions. Greatly enhances the accuracy of the SCF energies but not for correlation energies.
aug-ano-pVnZ	(n = D, T, Q) full augmentation with spd, spdf, spdfg set of polarization functions. Almost as expensive as the next higher basis set. In fact, aug-ano-pVnZ = ano-pV$(n+1)$Z with the highest angular momentum polarization function deleted.
BNANO-DZP	(synonym Bonn-ANO-DZP). A small DZP type ANO basis set from the Bonn group. It is based on the TZV primitives with more contracted polarization functions.
BNANO-TZ2P	(synonym Bonn-ANO-TZ2P). A slightly larger triple-zeta ANO basis set with two d-polarization sets from the Bonn group. Based on the same primitives (both from TZV).

BNANO-TZ3P	Same but with a contracted set of f-polarization functions on the heavy atoms.
NASA-AMES-ANO	The original NASA/AMES ANO basis set (quadruple-zeta).
BAUSCHLICHER ANO	First row transition metal ANO sets.
ROOS-ANO-DZP	A fairly large DZP basis set from Roos. It is of the same size as aug-ano-pVDZ.
ROOS-ANO-TZP	A fairly large TZP basis from Roos. It is of the same size as aug-ano-pVTZ.
F12 and F12-CABS basis sets	
cc-pVnZ-F12	(n = D, T, Q) Special orbital basis sets for F12 calculations (larger than the regular D, T, Q-zeta basis sets!)
cc-pVnZ-F12-CABS	(n = D, T, Q) Near-complete auxiliary basis sets for F12 calculations.
Miscellaneous and specialized basis sets	
DGAUSS or DZVP-DFT	DGauss polarized valence double zeta basis set.
SADLEJ-PVTZ	Sadlej's polarized triple zeta basis for polarizability and related calculations.
EPR-II	Barone's Basis set for EPR calculations (double zeta).
EPR-III	Barone's Basis set for EPR calculations (triple-zeta).
IGLO-II	Kutzelnigg's basis set for NMR and EPR calculations.
IGLO-III	Kutzelnigg's basis set for NMR and EPR calculations (accurate).
Partridge-1,2,3	Accurate uncontracted basis sets.
Wachters	Good first row transition metal basis set.

Scalar relativistic basis sets. For DKH and ZORA calculations ORCA provides relativistically recontracted versions of the SV, TZV and QZV basis sets (and their polarized counterparts) for elements up to Xe. Note that for other non-relativistic basis sets (for example Pople-style bases) no recontraction has been performed and consequently such calculations are inconsistent! The standard non-relativistic Ahlrichs basis sets are automatically replaced by their recontracted versions if *both* the basis set and the scalar relativistic Hamiltonian are specified in *the same* keyword line, for example:

```
! B3LYP ZORA TZVP ...
```

For more detailed control over basis set specification, see section 6.3. Using the same input as above, third-row transition metals, lanthanides, actinides and 6p elements are automatically treated by our SARC (segmented all-electron relativistically contracted) basis sets [5–9]. These were specifically developed for scalar relativistic calculations and are individually adapted to the DKH2 and ZORA Hamiltonians.

Auxiliary basis sets. Auxiliary basis sets for the RI-J and RI-MP2 approximations can also be specified directly in the simple input:

Table 3.5: Overview of auxiliary basis sets available in ORCA.

Auxiliary basis sets for Coulomb fitting	
AutoAux	Automatic construction of a general purpose fitting basis
DEMON/J	The DeMon/J Coulomb fitting basis
DGAUSS/J	The DGauss A1 Coulomb fitting basis
SV/J (=VDZ/J)	Ahlrichs Coulomb fitting basis for the SVP basis
TZV/J (=VTZ/J)	Ahlrichs Coulomb fitting basis for the TZV or TZVP basis

QZVPP/J	Ahlrichs Coulomb fitting for the QZVPP basis
Def2-SVP/J	Ahlrichs Coulomb fitting for def-SVP
Def2-TZVPP/J	Ahlrichs Coulomb fitting for def2-TZVPP/J
Def2-QZVPP/J	Ahlrichs Coulomb fitting for def2-QZVPP/J
SV/J(-f)	Same as SV/J but with the highest angular momentum function deleted
TZV/J(-f)	Same as TZV/J, with the highest angular momentum function deleted
Auxiliary basis sets for correlation calculations (all from the Ahlrichs group)	
AutoAux	Automatic construction of a general purpose fitting basis
SV/C (=VDZ/C)	Correlation fitting basis for MP2-RI with SVP
TZV/C (=VTZ/C)	Correlation fitting basis for MP2-RI with TZVP
TZVPP/C (=VTZPP/C)	Correlation fitting basis for MP2-RI with extended triple-ζ bases
QZVP/C	Correlation fitting for the QZVP basis
QZVPP/C	Correlation fitting for the QZVPP basis
Def2-SVP/C	Correlation fitting for the def2-SVP basis
Def2-TZVP/C	Correlation fitting for the def2-TZVP basis
Def2-TZVPP/C	Correlation fitting for the def2-TZVPP basis
Def2-QZVPP/C	Correlation fitting for the def2-QZVPP basis
cc-pVDZ/C	Aux-basis for the cc-pVDZ orbital basis
cc-pVTZ/C	Aux-basis for the cc-pVTZ orbital basis
cc-pVQZ/C	Aux-basis for the cc-pVQZ orbital basis
cc-pV5Z/C	Aux-basis for the cc-pV5Z orbital basis
cc-pV6Z/C	Aux-basis for the cc-pV6Z orbital basis
aug-cc-pVDZ/C	Aux-basis for the aug-cc-pVDZ orbital basis
aug-cc-pVTZ/C	Aux-basis for the aug-cc-pVTZ orbital basis
aug-SV/C	Aux basis for SVP and related bases but with diffuse functions
aug-TZV/C	Aux basis for TZVP and related bases but with diffuse functions
aug-TZVPP/C	Aux basis for TZVPP and related bases but with diffuse functions
Auxiliary basis sets for simultaneously fitting Coulomb and exchange	
Fitting basis sets developed by Weigend for fitting simultaneously Coulomb and exchange energies. They are quite large and accurate. They fit SCF energies very well but even if they are large they do not fit correlation as well as the dedicated '/C' auxiliary basis sets.	
SVP/JK	Coulomb+Exchange fitting for SVP
TZVPP/JK	Coulomb+Exchange fitting for TZVPP
QZVPP/JK	Coulomb+Exchange fitting for QZVPP
Def2-SVP/JK	Coulomb+Exchange fitting for def2-SVP
Def2-TZVPP/JK	Coulomb+Exchange fitting for def2-TZVPP
Def2-QZVPP/JK	Coulomb+Exchange fitting for def2-QZVPP
cc-pVDZ/JK	Coulomb+Exchange fitting for cc-pVDZ
cc-pVTZ/JK	Coulomb+Exchange fitting for cc-pVTZ
cc-pVQZ/JK	Coulomb+Exchange fitting for cc-pVQZ
cc-pV5Z/JK	Coulomb+Exchange fitting for cc-pV5Z
cc-pV6Z/JK	Coulomb+Exchange fitting for cc-pV6Z

NOTE: Starting from version 2.6.63 ORCA can deal with two auxiliary basis sets – one for Coulomb and one for correlation. The default is the Coulomb fitting aux-basis. If you select a separate correlation fitting basis, the correlation modules (RI-MP2, RI-MDCI, RI-MRCI) will replace this Coulomb fitting aux-basis with the correlation fitting basis. In order to use this feature you have to give the 'C'-fit basis via the 'simple input' lines.

3.2.5 Effective Core Potentials

Starting from version 2.8.0, ORCA features effective core potentials (ECPs). They are a good alternative to scalar relativistic all-electron calculations if heavy elements are involved. This pertains to geometry optimizations and energy calculations but may not be true for property calculations.

In order to reduce the computational effort, the usually highly contracted and chemically inert core basis functions can be eliminated by employing ECPs. ECP calculations comprise a 'valence-only' basis and thus are subject to the frozen core approximation. Contributions due to the core orbitals are accounted for by an effective one-electron operator U^{core} which replaces the interactions between core and valence electrons and accounts for the indistinguishability of the electrons. Its radial parts $U_l(r)$ are generally expressed as a linear combination of Gaussian functions, while the angular dependence is included through angular momentum projectors $|S_m^l\rangle$.

$$U^{core} = U_L(r) + \sum_{l=0}^{L-1} \sum_{m=-l}^{l} \left| S_m^l \right\rangle [U_l(r) - U_L(r)] \left\langle S_m^l \right|$$

$$U_l = \sum_k d_{kl} r^{n_{kl}} \exp(-\alpha_{kl} r^2)$$

The maximum angular momentum L is generally defined as $l_{max}^{atom} + 1$. The parameters n_{kl}, α_{kl} and d_{kl} that are necessary to evaluate the ECP integrals have been published by various authors, among them the well-known Los Alamos (LANL) [10] and Stuttgart–Dresden (SD) [11–56] parameter sets. Depending on the specific parametrization of the ECP, relativistic effects can be included in a semiempirical fashion in an otherwise nonrelativistic calculation. Introducing U^{core} into the electronic Hamiltonian yields two types of ECP integrals, the local (or type-1) integrals that arise because of the maximum angular momentum potential U_L and the semi-local (or type-2) integrals that result from the projected potential terms. The evaluation of these integrals in ORCA proceeds according to the scheme published by Flores-Moreno et al. [57].

A selection of ECP parameters and associated basis sets is directly accessible in ORCA through the internal ECP library (see table 3.6 for a listing of keywords).

Table 3.6: Overview of library keywords for ECPs and associated basis sets available in ORCA.

ECP keyword	Appendant ECP[1]	Elements	Valence basis sets
Stuttgart–Dresden ECPs			
def2-SD	SD(28,MWB)	Rb–Cd	available for Rb–La, Hf–Rn:
	SD(28,MDF)	In–Xe[2]	
	SD(46,MWB)	Cs–La	def2-SV(P), def2-SVP, def2-TZVP,
	SD(60,MWB)	Hf–Pt	def2-TZVPP, def2-QZVP, def2-QZVPP
	SD(60,MDF)	Au–Rn[3]	
			auxiliary basis sets are available and denoted as
			def2-SVP/J, def2-SVP/C, def2-SVP/JK etc.
def-SD	SD(28,MWB)	Rb–Cd	
	SD(46,MWB)	In–La	available for Rb–Lr:
	SD(28,MWB)	Ce–Lu	
	SD(60,MWB)	Hf–Pt	def-SV(P), def-SVP, def-TZVP,
	SD(60,MDF)	Au,[3] Hg,[3] Rn	def-TZVPP, def-QZVP, def-QZVPP
	SD(78,MWB)	Tl–At	
	SD(78,MDF)	Fr, Ra	auxiliary basis sets are available and denoted as
	SD(60,MWB)	Ac–Lr	def-SVP/J, def-SVP/C, def-SVP/JK etc.

SDD	SD(2,SDF)	Li, Be	SD(2,SDF)
	SD(2,MWB)	B–Ne	SD(2,MWB)
	SD(10,SDF)	Na, Mg	SD(10,SDF)
	SD(10,MWB)	Al–Ca	SD(10,MWB)
	SD(10,MDF)	Sc–Zn	RSC[4]
	SD(28,MWB)	Ga–Cd	Ga–Sr: SD(28,MWB), Y–Cd: SD(28,MHF)
	SD(28,MDF)	Ge–Br, Rb–Xe	SD(28,MDF)
	SD(46,MWB)	In–Ba	SD(46,MWB)
	SD(28,MWB)	La–Lu	SD(28,MWB)-SEG
	SD(60,MWB)	Hf–Hg	RSC[5]
	SD(78,MWB)	Tl–Rn	SD(78,MWB)
	SD(60,MWB)	Ac–Lr	RSC[6]
Los Alamos ECPs			
LANL2	LANL(10)	K–Cu	LANL-type basis sets are available only for the elements given in parentheses!
	LANL(28)	Rb–Ag	
	LANL(46)	Cs–La	
	LANL(60)	Hf–Au	
LANL1	LANL(10)	Na–Ar	LANLDZ (Na–La, Hf–Bi),
	LANL(18)	K–Zn	LANL08 (Na–La, Hf–Bi),
	LANL(28)	Ga–Kr	LANL08(d) (Ge–Br, Sn–I, Pb, Bi),
	LANL(36)	Rb–Cd	LANL08(f) (Sc–Cu, Y–Ag, La, Hf–Au),
	LANL(46)	In–Xe	LANL08+ (Sc–Cu),
	LANL(54)	Cs–La	LANLTZ (Sc–Zn, Y–Cd, La, Hf–Hg),
	LANL(68)	Hf–Tl	LANLTZ(f) (Sc–Cu, Y–Ag, La, Hf–Au),
	LANL(78)	Pb, Bi	LANLTZ+ (Sc–Zn)
LANL	LANL2	K–Cu, Rb–Ag, Cs–La, Hf–Au	LANL automatically assigns the following basis sets:
	LANL1	Na–Ca, Zn–Sr, Cd–Ba, Tl–Bi	LANLTZ for transition metals, LANL08 for main group elements

[1] The ECP names in the second column of the table consist of a token (SD, LANL), followed by the number of replaced core electrons in parentheses and, if necessary, tags for the reference method and data (S: single-valence-electron ion; M: neutral atom; HF: Hartree–Fock; WB: quasi-relativistic; DF: relativistic). [2] I: OLD-SD(28,MDF) due to compatibility with TURBOMOLE.

[3] Au, Hg: OLD-SD(60,MDF) due to compatibility with TURBOMOLE.

[4] Sc–Ni: truncated and modified SD(10,MDF) basis set. Cu, Zn: truncated SD(10,MWB) basis set.

[5] Truncated SD(60,MWB) basis set.

[6] Consider using SD(60,MHF), which has the same contraction scheme.

The simplest way to assign ECPs and associated valence basis sets is by using the `ECP{}` statement within the short keyword line. Of the keywords listed in the first column of table 3.6, only two of them (SDD and LANL) assign at the same time the appropriate ECP for each element as listed in the table, together with the corresponding valence basis set (e.g. `ECP{SDD}`). The other four keywords (DEF2-SD, DEF-SD, LANL2 and LANL1) only select ECP parameters but do not assign valence basis sets; these have to be specified explicitly within the `ECP{}` keyword following the ECP declaration (e.g. `ECP{def2-SD,def2-TZVP}`). An appropriate auxiliary basis set can also be explicitly specified within the `ECP{}` keyword as an additional argument (e.g. `ECP{def2-SD,def2-TZVP,def2-TZVP/J}`).

To facilitate the assignment of ECPs, ORCA offers convenient ways to load default combinations of high-quality Stuttgart–Dresden ECPs and appropriate basis sets. Thus, the statement

```
    ! ECP{def2-TZVP} ...
```

assigns a def2-TZVP basis set and loads the respective "def2-SD" ECPs automatically for the elements Rb–Rn (but omitting the lanthanides, see table 3.6 for keywords and scopes). Specifically, the command loads the SD(28,MWB) ECPs for Rb–Cd, SD(28,MDF) for In–Xe, etc (these ECPs have been tagged as "def2-SD"). For the lanthanides and actinides only "def" ECPs and basis sets are available at the moment.

To further ease the use of def/def2 ECPs and the respective basis sets, one can omit the def/def2 qualifier and simply write SV(P), SVP, TZVP, TZVPP, QZVP and QZVPP within the ECP{} keyword. The program then loads def2 parameters for Rb–La and Hf–Rn, and def ECPs and basis sets for Ce–Lu and Fr–Lr.

The ECP{} keyword further allows to load ECPs and basis sets only for a small selection of elements, defined either by a range or by a list:

```
    # Loads def2-SD ECPs and associated def2-TZVP valence basis sets for all 5d elements.
    #
    ! ECP{TZVP=Hf-Hg}
```

```
    # Does the same only for the three elements listed within the square brackets.
    #
    ! ECP{TZVP=[Ag,Pt,Au]}
```

NOTE:

- **All-electron basis sets must be explicitly assigned to all atoms that are not affected by the ECP statement! Contrary to some other program packages, all-electron basis sets e.g. on lighter atoms are not loaded automatically!**

Basis sets for light elements are specified as usual: in the following example the def2-TZVP keywords within the ECP statement affect only Au, while the def2-TZVP keywords outside the ECP statement refer to Cl. **It is important that both the keywords appear in the same line.**

```
    ! BP ECP{def2-TZVP,def2-TZVP/J} def2-TZVP def2-TZVP/J TightSCF

    * xyz -2 2
    Au  0     0     0
    Cl  2.5   0     0
    Cl -2.5   0     0
    Cl  0     2.5   0
    Cl  0    -2.5   0
    *
```

As another example, for a first-row transition metal that would not normally be affected by the ECP statement, one may use:

```
    #
    # NOTE: ECP{SDD} or ECP{LANL} would assign ECPs to both Cu and Cl
    #
    ! BP ECP{SDD=Cu,def2-TZVP/J} def2-TZVP def2-TZVP/J TightSCF


    * xyz -2 2
    Cu  0      0      0
    Cl  2.25   0      0
    Cl -2.25   0      0
    Cl  0      2.25   0
    Cl  0     -2.25   0
    *
```

If ECPs are to be used for the lighter atoms as well, then the ECP{SDD} keyword can be helpful. It employs large-core ECPs on main group elements in combination with basis sets of double zeta quality and small-core ECPs on transition metals and f elements together with basis sets of at least triple zeta quality. This comes close to what is obtained in a major commercial quantum chemistry program. However, in our opinion its use should be deferred in favor of the more systematic Ahlrichs-type basis sets in conjunction with the def-/def2-SD potentials.

The library also contains the Los Alamos (LANL) ECP parameters. Small-core ECPs are only available for the transition metals and are denoted as LANL2, while large-core potentials (LANL1) exist for Na-Bi with the exception of the lanthanides. For the LANL potentials, three families of valence basis sets exist:

- The often used basis sets of double zeta quality [10] are denoted LANLDZ in ORCA

- LANLTZ and variants thereof are recontractions of the original Hay-Wadt primitives [58] and exist only for the transition metals

- LANL08 and variants thereof are completely uncontracted basis sets [58] and should be preferentially used for the main group elements

The directive ECP{LANL} automatically loads LANL2 potentials if they are present and falls back to LANL1 otherwise. For the transition metals the LANLTZ basis sets are loaded, while the LANL08 basis sets are used for all main group elements.

NOTE:

- In some other program packages, the LANL-type ECPs and basis sets are all prefixed with the token LANL2. This is different in ORCA where they consistently carry the LANL prefix, whereas LANL2 explicitly references only the small-core potentials.

For further details in advanced specification of ECPs, including assigning ECPs to individual atoms, manually entering core potential parameters, and using ECPs as embedding potentials in cluster calculations, see sections 6.3.6 and 6.3.7.

3.3 ORCA and Symmetry

For most of its life, ORCA did not take advantage of molecular symmetry. Starting from version 2.8.0 there is at least limited use. On request, with the `UseSym` keyword, the program detects the point group, cleans up the coordinates, orients the molecule and produces symmetry-adapted orbitals in SCF/CASSCF calculations. Note however that the calculation time will not be reduced. Only D_{2h} and subgroups are currently supported. The only correlation module that makes use of this information so far is the MRCI module. Here and in CASSCF calculations, the use of symmetry helps to control the calculation and the interpretation of the results. More symmetry is likely to be implemented in the future, although it is unlikely that the program will ever take advantage of symmetry in a very big way.

If the automatic symmetry detection fails to find the expected point group, the coordinates specified are not absolutely symmetrical to that group, and one should take a careful look at the input coordinates, maybe using a visualization program. A problem often encountered when using coordinates generated from other jobs (e.g. geometry optimizations) is the detection of a "too low" symmetry because of numerical noise. This can be solved by increasing the detection threshold using an input line which looks like this:

```
%method SymThresh 5.0e-2 end
```

However, it is not recommended to run calculations on a very high threshold, since this may introduce some odd behavior. Instead, a method to symmetrize the coordinates is to do a "fake" run with `NoIter`, `XYZFile` and an increased threshold, and then to use the created `.xyz` file as input for the actual calculation. This has the additional benefit that the input coordinates stored in your data are already symmetrical. To give an example: the following coordinates for staggered ethane were obtained by geometry optimization NOT using the symmetry module. They are, however, not recognized as D_{3d} symmetrical due to numerical noise and instead are found to be of C_i symmetry (a subgroup of D_{3d}). To counter this, the detection threshold is increased and a symmetry perfected coordinate file is produced by the following input:

```
! RHF SVP UseSym NoIter XYZfile
%method SymThresh 1.0e-2 end
*xyz 0 1
  C    -0.002822   -0.005082   -0.001782
  C    -0.723141   -1.252323   -0.511551
  H     0.017157    0.029421    1.100049
  H     1.042121    0.030085   -0.350586
  H    -0.495109    0.917401   -0.350838
  H    -0.743120   -1.286826   -1.613382
  H    -0.230855   -2.174806   -0.162495
  H    -1.768085   -1.287489   -0.162747
*
```

3.4 Jobs with Multiple Steps

ORCA supports input files with multiple jobs. This feature is designed to simplify series of closely related calculations on the same molecule or calculations on different molecules. The objectives for implementing this feature include:

- Calculate of a molecular property using different theoretical methods and/or basis sets for one molecule.

- Calculations on a series of molecules with identical settings.

- Geometry optimization followed by more accurate single points and perhaps property calculations.

- Crude calculations to provide good starting orbitals that may then be used for subsequent calculations with larger basis sets.

For example consider the following job that in the first step computes the g-tensor of BO at the LDA level, and in the second step using the BP86 functional.

```
# -------------------------------------------------------
! UKS LSD SVP grid5 NoFinalGrid TightSCF KeepInts
# -------------------------------------------------------
%eprnmr gtensor 1 end

* int 0 2
   B  0  0  0   0      0  0
   O  1  0  0   1.2049  0  0
*


# ************************************************
# ****** This starts the input for the next job  *
# ************************************************
$new_job
# -------------------------------------------------------
! BP86 SVP SmallPrint ReadInts NoKeepInts
# -------------------------------------------------------
%eprnmr gtensor 1 end

* int 0 2
   B  0  0  0   0      0  0
   O  1  0  0   1.2049  0  0
*
```

What happens if you use the $new_job feature is that all calculation flags for the actual job are transferred from the previous job and that only the changes in the settings must be input by the user. Thus if you turn on some flags for one calculation that you do not want for the next, you have to turn them off again yourself (for example the use of the RI approximation)! In addition, the default is that the new job takes the orbitals from the old job as input. If you do not want this you have to overwrite this default by specifying your desired guess explicitly.

Changing the default BaseName

Normally the output files for MyJob.inp are returned in MyJob.xxx (any xxx, for example xxx=out). Sometimes, and in particular in multistep jobs, you will want to change this behavior. To this end there is the variable '%base' that can be user controlled. All filenames (also scratch files) will then be based on this default name.

4 Input of Coordinates

Coordinates can either be specified directly in the input file or read from an external file. There are two ways to specify coordinates in the input file. One of them is within the `%coords` block, which is organized as follows:

```
%coords
 CTyp   xyz  # the type of coordinates = xyz or internal
 Charge 0    # the total charge of the molecule
 Mult   2    # the multiplicity = 2S+1
 Units  Angs # the unit of length = angs or bohrs
 # the subblock coords is for the actual coordinates
 # for CTyp=xyz
  coords
     Atom1   x1  y1  z1
     Atom2   x2  y2  z2
  end
 # for CTyp=internal
  coords
     Atom1  0  0  0    0.0   0.0   0.0
     Atom2  1  0  0    R1    0.0   0.0
     Atom3  1  2  0    R2    A1    0.0
     Atom4  1  2  3    R3    A2    D1
      . . .
     AtomN  NA NB NC   RN    AN    DN
  end
end
```

The input of Cartesian coordinates in the 'xyz' option is straightforward. Each line consists of the label for a given atom type and three numbers that specify the coordinates of the atom. The units can be either Ångström or Bohr depending on the variable `Units` in the `%coords` main block. The default is to specify the coordinates in Ångströms.

Internal coordinates are specified in the form of the familiar "Z-matrix". A Z-matrix basically contains information about molecular connectivity, bond lengths, bond angles and dihedral angles. The program then constructs Cartesian coordinates from this information. Both sets of coordinates are printed in the output such that conversion between formats is facilitated. The rules for connectivity in the 'internal' mode (see above example) are as follows:

- `NA`: The atom that the actual atom has a distance (`RN`) with.

- `NB`: The actual atom has an angle (`AN`) with atoms `NA` and `NB`.

- `NC`: The actual atom has a dihedral angle (`DN`) with atoms `NA,NB` and `NC`. This is the angle between the actual atom and atom `NC` when looking down the `NA-NB` axis.

Angles are always given in degrees! The rules are compatible with those used in the well known MOPAC and ADF programs.

Besides the `%coord` block, there is a slightly simpler input syntax of the following form:

```
* CTyp Charge Multiplicity
  ... atoms as in the coord subblock
*
```

For example for CO^+ in a $S = 1/2$ state (multiplicity $= 2 \times 1/2 + 1 = 2$)

```
* xyz 1 2
  C  0 0 0
  O  0 0 1.1105
*
```

It is also possible to read the coordinates from a `.xyz` file, which can in principle contain more than one structure (see section 5.2.9 for this multiple XYZ feature):

```
* xyzfile 1 2 mycoords.xyz
```

If multiple jobs are specified in the same input file then new jobs can read the coordinates from previous jobs. If no filename is given as fourth argument then the name of the actual job is automatically used.

```
... specification for the first job

$new_job
! keywords
* xyzfile 1 2
```

In this way, optimization and single point jobs can be very conveniently combined in a single, simple input file. Examples are provided in the following sections.

Special cases

- **Dummy atoms** are defined in exactly the same way as any other atom, by using "DA" as the atomic symbol.

- **Ghost atoms** are specified by adding ":" right after the symbol of the element (see 5.1.6).

- **Point charges** are specified with the symbol "Q", followed by the charge (see 6.1.3).

- **Embedding potentials** are specified by adding a ">" right after the symbol of the element (see 6.3.7).

- **Non-standard** isotopes or nuclear charges are specified with the statements "M = ..." and "Z = ...", respectively, after the atomic coordinate definition.

- **Fragments** can be conveniently defined by declaring the fragment number a given atom belongs to in parentheses "(n)" following the element symbol (see 6.1.1).

5 Running Typical Calculations

Before entering the detailed documentation of the various features of ORCA it is instructive to provide a chapter that shows how 'typical' tasks may be performed. This should make it easier for the user to get started on the program and not get lost in the details of how-to-do-this or how-to-do-that. We hope that the examples are reasonably intuitive.

5.1 Single Point Energies and Gradients

5.1.1 Hartree-Fock

5.1.1.1 Standard Single Points

In general single point calculations are fairly easy to run. What is required is the input of a method, a basis set and a geometry. For example, in order run a single point Hartree-Fock calculation on the CO molecule with the SVP basis set type:

```
#
# My first ORCA calculation :-)
#
! HF SVP
* xyz 0 1
  C  0  0  0
  O  0  0  1.13
*
```

As an example consider this simple calculation on the cyclohexane molecule that may serve as a prototype for this type of calculation.

```
# Test a simple direct HF calculation
! HF RHF SV(P)
* xyz 0 1
C   -0.79263     0.55338    -1.58694
C    0.68078     0.13314    -1.72622
C    1.50034     0.61020    -0.52199
C    1.01517    -0.06749     0.77103
C   -0.49095    -0.38008     0.74228
C   -1.24341     0.64080    -0.11866
H    1.10490     0.53546    -2.67754
```

```
    H    0.76075   -0.97866   -1.78666
    H   -0.95741    1.54560   -2.07170
    H   -1.42795   -0.17916   -2.14055
    H   -2.34640    0.48232   -0.04725
    H   -1.04144    1.66089    0.28731
    H   -0.66608   -1.39636    0.31480
    H   -0.89815   -0.39708    1.78184
    H    1.25353    0.59796    1.63523
    H    1.57519   -1.01856    0.93954
    H    2.58691    0.40499   -0.67666
    H    1.39420    1.71843   -0.44053
    *
```

5.1.1.2 Basis Set Options

There is extensive flexibility in the specification of basis sets in ORCA. First of all, you are not only restricted to the basis sets that are built in ORCA, but can also read basis set definitions from files. In addition there is a convenient way to change basis sets on certain types of atoms or on individual atoms. Consider the following example:

```
! UHF HF
%basis  basis  SV
        newGTO Cl "DUNNING-DZP" end
        end
* xyz -2 2
  Cu  0     0     0  newGTO "TZVPP" end
  Cl  2.25  0     0
  Cl -2.25  0     0
  Cl  0     2.25  0
  Cl  0    -2.25  0
*
```

In this example the basis set is initialized as the Ahlrichs split valence basis. Then the basis set on all atoms of type Cl is changed to DUNNING-DZP and finally the basis set for only the copper atom is changed to the more accurate TZVPP set. In this way you could treat different atom types or even individual groups in a molecule according to the desired accuracy. Similar functionality regarding per-element or per-atom assignments exists for effective core potentials. More details are provided in section 6.3.

Sometimes you will like to change the ordering of the starting orbitals to obtain a different electronic state in the SCF calculation. For example, if we take the last input and want to converge to a ligand field excited state this can be achieved by:

```
! UHF HF SV
%basis  newGTO Cl "Dunning-DZP" end
        end
```

```
%scf    rotate {48, 49, 90, 1, 1} end
        end
* xyz -2 2
  Cu  0     0     0    newGTO "TZVPP" end
  Cl  2.25  0     0
  Cl -2.25  0     0
  Cl  0     2.25  0
  Cl  0    -2.25  0
*
```

In the present case, MO 48 is the spin-down HOMO and MO49 the spin-down LUMO. Since we do a calculation on a Cu(II) complex (d^9 electron configuration) the beta LUMO corresponds with the 'SOMO'. Thus, by changing the SOMO we proceed to a different electronic state (in this case the one with the 'hole' in the 'd$_{xy}$' orbital instead of the 'd$_{x^2-y^2}$' orbital). The interchange of the initial guess MO's is achieved by the command `rotate {48, 49, 90, 1, 1}` end. What this does is the following: take the initial guess MOs 48 and 49 and rotate them by an angle of 90 degree (this just interchanges them). The two last numbers mean that both orbitals are from the spin-down set. For RHF or ROHF calculations the operator would be 0. In general you would probably first take a look at the initial guess orbitals before changing them.

5.1.1.3 SCF and Symmetry

Upon request, the SCF program produces symmetry adapted orbitals. This can help to converge the SCF on specific excited states of a given symmetry. Take for example the cation H_2O^+: We first run the simple job:

```
! SVP UseSym

* xyz 1 2
  O      0.000000    0.000000     0.068897
  H      0.000000    0.788011    -0.546765
  H      0.000000   -0.788011    -0.546765
*
```

The program will recognize the C$_{2v}$ symmetry and adapt the orbitals to this:

```
------------------
SYMMETRY DETECTION
------------------
Preparing Data                  ... done
Detection Threshold:   SymThresh ... 1.0000e-04

Point Group will now be determined:
Moving molecule to center of mass ... done

POINT GROUP                     ... C2v

The coordinates will now be cleaned:
Moving to standard coord frame    ... done
```

```
     (Changed main axis to z and one of the symmetry planes to xz plane)
     Structure cleanup requested       ... yes
     Selected point group              ... C2v
     Cleaning Tolerance      SymThresh ... 1.0000e-04

     Some missing point group data is constructed:
     Constructing symmetry operations  ... done
     Creating atom transfer table      ... done
     Creating asymmetric unit          ... done

     Cleaning coordinates              ... done


     ----------------------------------------------
     SYMMETRY-PERFECTED CARTESIAN COORDINATES (A.U.)
     ----------------------------------------------

        0 O     0.00000000   0.00000000   0.13019595
        1 H     0.00000000   1.48912498  -1.03323662
        2 H     0.00000000  -1.48912498  -1.03323662


     ------------------
     SYMMETRY REDUCTION
     ------------------
     ORCA supports only abelian point groups.
     It is now checked, if the determined point group is supported:
     Point Group ( C2v   ) is          ... supported

     (Re)building abelian point group:
     Creating Character Table          ... done
     Making direct product table       ... done


     ---------------------
     ASYMMETRIC UNIT IN C2v
     ---------------------

       # AT     MASS            COORDS                    BAS
       0 O    15.9990   0.00000000   0.00000000   0.13019595   0
       1 H     1.0080   0.00000000   1.48912498  -1.03323662   0


     ---------------------
     SYMMETRY ADOPTED BASIS
     ---------------------
     The coefficients for the symmetry adopted linear combinations (SALCS)
     of basis functions will now be computed:
     Number of basis functions         ...    24
     Preparing memory                  ... done
     Constructing Gamma(red)           ... done
     Reducing Gamma(red)               ... done
     Constructing SALCs                ... done
     Checking SALC integrity           ... nothing suspicious
     Normalizing SALCs                 ... done

     Storing the symmetry object:
     Symmetry file                     ... Test-SYM-H2O+.sym.tmp
     Writing symmetry information       ... done
```

The initial guess in the SCF program will then recognize and freeze the occupation numbers in each irreducible representation of the C_{2v} point group.

```
    The symmetry of the initial guess is 2-B1
    Irrep occupations for operator 0
      A1 -    3
      A2 -    0
      B1 -    1
      B2 -    1
    Irrep occupations for operator 1
      A1 -    3
      A2 -    0
      B1 -    0
      B2 -    1
```

The calculation converges smoothly to

```
    Total Energy      :           -75.56349710 Eh          -2056.18729 eV
```

With the final orbitals being:

```
    SPIN UP ORBITALS
     NO   OCC         E(Eh)          E(eV)    Irrep
      0   1.0000     -21.127827      -574.9174   1-A1
      1   1.0000      -1.867576       -50.8193   2-A1
      2   1.0000      -1.192139       -32.4397   1-B2
      3   1.0000      -1.124657       -30.6035   1-B1
      4   1.0000      -1.085062       -29.5260   3-A1
      5   0.0000      -0.153303        -4.1716   4-A1
      6   0.0000      -0.071324        -1.9408   2-B2
    ...
                 SPIN DOWN ORBITALS
     NO   OCC         E(Eh)          E(eV)    Irrep
      0   1.0000     -21.081198      -573.6486   1-A1
      1   1.0000      -1.710193       -46.5367   2-A1
      2   1.0000      -1.152855       -31.3708   1-B2
      3   1.0000      -1.032556       -28.0973   1-B1
      4   0.0000      -0.306683        -8.3453   3-A1
      5   0.0000      -0.139418        -3.7937   4-A1
      6   0.0000      -0.062261        -1.6942   2-B2
      7   0.0000       0.374727        10.1968   3-B2
    ...
```

Suppose now that we want to converge on an excited state formed by flipping the spin-beta HOMO and LUMO that have different symmetries.

```
    ! SVP UseSym
    ! moread
    %moinp "Test-SYM-H2O+.gbw"
    %scf rotate {3,4,90,1,1}
             end
         end
```

```
   * xyz 1 2
     O      0.000000     0.000000     0.068897
     H      0.000000     0.788011    -0.546765
     H      0.000000    -0.788011    -0.546765
   *
```

The program now finds:

```
   Irrep occupations for operator 0
      A1 -     3
      A2 -     0
      B1 -     1
      B2 -     1
   Irrep occupations for operator 1
      A1 -     2
      A2 -     0
      B1 -     1
      B2 -     1
```

And converges smoothly to

```
   Total Energy     :          -75.48231924 Eh          -2053.97833 eV
```

Which is obviously an excited state of the H_2O^+ molecule. In this situation (and in many others) it is an advantage to have symmetry adapted orbitals.

5.1.2 MP2

5.1.2.1 MP2 and RI-MP2 Energies

You can do conventional or integral direct MP2 calculations for RHF, UHF or high-spin ROHF reference wavefunctions. For RHF and UHF, the analytic gradients are available

```
   MP2 RHF TZVP TightSCF
   %mp2    MaxCore 100
           end
   %paras  rCO  =   1.20
           ACOH = 120
           rCH  =   1.08
           end
   * int 0 1
    C(1)  0 0 0 0.00   0.0      0.00
    O(2)  1 0 0 {rCO}  0.0      0.00
    H(3)  1 2 0 {rCH} {ACOH}    0.00
    H(3)  1 2 3 {rCH} {ACOH}    180.00
   *
```

This example also shows how to define geometry parameters and how to divide the molecule into fragments. It is of course not necessary to do this in order to perform a MP2 calculation.

NOTE:

- There are two algorithms for MP2 calculations without the RI approximation. The first one uses main memory as much as possible. The second one uses more disk space and is usually faster (in particular, if you run the calculations in single precision using ! FLOAT, UCFLOAT or CFLOAT). The memory algorithm is used using Q1Opt >0 and the disk based algorithm uses Q1Opt = -1. Gradients are only available presently for the memory based algorithm.

The RI approximation to MP2 is fairly easy to use too. For example consider the same calculation as before:

```
# only the auxiliary basis set TZV/C is added to
# the keyword line
#
! MP2 RHF TZVP TZV/C TightSCF
%mp2     MaxCore 100
         RI      true
         end
%paras   rCO  =   1.20
         ACOH = 120
         rCH  =   1.08
         end
* int 0 1
 C(1)  0 0 0 0.00   0.0     0.00
 O(2)  1 0 0 {rCO}  0.0     0.00
 H(3)  1 2 0 {rCH} {ACOH}   0.00
 H(3)  1 2 3 {rCH} {ACOH}   180.00
*
```

The MP2 module can also do Grimme's spin-component scaled MP2 [59]. This is a semi-empirical modification of MP2 which, however, gives a fair bit better results than MP2 itself.

```
#
# Spin-component scaled MP2 example
#
! SCS-MP2 RHF TZV(2d,2p) TightSCF
%paras   rCO  =   1.20
         ACOH = 120
         rCH  =   1.08
         end
* int 0 1
 C(1)  0 0 0 0.00   0.0     0.00
 O(2)  1 0 0 {rCO}  0.0     0.00
 H(3)  1 2 0 {rCH} {ACOH}   0.00
 H(3)  1 2 3 {rCH} {ACOH}   180.00
*
```

Energy differences with SCS-MP2 appear to be much better than from MP2 itself according to Grimme's detailed evaluation study.

5.1.2.2 Orbital Optimized MP2 Methods

By making the Hylleraas functional stationary with respect to the orbital rotations one obtains the orbital-optimized MP2 method that is implemented in ORCA in combination with the RI approximation (OO-RI-MP2). One obtains from these calculations orbitals that are adjusted to the dynamic correlation field at the level of second order many-body perturbation theory. Also, the total energy of the OO-RI-MP2 method is lower than that of the RI-MP2 method itself. One might think of this method as a special form of multiconfigurational SCF theory except for the fact that the Hamiltonian is divided into a 0th order term and a perturbation.

The main benefit of the OO-RI-MP2 method is that it 'repairs' the poor Hartree–Fock orbitals to some extent which should be particularly beneficial for systems which suffer from the inbalance in the Hartree-Fock treatment of the Coulomb and the Exchange hole. Based on the experience gained so far, the OO-RI-MP2 method is no better than RI-MP2 itself for the thermochemistry of organic molecules. However, for reactions barriers and radicals the benefits of OO-MP2 over MP2 are substantial. This is particularly true with respect to the spin-component scaled variant of OO-RI-MP2 that is OO-RI-SCS-MP2. Furthermore, the OO-RI-MP2 method substantially reduces the spin contamination in UHF calculations on radicals.

Since every iteration of the OO-MP2 method is as expensive as a RI-MP2 relaxed density calculation, the computational cost is much higher than for RI-MP2 itself. One should estimate about a factor of 10 increase in computation time with respect to the RI-MP2 time of a normal calculation. This may still be feasible for calculations in the range of 1000–2000 basis functions (the upper limit, however, implies very significant computational costs).

A full assessment of the orbital optimized MP2 method can be found in the following paper:

> Neese, F.; Schwabe, T.; Kossmann, S.; Schirmer, B.; Grimme, S. (**2009**) Assessment of Orbital Optimized, Spin-Component Scaled Second Order Many Body Perturbation Theory for Thermochemistry and Kinetics. *J. Chem. Theory Comput.*, 5, 3060-3073

The OO-RI-MP2 method comes with new variables:

```
%mp2 OrbOpt   true   # turns on the orbital optimization
     CalcS2   false  # calculate the S**2 expectation value
                     # in spin-unrestricted calculations
     MaxOrbIter 64   # Max. number of iterations
     MP2Shift  0.1   # Level shift for the procedure
     end
```

The solver is a simple DIIS type scheme with additional level shifting. We have found that it is not really beneficial to first converge the Hartree-Fock equations. Thus it is sensible to additionally use the keyword ! `noiter` in order to turn off the standard Hartree-Fock SCF process before entering the orbital optimizations.

The OO-RI-MP2 method is implemented for RHF and UHF reference wavefunctions. Analytic gradients are available.

The density does not need to be requested separately in OO-RI-MP2 calculations because it is automatically calculated. Also, there is no distinction between relaxed and unrelaxed densities because the OO-RI-MP2 energy is fully stationary with respect to all wavefunction parameters and hence the unrelaxed and relaxed densities coincide.

5.1.2.3 MP2 and RI-MP2 Gradients

Geometry optimization with MP2, RI-MP2, SCS-MP2 and RI-SCS-MP2 proceeds just as with any SCF method and numerical frequencies are available in the same way too. The frozen core option is also supported for RI-MP2 gradients in both serial and parallel runs. The RIJCOSX approximation (section 5.1.4.3) is supported in RI-MP2 and hence also double hybrid DFT gradient runs. This leads to large speedups in larger calculations, particularly if the basis sets are accurate.

```
#
# MP2 optimization example
#
! RHF SCS-MP2 TZVP TZV/C TightSCF Opt NoFrozenCore
* int 0 1
  C  0 0 0 0.00   0.0      0.00
  O  1 0 0 1.20   0.0      0.00
  H  1 2 0 1.09 120.0      0.00
  H  1 2 3 1.09 120.0    180.00
*
```

This job results in:

```
--------------------------------------------------------------------------
                     Redundant Internal Coordinates

                     --- Optimized Parameters ---
                       (Angstroem and degrees)

       Definition                OldVal   dE/dq    Step    FinalVal
--------------------------------------------------------------------------
   1. B(O   1,C   0)             1.2115  0.000515 -0.0003   1.2112
   2. B(H   2,C   0)             1.1004 -0.000033  0.0001   1.1005
   3. B(H   3,C   0)             1.1004 -0.000033  0.0001   1.1005
   4. A(H   2,C   0,O   1)       121.93  0.000028   -0.00   121.93
   5. A(H   3,C   0,O   1)       121.93  0.000028   -0.00   121.93
   6. A(H   3,C   0,H   2)       116.14 -0.000056    0.01   116.14
   7. I(O   1,H   3,H   2,C   0)  -0.00 -0.000000    0.00    -0.00
--------------------------------------------------------------------------
```

Just to demonstrate the accuracy of RI-MP2, here is the result with RI-SCS-MP2 instead of SCS-MP2:

```
--------------------------------------------------------------------------
                     Redundant Internal Coordinates

                     --- Optimized Parameters ---
                       (Angstroem and degrees)

       Definition                OldVal   dE/dq    Step    FinalVal
--------------------------------------------------------------------------
   1. B(O   1,C   0)             1.2115  0.000515 -0.0003   1.2112
   2. B(H   2,C   0)             1.1004 -0.000033  0.0001   1.1005
```

```
   3. B(H   3,C   0)                1.1004 -0.000033  0.0001    1.1005
   4. A(H   2,C   0,O   1)          121.93  0.000028   -0.00    121.93
   5. A(H   3,C   0,O   1)          121.93  0.000028   -0.00    121.93
   6. A(H   3,C   0,H   2)          116.14 -0.000056    0.01    116.14
   7. I(O   1,H   3,H   2,C   0)     -0.00 -0.000000    0.00      0.00
   ----------------------------------------------------------------------
```

You see that *nothing* is lost in the optimized geometry through the RI approximation thanks to the efficient and accurate RI-auxiliary basis sets of the Karlsruhe group (in general the deviations in the geometries between standard MP2 and RI-MP2 are very small). Thus, RI-MP2 really is a substantial improvement in efficiency over standard MP2.

5.1.2.4 RIJCOSX-RI-MP2 Gradients

The RIJCOSX-MP2 gradient is easily called with:

```
RI-MP2 def2-TZVPP def2-TZVPP/J def2-TZVPP/C TightSCF RIJCOSX
! TightOpt
...
```

The use of two different fitting basis sets is strongly recommended! In RIJCOSX-MP2 gradient runs two different grids for the seminumeric exchange integration are employed. The first grid is required for the solution of the z-vector equations and depends on the previously defined SCF grids.

```
%method
   Z_GridX 1  # smallest SCF grid (default)
           2  # medium SCF grid
           3  # largest SCF grid
 end
```

The second grid is independent of the SCF grids and selects the integration grid to evaluate the basis function derivatives. This grid can be chosen with ! MP2GridXn ($n = 1$–3), default is MP2GridX2.

For expert users, the grid parameters for the exchange grids can be even more finely controlled:

```
%method
   IntAccX     Acc1,  Acc2,  Acc3,  Acc4
   GridX       Ang1,  Ang2,  Ang3,  Acc4
   XCorrection Corr1, Corr2, Corr3, Corr4
   UseFinalGridX true
 end
```

The fourth entries for IntAccX, GridX and XCorrection refer to the MP2GridX.

5.1.2.5 MP2 and RI-MP2 Second Derivatives

The calculation of second-order properties with the MP2, RI-MP2 and double-Hybrid DFT methods is also available in ORCA. The most expensive term in the second derivative calculations is the 4-external contribution which can be evaluated either via an AO direct or a semi-numerical Chain-of-Spheres approach. The KC_GridX for the semi-numerical evaluation can be chosen as follows:

```
   %mp2 KCOpt _AOBLAS          # (default) AO direct with BLAS routines
             _COSX             # semi-numerical evaluation using the COSX method
        KC_UseXGrid 1..9       # default 4
        KC_GridX    1..7       # default 2 (= angular Lebedev grid 110)
        KC_IntAccX  3.34..4.67 # default 3.34
   end
```

Electric and magnetic second derivatives are only calculated at MP2 level, if a MP2 density has been calculated %mp2 density relaxed/unrelaxed end otherwise the SCF density is used.

5.1.2.6 Frozen Core Options

In MP2 energy and gradient runs the Frozen Core (FC) approximation is applied by default. This implies that the core electrons are not included in the perturbation treatment, since the inclusion of dynamic correlation in the core electrons usually effects relative energies or geometry parameters insignificantly. The FC approximation can be adjusted in the %method block.

```
   %method  FrozenCore FC_ELECTRONS  # Freeze all core electrons
                       FC_EWIN       # Freeze selected core electrons via
                                     # %mp2 EWin EMin,EMax
                       FC_NONE       # full MP2
   end
```

If a MP2 calculation including all electrons is desired, the ! NoFrozenCore keyword can be simply inserted. Frozen virtual orbitals are not allowed in gradient runs or geometry optimization!

5.1.2.7 MP2 Densities and Natural Orbitals

In (RI-)MP2 gradient runs two types of MP2 densities are constructed - an "unrelaxed" density (which basically corresponds to the MP2 expectation value density) and a "relaxed" density which incorporates orbital relaxation. For both sets of densities a population analysis is printed if the SCF calculation also requested this population analysis. These two densities are stored as JobName.pmp2ur.tmp and JobName.pmp2re.tmp respectively. The corresponding spin densities are also constructed.

In addition to the density options, the user has the ability to construct MP2 natural orbitals. If relaxed densities are available, the program uses the relaxed densities and otherwise the unrelaxed ones. The natural orbitals are stored as JobName.mp2nat which is a GBW type file that can be read as input for other jobs (for example, it is sensible to start CASSCF calculations from MP2 natural orbitals). The density construction can be controlled separately in the input file (even without running a gradient or optimization) by:

```
        #
        # MP2 densities and natural orbitals
        #
        %mp2  Density none          # no density
                     unrelaxed      # unrelaxed density
                     relaxed        # relaxed density
              NatOrbs true          # Natural orbital construction on or off
              end
```

NOTE

- In very large RI-MP2 runs you can cut down the amount of main memory used by a factor of two if you use the keyword ! `FLOAT`. This is more important in gradient runs than in single point runs. Deviations from double precision values for energies and gradients should be in the microhartree and sub-microhartree range. However, we have met cases where this option introduced a large and unacceptable error, in particular in transition metal calculations. You are therefore adviced to be careful and check things out beforehand.

5.1.2.8 Explicitly correlated MP2 calculations

ORCA features an efficient explicit correlation module that is available for MP2 and coupled-cluster calculations (section 5.1.3.6). It is described below in the context of coupled cluster calculations

5.1.3 Coupled Cluster and Coupled Pair Methods

5.1.3.1 Basics

The coupled cluster method is presently available for RHF and UHF references. The implementation is fairly efficient and suitable for large-scale calculations. The most elementary use of this module is fairly simple.

```
    !  METHOD
    # where METHOD is:
    #   CCSD CCSD(T) QCISD QCISD(T) CPF/n NCPF/n CEPA/n NCEPA/n
    #    (n=1,2,3 for all variants) ACPF NACPF AQCC CISD

    !  AOX-METHOD
    # computes contributions from integrals with 3- and 4-external
    # labels directly from AO integrals that are pre-stored in a
    # packed format suitable for efficient processing

    !  AO-METHOD
    # computes contributions from integrals with 3- and 4-external
    # labels directly from AO integrals. Can be done for integral
    # direct and conventional runs. In particular, the conventional
    # calculations can be very efficient
```

```
   !  MO-METHOD   (this is the default)
   # performs a full four index integral transformation. This is
   # also often a good choice


   !  RI-METHOD
   # selects the RI approximation for all integrals. Rarely advisable


   !  RI34-METHOD
   # selects the RI approximation for the integrals with 3- and 4-
   # external labels
   #
   # The module has many additional options that are documented
   # later in the manual.
```

NOTE

- The same FrozenCore options as for MP2 are applied in the MDCI module.

The computational effort for these methods is high — $O(N^6)$ for all methods and $O(N^7)$ if the triples correction is to be computed (calculations based on an unrestricted determinant are roughly 3 times more expensive than closed shell calculations and approximately six times more expensive if triple excitations are to be calculated). This restricts the calculations somewhat: on presently available PCs 300–400 basis functions are feasible and if you are patient and stretch it to the limit it may be possible to go up to 500–600; if not too many electrons are correlated may be even up to 800–900 basis functions (then using AO-direct methods).

TIP

- For calculations on small molecules and large basis sets the MO-METHOD option is usually the most efficient; say perhaps up to about 300 basis functions. For integral conventional runs the AO-METHOD may even more efficient.

- For large calculations (>300 basis functions) the AO-METHOD option is a good choice. If, however, you use very deeply contracted basis sets such as ANOs these calculations should be run in the integral conventional mode.

- AOX-METHOD is usually slightly less efficient than MO-METHOD or AO-METHOD.

- RI-METHOD is seldom the most efficient choice. If the integral transformation time is an issue than you can select `%mdci trafotype trafo_ri` or choose RI-METHOD and then `%mdci kcopt kc_ao`.

To put this into perspective, consider a calculation on serine with the cc-pVDZ basis set — a basis on the lower end of what it suitable for a highly correlated calculation. The time required to solve the equations is listed in Table 5.1. We can draw the following conclusions:

- As long as one can store the integrals and the I/O system of the computer is not the bottleneck, the most efficient way to do coupled-cluster type calculations is usually to go via the full transformation (it scales as $O(N^5)$ whereas the later steps scale as $O(N^6)$ and $O(N^7)$ respectively).

- AO-based coupled cluster calculations are not much inferior. For larger basis sets (i.e. when the ratio of virtual to occupied orbitals is larger), the computation times will be even more favorable for the AO based implementation. The AO direct method uses much less disk space. However, when you use a very expensive basis set the overhead will be larger than what is observed in this example. Hence, conventionally stored integrals — if affordable — are a good choice.

- AOX based calculations runs at essentially the same speed as AO based calculations. Since AOX based calculations take four times as much disk space they are pretty much outdated and the AOX implementation is only kept for historical reasons.

- RI based coupled cluster methods are significantly slower. There are some disk space savings but the computationally dominant steps are executed less efficiently.

- CCSD is at most 10% more expensive than QCISD. With the latest AO implementation the awkward coupled cluster terms are handled efficiently.

- CEPA is not much more than 20% faster than CCSD. In many cases CEPA results wil be better than CCSD and then it is a real saving compared to CCSD(T), which is the most rigorous.

- If triples are included practically the same comments apply for MO versus AO based implementations as in the case of CCSD.

ORCA is quite efficient in this type of calculation but it is also clear that the range of application of these rigorous methods is limited. Future releases of ORCA introduce approximations to make calculations on at least medium sized molecules feasible.

Table 5.1: Computer times (minutes) for solving the coupled cluster/coupled pair equations for Serine (cc-pVDZ basis set)

Method	SCFMode	Time (min)
MO-CCSD	Conv	38.2
AO-CCSD	Conv	47.5
AO-CCSD	Direct	50.8
AOX-CCSD	Conv	48.7
RI-CCSD	Conv	64.3
AO-QCISD	Conv	44.8
AO-CEPA/1	Conv	40.5
MO-CCSD(T)	Conv	147.0
AO-CCSD(T)	Conv	156.7

All of these methods are designed to cover dynamic correlation in systems where the Hartree-Fock determinant dominates the wavefunctions. The least attractive of these methods is CISD which is not size-consistent and therefore practically useless. The most rigorous are CCSD(T) and QCISD(T). The former is perhaps to be preferred since it is more stable in difficult situations.[1] One can get highly accurate results from such calculations. However, one only gets this accuracy in conjunction with large basis sets. It is perhaps not very meaningful to perform a CCSD(T) calculation with a double-zeta basis set (see Table 5.2). The very least basis set quality required for meaningful results would perhaps be something like def2-TZVP(-f) or preferably def2-TZVPP (cc-pVTZ, ano-pVTZ). For accurate

---

[1]The exponential of the T1 operator serves to essentially fully relax the orbitals of the reference wavefunction. This is not included in the QCISD model that only features at most a blinear T1T2 term in the singles residuum. Hence, if the Hartree-Fock wavefunction is a poor starting point but static correlation is not the main problem, CCSD is much preferred over QCISD. This is not uncommon in transition metal complexes.

results quadruple-zeta and even larger basis sets are required and at this stage the method is restricted to rather small systems.

Let us look at the case of the potential energy surface of the N_2 molecule. We study it with three different basis sets: TZVP, TZVPP and QZVP. The input is the following:

```
! RHF TZVPP CCSD(T)
%paras R= 1.05,1.13,8
        end
* xyz 0 1
N 0 0 0
N 0 0 {R}
*
```

For even higher accuracy we would need to introduce relativistic effects and - in particular - turn the core correlation on. [2]

Table 5.2: Computed spectroscopic constants of N_2 with coupled-cluster methods.

Method	Basis set	R_e (pm)	ω_e (cm^{-1})	$\omega_e x_e$ (cm^{-1})
CCSD(T)	SVP	111.2	2397	14.4
	TZVP	110.5	2354	14.9
	TZVPP	110.2	2349	14.1
	QZVP	110.0	2357	14.3
	ano-pVDZ	111.3	2320	14.9
	ano-pVTZ	110.5	2337	14.4
	ano-pVQZ	110.1	2351	14.5
CCSD	QZVP	109.3	2437	13.5
Exp		**109.7**	**2358.57**	**14.32**

One can see from Table 5.2 that for high accuracy - in particular for the vibrational frequency - one needs both - the connected triple-excitations and large basis sets (the TZVP result is fortuitously good). While this is an isolated example, the conclusion holds more generally. If one pushes it, CCSD(T) has an accuracy (for reasonably well-behaved systems) of approximately 0.2 pm in distances, <10 cm^{-1} for harmonic frequencies and a few kcal/mol for atomization energies.[3] It is also astonishing how well the Ahlrichs basis sets do in these calculations — even slightly better than the much more elaborate ANO bases.

NOTE:

---

[2]Note that core correlation is not simply introduced by including the core orbitals in the correlation problem. In addition, special correlation core-polarization functions are needed. They have been standardized for a few elements in the cc-pCVxZ (X=D,T,Q,5,6) basis sets.

[3]However, in recent years it became more evident that even CCSD(T) achieves its high apparent accuracy through error cancellations. The full CCSDT method (triples fully included) usually performs worse than CCSD(T). The reason is that the (T) correction undershoots the effects of the triples to some extent and thereby compensates for the neglect of connected quadruple excitations. For very high accuracy quantum chemistry, even these must be considered. The prospects for treating chemically more relevant molecules with such methods is not particularly bright for the foreseeable future...

- The quality of a given calculation is not always high because it carries the label "Coupled Cluster". Accurate results are only obtained in conjunction with large basis sets and for systems where the HF approximation is a good 0th order starting point.

5.1.3.2 Coupled Cluster Densities

If one is mainly accustomed to Hartree-Fock or DFT calculations, the calculation of the density matrix is more or less a triviality and is automatically done together with the solution of the self-consistent field equations. Unfortunately, this is not the case in coupled cluster theory (and also not in MP2 theory). The underlying reason is that in coupled cluster theory, the expansion of the exponential $e^{\hat{T}}$ in the expectation value

$$D_{pq} = \frac{\langle \Psi | E_p^q | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\langle e^{\hat{T}} \Psi_0 | E_p^q | e^{\hat{T}} \Psi_0 \rangle}{\langle e^{\hat{T}} \Psi_0 | e^{\hat{T}} \Psi_0 \rangle}$$

only terminates if all possible excitation levels are exhausted, i.e., if all electrons in the reference determinant Ψ_0 (typically the HF determinant) are excited from the space of occupied to the space of virtual orbitals (here D_{pq} denotes the first order density matrix, E_p^q are the spin traced second quantized orbital replacement operators, and $\hat{T}$ is the cluster operator). Hence, the straightforward application of these equations is far too expensive. It is, however, possible to expand the exponentials and only keep the linear term. This then defines a linearized density which coincides with the density that one would calculate from linearized coupled cluster theory (CEPA/0). The difference to the CEPA/0 density is that converged coupled cluster amplitudes are used for its evaluation. This density is straightforward to compute and the computational effort for the evaluation is very low. Hence, this is a density that can be easily produced in a coupled cluster run. It is not, however, what coupled cluster aficionados would accept as a density.

The subject of a density in coupled cluster theory is approached from the viewpoint of response theory. Imagine one adds a perturbation of the form

$$H^{(\lambda)} = \lambda \sum_{pq} h_{pq}^\lambda E_p^q$$

to the Hamiltonian. Then it is always possible to cast the first derivative of the total energy in the form:

$$\frac{dE}{d\lambda} = \sum_{pq} D_{pq}^{(response)} h_{pq}^\lambda$$

This is a nice result. The quantity $D_{pq}^{(response)}$ is the so-called response density. In the case of CC theory where the energy is not obtained by variational optimization of an energy functional, the energy has to be replaced by a Lagrangian reading as follows:

$$L_{CC} = E_{HF} + \sum_{ai} z_{ai} F_{ai} + \tfrac{1}{4} \sum_{ijab} \langle ij || ab \rangle t_{ab}^{ij} + \sum_\mu z_\mu \langle \mu | \hat{H} e^{\hat{T}} | \Psi_0 \rangle$$

Here μ denotes any excited determinant (singly, doubly, triply, ...). There are two sets of Lagrange multipliers: the quantities z_{ai} that guarantee that the perturbed wavefunction fulfills the Hartree-Fock conditions by making the off-diagonal Fock matrix blocks zero and the quantities z_μ that guarantee that the coupled-cluster projection equations for the amplitudes are fulfilled. If both sets of conditions are fulfilled then the coupled cluster Lagrangian simply evaluates to the coupled cluster energy. The coupled cluster Lagrangian can be made stationary with respect to the Lagrangian multipliers z_{ai} and z_μ. The response density is then defined through:

$$\frac{dL_{CC}}{d\lambda} = \sum_{pq} D_{pq}^{(response)} h_{pq}^\lambda$$

The density D_{pq} appearing in this equation does not have the same properties as the density that would arise from an expectation value. For example, the response density can have eigenvalues lower than 0 or larger than 2. In practice, the response density is, however, the best 'density' there is for coupled cluster theory.

Unfortunately, the calculation of the coupled cluster response density is quite involved because additional sets of equations need to be solved in order to determine the z_{ai} and z_μ. If only the equations for z_μ are solved one speaks of an "unrelaxed" coupled cluster density. If both sets of equations are solved, one speaks of a "relaxed" coupled cluster density. For most intents and purposes, the orbital relaxation effects incorporated into the relaxed density are small for a coupled cluster density. This is so, because the coupled cluster equations contain the exponential of the single excitation operator $e^{\hat{T}_1} = \exp(\sum_{ai} t_a^i E_i^a)$. This brings in most of the effects of orbital relaxation. In fact, replacing the $\hat{T}_1$ operator by the operator $\hat{\kappa} = \sum_{ai} \kappa_a^i (E_i^a - E_a^i)$ would provide all of the orbital relaxation thus leading to "orbital optimized coupled cluster theory" (OOCC).

Not surprisingly, the equations that determine the coefficients z_μ (the Z vector equations) are as complicated as the coupled-cluster amplitude equations themselves. Hence, the calculation of the unrelaxed coupled cluster density matrix is about twice as expensive as the calculation of the coupled cluster energy (but not quite as with proper program organization terms can be reused and the Z vector equations are linear equations that converge somewhat better than the non-linear amplitude equations).

ORCA features the calculation of the unrelaxed coupled cluster density on the basis of the Z vector equations for closed- and open-shell systems. If a fully relaxed coupled cluster density is desired then ORCA still features the orbital-optimized coupled-cluster doubles method (OOCCD). This is not exactly equivalent to the fully relaxed CCSD density matrix because of the operator $\hat{\kappa}$ instead of $\hat{T}_1$. However, results are very close and orbital optimized coupled cluster doubles is the method of choice if orbital relaxation effects are presumed to be large.

In terms of ORCA keywords, the coupled cluster density is obtained through the following keywords:

```
#
# Coupled cluster density
#
%mdci   density    none
                    linearized
                    unrelaxed
                    orbopt
        end
```

which will work together with CCSD or QCISD (QCISD and CCSD are identical in the case of OOCCD because of the absence of single excitations). Note, that an unrelaxed density for CCSD(T) is NOT available.

Instead of using the density option "orbopt" in the mdci-block, OOCCD can also be invoked by using the keyword:

```
! OOCD
```

5.1.3.3 Static versus Dynamic Correlation

Having said that, let us look at an "abuse" of the single reference correlation methods by studying (very superficially) a system which is not well described by a single HF determinant. This already occurs for the twisting of the double-bond of C_2H_4. At a 90° twist angle the system behaves like a diradical and should be described by a multireference method (see section 5.1.7)
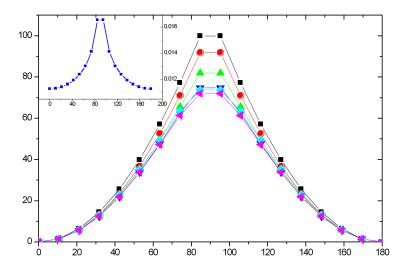
Figure 5.1: A rigid scan along the twisting coordinate of C_2H_4. The inset shows the T_1 diagnostic for the CCSD calculation.

As can be seen in Figure 5.1, there is a steep rise in energy as one approaches a 90° twist angle. The HF curve is actually discontinuous and has a cusp at 90°. This is immediately fixed by a simple CASSCF(2,2) calculation which gives a smooth potential energy surface. Dynamic correlation is treated on top of the CASSCF(2,2) method with the MRACPF approach as follows:

```
#
# twisting the double bond of C2H4
#
! RHF    SV(P) SV/C SmallPrint  NoPop   MRACPF
%casscf nel        2
        norb       2
        mult       1
        nroots     1
        TrafoStep  RI
        switchstep nr
        end
%mrci   tsel 1e-10
        tpre 1e-10
        end
%method scanguess pmodel
        end
%paras  R= 1.3385
        Alpha=0,180,18
        end
* int 0 1
C  0  0  0  0      0   0
C  1  0  0 {R}     0   0
H  1  2  0 1.07  120   0
H  1  2  3 1.07  120 180
```

```
H  2  1  3 1.07  120 {Alpha}
H  2  1  3 1.07  120 {Alpha+180}
*
```

This is the reference calculation for this problem. One can see that the RHF curve is far from the MRACPF reference but the CASSCF calculation is very close. Thus, dynamic correlation is not important for this problem! It only appears to be important since the RHF determinant is such a poor choice. The MP2 correlation energy is insufficient in order to repair the RHF result. The CCSD method is better but still fall short of quantitative accuracy. Finally, the CCSD(T) curve is very close the MRACPF. This even holds for the total energy (inset of Figure 5.2) which does not deviate by more than 2–3 mEh from each other. Thus, in this case one uses the powerful CCSD(T) method in an inappropriate way in order to describe a system that has multireference character. Nevertheless, the success of CCSD(T) shows how stable this method is even in tricky situations. The "alarm" bell for CCSD and CCSD(T) is the so-called "T_1-diagnostic"[4] that is also shown in Figure 5.2. A rule of thumb says, that for a value of the diagnostic of larger than 0.02 the results are not to be trusted. In this calculation we have not quite reached this critical point although the T_1 diagnostic blows up around 90° twist.
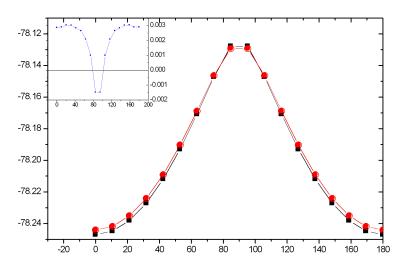
Figure 5.2: Comparison of the CCSD(T) and MRACPF total energies of the C_2H_4 along the twisting coordinate. The inset shows the difference E(MRACPF)-E(CCSD(T)).

The computational cost (disregarding the triples) is such that the CCSD method is the most expensive followed by QCISD ($\sim$10% cheaper) and all other methods (about 50% to a factor of two cheaper than CCSD). The most accurate method is generally CCSD(T). However, this is not so clear if the triples are omitted and in this regime the coupled pair methods (in particular CPF/1 and NCPF/1[5]) can compete with CCSD.

Let us look at the same type of situation from a slightly different perspective and dissociate the single bond of F_2. As is well known, the RHF approximation fails completely for this molecule and predicts it to be unbound. Again we use a much too small basis set for quantitative results but it is enough to illustrate the principle.

We first generate a "reference" PES with the MRACPF method:

---

[4]It is defined as $\|T_1\| / N^{1/2}$ where T_1 are the singles amplitudes and N the number of correlated electrons. The original reference is [60]

[5]The "N" methods have been suggested by [61] and are exclusive to ORCA. Please note that our NCPF/1 is different from the MCPF method in the literature [62]. The original CPF method — which we prefer — is from [63]; see also [64] for a nice review about the coupled pair approaches and the underlying philosophy.

```
    ! RHF SVP SV/C MRACPF
    %casscf nel 2 norb 2 nroots 1 mult 1
            shiftup 1.0 shiftdn 1.0
            switchstep nr
            end
    %mrci   tsel    1e-10
            tpre    1e-10
            end
    %paras  R= 3.0,1.3,35
            end
    * xyz 0 1
    F 0 0 0
    F 0 0 {R}
    *
```

Note that we scan from outward to inward. This helps the program to find the correct potential energy surface since at large distances the σ and σ^* orbitals are close in energy and fall within the desired 2×2 window for the CASSCF calculation (see section 5.1.7). Comparing the MRACPF and CASSCF curves it becomes evident that the dynamic correlation brought in by the MRACPF procedure is very important and changes the asymptote (loosely speaking the binding energy) by almost a factor of two. Around the minimum (roughly up to 2.0 Å) the CCSD(T) and MRACPF curves agree beautifully and are almost indistinguishable. Beyond this distance the CCSD(T) calculation begins to diverge and shows an unphysical behavior while the multireference method is able to describe the entire PES up to the dissociation limit. The CCSD curve is qualitatively ok but has pronounced quantitative shortcomings: it predicts a minimum that is much too short and a dissociation energy that is much too high. Thus, already for this rather "simple" molecule, the effect of the connected triple excitations is very important. Given this (rather unpleasant) situation, the behavior of the much simpler CEPA method is rather satisfying since it predicts a minimum and dissociation energy that is much closer to the reference MRACPF result than CCSD or CASSCF. It appears that in this particular case CEPA/1 and CEPA/2 bracket the correct result.
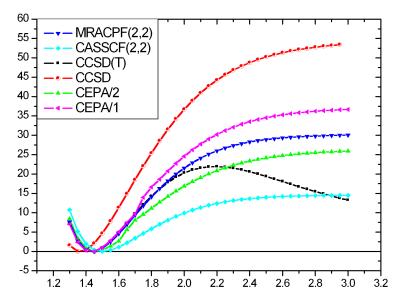
Figure 5.3: Potential energy surface of the F_2 molecule calculated with some single-reference methods and compared to the MRACPF reference.

As for MP2 calculations, the RI approximation can be introduced. It does not lead to spectacular performance gains but easens the burden for the integral transformation in larger calculations. The error introduced by the RI-approximation is usually negligible.For larger systems the AO or AOX methods are usually used.

5.1.3.4 Basis Sets for Correlated Calculations. The case of ANOs.

In HF and DFT calculations the generation and digestion of the two-electron repulsion integrals is usually the most expensive step of the entire calculation. Therefore, the most efficient approach is to use loosely contracted basis sets with as few primitives as possible — the Ahlrichs basis sets (SVP, TZVP, TZVPP, QZVP, def2-TZVPP, def2-QZVPP) are probably the best in this respect. Alternatively, the polarization consistent basis sets pc-1 through pc-4 could be used, but they are only available for H-Ar. For large molecules such basis sets also lead to efficient prescreening and consequently efficient calculations.

This situation is different in highly correlated calculations such as CCSD and CCSD(T) where the effort scales steeply with the number of basis functions. In addition, the calculations are usually only feasible for a limited number of basis functions and are often run in the integral conventional mode since high angular momentum basis functions are present and these are expensive to recomputed all the time. Hence, a different strategy concerning the basis set design seems logical. It would be good to use as few basis functions as possible but make them as accurate as possible. This is compatible with the philosophy of atomic natural orbital (ANO) basis sets. Such basis sets are generated from correlated atomic calculations and replicate the primitives of a given angular momentum for each basis function. Therefore, these basis sets are deeply contracted and expensive but the natural atomic orbitals form a beautiful basis for molecular calculations. In ORCA an accurate and systematic set of ANOs (ano-pVnZ, n = D, T, Q, 5 is incorporated). A related strategy underlies the design of the correlation consistent basis sets (cc-pVnZ, n = D, T, Q, 5, 6,...) that are also generally contracted except for the outermost primitives of the "principal" orbitals and the polarization functions that are left uncontracted.

Let us study this subject in some detail using the H_2CO molecule at a standard geometry and compute the SCF and correlation energies with various basis sets. In judging the results one should view the total energy in conjunction with the number of basis functions and the total time elapsed. Looking at the data in the Table below, it is obvious that the by far lowest SCF energies for a given cardinal number (2 for double-zeta, 3 for triple zeta and 4 for quadruple zeta) are provided by the ANO basis sets. Using specially optimized ANO integrals that is available since ORCA 2.7.0, the calculations are not even much more expensive than those with standard basis sets. Obviously, the correlation energies delievered by the ANO bases are also the best of all 12 basis sets tested. Hence, ANO basis sets are a very good choice for highly correlated calculations. The advantages are particularly large for the early members (DZ/TZ).

Table 5.3: Comparison of various basis sets for highly correlated calculations

Basis set	No. Basis Fcns	E(SCF)	E_C(CCSD(T))	E_{tot}(CCSD(T))	Total Time
cc-pVDZ	38	-113.876184	-0.34117952	-114.217364	2
cc-pVTZ	88	-113.911871	-0.42135475	-114.333226	40
cc-pVQZ	170	-113.920926	-0.44760332	-114.368529	695
def2-SVP	38	-113.778427	-0.34056109	-114.118988	2
def2-TZVPP	90	-113.917271	-0.41990287	-114.337174	46
def2-QZVPP	174	-113.922738	-0.44643753	-114.369175	730
pc-1	38	-113.840092	-0.33918253	-114.179274	2
pc-2	88	-113.914256	-0.41321906	-114.327475	43

pc-3	196	-113.922543	-0.44911659	-114.371660	1176
ano-pVDZ	38	-113.910571	-0.35822337	-114.268795	12
ano-pVTZ	88	-113.920389	-0.42772994	-114.348119	113
ano-pVQZ	170	-113.922788	-0.44995355	-114.372742	960
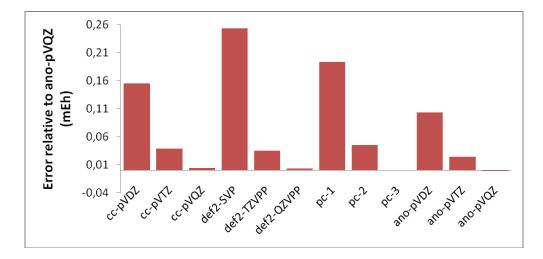
Figure 5.4: Error in mEh for various basis sets for highly correlated calculations relative to the ano-pVQZ basis set.

Let us look at one more example: the optimized structure of the N_2 molecule as a function of basis set using the MP2 method *(these calculations are a bit older from the time when the ano-pVnZ basis sets did not yet exist. Today, the ano-pVnZ would be prefered)*:

Table 5.4: Comparison of various basis sets for correlated calculations

Basis set	R_{eq} (pm)	$E(2N-N_2)$ (eV)	$IP(N/N^+)$ (eV)	$E(MP2)$ (Eh)
SVP	112.2	-9.67	14.45	**-109.1677**
cc-pVDZ	112.9	9.35	14.35	**-109.2672**
Bonn-ANO-DZP	112.1	9.45	14.58	**-109.3098**
TZVP	111.5	10.41	14.37	**-109.3423**
TZV(2d,2p)	111.4	10.61	14.49	**-109.3683**
Bonn-ANO-TZ2P	111.1	10.80	14.56	**-109.3791**
TZVPP	111.1	10.94	14.56	**-109.3973**
Bonn-ANO-TZ3P	110.9	11.18	14.65	**-109.4108**
QZVP	110.9	11.52	14.60	**-109.4389**

The highest quality basis set here is QZVP and it also gives the lowest total energy. However, this basis set contains up to g-functions and is very expensive. The Bonn-ANO-TZ3P is of the same size as TZVPP and gives the same result as QZVP for the geometry and an energy that is intermediate between TZVPP and QZVP. To not use a set of f-functions has still a noticeable effect on the outcome of the calculations and leads to an overestimation of the bond distance of 0.2 pm — a small change but for benchmark calculations of this kind still significant. Among these spd-only basis sets the Bonn-ANO-TZ2P basis set gives a better result than TZV(2d,2p) and a lower energy as well. In fact, similarly as for the Bonn-ANO-TZ3P, the distances is as good as that from TZVPP and the energy is

intermediate between TZV(2d,2p) and TZVPP($\equiv$TZV(2df,2pd)). The error made by the TZVP basis set that lacks the second set of d-functions is surprisingly small even though the deletion of the second d-set "costs" more than 20 mEh in the total energy.

A significant error on the order of 1-2 pm in the calculated distances is produced by smaller DZP type basis sets which underlines once more that such basis sets are really too small for correlated molecular calculations — the ANO-DZP basis sets are too strongly biased towards the atom and the "usual" molecule targeted DZP basis sets like SVP have the d-set designed to cover polarization but not correlation (the correlating d-functions are steeper than the polarizing ones). Among the three tested basis sets the Bonn-ANO-DZP fares best in this test and cc-pVDZ fares worst. The relatively good energy of the Bonn-ANO-DZP basis certainly comes from the good description of the atoms. The performance of the very economical SVP basis set should be considered as very good.

Essentially the same picture is obtained by looking at the (uncorrected for ZPE) binding energy calculated at the MP2 level – the largest basis set, QZVP gives the largest binding energy while the small basis set underestimate it. The error of the DZP type basis sets is fairly large (≈ 2 eV) and therefore caution is advisable when using such bases. In all cases it was found that the Bonn-ANO bases do slightly better than the segmented contracted basis sets of the same size. This still holds for the calculated ionization potential of the nitrogen atom. In principle, this is a worst case scenario for the ANO basis sets since they are supposedly strongly biased towards the neutral atom. Yet, The fare no worse than the segmented contracted basis sets. Here, an error cancellation is likely: The ANO bases recover more correlation (larger for the neutral) but are biased towards the neutral (underestimating the SCF energy for the cation). This bias perhaps shows up most clearly for the ANO-DZP basis set which gives a calculated IP that is accidentally almost right. The largest ANO-TZ3P even slightly overshoots relative to the QZVP basis set, which is expected.

5.1.3.5 Automatic extrapolation to the basis set limit

As eluded to in the previous section, one of the biggest problems with correlation calculations is the slow convergence to the basis set limit. One possibility to overcome this problem is the use of explicitly correlated methods. The other possibility is to use basis set extrapolation techniques. Since this involves some fairly repetitive work, some procedures were hardwired into the ORCA program. So far, only energies are supported. For extrapolation, a systematic series of basis sets is required. This is, for example, provided by the cc-pVnZ, aug-cc-pVnZ or the corresponding ANO basis sets. Here n is the 'cardinal number' that is 2 for the double-zeta basis sets, 3 for triple-zeta, etc.

The convergence of the HF energy to the basis set limit is assumed to be given by:

$$E_{\text{SCF}}^{(X)} = E_{\text{SCF}}^{(\infty)} + A \exp\left(-\alpha\sqrt{X}\right) \tag{5.1}$$

Here, $E_{\text{SCF}}^{(X)}$ is the SCF energy calculated with the basis set with cardinal number X, $E_{\text{SCF}}^{(\infty)}$ is the basis set limit SCF energy and A and α are constants. The approach taken in ORCA is to do a two-point extrapolation. This means that either A or α have to be known. Here, we take A as to be determined and α as a basis set specific constant.

The correlation energy is supposed to converge as:

$$E_{\text{corr}}^{(\infty)} = \frac{X^\beta E_{\text{corr}}^{(X)} - Y^\beta E_{\text{corr}}^{(Y)}}{X^\beta - Y^\beta} \tag{5.2}$$

The theoretical value for β is 3.0. However, it was found by Truhlar and confirmed by us, that for 2/3 extrapolations $\beta = 2.4$ performs considerably better.

For a number of basis sets, we have determined the optimum values for α and β:

	α_{23}	β_{23}	α_{34}	β_{34}
cc-pVnZ	4.42	2.46	5.46	3.05
pc-n	7.02	2.01	9.78	4.09
def2	10.39	2.40	7.88	2.97
ano-pVnZ	5.41	2.43	4.48	2.97
saug-ano-pVnZ	5.48	2.21	4.18	2.83
aug-ano-pVnZ	5.12	2.41		

Since the β values for 2/3 are close to 2.4, we always take this value. Likewise, all 3/4 and higher extrapolations are done with $\beta = 3$. However, the optimized values for α are taken throughout.

Using the keyword ! `Extrapolate(X/Y,basis)`, where `X` and `Y` are the corresponding successive cardinal numbers and `basis` is the type of basis set requested (= `cc`, `aug-cc`, `ano`, `saug-ano`, `aug-ano`, `def2`) ORCA will calculate the SCF and optionally the MP2 or MDCI energies with two basis sets and separately extrapolate.

Let us take the example of the H2O molecule at the B3LYP/TZVP optimized geometry. The reference values have been determined from a HF calculation with the decontracted aug-cc-pV6Z basis set and the correlation energy was obtained from the cc-pV5Z/cc-pV6Z extrapolation. This gives:

```
E(SCF,CBS)        = -76.066958 Eh
EC(CCSD(T),CBS)   =  -0.30866 Eh
Etot(CCSD(T),CBS) = -76.37561 Eh
```

Now we can see what extrapolation can bring in:

```
! RHF MP2 CCSD(T) Extrapolate(2/3) TightSCF Conv Bohrs
* int 0 1
O 0 0 0  0 0 0
H 1 0 0  1.81975 0 0
H 1 2 0  1.81975 105.237 0
*
```

NOTE:

- The RI-JK and RIJCOSX approximations work well together with this option and RI-MP2 is also possible. Auxiliary basis sets are automatically chosen and can not be changed.

- All other basis set choices, externally defined bases etc. will be ignored — the automatic procedure only works with the default basis sets!

- So far, the extrapolation is only implemented for single points and not for gradients. Hence, geometry optimizations cannot be done in this way.

This gives:

```
    Cardinal #: 2 Basis-set:        cc-pVDZ
    Cardinal #: 3 Basis-set:        cc-pVTZ
    Alpha    :   4.420 (SCF Extrapolation)
    Beta     :   2.400 (correlation extrapolation)


    SCF-basis set extrapolation:
                                E(SCF,2)      =       -76.026430944
                                E(SCF,3)      =       -76.056728252
    Extrapolated SCF energy     E(SCF,2/3)    =       -76.066581429 (-0.009853177)


    MP2-basis set extrapolation:
                                EC(MP2,2)     =        -0.201959623
                                EC(MP2,3)     =        -0.261813141
    Extrapolated MP2 corr. en.  EC(MP2,2/3)   =        -0.298172225 (-0.036359084)
    Extrapolated MP2 energy     EMP2(2/3)     =       -76.364753654 (-0.046212261)


    MDCI-basis set extrapolation:
                                EC(MDCI,2)    =        -0.214590888
                                EC(MDCI,3)    =        -0.275382820
    Extrapolated MDCI corr. en. EC(MDCI,2/3)  =        -0.312311961 (-0.036929141)
    Extrapolated MDCI energy    EMDCI(2/3)    =       -76.378893390 (-0.046782318)
```

Thus, the error in the total energy is indeed strongly reduced. Let us look at the more rigorous 3/4 extrapolation:

```
    Cardinal #: 3 Basis-set:        cc-pVTZ
    Cardinal #: 4 Basis-set:        cc-pVQZ
    Alpha    :   5.460 (SCF Extrapolation)
    Beta     :   3.000 (correlation extrapolation)


    SCF-basis set extrapolation:
                                E(SCF,3)      =       -76.056728252
                                E(SCF,4)      =       -76.064381268
    Extrapolated SCF energy     E(SCF,3/4)    =       -76.066687152 (-0.002305884)


    MP2-basis set extrapolation:
                                EC(MP2,3)     =        -0.261813115
                                EC(MP2,4)     =        -0.283155604
    Extrapolated MP2 corr. en.  EC(MP2,3/4)   =        -0.298729852 (-0.015574249)
    Extrapolated MP2 energy     EMP2(3/4)     =       -76.365417005 (-0.017880133)


    MDCI-basis set extrapolation:
                                EC(MDCI,3)    =        -0.275382818
                                EC(MDCI,4)    =        -0.295324278
    Extrapolated MDCI corr. en. EC(MDCI,3/4)  =        -0.309876154 (-0.014551876)
    Extrapolated MDCI energy    EMDCI(3/4)    =       -76.376563307 (-0.016857760)
```

In our experience, the ANO basis sets extrapolate quite a bit better than the cc-basis sets. Hence, repeating the entire calculation with `Extrapolate(3,ANO)` gives:

```
    Extrapolated MDCI energy    EMDCI(2/3)    =       -76.378256257 (-0.032505113)
    Extrapolated MDCI energy    EMDCI(3/4)    =       -76.376801619 (-0.012800873)
```

Which is within 1 mEh of the estimated CCSD(T) basis set limit energy in the case of the 3/4 extrapolation and within 3 mEh for the 2/3 extrapolation.

For larger molecules, the bottleneck of the calculation will be the CCSD(T) calculation with the larger basis set. In order to avoid this expensive (or prohibitive) calculation, it is possible to estimate the CCSD(T) energy at the basis set limit as:

$$E_{\text{corr}}^{(\text{CCSD(T)};Y)} \approx E_{\text{corr}}^{(\text{CCSD(T)};X)} + E_{\text{corr}}^{(\text{MP2};\infty)} - E_{\text{corr}}^{(\text{MP2};X)} \tag{5.3}$$

This assumes that the basis set dependence of MP2 and CCSD(T) is similar. One can then extrapolatet as before. Alternatively, the standard way — as extensively exercised by Hobza and co-workers — is to simply use:

$$E_{\text{total}}^{(\text{CCSD(T)};\text{CBS})} \approx E_{\text{SCF}}^{(Y)} + E_{\text{corr}}^{(\text{CCSD(T)};X)} + E_{\text{corr}}^{(\text{MP2};\infty)} - E_{\text{corr}}^{(\text{MP2};X)} \tag{5.4}$$

The appropriate keyword is:

```
! RHF MP2 CCSD(T) ExtrapolateEC(2/3,ANO) TightSCF Conv Bohrs
* int 0 1
O 0 0 0   0 0 0
H 1 0 0   1.81975 0 0
H 1 2 0   1.81975 105.237 0
*
```

This creates the following output:

```
Standard MP2/MDCI extrapolation:
Small basis SCF energy              =      -76.059178452
Large basis SCF energy              =      -76.064774379
Small basis MP2 energy              =       -0.219202852
Large basis MP2 energy              =       -0.267058613
Extrapolated CBS MP2 energy         =       -0.296129447 (-0.076926594)
Small basis MDCI energy             =       -0.229478138
Estimated CBS-MDCI energy           =       -0.306404733
Estimated large basis total energy  =      -76.371179112
```

The estimated correlation energy is not really bad — within 2 mEh from the basis set limit. Thus, in this example about half of the error comes from the SCF energy (not extrapolated in this scheme) and half of the error stems from the extrapolation of the correlation energy.

Using the `ExtrapolateEP2(n/m,bas)` keyword one can use a generalization of the above method where instead of MP2 any available correlation method can be used as described in Ref. [65].

$$E_{\text{corr}}^{(\text{CCSD(T)};CBS)} \approx E_{\text{corr}}^{(\text{CCSD(T)};X)} + E_{\text{corr}}^{(\text{M};CBS)}(X, X+1) - E_{\text{corr}}^{(\text{M};X)} \tag{5.5}$$

Here M represents any correlation method one would like to use. For the previous water molecule the input of a calculation that uses LPNO-CEPA/1 instead of MP2 would look like:

```
! RHF LPNO-CEPA/1 ExtrapolateEP2(2/3,cc) TightSCF Conv Bohrs
* int 0 1
O 0 0 0  0 0 0
H 1 0 0  1.81975 0 0
H 1 2 0  1.81975 105.237 0
*
```

and it would produce the following output:

```
Alpha    :  4.420 (SCF Extrapolation)
Beta     :  2.460 (correlation extrapolation)

SCF energy with basis      cc-pVDZ            =      -76.026430944
                           cc-pVTZ            =      -76.056728252
Extrapolated CBS SCF energy                   =      -76.066581429

M energy with basis        cc-pVDZ            =       -0.211813810
                           cc-pVTZ            =       -0.268899833
Extrapolated CBS M energy                     =       -0.302257269 (-0.033357436)

CCSD(T) energy with basis    cc-pVDZ          =       -0.214590888
CCSD(T) - M with basis       cc-pVDZ          =       -0.002777078

Estimated CBS CCSD(T) energy                  =       -0.305034347
Estimated CBS total energy                    =      -76.371615776
```

Finally it was shown [65] that instead of extrapolating the cheap method, M, using cardinal numbers X and $X + 1$ it is better to use cardinal numbers $X + 1$ and $X + 2$.

$$E_{\text{corr}}^{(\text{CCSD(T)});CBS} \approx E_{\text{corr}}^{(\text{CCSD(T)});X} + E_{\text{corr}}^{(\text{M};CBS)}(X + 1, X + 2) - E_{\text{corr}}^{(\text{M};X)} \tag{5.6}$$

This can be done using the `ExtrapolateEP3(bas)` keyword:

```
! RHF LPNO-CEPA/1 ExtrapolateEP3(cc) TightSCF Conv Bohrs
```

and the corresponding output would be:

```
Alpha    :  5.460 (SCF Extrapolation)
Beta     :  3.050 (correlation extrapolation)

SCF energy with basis      cc-pVDZ            =      -76.026430944
                           cc-pVTZ            =      -76.056728252
                           cc-pVQZ            =      -76.064381268
Extrapolated CBS SCF energy                   =      -76.064381268

M energy with basis        cc-pVDZ            =       -0.211813813
                           cc-pVTZ            =       -0.268899833
                           cc-pVQZ            =       -0.287854095
```

```
    Extrapolated CBS M energy                            =        -0.301347434 (-0.013493340)

    CCSD(T) energy with basis        cc-pVDZ             =        -0.214590888
    CCSD(T) - M with basis           cc-pVDZ             =        -0.002777075

    Estimated CBS CCSD(T) energy                         =        -0.304124509
    Estimated CBS total energy                           =       -76.368505778
```

5.1.3.6 Explicitly Correlated MP2 and CCSD(T) Calculations

A physically perhaps somewhat more satisfying alternative to basis set extrapolation is the theory of explicit correlation. In this method terms are added to the wavefunction Ansatz that contain the interelectronic coordinates explicitly (Hence the name 'explicit correlation"). Initially these terms were linear in the interelectronic distances ('R12-Methods'). However, it has later been found that better results can be obtained by using other functions, such as an exponential, of the the interelectronic distance ('F12-Method'). These methods are known to yield near basis set limit results for correlation energies in conjunction with much smaller orbital basis sets.

In applying these methods several points are important:

- Special orbital basis sets are at least advantageous. The development of such basis sets is still in its infancy. For a restricted range of elements the basis sets cc-pVnZ-F12 are available (where n=D,T,Q) and are recommended. Note, that other than their names suggest, these are a fair bit larger than regular double-, triple- or quadruple zeta basis sets

- In addition to an orbital basis set, a near-complete auxiliary basis set must be specified. This is the so-called 'CABS' basis. For the three basis sets mentioned above these are called cc-pVnZ-F12-CABS. If you have elements that are not covered you are on your own to supply a CABS basis set. CABS basis sets can be read into ORCA in a way analogous to RI auxiliary basis sets (replace 'AUX' by 'CABS' in the input

- if the RI approximation is used in conjunction with F12, a third basis set is required - this can be the regular auxiliary '/C' basis, but we recommend to step one level up in the auxiliary basis set (e.g. use a cc-pVTZ/C fitting basis in conjunction with cc-pVDZ-F12)

- It is perfectly feasible to use RIJCOSX or RI-JK at the same time. In this case, you should provide a fourth basis set for the Coulomb fitting

- RHF and UHF are available, ROHF not.

- Gradients are not available

- F12 in conjunction with COSMO and DKH/ZORA can be done, but this is not well tested methodology.

Doing explicitly correlated MP2 calculations is straightforward. For example look at the following calculation on the water molecule at a given geometry:

```
#
! F12-MP2 cc-pVDZ-F12 cc-pVDZ-F12-CABS  VeryTightSCF PModel

* xyz 0 1
    O     0.000000000000      0.000000000000      0.369372944000
```

```
       H     0.783975899000        0.000000000000      -0.184686472000
       H    -0.783975899000        0.000000000000      -0.184686472000
    *
```

and similary in conjunction with the RI approximation:

```
    #
    ! F12-RI-MP2 cc-pVDZ-F12 cc-pVDZ-F12-CABS cc-pVTZ/C VeryTightSCF PModel

    * xyz 0 1
       O     0.000000000000        0.000000000000       0.369372944000
       H     0.783975899000        0.000000000000      -0.184686472000
       H    -0.783975899000        0.000000000000      -0.184686472000
    *
```

The output is relatively easy to interprete:

```
    -----------------
    RI-MP2-F12 ENERGY
    -----------------


    EMP2 correlation Energy        :      -0.286878725335
    F12 correction                 :      -0.013254851100
                                          -----------------
    MP2 basis set limit estimate   :      -0.300133576436


    Hartree-Fock energy            :     -76.066649848665
    (2)_CABS correction to EHF     :      -0.000124410174
                                          -----------------
    HF basis set limit estimate    :     -76.066774258839


    MP2 total energy before F12    :     -76.353528574000
    Total F12 correction           :      -0.013379261275
                                          -----------------
    Final basis set limit MP2 estimate :  -76.366907835275
```

It consists of several parts. The first is the regular (RI-)MP2 correlation energy in the orbitals basis followed by the additive MP2 correction that are combined to provide a MP2 correlation energy basis set limit estimate. The second part consists of an estimate in the error in the underlying SCF energy. This is the "(2) CABS" correction. The combination of the SCF energy with this correction yields an estimate of the SCF basis set limit. The correction will typically undershoot somewhat, but the error is very smooth. Finally, the corrected correlation energy and the corrected SCF energy are added to yield the F12 total energy estimate at the basis set limit.

Let's look at some results and compare to extrapolation:

```
    #
    # Correlation energies of the water molecule: extrapolation versus F12
    #
    # cc-pVDZ MP2: -0.201 380 894
```

```
#     T     : -0.261 263 141
#     Q     : -0.282 661 311
#    T/Q    : -0.298 276 192
#    Q/5    : -0.300 598 282
# F12-DZ    : -0.295 775 804
# RI-F12-DZ : -0.295 933 560 (cc-pVDZ/C)
#             -0.295 774 489 (cc-pVTZ/C)
# F12-TZ    : -0.299 164 006
# RI-F12-TZ : -0.299 163 478 (cc-pVQZ/C)
# F12-QZ    : -0.300 130 086
```

It is obvious that extrapolated and F12 correlation energies converge to the same number (in this case around 300 mEh). The best extreapolated result is still below the F12 result (this would primarily be meaningful in a variational calculation). However, first of all this was an expensive extrapolation and second, the small residual F12 error is very smooth and cancels in energy differences. In any case, already the F12-double zeta (where 'double zeta' is to be interpreted rather loosely) brings one into within 5 mEh of the basis set limit correlation energy and the F12-triple zeta calculation to within 1 mEh, which is impressive.

The additional effort for the F12 calculation is rather high, since five types of additional two-electron integrals need to be calculated. Both, integrals in CABS space and in the original orbital (OBS) space must be calculated and mixed Fock matrices are also required. Hence, one may wonder, whether a double-zeta F12 calculation actually saves any time over, say, a quadruple-zeta regular calculation. The actual answer to this question is: 'NO'. Given all possibilities of obtained approximate MP2 and SCF energies, we have investigated the question of how to obtain MP2 basis set limit energies most efficiently in some detail. The results have been summarized in a paper (Liakos et al., Molec. Phys., 2013) and show that in terms of timings basis set extrapolation in combination with RI-JK is the method of choice in combination for MP2. However, energy differences are more reliable with F12-MP2. In combination with RI-JK or RIJCOSX F12-MP2 becomes also competitive in terms of computational efficiency.

This situation is different in the case of coupled cluster methods, where F12 methods outperform extrapolation and are the method of choice.

For coupled cluster theory, everything works in a very similar fashion:

```
# the keywords
! F12-CCSD(T)
# and
! CCSD(T)-F12
# are equivalent
```

A special feature of ORCA that can use large amounts of time, is to use the RI approximation only for the F12-part. The keyword here is:

```
! F12/RI-CCSD(T)
# or
! CCSD(T)-F12/RI
```

Everything else works as described for F12-MP2.

5.1.3.7 Local Coupled Pair and Coupled Cluster Calculations

ORCA features a special set of local correlation methods. The prevalent local coupled cluster approaches date back to ideas of Pulay and have been extensively developed by Werner, Sch"utz and co-workers. They use the concept of correlation domains in order to achieve linear scaling with respect to CPU, disk and main memory. While the central concept of electron pairs is very similar in both approaches, the local correlation methods in ORCA follow a completely different and original philosophy.

In ORCA rather than trying to use sparsity, we exploit data compression. To this end two concepts are used: (a) localization of internal orbitals, which reduces the number of electron pairs to be correlated since the pair correlation energies are known to fall off sharply with distance; (b) use of a truncated pair specific natural orbital basis to span the significant part of the virtual space for each electron pair. This guarantees the fastest convergence of the pair wavefunction and a nearly optimal convergence of the pair correlation energy while not introducing any real space cut-offs or geometrically defined domains. These PNOs have been used previously by the pioneers of correlation theory. However, as discussed in the original papers, the way in which they have been implemented into ORCA is very different.

For a full description of technical details and numerical tests see:

- Neese, F.; Liakos, D.; Hansen, A. (**2009**) Efficient and accurate local approximations to the coupled cluster singles and doubles method using a truncated pair natural orbital basis *J. Chem. Phys.*, <u>131</u>, 064103.

- Neese, F.; Wennmohs, F.; Hansen, A.; Grimme, S. (**2009**) Accurate Theoretical Chemistry with Coupled Electron Pair Models *Acc. Chem. Res.*, <u>42</u>(5), 641-648

- Neese, F.; Wennmohs, F.; Hansen, A. (**2009**) Efficient and accurate local approximations to coupled electron pair approaches. An attempt to revive the pair-natural orbital method *J. Chem. Phys.*, <u>130</u>, 114108

- Liakos, D. G.; Hansen, A.; Neese, F. (**2011**) Weak molecular interactions studied with parallel implementations of the local pair natural orbital coupled pair and coupled cluster methods *J. Chem. Theory Comput.*, <u>7</u>(1), 76-87

- Hansen, A.; Liakos, D.G.; Neese, F. (**2011**) Efficient and accurate local single reference correlation methods for high-spin open-shell molecules using pair natural orbitals *J. Chem. Phys.*, <u>135</u>, 214102

- Riplinger, C.; Neese, F. (**2013**) An efficient and near linear scaling pair natural orbital based local coupled cluster method *J. Chem. Phys.*, <u>138</u>, 034106

- Riplinger, C.; Sandhoefer, B.; Hansen, A.; Neese, F. (**2013**) Natural triple excitations in local coupled cluster calculations with pair natural orbitals *J. Chem. Phys.*, <u>139</u>, 134101

The paper from 2013 introduces the so-called DLPNO-CCSD method ('domain based local pair natural orbital method'). This method is near linear scaling with system size and allows for giant calculations to be performed. The principle idea behind DLPNO is the following: it became clear early on that the PNO space for a given electron pair (ij) is local and located in the same region of space as the electron pair (ij). in LPNO-CCSD this locality was partially used in the local fitting to the PNOs (controlled by the parameter TCutMKN). However, the PNOs were expanded in localized, canonical virtual orbitals which led to some higher order scaling steps. In DLPNO, the PNOs are expanded in the set of projected atomic orbitals:

$$|\tilde{\mu}\rangle = \left(1 - \sum_i |i\rangle\langle i|\right)|\mu\rangle \tag{5.7}$$

where $|\mu\rangle$ is an atomic orbital and $|i\rangle$ refes to an occupied molecular orbital. Such projected orbitals are an overcomplete representation of the virtual space. The projected orbital $|\tilde{\mu}\rangle$ is located in the same region of space as $|\mu\rangle$ and hence can be assigned to atomic centers. This has first been invented and used by Pulay and Saebo [66] in their pioneering work on local correlation methods and widely exploited by Werner, Schütz and co-workers in their local correlation approaches. [67,68] DLPNO-CCSD goes one step further in expanding the PNOs $|\tilde{a}_{ij}\rangle$ of a given pair (ij) as:

$$|\tilde{a}_{ij}\rangle = \sum_{\tilde{\mu} \in \{ij\}} d^{ij}_{\tilde{\mu}\tilde{a}} |\tilde{\mu}\rangle \tag{5.8}$$

where $\tilde{\mu} \in \{ij\}$ is the domain of atoms (range of $\tilde{\mu}$) that is associated with the electron pair ij. The advantage of the PNO method is, that these domains can be chosen to be large ($>$15-20 atoms) without compromising the efficiency of the method.

The comparison between LPNO-CCSD and DLPNO-CCSD is shown in Figure 5.5. It is obvious that DLPNO-CCSD is (almost) never slower than LPNO-CCSD. However, its true advantages do become most apparent for molecules with more than approximately 60 atoms. The triples correction, that was added with our second paper from 2013, shows a perfect linear scaling, as is shown in Figure 5.5. For large systems it adds about 10%–20% to the DLPNO-CCSD computation time, hence its addition is possible for all systems for which the latter can still be obtained.
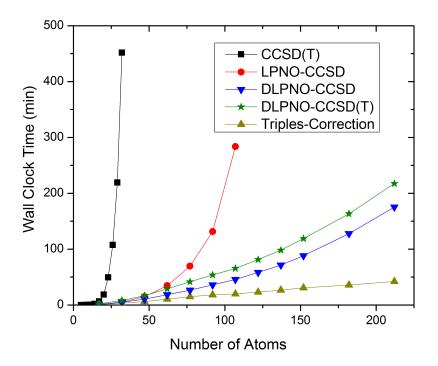
Figure 5.5: Scaling behaviour of the canonical CCSD, LPNO-CCSD and DLPNO-CCSD(T) methods. It is obvious that only DLPNO-CCSD and DLPNO-CCSD(T) can be applied to truly large molecules. The advantages of DLPNO-CCSD over LPNO-CCSD do not show before the system has reached a size of about 60 atoms

Using the DLPNO-CCSD(T) approach it was possible for the first time (in 2013) to perform a CCSD(T) level calculation on an entire protein (Crambin with more than 650 atoms, Figure 5.6)
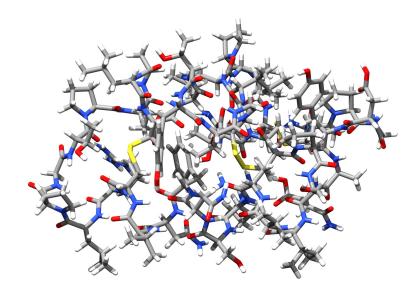
Figure 5.6: Structure of the Crambin protein - the first protein to be treated with a CCSD(T) level ab initio method

The use of the LPNO (and DLPNO) methods is simple and does not require special attention from the user:

```
# Local Pair Natural Orbital Test
! cc-pVTZ cc-pVTZ/C LPNO-CCSD TightSCF
# or
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD TightSCF
%maxcore 2000

# these are the default values - they need not to be touched!
%mdci  TCutPNO   3.33e-7  # cutoff for PNO occupation numbers. This
                                is the main truncation parameter
       TCutPairs 1e-4  # cut-off for estimated pair correlation energies.
                         This exploits the locality in the internal space
       TCutMKN   1e-3  # this is a technical parameter here that controls the domain
                         size for the local fit to the PNOs. It is conservative.
       Localtrafo 2    # this is a technical parameter that can have values 1, 2, 3.
                         Its default value for smaller systems is 1, for systems
                         with more than 60 atoms is 2. For parallel runs on large
                         systems with more than 8 CPUs Localtrafo 3 instead of 2
                         can lead to speed-ups.
       end

* xyz 0 1
... (coordinates)
*
```

Using the well tested default settings, the LPNO-CEPA (LPNO-CPF, LPNO-VCEPA), LPNO-QCISD and LPNO-

CCSD (LPNO-pCCSD) methods[6] can be run in strict analogy to canonical calculations and should approximate the canonical result very closely. In fact, one should not view the LPNO methods as new model chemistry - they are designed to reproduce the canonical results, including BSSE. This is different from the domain based local correlation methods that do constitute a new model chemistry with properties that are different from the original methods.
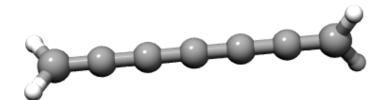
There are a few things to be noticed about (D)LPNO methods:

- The LPNO methods obligatorily make use of the RI approximation. Hence, a correlation fit set must be provided.

- For DLPNO-CCSD and DLPNO-CCSD(T) presently only the closed-shell variants are available.

- Besides the closed-shell version which uses a RHF or RKS reference determinant there is an open-shell version for high-spin open-shell molecules (see original paper) using an UHF or UKS reference determinant build from quasi-restricted orbitals (QROs, see section 7.8.3). Since the results of the current open-shell version are slightly less accurate than that of the closed-shell version it is mandatory to specify if you want to use the closed-shell or open-shell version for calculations of closed-shell systems, i.e. always put the "RHF" ("RKS") or "UHF" ("UKS") keyword in the simple keyword line. Open-shell systems can be of course only treated by the open-shell version. **Do not mix results of the closed- and open-shell version** (e.g. if you calculate reaction energies of a reaction in which both closed- and open-shell molecules take part, you should use the open-shell version throughout).

- F12-variants will become available in the foreseeable future.

- Parallelization is done.

- There are three thresholds that can be user controlled that can all be adjusted in the %mdci block: (a) T_{CutPNO} controls the number of PNOs per electron pair. This is the most critical parameter and has a default value of 3.33e−7. (b) $T_{CutPairs}$ controls a perturbative selection of significant pairs and has a default value of 1e-4. (c) T_{CutMKN} is a technical parameter and controls the size of the fit set for each electron pair. It has a default value of 1e−3. All of these default values are conservative. Hence, no adjustment of these parameters is necessary. All DLPNO-CCSD truncations are bound to these three truncation parameters and should not be touched (Hence they are also not documented :-)).

- LPNO-VCEPA/n (n=1,2,3) methods are only available in the open-shell version yet.

- LPNO variants of the parameterized coupled cluster methods (pCCSD, see section 7.8.1) are also available (e.g. LPNO-pCCSD/1a and LPNO-pC CSD/2a).

- The LPNO methods reproduce the canonical energy differences typically better than 1 kcal/mol. This accuracy exists over large parts of the potential energy surface. Tightening TCutPairs to 1e-5 gives more accurate results but also leads to significantly longer computation times.

- Potential energy surfaces are virtually but not perfectly smooth (like any method that involves cut-offs). Numerical gradient calculations have been attempted and reported to have been succesful.

---
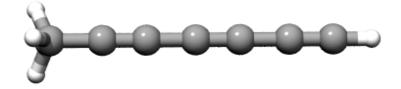
[6]As a technical detail: The closed-shell LPNO QCISD and CCSD come in two technical variants - LPNO1-CEPA/QCISD/CCSD and LPNO2-CEPA/CCSD/QCISD. The '2' variants consume less disk space but are also slightly less accurate than the '1' variants. This is discussed in the original paper in the case of QCISD and CCSD. For the sake of accuracy, the '1' variants are the default. In those cases, where '1' can still be performed, the computational efficiency of both approaches is not grossly different. For LPNO CCSD there is also a third variant (LPNO3-CCSD, also in the open-shell version) which avoids neglecting the dressing of the external exchange operator. However, the results do not differ significantly from variant 1 but the calculations will become more expensive. Thus it is not recommend to use variant 3. Variant 2 is not available in the open-shell version.

- The LPNO methods do work together with RIJCOSX, RI-JK and also with ANO basis sets and basis set extrapolation. They also work for conventional integral handling.

- The methods behave excellently with large basis sets. Thus, they stay efficient even when large basis sets are used that are necessary to obtain accurate results with wavefunction based *ab initio* methods. This is a prerequisite for efficient computational chemistry applications.

- For LPNO-CCSD, calculations with about 1000 basis functions are routine, calculations with about 1500 basis functions are possible and calculations with 2000-2500 basis functions are the limit on powerful computers. For DLPNO-CCSD much larger calculations are possible. There is virtually no crossover and DLPNO-CCSD is essentially always more efficient than LPNO-CCSD. Starting from about 50 atoms the differences become large. The largest DLPNO-CCSD calculation to date featured >450 atoms and almost 8900 basis functions!

- Using large main memory is not mandatory but advantageous since it speeds up the initial integral transformation significantly (controlled by "MaxCoreWork" and "MaxCoreIntAmp" in the %mdci block, see section 7.8.4).

- The open-shell versions are about twice as expensive as the corresponding closed-shell versions.

- Analytic gradients are not available.

As an example, see the following isomerization reaction that appears to be particularly difficult for DFT:

Isomerizes to:

The results of the calculations (closed-shell versions) with the def2-TZVP basis set (about 240 basis functions) are shown below:

Method	Energy Difference (kcal/mol)	Time (min)
CCSD(T)	-14.6	92.4
CCSD	-18.0	55.3
LPNO-CCSD	-18.6	20.0
CEPA/1	-12.4	42.2
LPNO-CEPA/1	-13.5	13.4

The calculations are typical in the sense that: (a) the LPNO methods provide answers that are within 1 kcal/mol of the canonical results, (b) CEPA approximates CCSD(T) more closely than CCSD. The speedups of a factor of 2-5 are

moderate in this case. However, this is also a fairly small calculation. For larger systems, speedups of the LPNO methods compared to their canonical counterparts are on the order of a factor >100-1000.

5.1.3.8 Arbitrary Order Coupled Cluster Calculations

ORCA features an interface to Kallay's powerful MRCC program. This program must be obtained separately. The interface is restricted to single point energies but can be used for rigid scan calculations or numerical frequencies.

The use of the interface is simple:

```
#
# Test the MRCC code of Mihael Kallay
#
! cc-pVDZ Conv SCFConv10 UseSym

%mrcc method    "CCSDT"
      ETol      9
      end

* xyz 0 1
F 0 0 0
H 0 0 0.95
*
```

The Method string can be any of:

```
# The excitation level specification can be anything
# like SD, SDT, SDTQ, SDTQP etc.
 %mrcc method    "CCSDT"
                 "CCSD(T)"
                 "CCSD[T]"
                 "CCSD(T)_L"   (the lambda version)
                 "CC3"
                 "CCSDT-1a"
                 "CCSDT-1b"
                 "CISDT"
```

It is not a good idea, of course, to use this code for CCSD or CCSD(T) or CISD. Its real power lies in performing the higher order calculations. Open shell calculations can presently not be done with the interface.

5.1.4 Density Functional Theory

5.1.4.1 Standard Density Functional Calculations

Density functional calculations are as simple to run as HF calculations. In this case you may want to adjust the integration grid and you almost certainly will want to use the RI-J approximation in case that you have a LDA, GGA

or meta-GGA functional (non-hybrid functional). For hybrid functionals, the RIJCOSX and RI-JK approximations offer large speedups.

For example, consider this B3LYP calculation on the cyclohexane molecule.

```
# Test a simple DFT calculation
! RKS B3LYP SVP Grid4
* xyz 0 1
C   -0.79263     0.55338    -1.58694
C    0.68078     0.13314    -1.72622
C    1.50034     0.61020    -0.52199
C    1.01517    -0.06749     0.77103
C   -0.49095    -0.38008     0.74228
C   -1.24341     0.64080    -0.11866
H    1.10490     0.53546    -2.67754
H    0.76075    -0.97866    -1.78666
H   -0.95741     1.54560    -2.07170
H   -1.42795    -0.17916    -2.14055
H   -2.34640     0.48232    -0.04725
H   -1.04144     1.66089     0.28731
H   -0.66608    -1.39636     0.31480
H   -0.89815    -0.39708     1.78184
H    1.25353     0.59796     1.63523
H    1.57519    -1.01856     0.93954
H    2.58691     0.40499    -0.67666
H    1.39420     1.71843    -0.44053
*
```

If you want an accurate single point energy then it is wise to choose 'TightSCF', select a basis set of at least valence triple-zeta plus polarization quality (e.g. def2-TZVP) and also to move one step up in the DFT integration grid (i.e. 'Grid4').

5.1.4.2 DFT Calculations with RI

DFT calculations that do not require the HF exchange to be calculated (non-hybrid DFT) can be *very* efficiently executed with the RI-J approximation. It leads to very large speedups at essentially no loss of accuracy. The use of the RI-J approximation may be illustrated for a medium sized organic molecule - Penicillin:

```
# RI-DFT calculation on the Penicillin molecule
! RKS BP86 RI SVP SV/J TightSCF

* xyz 0 1
N    3.17265     1.15815    -0.09175
C    2.66167     0.72032     1.18601
C    4.31931     0.59242    -0.73003
```

```
    C      2.02252      1.86922     -0.54680
    C      1.37143      1.52404      0.79659
    S      2.72625     -1.05563      0.80065
    C      4.01305     -0.91195     -0.52441
    C      5.58297      1.09423     -0.06535
    O      1.80801      2.36292     -1.62137
    N      0.15715      0.73759      0.70095
    C      5.25122     -1.72918     -0.12001
    C      3.41769     -1.50152     -1.81857
    O      6.60623      1.14077     -0.91855
    O      5.72538      1.40990      1.08931
    C     -1.08932      1.35001      0.75816
    C     -2.30230      0.45820      0.54941
    O     -1.19855      2.53493      0.96288
    O     -3.48875      1.21403      0.57063
    C     -4.66939      0.59150      0.27339
    C     -4.84065     -0.79240      0.11956
    C     -5.79523      1.39165      0.03916
    C     -6.07568     -1.34753     -0.22401
    C     -7.03670      0.85454     -0.30482
    C     -7.18253     -0.52580     -0.43612
    H      3.24354      1.09074      2.02120
    H      4.33865      0.87909     -1.77554
    H      1.26605      2.42501      1.39138
    H      0.17381     -0.25857      0.47675
    H      6.05024     -1.64196     -0.89101
    H      5.67754     -1.39089      0.85176
    H      5.01118     -2.81229     -0.01401
    H      2.50304     -0.95210     -2.14173
    H      4.15186     -1.44541     -2.65467
    H      3.14138     -2.57427     -1.69700
    H      7.29069      1.46408     -0.31004
    H     -2.21049     -0.02915     -0.44909
    H     -2.34192     -0.28647      1.37775
    H     -4.00164     -1.48999      0.26950
    H     -5.69703      2.48656      0.12872
    H     -6.17811     -2.44045     -0.33185
    H     -7.89945      1.51981     -0.47737
    H     -8.15811     -0.96111     -0.71027
    *
```

The job has 42 atoms and 430 contracted basis functions. Yet, it executes in just a few minutes elapsed time on any reasonable personal computer.

NOTES:

- The RI-J approximation requires an "auxiliary basis set" in addition to a normal orbital basis set. For the Karlsruhe basis sets and the "cc" bases there are tailored fit sets that are chosen by appending a "/J" to the basis set name (e.g. def2-TZVP def2-TZVP/J, cc-pVDZ cc-pVDZ/J. Other choices are documented in sections 6.2.2.5 and 6.2.2.9.

- For "pure" functionals the use of RI-J is the default. Appropriate fitting bases can usually be chosen automatically. Only if non standard basis sets are used is the explicit specification of the fitting bases strongly recommended. For basis sets like SV(P), SVP, TZVP or TZVPP the defaults are appropriate.

Since DFT is frequently applied to open shell transition metals we also show one (more or less trivial) example of a Cu(II) complex treated with DFT.

```
! UKS BP86 RI SV DeMon/J SlowConv
%base "temp"
* xyz -2 2
  Cu  0      0      0
  Cl  2.25   0      0
  Cl -2.25   0      0
  Cl  0      2.25   0
  Cl  0     -2.25   0
*


$new_job


! UKS B3LYP NoRI TZVP TightSCF MORead
%moinp "temp.gbw"
%scf GuessMode CMatrix
     end
* xyz -2 2
  Cu  0      0      0
  Cl  2.25   0      0
  Cl -2.25   0      0
  Cl  0      2.25   0
  Cl  0     -2.25   0
*
```

Although it would not have been necessary for this example, it shows a possible strategy how to converge such calculations. First a less accurate but fast job is performed using the RI approximation, a GGA functional and a small basis set without polarization functions. Note that a larger damping factor has been used in order to guide the calculation (`SlowConv`). The second job takes the orbitals of the first as input and performs a more accurate hybrid DFT calculation. A subtle point in this calculation on a dianion in the gas phase is the command `GuessMode CMatrix` that causes the corresponding orbital transformation to be used in order to match the orbitals of the small and the large basis set calculation. This is always required when the orbital energies of the small basis set calculation are positive as will be the case for anions.

5.1.4.3 Hartree–Fock and Hybrid DFT Calculations with RIJCOSX

Frustrated by the large difference in execution times between pure and hybrid functionals, we have been motivated to study approximations to the Hartree-Fock exchange term. The method that we have finally come up with is called the 'chain of spheres' COSX approximation and may be thought of as a variant of the pseudo-spectral philosophy. Essentially, in performing two electron integrals, the first integration is done numerically on a grid and the second (involving the Coulomb singularity) is done analytically.[7] Upon combining this treatment with the Split-RI-J method for the Coulomb term (thus, you do need a Coulomb fitting basis!), we have designed the RIJCOSX approximation that can be used to accelerate Hartree-Fock and hybrid DFT calculations. Note that this introduces another grid on top of the DFT integration grid which is usually significantly smaller.

In particular for large and accurate basis sets, the speedups obtained in this way are very large - we have observed up to a factor of sixty! The procedure is essentially linear scaling such that large and accurate calculations become possible with high efficiency. The RIJCOSX approximation is basically available throughout the program. The errors are on the order of 1 kcal mol^{-1} or less in the total energies as well as in energy differences and can be made smaller with larger than the default grids or by running the final SCF cycle without this approximation. The impact on bond distances is a fraction of a pm, angles are better than a few tenth of a degree and soft dihedral angles are good to about 1 degree. To the limited extent to which it has been tested, vibrational frequencies are roughly good to 2-10 wavenumbers with the default settings.

The use of RIJCOSX is very simple:

```
! B3LYP def2-TZVPP def2-TZVPP/J TightSCF RIJCOSX
* xyz 0 1
C   -0.79263     0.55338    -1.58694
C    0.68078     0.13314    -1.72622
C    1.50034     0.61020    -0.52199
C    1.01517    -0.06749     0.77103
C   -0.49095    -0.38008     0.74228
C   -1.24341     0.64080    -0.11866
H    1.10490     0.53546    -2.67754
H    0.76075    -0.97866    -1.78666
H   -0.95741     1.54560    -2.07170
H   -1.42795    -0.17916    -2.14055
H   -2.34640     0.48232    -0.04725
H   -1.04144     1.66089     0.28731
H   -0.66608    -1.39636     0.31480
H   -0.89815    -0.39708     1.78184
H    1.25353     0.59796     1.63523
H    1.57519    -1.01856     0.93954
H    2.58691     0.40499    -0.67666
H    1.39420     1.71843    -0.44053
*
```

One thing to be mentioned in correlation calculations with RIJCOSX is that the requirements for the SCF and correlation fitting bases are quite different. We therefore now support two different auxiliary basis sets in the same run:

---

[7]For algorithmic and theoretical details see: [69].

```
    ! RI-MP2 def2-TZVPP def2-TZVPP/J def2-TZVPP/C TightSCF RIJCOSX
    ...
```

CAUTION:

- This feature does NOT work in multiple job inputs

5.1.4.4 Hartree-Fock and Hybrid DFT Calculations with RI-JK

An alternative algorithm for accelerating the HF exchange in hybrid DFT or HF calculations is to use the RI approximation for both Coulomb and exchange. This is implemented in ORCA for SCF single point energies but not for gradients.

```
    ! RHF def2-TZVPP def2-TZVPP/JK RI-JK
    ...
```

The speedups for small molecules are better than for RIJCOSX, for medium sized molecules (e.g. $(gly)_4$ similar and for larger molecules RI-JK is less efficient than RIJCOSX). The errors of RI-JK are usually below 1 mEh and the error is very smooth (smoother than for RIJCOSX). Hence, for small calculations with large basis sets, RI-JK is a good idea, for large calculations on large molecules RIJCOSX is better.

NOTES:

- For RI-JK you will need a larger auxiliary basis set. For the Kralsruhe basis set, the def2-SVP/JK etc. basis sets are available. They are large and accurate.

- For UHF RI-JK is roughly twice as expensive as for RHF. This is not true for RIJCOSX.

- RI-JK is available for conventional and direct runs and also for ANO bases. There the conventional mode is recommended.

A comparison of the RIJCOSX and RI-JK methods (taken from Ref. [70]) for the $(gly)_2$, $(gly)_4$ and $(gly)_8$ is shown below (wall clock times in second for performing the entire SCF):

		Def2-SVP	Def2-TZVP(-df)	Def2-TZVPP	Def2-QZVPP
$(gly)_2$	*Default*	105	319	2574	27856
	RI-JK	44	71	326	3072
	RIJCOSX	70	122	527	3659
$(gly)_4$	*Default*	609	1917	13965	161047
	RI-JK	333	678	2746	30398
	RIJCOSX	281	569	2414	15383
$(gly)_8$	*Default*	3317	12505	82774	
	RI-JK	3431	5452	16586	117795
	RIJCOSX	1156	2219	8558	56505

It is obvious from the data that for small molecules the RI-JK approximation is the most efficient choice. For $(gly)_4$ this is already no longer obvious. For up to the def2-TZVPP basis set, RI-JK and RIJCOSX are almost identical and for def2-QZVPP RIJCOSX is already a factor of two faster than RI-JK. For large molecules like $(gly)_8$ with

small basis sets RI-JK is not a big improvement but for large basis set it still beats the normal 4-index calculation. RIJCOSX on the other hand is consistently faster. It leads to speedups of around 10 for def2-TZVPP and up to 50-60 for def2-QZVPP. Here it outperforms RI-JK by, again, a factor of two.

5.1.4.5 DFT Calculations with Second Order Perturbative Correction (Double-Hybrid Functionals)

There is a family of functionals which came up in 2006 and were proposed by Grimme [71]. They consist of a semi-empirical mixture of DFT components and the MP2 correlation energy calculated with the DFT orbitals and their energies. Grimme referred to his functional as B2PLYP (B88 exchange, 2 parameters that were fitted and perturbative mixture of MP2 and LYP) – a version with improved performance (in particular for weak interactions) is mPW2PLYP [72] and is also implemented. From the extensive calibration work, the new functionals appear to give better energetics and a narrower error distribution than B3LYP. Thus, the additional cost of the calculation of the MP2 energy may be well invested (and is quite limited in conjunction with density fitting in the RI part). Martin has reported reparameterizations of B2PLYP (B2GP-PLYP, B2K-PLYP and B2T-PLYP) that are optimized for 'general-purpose', 'kinetic' and 'thermochemistry' applications. A fairly new variant of these double hybrids is the PWPB95 functional by Grimme. For more information, see the detailed input section.

In this version of ORCA, the method is available for single points, geometry optimizations [73], dipole moments and other first order properties, as well as for numerical polarizabilities and frequencies.

5.1.4.6 DFT Calculations with Atom-pairwise Dispersion Correction

It is well known that DFT does not include dispersion forces. It is possible to use a simple atom-pairwise correction to account for the major parts of this contribution to the energy [1, 2, 74]. We have adopted the code and method developed by Stefan Grimme in this ORCA version. The method is parameterized for many established functionals (e.g. BLYP, BP86, PBE, TPSS, B3LYP, B2PLYP).[8] The atom-pairwise dispersion correction with Becke-Johnson damping (keyword ! D3BJ) is the default option and will automatically be invoked by the simple keyword ! D3.

```
! BLYP D3 def2-QZVP RI def2-QZVP/J Opt Grid4
%paras R= 2.5,4.0,16
end
%geom Constraints
{ C 0 C }
{ C 1 C }
end
end
* xyz 0 1
Ar    0.0000000    0.0000000    {R}
H     0.0000000    0.0000000     0.0000000
C     0.0000000    0.0000000    -1.0951073
H     0.5163499    0.8943443    -1.4604101
```

---

[8] For expert users: The keyword D2, D3ZERO and D3BJ select the empirical 2006 and the atom-pairwise 2010 model, repsectively, with either zero-damping or Becke-Johnson damping. The default is the most accurate D3BJ model. The outdated model from 2004 [75] is no longer supported and can only be invoked by setting DFTDOPT = 1. The C6-scaling coefficient can be user defined using e.g. "%method DFTDScaleC6 1.2 end"

```
    H      0.5163499    -0.8943443    -1.4604101
    H     -1.0326998     0.0000000    -1.4604101
    *
```

In this example, a BLYP calculation without dispersion correction will show a repulsive potential between the argon atom and the methane molecule. Using the D3 dispersion correction as shown above, the potential curve shows a minimum at about $3.1-3.2$ Å. The atom-pairwise correction is quite successful and Grimme's work suggests that this is more generally true. For many systems like stacked DNA basepairs, hydrogen bond complexes and other weak interactions the atom-pairwise dispersion correction will improve substantially the results of standard functionals at essentially no extra cost.

5.1.4.7 DFT Calculations with Range-Separated Hybrid Functionals

All range-separated functionals in ORCA use the error function based approach according to Hirao and coworkers. [76] This allows the definition of DFT functionals that dominate the short-range part by an adapted exchange functional of LDA, GGA or meta-GGA level and the long-range part by Hartree-Fock exchange.

CAM-B3LYP, [77] LC-BLYP [78] and members of the ωB97-family of functionals have been implemented into ORCA, namely ωB97, ωB97X [79] and ωB97X-D3. [80] Some of them incorporate fixed amounts of Hartree-Fock exchange (EXX) and/or DFT exchange and they differ in the RS-parameter μ. In the case of ωB97X-D3, the proper D3 correction (employing the zero-damping scheme) should be calculated automatically. The user is encouraged to check this.

Several restrictions apply to these functionals at the moment. They have only been implemented and tested for use with the `libint` integral package and for RHF and UHF single-point and TDDFT calculations. Only the standard integral handling (NORI in direct or conventional mode) and RIJONX are supported. **Do not use these functionals with any other options**.

5.1.5 Quadratic Convergence

Convergence in SCF calculations is not always easy to achieve. One way to go if the calculations converges only slowly or "creeps" towards the end of the SCF cycles such that many almost useless cycles are being performed is to switch on a powerful but somewhat expensive convergence helper – the full Newton-Raphson method. In the neighbourhood of a stable SCF solution this method converges quadratically which means that after 3-4 Newton-Raphson cycles the calculations are normally converged. However, each cycle consists of microiterations which are roughly as expensive as one SCF iteration. Thus, each SCF iteration becomes somewhat expensive but this may be more than compensated by the reduced number of cycles.

Consider the following example of a molecule with a small HOMO–LUMO gap (≈ 1 eV):

```
    ! PModel RKS BP86 RI SemiDirect SV(P) DeMon/J TightSCF

    * xyz 0 1
    C -2.65720  1.06150 -0.05713
    C -1.26906  0.45978  0.20734
    C  2.30892  0.84241 -0.90960
```

```
  C  3.66808  0.62825 -0.22855
  C -3.24744  1.63074  1.22706
  C  4.55462  1.85858 -0.37334
  O -4.34718  2.12974  1.26070
  O  4.32817  2.77288 -1.12962
  S -0.52934 -0.10251 -1.36344
  S  1.32901 -0.69440 -0.81445
  N -3.52304 -0.01967 -0.66394
  N  3.40865  0.27323  1.21839
  O -2.67918  1.59541  2.29262
  O  5.54649  1.99821  0.30260
  H -2.59274  1.90623 -0.78323
  H -3.61570 -0.84473 -0.06839
  H -3.16141 -0.34892 -1.56042
  H -4.47232  0.30853 -0.85012
  H  4.27898  0.16311  1.74252
  H  2.86502  0.97910  1.71822
  H  2.90558 -0.60935  1.32107
  H  4.22252 -0.21720 -0.70036
  H  2.45549  1.12172 -1.97850
  H  1.78000  1.68539 -0.40747
  H -0.61420  1.23221  0.67374
  H -1.35568 -0.39162  0.92184
  *


  $new_job
  ! PModel RKS BP86 RI SemiDirect SV(P) DeMon/J TightSCF NRSCF


  * xyz 0 1
   ... etc, coordinates repeated
  *
```

The first job converges as follows:

```
  ITER      Energy         Delta-E         Max-DP      RMS-DP     [F,P]    Damp
   0  -1441.4978981358   0.000000000000 0.40179217  0.00763633  0.2989113 0.7000
                         ***Turning on DIIS***
   1  -1441.7605061369  -0.262608001152 0.88234063  0.01388441  0.0944375 0.0000
   2  -1441.4732415586   0.287264578307 0.11142170  0.00283563  0.2353949 0.7000
   3  -1441.7133405526  -0.240098993967 0.13296844  0.00300970  0.1969393 0.7000
   4  -1441.8756771704  -0.162336617820 0.12684567  0.00251575  0.1502491 0.7000
   5  -1441.9617903323  -0.086113161899 0.09818570  0.00195646  0.1031261 0.7000
   6  -1442.0049616788  -0.043171346504 0.26817199  0.00537057  0.0669699 0.0000
   7  -1442.0598897762  -0.054928097428 0.04412999  0.00080076  0.0423744 0.0000
   8  -1442.0665608972  -0.006671121001 0.04285625  0.00064591  0.0175566 0.0000
   9  -1442.0686166076  -0.002055710323 0.02519628  0.00048681  0.0045225 0.0000
  10  -1442.0690784815  -0.000461873967 0.00652186  0.00017711  0.0024081 0.0000
```

```
   11  -1442.0691994344  -0.000120952896 0.00318862  0.00009817  0.0013287 0.0000
   12  -1442.0692370986  -0.000037664145 0.00289703  0.00006658  0.0010036 0.0000
   13  -1442.0692557561  -0.000018657537 0.00872807  0.00026481  0.0006098 0.0000
   14  -1442.0692790592  -0.000023303114 0.00136738  0.00001713  0.0019054 0.0000
   15  -1442.0692841649  -0.000005105642 0.00194277  0.00004165  0.0007584 0.0000
   16  -1442.0692857367  -0.000001571858 0.00043978  0.00001226  0.0002705 0.0000
   17  -1442.0692859394  -0.000000202716 0.00028171  0.00000515  0.0001447 0.0000
   18  -1442.0692859977  -0.000000058216 0.00014881  0.00000277  0.0000886 0.0000
                  **** Energy Check signals convergence ****
```

Thus, the job converges almost monotonically but it does so somewhat slowly towards the end. The second job, however:

```
   ITER      Energy        Delta-E       Max-DP      RMS-DP      [F,P]     Damp
            *** Starting incremental Fock matrix formation  ***
    0  -1441.4978981358   0.000000000000 0.40179217  0.00763633  0.2989113 0.7000
                       ***Turning on DIIS***
    1  -1441.7605061369  -0.262608001152 0.88234063  0.01388441  0.0944375 0.0000
    2  -1441.4732415586   0.287264578307 0.11142170  0.00283563  0.2353949 0.7000
    3  -1441.7133405526  -0.240098993967 0.13296844  0.00300970  0.1969393 0.7000
    4  -1441.8756771704  -0.162336617820 0.12684567  0.00251575  0.1502491 0.7000
    5  -1441.9617903323  -0.086113161899 0.09818570  0.00195646  0.1031261 0.7000
    6  -1442.0049616788  -0.043171346504 0.26817199  0.00537057  0.0669699 0.0000
    7  -1442.0598897762  -0.054928097428 0.04412999  0.00080076  0.0423744 0.0000
    8  -1442.0665608972  -0.006671121001 0.04285625  0.00064591  0.0175566 0.0000
                *** Initiating the Newton-Raphson procedure ***
                      *** Shutting down DIIS ***
                *** Removing any level shift ***
   ITER      Energy      Delta-E       Grad       Rot      Max-DP     RMS-DP
    9  -1442.06861661  -0.0020557103  0.002665  0.000000  0.025196  0.000487
       CP-SCF ITERATION   0:
       CP-SCF ITERATION   1:       0.000033353
       CP-SCF ITERATION   2:       0.000001802
       CP-SCF ITERATION   3:       0.000000887
       CP-SCF ITERATION   4:       0.000000014
   10  -1442.06907848  -0.0004618740  0.008069  0.035908  0.017951  0.000570
       CP-SCF ITERATION   0:
       CP-SCF ITERATION   1:       0.000000367
       CP-SCF ITERATION   2:       0.000000013
   11  -1442.06928354  -0.0002050609  0.000678  0.004293  0.002640  0.000063
       CP-SCF ITERATION   0:
       CP-SCF ITERATION   1:       0.000000001
   12  -1442.06928602  -0.0000024726  0.000066  0.000109  0.000051  0.000002
                <<< The NR Solver signals convergence >>>
```

Thus, after reaching the threshold for initiating the Newton-Raphson procedure after nine iterations, the job takes only three more iterations to converge to the correct solution. Since each micro-iteration in the CP-SCF procedure roughly corresponds to the formation of one Fock-matrix the second job is still somewhat more expensive. You cannot always expect the Newton-Raphson procedure to converge[9] and it will not converge at all if your SCF solution is not stable (i.e. if the orbital Hessian has negative eigenvalues) – still, it is a powerfull technique to try if other alternatives

---

[9] For example try `%scf nr start 0.1 end end` to turn on the NRSCF after two iterations in the example above – it will run into trouble.

do not do well. It is important, however, to bring the SCF into the radius of convergence of the Newton-Raphson procedure for it to be efficient.

The implementation covers closed-shell and spin-unrestricted Hartree-Fock and DFT calculations.

5.1.6 Counterpoise Correction

In calculating weak molecular interactions the nasty subject of the basis set superposition error (BSSE) arises. It consists of the fact that if one describes a dimer, the basis functions on A help to lower the energy of fragment B and vice versa. Thus, one obtains an energy that is biased towards the dimer formation due to basis set effects. Since this is unwanted, the Boys and Bernardi procedure aims to correct for this deficiency by estimating what the energies of the monomers would be if they had been calculated with the dimer basis set. This will stabilize the monomers relative to the dimers. The effect can be a quite sizeable fraction of the interaction energy and should therefore be taken into account. The original Boys and Bernardi formula for the interaction energy between fragments A and B is:

$$\Delta E = E_{AB}^{AB}(AB) - E_A^A(A) - E_B^B(B) - \left[E_A^{AB}(AB) - E_A^{AB}(A) + E_B^{AB}(AB) - E_B^{AB}(B) \right] \qquad (5.9)$$

Here $E_X^Y(Z)$ is the energy of fragment X calculated at the optimized geometry of fragment Y with the basis set of fragment Z. Thus, you need to do a total the following series of calculations: (1) optimize the geometry of the dimer and the monomers with some basis set Z. This gives you $E_{AB}^{AB}(AB)$, $E_A^A(A)$ and $E_B^B(B)$ (2) delete fragment A (B) from the optimized structure of the dimer and re-run the single point calculation with basis set Z. This gives you $E_B^{AB}(B)$ and $E_A^{AB}(A)$. (3) Now, the final calculation consists of calculating the energies of A and B at the dimer geometry but with the dimer basis set. This gives you $E_A^{AB}(AB)$ and $E_B^{AB}(AB)$.

In order to achieve the last step efficiently, a special notation was put into ORCA which allows you to delete the electrons and nuclear charges that come with certain atoms but retain the assigned basis set. This trick consists of putting a ":" after the atoms name. Here is an example of how to run such a calculation of the water dimer at the MP2 level (with frozen core):

```
#
# BSSE test
#


# ------------------------------------------
# First the monomer. It is a waste of course
# to run the monomer twice ...
# ------------------------------------------
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "monomer"
* xyz 0 1
  O     7.405639     6.725069     7.710504
  H     7.029206     6.234628     8.442160
  H     8.247948     6.296600     7.554030
*


$new_job
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "monomer"
* xyz 0 1
  O     7.405639     6.725069     7.710504
  H     7.029206     6.234628     8.442160
```

```
 H      8.247948      6.296600      7.554030
*


# -----------------------------------------
# now the dimer
# -----------------------------------------
$new_job
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "dimer"
* xyz 0 1
O   7.439917   6.726792   7.762120
O   5.752050   6.489306   5.407671
H   7.025510   6.226170   8.467436
H   8.274883   6.280259   7.609894
H   6.313507   6.644667   6.176902
H   5.522285   7.367132   5.103852
*


# -----------------------------------------
# Now the calculations of the monomer at the
# dimer geometry
# -----------------------------------------
$new_job
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "monomer_1"

* xyz 0 1
O   7.439917   6.726792   7.762120
H   7.025510   6.226170   8.467436
H   8.274883   6.280259   7.609894
*


$new_job
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "monomer_1"
* xyz 0 1
O   5.752050   6.489306   5.407671
H   6.313507   6.644667   6.176902
H   5.522285   7.367132   5.103852
*


# -----------------------------------------
# Now the calculation of the monomer at the
# dimer geometry but with the dimer basis set
# -----------------------------------------
$new_job
! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
%id "monomer_2"
* xyz 0 1
O   7.439917   6.726792   7.762120
O : 5.752050   6.489306   5.407671
H   7.025510   6.226170   8.467436
H   8.274883   6.280259   7.609894
H : 6.313507   6.644667   6.176902
H : 5.522285   7.367132   5.103852
*
```

```
    $new_job
    ! RHF MP2 TZVPP VeryTightSCF XYZFile PModel
    %id "monomer_2"
    * xyz 0 1
    O : 7.439917    6.726792    7.762120
    O   5.752050    6.489306    5.407671
    H : 7.025510    6.226170    8.467436
    H : 8.274883    6.280259    7.609894
    H   6.313507    6.644667    6.176902
    H   5.522285    7.367132    5.103852
    *
```

You obtain the energies:

```
    Monomer :    -152.647062118 Eh
    Dimer :    -152.655623625 Eh    -5.372 kcal/mol
    Monomer at dimer geometry:   -152.647006948 Eh    0.035 kcal/mol
    Same with AB Basis set :   -152.648364970 Eh   -0.818 kcal/mol


    Thus, the corrected interaction energy is:
    -5.372 kcal/mol - (-0.818-0.035)=-4.52 kcal/mol
```

5.1.7 Complete Active Space Self-Consistent Field Method

5.1.7.1 Introduction

There are several situations where a complete-active space self-consistent field (CASSCF) treatment is a good idea:

- Wavefunctions with significant multireference character arising from several nearly degenerate configurations (static correlation)

- Wavefunctions which require a multideterminantal treatment (for example multiplets of atoms, ions, transition metal complexes, . . .)

- Situations in which bonds are broken or partially broken.

- Generation of orbitals which are a compromise between the requirements for several states.

- Generation of start orbitals for treatment of dynamic correlation via the MRCI module.

- Generation of genuine spin eigenfunctions for multideterminantal/multireference wavefunctions.

In all of these cases the single-determinantal Hartree-Fock method fails badly and in most of these cases DFT methods will also fail. In these cases a CASSCF method is a good starting point. CASSCF is a special case of multconfigurational SCF (MCSCF) methods which specialize to the situation where the orbitals are divided into three-subspaces: (a) the internal orbitals which are doubly occupied in all configuration state functions (CSFs), (b) partially occupied (active) orbitals and (c) virtual (external) orbitals which are empty in all CSFs. A fixed number of electrons is assigned to the internal subspace and the active subspace. If N-electrons are "active" in M orbitals one speaks of a CASSCF(N,M) wavefunctions. All spin-eigenfunctions for N-electrons in M orbitals are included in the configuration interaction step and the energy is made stationary with respect to variations in the MO and the CI

coefficients. Any number of roots of any number of different multiplicities can be calculated and the CASSCF energy may be optimized with respect to a user defined average of these states.

The CASSCF method has the nice advantage that it is fully variational which renders the calculation of analytical gradients relatively easy. Thus, the CASSCF method may be used for geometry optimizations and numerical frequency calculations.

The price to pay for this strongly enhanced flexibility relative to the single-determinantal HF method is that the CASSCF method requires more computational ressources and also more insight and planning from the user side. The technical details are explained in section 6.9. Here we explain the use of the CASSCF method by examples.

Properties such as EPR parameters are available in the framework of the quasi-degenerate perturbation theory. For a detailed description of the available properties and options see section 6.9.2. **NOTE: Properties that do not require the response of the wavefunction can be computed with the EPRNMR module as well e.g. Mößbauer parameters**

5.1.7.2 A simple Example

One standard example of a multireference system is the Be atom. Let us run two calculations, a standard closed shell calculation $(1s^2 2s^2)$. The second calculation is a CASSCF(2,4) calculation which also includes the $(1s^2 2s^1 2p^1)$ and $(1s^2 2s^0 2p^2)$ configurations.

```
! TZVPP TightSCF
* xyz 0 1
Be 0 0 0
*
```

This standard closed shell calculations yields the energy `-14.56213241 Eh`. The CASSCF calculation:

```
! TZVPP TightSCF
%casscf nel   2
        norb 4
        end
* xyz 0 1
Be 0 0 0
*
```

yields the energy `-14.605381525 Eh`. Thus, the inclusion of the 2p shell results in an energy lowering of 43 mEh which is considerable. The CASSCF program also prints the composition of the wavefunction:

```
    ----------------------------------------------
    CAS-SCF STATES FOR BLOCK  1 MULT= 1 NROOTS= 1
    ----------------------------------------------


    ROOT   0:  E=      -14.6053815294 Eh
        0.90060 [    0]: 2000
```

```
        0.03313 [    4]: 0200
        0.03313 [    9]: 0002
        0.03313 [    7]: 0020
```

This information is to be read as follows: The lowest state is composed of 90% of the configuration which has the active space occupation pattern 2000 which means that the first active orbital doubly occupied in this configuration while the other three are empty. The MO vector composition tells us what these orbitals are (ORCA uses natural orbitals to canonicalize the active space).

```
                        0         1         2         3         4         5
                    -4.70502  -0.27270   0.11579   0.11579   0.11579   0.16796
                     2.00000   1.80121   0.06626   0.06626   0.06626   0.00000

                    --------  --------  --------  --------  --------  --------
    0 Be  s           100.0     100.0       0.0       0.0       0.0     100.0
    0 Be  pz            0.0       0.0      13.6       6.1      80.4       0.0
    0 Be  px            0.0       0.0       1.5      93.8       4.6       0.0
    0 Be  py            0.0       0.0      84.9       0.1      15.0       0.0
```

Thus, the first active space orbital has occupion number 1.80121 and is the Be-2s orbital. The other three orbitals are 2p in character and all have the same occupation number 0.06626. Since they are degenerate in occupation number space, they are arbitrary mixtures of the three 2p orbitals. It is then clear that the other components of the wavefunction (each with 3.31%) are those in which one of the 2p orbitals is doubly occupied.

How did we know how to put the 2s and 2p orbitals in the active space? The answer is – WE DID NOT KNOW! In this case it was "good luck" that the initial guess produced the orbitals in such an order that we had the 2s and 2p orbitals active. **IN GENERAL IT IS YOUR RESPONSIBILITY THAT THE ORBITALS ARE ORDERED SUCH THAT THE ORBITALS THAT YOU WANT IN THE ACTIVE SPACE COME IN THE DESIRED ORDER**. In many cases this will require re-ordering and **CAREFUL INSPECTION** of the starting orbitals.

ATTENTION:

- If you include orbitals in the active space that are nearly empty or nearly doubly occupied, convergence problems are likely (in many cases these are overcome by the Newton-Raphson method).

5.1.7.3 Starting Orbitals

TIP

- In many cases natural orbitals of a simple correlated calculation of some kind provide a good starting point for CASSCF.

Let us illustrate this principle with a calculation on the Benzene molecule where we want to include all six π-orbitals in the active space. After doing a RHF calculation:

```
! RHF SV(P)

* int 0 1
C 0 0 0 0.000000 0.000 0.000
C 1 0 0 1.389437 0.000 0.000
C 2 1 0 1.389437 120.000 0.000
C 3 2 1 1.389437 120.000 0.000
C 4 3 2 1.389437 120.000 0.000
C 5 4 3 1.389437 120.000 0.000
H 1 2 3 1.082921 120.000 180.000
H 2 1 3 1.082921 120.000 180.000
H 3 2 1 1.082921 120.000 180.000
H 4 3 2 1.082921 120.000 180.000
H 5 4 3 1.082921 120.000 180.000
H 6 5 4 1.082921 120.000 180.000
*
%Output
        Print[P_ReducedOrbPopMO_L]       1
End
```

We can look at the orbitals around the HOMO/LUMO gap:

```
                  12        13        14        15        16        17
               -0.63810  -0.62613  -0.59153  -0.59153  -0.50570  -0.49833
                2.00000   2.00000   2.00000   2.00000   2.00000   2.00000
               --------  --------  --------  --------  --------  --------
   0 C  s         2.9       0.0       0.3       0.1       0.0       0.0
   0 C  pz        0.0       0.0       0.0       0.0      16.5       0.0
   0 C  px        1.4      12.4       5.9       0.3       0.0      11.2
   0 C  py        4.2       4.1      10.1       5.9       0.0       0.1
   0 C  dyz       0.0       0.0       0.0       0.0       0.1       0.0
   0 C  dx2y2     0.1       0.1       0.2       0.2       0.0       0.5
   0 C  dxy       0.4       0.0       0.0       0.2       0.0       0.0
   1 C  s         2.9       0.0       0.3       0.1       0.0       0.0
   1 C  pz        0.0       0.0       0.0       0.0      16.5       0.0
   1 C  px        1.4      12.4       5.9       0.3       0.0      11.2
   1 C  py        4.2       4.1      10.1       5.9       0.0       0.1
   1 C  dyz       0.0       0.0       0.0       0.0       0.1       0.0
   1 C  dx2y2     0.1       0.1       0.2       0.2       0.0       0.5
   1 C  dxy       0.4       0.0       0.0       0.2       0.0       0.0
   2 C  s         2.9       0.0       0.0       0.4       0.0       0.1
   2 C  pz        0.0       0.0       0.0       0.0      16.5       0.0
   2 C  px        5.7       0.0       0.0      20.9       0.0      10.1
   2 C  py        0.0      16.5       1.3       0.0       0.0       0.0
   2 C  dxz       0.0       0.0       0.0       0.0       0.1       0.0
   2 C  dx2y2     0.6       0.0       0.0       0.2       0.0       1.2
   2 C  dxy       0.0       0.1       0.5       0.0       0.0       0.0
   3 C  s         2.9       0.0       0.3       0.1       0.0       0.0
   3 C  pz        0.0       0.0       0.0       0.0      16.5       0.0
```

3	C	px	1.4	12.4	5.9	0.3	0.0	11.2
3	C	py	4.2	4.1	10.1	5.9	0.0	0.1
3	C	dyz	0.0	0.0	0.0	0.0	0.1	0.0
3	C	dx2y2	0.1	0.1	0.2	0.2	0.0	0.5
3	C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
4	C	s	2.9	0.0	0.3	0.1	0.0	0.0
4	C	pz	0.0	0.0	0.0	0.0	16.5	0.0
4	C	px	1.4	12.4	5.9	0.3	0.0	11.2
4	C	py	4.2	4.1	10.1	5.9	0.0	0.1
4	C	dyz	0.0	0.0	0.0	0.0	0.1	0.0
4	C	dx2y2	0.1	0.1	0.2	0.2	0.0	0.5
4	C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
5	C	s	2.9	0.0	0.0	0.4	0.0	0.1
5	C	pz	0.0	0.0	0.0	0.0	16.5	0.0
5	C	px	5.7	0.0	0.0	20.9	0.0	10.1
5	C	py	0.0	16.5	1.3	0.0	0.0	0.0
5	C	dxz	0.0	0.0	0.0	0.0	0.1	0.0
5	C	dx2y2	0.6	0.0	0.0	0.2	0.0	1.2
5	C	dxy	0.0	0.1	0.5	0.0	0.0	0.0
6	H	s	7.5	0.0	7.5	2.5	0.0	2.5
7	H	s	7.5	0.0	7.5	2.5	0.0	2.5
8	H	s	7.5	0.0	0.0	10.0	0.0	9.9
9	H	s	7.5	0.0	7.5	2.5	0.0	2.5
10	H	s	7.5	0.0	7.5	2.5	0.0	2.5
11	H	s	7.5	0.0	0.0	10.0	0.0	9.9

			18	19	20	21	22	23
			-0.49833	-0.33937	-0.33937	0.13472	0.13472	0.18198
			2.00000	2.00000	2.00000	0.00000	0.00000	0.00000
			--------	--------	--------	--------	--------	--------
0	C	s	0.1	0.0	0.0	0.0	0.0	2.2
0	C	pz	0.0	8.1	24.4	7.8	23.4	0.0
0	C	px	0.1	0.0	0.0	0.0	0.0	0.6
0	C	py	10.4	0.0	0.0	0.0	0.0	1.7
0	C	dxz	0.0	0.4	0.2	0.7	0.7	0.0
0	C	dyz	0.0	0.2	0.0	0.7	0.0	0.0
0	C	dx2y2	0.0	0.0	0.0	0.0	0.0	0.2
0	C	dxy	1.0	0.0	0.0	0.0	0.0	0.5
1	C	s	0.1	0.0	0.0	0.0	0.0	2.2
1	C	pz	0.0	8.1	24.4	7.8	23.4	0.0
1	C	px	0.1	0.0	0.0	0.0	0.0	0.6
1	C	py	10.4	0.0	0.0	0.0	0.0	1.7
1	C	dxz	0.0	0.4	0.2	0.7	0.7	0.0
1	C	dyz	0.0	0.2	0.0	0.7	0.0	0.0
1	C	dx2y2	0.0	0.0	0.0	0.0	0.0	0.2
1	C	dxy	1.0	0.0	0.0	0.0	0.0	0.5
2	C	s	0.0	0.0	0.0	0.0	0.0	2.2
2	C	pz	0.0	32.5	0.0	31.2	0.0	0.0
2	C	px	0.0	0.0	0.0	0.0	0.0	2.2
2	C	py	11.6	0.0	0.0	0.0	0.0	0.0
2	C	dxz	0.0	0.1	0.0	0.3	0.0	0.0
2	C	dyz	0.0	0.0	0.8	0.0	1.8	0.0
2	C	dx2y2	0.0	0.0	0.0	0.0	0.0	0.7
2	C	dxy	0.4	0.0	0.0	0.0	0.0	0.0
3	C	s	0.1	0.0	0.0	0.0	0.0	2.2
3	C	pz	0.0	8.1	24.4	7.8	23.4	0.0

```
   3 C  px           0.1      0.0      0.0      0.0      0.0      0.6
   3 C  py          10.4      0.0      0.0      0.0      0.0      1.7
   3 C  dxz          0.0      0.4      0.2      0.7      0.7      0.0
   3 C  dyz          0.0      0.2      0.0      0.7      0.0      0.0
   3 C  dx2y2        0.0      0.0      0.0      0.0      0.0      0.2
   3 C  dxy          1.0      0.0      0.0      0.0      0.0      0.5
   4 C  s            0.1      0.0      0.0      0.0      0.0      2.2
   4 C  pz           0.0      8.1     24.4      7.8     23.4      0.0
   4 C  px           0.1      0.0      0.0      0.0      0.0      0.6
   4 C  py          10.4      0.0      0.0      0.0      0.0      1.7
   4 C  dxz          0.0      0.4      0.2      0.7      0.7      0.0
   4 C  dyz          0.0      0.2      0.0      0.7      0.0      0.0
   4 C  dx2y2        0.0      0.0      0.0      0.0      0.0      0.2
   4 C  dxy          1.0      0.0      0.0      0.0      0.0      0.5
   5 C  s            0.0      0.0      0.0      0.0      0.0      2.2
   5 C  pz           0.0     32.5      0.0     31.2      0.0      0.0
   5 C  px           0.0      0.0      0.0      0.0      0.0      2.2
   5 C  py          11.6      0.0      0.0      0.0      0.0      0.0
   5 C  dxz          0.0      0.1      0.0      0.3      0.0      0.0
   5 C  dyz          0.0      0.0      0.8      0.0      1.8      0.0
   5 C  dx2y2        0.0      0.0      0.0      0.0      0.0      0.7
   5 C  dxy          0.4      0.0      0.0      0.0      0.0      0.0
   6 H  s            7.4      0.0      0.0      0.0      0.0     11.5
   7 H  s            7.4      0.0      0.0      0.0      0.0     11.5
   8 H  s            0.0      0.0      0.0      0.0      0.0     11.5
   9 H  s            7.4      0.0      0.0      0.0      0.0     11.5
  10 H  s            7.4      0.0      0.0      0.0      0.0     11.5
  11 H  s            0.0      0.0      0.0      0.0      0.0     11.5
```

We see that the occupied π-orbitals number 16,19,20 and the unoccupied ones start with 21 and 22. However, the six'th high-lying π^*-orbital cannot easily be found. Thus, let us run a simple selected CEPA/2 calculation and look at the natural orbitals.

```
! RHF SV(P) SV/C
! moread
%moinp "Test-CASSCF-Benzene-1.gbw"

%mrci citype cepa2
      tsel 1e-5
      natorbiters 1
      newblock 1 *
        nroots 1
        refs cas(0,0) end
        end
      end
# ...etc, input of coordinates
```

The calculation prints the occupation numbers:

```
N[  6] =    1.98784765
N[  7] =    1.98513069
N[  8] =    1.98508633
N[  9] =    1.97963799
N[ 10] =    1.97957039
N[ 11] =    1.97737886
N[ 12] =    1.97509724
N[ 13] =    1.97370616
N[ 14] =    1.97360821
N[ 15] =    1.96960145
N[ 16] =    1.96958645
N[ 17] =    1.96958581
N[ 18] =    1.95478929
N[ 19] =    1.91751184
N[ 20] =    1.91747498
N[ 21] =    0.07186879
N[ 22] =    0.07181758
N[ 23] =    0.03203528
N[ 24] =    0.01766832
N[ 25] =    0.01757735
N[ 26] =    0.01708578
N[ 27] =    0.01707675
N[ 28] =    0.01671912
N[ 29] =    0.01526139
N[ 30] =    0.01424982
```

From these occupation number it becomes evident that there are several natural orbitals which are not quite doubly occupied MOs. Those with an occupation number of 1.95 and less should certainly be taken as active. In addition the rather strongly occupied virtual MOs 21-23 should also be active leading to CASSCF(6,6). Let us see what these orbitals are before starting CASSCF:

```
! RHF SV(P) SV/C
! moread noiter
%moinp "Test-CASSCF-Benzene-2.mrci.nat"
```

Leading to:

		18	19	20	21	22	23
		1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
		1.95479	1.91751	1.91747	0.07187	0.07182	0.03204
		--------	--------	--------	--------	--------	--------
0 C	pz	16.5	8.1	24.4	23.4	7.8	16.1
0 C	dxz	0.0	0.4	0.2	0.6	0.9	0.1
0 C	dyz	0.1	0.2	0.0	0.0	0.6	0.4
1 C	pz	16.5	8.1	24.4	23.5	7.8	16.1

```
   1 C   dxz           0.0      0.4      0.2      0.6      0.9      0.1
   1 C   dyz           0.1      0.2      0.0      0.0      0.6      0.4
   2 C   pz           16.5     32.5      0.0      0.0     31.3     16.3
   2 C   dxz           0.1      0.1      0.0      0.0      0.2      0.5
   2 C   dyz           0.0      0.0      0.8      1.9      0.0      0.0
   3 C   pz           16.5      8.1     24.4     23.4      7.8     16.1
   3 C   dxz           0.0      0.4      0.2      0.6      0.9      0.1
   3 C   dyz           0.1      0.2      0.0      0.0      0.6      0.4
   4 C   pz           16.5      8.1     24.4     23.5      7.8     16.1
   4 C   dxz           0.0      0.4      0.2      0.6      0.9      0.1
   4 C   dyz           0.1      0.2      0.0      0.0      0.6      0.4
   5 C   pz           16.5     32.5      0.0      0.0     31.3     16.3
   5 C   dxz           0.1      0.1      0.0      0.0      0.2      0.5
   5 C   dyz           0.0      0.0      0.8      1.9      0.0      0.0
```

This shows us that these six orbitals are precisely the π/π^* orbitals that we wanted to have active (you can also plot them to get even more insight).

Now we know that the desired orbitals are in the correct order, we can do CASSCF:

```
! RHF SV(P)
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"

%casscf  nel    6
         norb   6
         nroots 1
         mult   1
         switchstep nr
         end
```

Here we have also used a special feature of the CASSCF program – the second order convergence provided by the Newton-Raphson method after a certain convergence has been reached (the `switchstep nr` statement). The output of the CASSCF program is:

```
------------------
CAS-SCF ITERATIONS
------------------


MACRO-ITERATION   1:
   --- Canonicalize Internal Space
   --- Canonicalize External Space
   --- Inactive Energy E0 = -224.09726145 Eh
CI-ITERATION    0:
  -230.588262104   0.000000000000 (     0.01)
   CI-PROBLEM SOLVED
   DENSITIES MADE
```

```
                   <<<<<<<<<<<<<<<<INITIAL CI STATE CHECK>>>>>>>>>>>>>>>>>

   BLOCK  1 MULT= 1 NROOTS= 1
   ROOT  0: E=     -230.5882621045 Eh
        0.89515 [     0]: 222000
        0.02886 [    14]: 211110
        0.01975 [    29]: 202020
        0.01971 [     4]: 220200
        0.01178 [    65]: 112011
        0.01170 [    50]: 121101


                   <<<<<<<<<<<<<<<<INITIAL CI STATE CHECK>>>>>>>>>>>>>>>>>

     E(CAS)=  -230.588262104 Eh DE=      0.000000000
     N(occ)=  1.96394 1.90967 1.90990 0.09156 0.09174 0.03318
     ||g|| =      0.105153413 Max(G)=    -0.051994667 Rot=41,19
     --- Orbital Update [   SuperCI]
     SUPERCI-ITER   0: DE=    -0.001066040 <r|r>=      0.000008900
     SUPERCI-ITER   1: DE=    -0.000010417 <r|r>=      0.000001330
     SUPERCI-ITER   2: DE=    -0.000000878 <r|r>=      0.000000019
     SUPERCI-ITER   3: DE=    -0.000000010
     <Psi(SX)|Psi(SX)>=      0.999999875 DE(SX)=-0.001077344
     Largest coefficient in the normalized |SX> wavefunction= 0.012026247
     --- Forming Natural Orbitals
     --- Density off from convergence (9.668e-01). Exact active Fock build


   MACRO-ITERATION   2:
     --- Canonicalize Internal Space
     --- Canonicalize External Space
     --- Inactive Energy E0 = -224.09482620 Eh
   CI-ITERATION   0:
     -230.589517156   0.000000000000 (    0.01)
     CI-PROBLEM SOLVED
     DENSITIES MADE
     E(CAS)=  -230.589517156 Eh DE=     -0.001255051
     N(occ)=  1.96277 1.90651 1.90639 0.09498 0.09490 0.03445
     ||g|| =      0.060116073 Max(G)=     0.028742952 Rot=45,20
     --- Orbital Update [   SuperCI]
     SUPERCI-ITER   0: DE=    -0.000434389 <r|r>=      0.000000001
     <Psi(SX)|Psi(SX)>=      1.000000014 DE(SX)=-0.000434389
     Largest coefficient in the normalized |SX> wavefunction= 0.007779462
     --- Forming Natural Orbitals
     --- Density close to convergence (3.348e-03). Approximate active Fock build


   MACRO-ITERATION   3:
     --- Canonicalize Internal Space
     --- Canonicalize External Space
     --- Inactive Energy E0 = -224.08994424 Eh
   CI-ITERATION   0:
     -230.589969060   0.000000000000 (    0.01)
     CI-PROBLEM SOLVED
     DENSITIES MADE
     E(CAS)=  -230.589969060 Eh DE=     -0.000451905
     N(occ)=  1.96215 1.90471 1.90465 0.09670 0.09666 0.03513
     ||g|| =      0.035627381 Max(G)=     0.016592318 Rot=44,19
     --- Orbital Update [   SuperCI]
```

```
     SUPERCI-ITER   0: DE=     -0.000187355 <r|r>=        0.000000000
     <Psi(SX)|Psi(SX)>=        1.000000001 DE(SX)=-0.000187355
     Largest coefficient in the normalized |SX> wavefunction= 0.005579610
     --- Forming Natural Orbitals
     --- Density close to convergence (1.773e-03). Approximate active Fock build


  MACRO-ITERATION   4:
     --- Canonicalize Internal Space
     --- Canonicalize External Space
     --- Inactive Energy E0 = -224.08604939 Eh
  CI-ITERATION   0:
   -230.590144691   0.000000000000 (    0.01)
     CI-PROBLEM SOLVED
     DENSITIES MADE
     E(CAS)= -230.590144691 Eh DE=    -0.000175631
     N(occ)=  1.96181 1.90376 1.90373 0.09761 0.09759 0.03551
     ||g|| =      0.021716727 Max(G)=    -0.009927233 Rot=44,19
     ===>>> Convergence to 3.0e-02 achieved - switching to Step=NR
     --- Orbital Update [        NR]
     --- S-Fitting active orbitals
     --- Density close to convergence (9.630e-04). Approximate active Fock build


  MACRO-ITERATION   5:
     --- Inactive Energy E0 = -224.07780923 Eh
  CI-ITERATION   0:
   -230.590271361   0.000000000000 (    0.01)
     CI-PROBLEM SOLVED
     DENSITIES MADE
     E(CAS)= -230.590271361 Eh DE=    -0.000126670
     N(occ)=  1.96137 1.90273 1.90273 0.09861 0.09860 0.03596
     ||g|| =      0.000734961 Max(G)=    -0.000332998 Rot=44,19
                    ---- THE CAS-SCF GRADIENT HAS CONVERGED ----
                        --- FINALIZING ORBITALS ---
                    ---- DOING ONE FINAL ITERATION FOR PRINTING ----
     --- Forming Natural Orbitals
     --- Canonicalize Internal Space
     --- Canonicalize External Space


  MACRO-ITERATION   6:
     --- Inactive Energy E0 = -224.07780923 Eh
  CI-ITERATION   0:
   -230.590271360   0.000000000000 (    0.01)
     CI-PROBLEM SOLVED
     DENSITIES MADE
     E(CAS)= -230.590271360 Eh DE=     0.000000001
     N(occ)=  1.96137 1.90273 1.90273 0.09861 0.09860 0.03596
     ||g|| =      0.000798884 Max(G)=     0.000333015 Rot=44,19
  --------------
  CASSCF RESULTS
  --------------


  Final CASSCF energy       : -230.590271360 Eh   -6274.6803 eV
```

First of all you can see how the program cycles between CI-vector optimization and orbital optimization steps (so-called unfolded two-step procedure). After 4 iterations, the program switches to the Newton-Raphson solver which then converges very rapidly. In the CASSCF program this produces lengthy integrals and Hessian files and therefore the

size of molecule that can be treated is somewhat limited.

5.1.7.4 CASSCF and Symmetry

The CASSCF program can make some use of symmetry. Thus, it is possible to do the CI calculations separated by irreducible representations. This allows one to calculate electronic states in a more controlled fashion.

Let us look at a simple example: C_2H_4. We first generate symmetry adapted MP2 natural orbitals:

```
! def2-TZVP def2-TZV/C VeryTightSCF Conv UseSym RI-MP2
%mp2 density relaxed donatorbs true end
* int 0 1
C 0 0 0  0 0 0
C 1 0 0  1.35 0 0
H 1 2 0  1.1 120 0
H 1 2 3  1.1 120 180
H 2 1 3  1.1 120 0
H 2 1 3  1.1 120 180
*
```

The program does the following. It first identifies the group correctly as D_{2h} and sets up its irreducible representations:

```
------------------
SYMMETRY DETECTION
------------------
Preparing Data                  ... done
Detection Threshold:    SymThresh ... 1.0000e-04

Point Group will now be determined:
Moving molecule to center of mass ... done

POINT GROUP                     ... D2h

The coordinates will now be cleaned:
Moving to standard coord frame    ... done
(Changed the main axis to z and one of the C2's to x)
Structure cleanup requested       ... yes
Selected point group              ... D2h
Cleaning Tolerance     SymThresh ... 1.0000e-04

Some missing point group data is constructed:
Constructing symmetry operations  ... done
Creating atom transfer table      ... done
Creating asymmetric unit          ... done

Cleaning coordinates              ... done


----------------------------------------------
SYMMETRY-PERFECTED CARTESIAN COORDINATES (A.U.)
----------------------------------------------
```

```
   0 C     1.27556514    0.00000000    0.00000000
   1 C    -1.27556514    0.00000000    0.00000000
   2 H     2.31491451    1.80020592    0.00000000
   3 H     2.31491451   -1.80020592    0.00000000
   4 H    -2.31491451    1.80020592    0.00000000
   5 H    -2.31491451   -1.80020592    0.00000000


------------------
SYMMETRY REDUCTION
------------------
ORCA supports only abelian point groups.
It is now checked, if the determined point group is supported:
Point Group ( D2h   ) is           ... supported

(Re)building abelian point group:
Creating Character Table           ... done
Making direct product table        ... done


---------------------
ASYMMETRIC UNIT IN D2h
---------------------

  # AT    MASS             COORDS (A.U.)               BAS
   0 C   12.0110    1.27556514    0.00000000    0.00000000    0
   2 H    1.0080    2.31491451    1.80020592    0.00000000    0


---------------------
SYMMETRY ADOPTED BASIS
---------------------

The coefficients for the symmetry adopted linear combinations (SALCS)
of basis functions will now be computed:
Number of basis functions          ...    86
Preparing memory                   ... done
Constructing Gamma(red)            ... done
Reducing Gamma(red)                ... done
Constructing SALCs                 ... done
Checking SALC integrity            ... nothing suspicious
Normalizing SALCs                  ... done

Storing the symmetry object:
Symmetry file                      ... Test-SYM-CAS-C2H4-1.sym.tmp
Writing symmetry information        ... done
```

It then performs the SCF calculation and keeps the symmetry in the molecular orbitals.

```
   NO   OCC        E(Eh)        E(eV)    Irrep
    0   2.0000    -11.236728   -305.7669    1-Ag
    1   2.0000    -11.235157   -305.7242    1-B3u
    2   2.0000     -1.027144    -27.9500    2-Ag
    3   2.0000     -0.784021    -21.3343    2-B3u
    4   2.0000     -0.641566    -17.4579    1-B2u
    5   2.0000     -0.575842    -15.6694    3-Ag
    6   2.0000     -0.508313    -13.8319    1-B1g
    7   2.0000     -0.373406    -10.1609    1-B1u
    8   0.0000      0.139580      3.7982    1-B2g
    9   0.0000      0.171982      4.6799    4-Ag
```

```
   10    0.0000      0.195186       5.3113    3-B3u
   11    0.0000      0.196786       5.3548    2-B2u
   12    0.0000      0.242832       6.6078    2-B1g
   13    0.0000      0.300191       8.1686    5-Ag
   14    0.0000      0.326339       8.8801    4-B3u
  ... etc
```

The MP2 module does not take any advantage of this information but produces natural orbitals that are symmetry adapted:

```
 N[  0](B3u) =   2.00000360
 N[  1]( Ag) =   2.00000219
 N[  2]( Ag) =   1.98056435
 N[  3](B3u) =   1.97195041
 N[  4](B2u) =   1.96746753
 N[  5](B1g) =   1.96578954
 N[  6]( Ag) =   1.95864726
 N[  7](B1u) =   1.93107098
 N[  8](B2g) =   0.04702701
 N[  9](B3u) =   0.02071784
 N[ 10](B2u) =   0.01727252
 N[ 11]( Ag) =   0.01651489
 N[ 12](B1g) =   0.01602695
 N[ 13](B3u) =   0.01443373
 N[ 14](B1u) =   0.01164204
 N[ 15]( Ag) =   0.01008617
 N[ 16](B2u) =   0.00999302
 N[ 17]( Ag) =   0.00840326
 N[ 18](B3g) =   0.00795053
 N[ 19](B3u) =   0.00532044
 N[ 20]( Au) =   0.00450556
 etc.
```

From this information and visual inspection you will know what orbitals you will have in the active space:

These natural orbitals can then be fed into the CASSCF calculation. We perform a simple calculation in which we keep the ground state singlet (A_{1g} symmetry, irrep=0) and the first excited triplet state (B_{3u} symmetry, irrep=7). In general the ordering of irreps follows standard conventions and in case of doubt you will find the relevant number for each irrep in the output.

For example, here (using LargePrint):

```
    ---------------------------
    CHARACTER TABLE OF GROUP D2h
    ---------------------------
    GAMMA  01   02   03   04   05   06   07   08
     Ag :  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
     B1g:  1.0  1.0 -1.0 -1.0  1.0  1.0 -1.0 -1.0
     B2g:  1.0 -1.0  1.0 -1.0  1.0 -1.0  1.0 -1.0
     B3g:  1.0 -1.0 -1.0  1.0  1.0 -1.0 -1.0  1.0
     Au :  1.0  1.0  1.0  1.0 -1.0 -1.0 -1.0 -1.0
     B1u:  1.0  1.0 -1.0 -1.0 -1.0 -1.0  1.0  1.0
     B2u:  1.0 -1.0  1.0 -1.0 -1.0  1.0 -1.0  1.0
```

```
   B3u:  1.0 -1.0 -1.0  1.0 -1.0  1.0  1.0 -1.0


   --------------------------------
   DIRECT PRODUCT TABLE OF GROUP D2h
   --------------------------------

      **   Ag  B1g B2g B3g Au  B1u B2u B3u

   Ag      Ag  B1g B2g B3g Au  B1u B2u B3u
   B1g     B1g Ag  B3g B2g B1u Au  B3u B2u
   B2g     B2g B3g Ag  B1g B2u B3u Au  B1u
   B3g     B3g B2g B1g Ag  B3u B2u B1u Au
   Au      Au  B1u B2u B3u Ag  B1g B2g B3g
   B1u     B1u Au  B3u B2u B1g Ag  B3g B2g
   B2u     B2u B3u Au  B1u B2g B3g Ag  B1g
   B3u     B3u B2u B1u Au  B3g B2g B1g Ag
```

Our CASSCF input is:

```
! def2-TZVP def2-TZV/C VeryTightSCF Conv NormalPrint UseSym
! moread
%moinp "Test-SYM-CAS-C2H4-1.mp2nat"
%casscf nel            4
        norb           4
        # This is only here to show that KDIIS can be used
        # together with CASSCF
        orbstep    kdiis
        switchstep kdiis
        # the lowest singet and triplet states. The new feature
        # is the array 'irrep' that lets you give the irrep for
        # a given block. Thus, now you can have several blocks of
        # the same multiplicity but different spatial symmetry
        irrep       0,7
        mult        1,3
        nroots      1,1
        gtol        1e-6
        etol        1e-11
        end

* int 0 1
C 0 0 0  0 0 0
C 1 0 0  1.35 0 0
H 1 2 0  1.1 120 0
H 1 2 3  1.1 120 180
H 2 1 3  1.1 120 0
H 2 1 3  1.1 120 180
*
```

And gives:

```
     ------------
     SCF SETTINGS
     ------------
     Hamiltonian:
      Ab initio Hamiltonian  Method          .... Hartree-Fock(GTOs)


     General Settings:
      Integral files         IntName          .... Test-SYM-CAS-C2H4-1
      Hartree-Fock type      HFTyp            .... CASSCF
      Total Charge           Charge           ....    0
      Multiplicity           Mult             ....    1
      Number of Electrons    NEL              ....   16
      Basis Dimension        Dim              ....   86
      Nuclear Repulsion      ENuc             ....      32.9609050695 Eh


      Symmetry handling      UseSym           .... ON
      Point group                             .... D2h
      Used point group                        .... D2h
      Number of irreps                        .... 8
        Irrep   Ag has   19 symmetry adapted basis functions (ofs=   0)
        Irrep  B1g has   12 symmetry adapted basis functions (ofs=  19)
        Irrep  B2g has    8 symmetry adapted basis functions (ofs=  31)
        Irrep  B3g has    4 symmetry adapted basis functions (ofs=  39)
        Irrep   Au has    4 symmetry adapted basis functions (ofs=  43)
        Irrep  B1u has    8 symmetry adapted basis functions (ofs=  47)
        Irrep  B2u has   12 symmetry adapted basis functions (ofs=  55)
        Irrep  B3u has   19 symmetry adapted basis functions (ofs=  67)
```

And further in the CASCSF program:

```
      Symmetry handling      UseSym        ... ON
      Point group                          ... D2h
      Used point group                     ... D2h
      Number of irreps                     ... 8
        Irrep   Ag has   19 SALCs (ofs=   0) #(closed)=   2 #(active)=   1
        Irrep  B1g has   12 SALCs (ofs=  19) #(closed)=   1 #(active)=   0
        Irrep  B2g has    8 SALCs (ofs=  31) #(closed)=   0 #(active)=   1
        Irrep  B3g has    4 SALCs (ofs=  39) #(closed)=   0 #(active)=   0
        Irrep   Au has    4 SALCs (ofs=  43) #(closed)=   0 #(active)=   0
        Irrep  B1u has    8 SALCs (ofs=  47) #(closed)=   0 #(active)=   1
        Irrep  B2u has   12 SALCs (ofs=  55) #(closed)=   1 #(active)=   0
        Irrep  B3u has   19 SALCs (ofs=  67) #(closed)=   2 #(active)=   1
      Symmetries of active orbitals:
        MO =    6  IRREP= 0 (Ag)
        MO =    7  IRREP= 5 (B1u)
        MO =    8  IRREP= 2 (B2g)
        MO =    9  IRREP= 7 (B3u)

      Setting up the integral package      ... done
      Building the CAS space               ... done (7 configurations for Mult=1 Irrep=0)
      Building the CAS space               ... done (4 configurations for Mult=3 Irrep=7)
```

Note that the irrep occupations and active space irreps will be frozen to what they are upon entering the CASSCF program. This helps to avoid flipping electronic states or active orbitals in and out of the active space.

After which it smoothly converges to give:

```
6:  1.986258     -0.753012    -20.4905   3-Ag
7:  1.457849     -0.291201     -7.9240   1-B1u
8:  0.541977      0.100890      2.7454   1-B2g
9:  0.013915      0.964186     26.2368   3-B3u
```

As well as:

```
----------------------------
SA-CASSCF TRANSITION ENERGIES
----------------------------


LOWEST ROOT =    -78.110314788 Eh -2125.490 eV


STATE    ROOT MULT IRREP DE/a.u.    DE/eV    DE/cm**-1
  1:     0    3   B3u   0.163741    4.456   35937.1
```

5.1.7.5 RI and RIJCOSX approximations for CASSCF

A significant speedup of CASSCF calculations on larger molecules can be achieved with the RI and RIJCOSX approximation. Thus, simply choose:

```
! RHF SV(P) SV/C
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"

%casscf  nel    6
         norb   6
         nroots 1
         mult   1
         switchstep nr
         trafostep rimo  #(or simply ri if memory becomes tight)
         end
```

The energy of this calculation is `-230.590328 Eh` compared to the previous result `-230.590271 Eh`. Thus, the RI error is only 0.06 mEh which is certainly negligible for all intents and purposes. We can compare to a calculation done without the NR solver:

```
! RHF SV(P)
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"

%casscf  nel    6
         norb   6
         nroots 1
```

```
              mult    1
              end
```

This calculation takes 9 cycles to converge. In many (less trivial) cases, first order methods like SuperCI and DIIS may take many more iterations to converge if they converge at all. On the other hand, these iterations are much cheaper than NR iterations and therefore it may pay off to do a few iterations more rather than switching to the expensive second order methods. On top of that, second order methods are also less robust and may diverge in certain circumstances.

The **TrafoStep rimo** does only affect the integral transformation in CASSCF calculations while the Fock operators are still calculated in the standard way using four index integrals. In order to fully avoid any four-index integral evaluation you can significantly speed up the time needed in each iteration by specifying **! RIJCOSX** in addition to **TrafoStep rimo**. This leads to the same type of speedup that RIJCOSX shows for RHF and UHF calculations while introducing a very similar (small) error.

5.1.7.6 Breaking Chemical Bonds

Let us turn to the breaking of chemical bonds. The first (trivial) example is H_2:

```
   ! RHF SVP TightSCF

   %casscf nel        2
           norb       2
           mult       1
           nroots     1
           end

   %paras R [0.6 0.65 0.70 0.75 0.80 0.90 1.0 1.1 1.3 1.5 1.7
             2.0 2.2 2.4 2.6 2.9 3.2 3.5 3.8 4.1]
          end

   * xyz 0 1
   h 0 0 0
   h 0 0 {R}
   *
```

The potential energy surface may be contrasted with the ones obtained by RHF and broken-symmetry UHF calculations generated by:

```
   ! RHF SVP SV/C TightSCF
   # etc...
```

And

```
    ! UHF SVP SV/C TightSCF


%scf  FlipSpin 1 FinalMs 0.0
      end
```
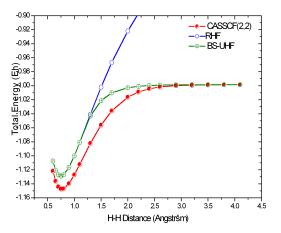

Figure 5.7: Potential Energy Surface of the H_2 molecule from RHF, UHF and CASSCF(2,2) calculations (SVP basis).

It is obvious, that the CASSCF surface is concise and yields the correct dissociation behavior. The RHF surface is roughly parallel to the CASSCF surface in the vicinity of the minimum but then starts to fail badly as the H-H bond starts to break. The broken-symmetry UHF solution is identical to RHF in the vicinity of the minimum and dissociates correctly. It is, however, of rather mediocre quality in the intermediate region wehre it follows the RHF surface too long too closely.

A more challenging case is to dissociate the N-N bond of the N_2 molecule correctly. Using CASSCF with the six p-orbitals we get a nice potential energy curve (The depth of the minimum is still too shallow compared to experiment by some 1 eV or so. A good dissociation energy requires a dynamic correlation treatment on top of CASSCF and a larger basis set).

One can use the H_2 example to illustrate the state-averaging feature. Since we have two active electrons we have two singlets and one triplet. Let us average the orbitals over these three states (we take equal weights for all states):

```
    ! RHF SVP TightSCF


%casscf nel          2
        norb         2
        mult         1,3
        nroots       2,1
        bweight    = 1,1
        weights[0] = 1,1
        weights[1] = 1
        end
```
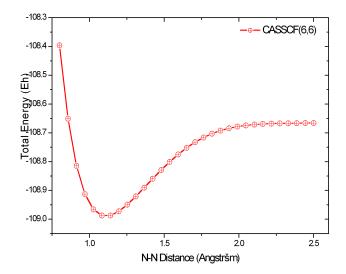
Figure 5.8: Potential Energy Surface of the N_2 molecule from CASSCF(6,6) calculations (SVP basis).

which gives:
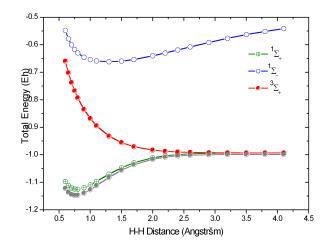
Figure 5.9: State averaged CASSCF(2,2) calculations on H_2 (two singlets, one triplet; SVP basis). The grey curve is the ground state CASSCF(2,2) curve

One observes, that the singlet and triplet ground states become degenerate for large distances (as required) while the second singlet becomes the ionic singlet state which is high in energy. If one compares the lowest root of the state-averaged calculation (in green) with the dedicated ground state calculation (in grey) one gets an idea of the energetic penalty that is associated with averaged as opposed to dedicated orbitals.

A more involved example is the rotation around the double bond in C_2H_4. Here, the π-bond is broken as one twists the molecule. The means the proper active space consists of two active electron in two orbitals.

The input is (for fun, we average over the lowest two singlets and the triplet):

```
    ! RHF   SV(P) SV/C SmallPrint   NoPop NoMOPrint


    %casscf nel         = 2
            norb        = 2
            mult        = 1,3
            nroots      = 2,1
            bweight     = 2,1
            weights[0]  = 1,1
            weights[1]  = 1
            TrafoStep   RI
            switchstep nr
            end


    %paras R= 1.3385
            Alpha=0,180,37
            end


    * int 0 1
    C  0  0  0  0       0    0
    C  1  0  0  {R}     0    0
    H  1  2  0 1.07   120    0
    H  1  2  3 1.07   120  180
    H  2  1  3 1.07   120  {Alpha}
    H  2  1  3 1.07   120  {Alpha+180}
    *
```

We can see from this plot, that the CASSCF method produces a nice ground state surface with the correct periodicity and degeneracy at the end points, which represent the planar ethylene molecule. At $90°$ one has a weakly coupled diradical and the singlet and triplet states become nearly degenerate, again as expected. Calculations with larger basis sets and inclusion of dynamic correlation would give nice quantitative results. We have also plotted the RHF energy (in purple) which gives a qualitatively wrong surface and doesn't return to the correct solution for planar ethylene. It is evident that even high quality dynamic correlation treatments like CC or CI would hardly be able to repair the shortcomings of the poor RHF reference state. In all these cases, CASSCF is the proper starting point for higher accuracy.

5.1.7.7 Excited States

As a final example, we do a state-average calculation on H_2CO in order to illustrate excited state treatments. We expect from the ground state (basically closed shell) a $n \rightarrow \pi^*$ and a $\pi \rightarrow \pi^*$ excited state which we want to describe. For the $n \rightarrow \pi^*$ we also want to calculate the triplet since it is well known experimentally. First we take DFT orbitals as starting guess.
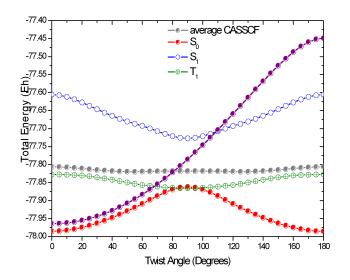
Figure 5.10: State averaged CASSCF(2,2) calculations on C_2H_4 (two singlets, one triplet; SV(P) basis). The grey curve is the state averaged energy and the purple curve corresponds to RHF.

```
! BP86 aug-SVP TightSCF
%base "1"


*int 0 1
C  0 0 0 0.00   0.0    0.00
O  1 0 0 1.20   0.0    0.00
H  1 2 0 1.10 120.0    0.00
H  1 2 3 1.10 120.0  180.00
*


$new_job

! RHF aug-SVP TightSCF
! moread
%moinp "1.gbw"
%base "Test-CASSCF.H2CO-1"
%casscf nel        4
        norb       3
        switchstep nr
        mult       1,3
        nroots     3,1
        bweight   = 1,1
        weights[0] = 1,1,1
        weights[1] = 1
        end
```

```
    *int 0 1
    C  0 0 0 0.00   0.0    0.00
    O  1 0 0 1.20   0.0    0.00
    H  1 2 0 1.10 120.0    0.00
    H  1 2 3 1.10 120.0  180.00
    *
```

We get:

```
    ----------------------------
    SA-CASSCF TRANSITION ENERGIES
    ----------------------------


    LOWEST ROOT =   -113.736217779 Eh -3094.842 eV


    STATE   ROOT MULT DE/a.u.    DE/eV    DE/cm**-1
      1:    0    3   0.056271    1.531   12349.9
      2:    1    1   0.075355    2.050   16538.5
      3:    2    1   0.272589    7.417   59826.3
```

Which is *very* poor. So, here the treatment based on average orbitals does not work at all. So let us do the states one after the other by adjusting the weights:

```
    Ground state:
    %casscf nel          4
            norb         3
            switchstep nr
            mult         1,3
            nroots       3,1
            bweight    = 1,0
            weights[0] = 1,0,0
            weights[1] = 0
            end
    First-Singlet:
    %casscf nel          4
            norb         3
            switchstep diis
            switchconv 0.003
            mult         1,3
            nroots       3,1
            bweight    = 1,0
            weights[0] = 0,1,0
            weights[1] = 0
            end
```

```
    Second-Singlet:
%casscf nel        4
        norb       3
        switchstep diis
        switchconv 0.003
        mult       1,3
        nroots     3,1
        bweight   = 1,0
        weights[0] = 0,0,1
        weights[1] = 0
        end
First-Triplet
%casscf nel        4
        norb       3
        switchstep diis
        switchconv 0.003
        mult       1,3
        nroots     3,1
        bweight   = 0,1
        weights[0] = 0,0,0
        weights[1] = 1
        end
```

The total energies are:

```
    Ground State: -113.825085148
    First-Singlet -113.671020811 = 4.19 eV
    Second-Singlet -113.479355179  = 9.41 eV
    First Triplet -113.684545957 = 3.82 eV
```

These results are much better and as soon as we include dynamic correlation one would have good transition energies (future releases will include state specific perturbation corrections to achieve this). This calculation shows the limits of state averaging – if the orbitals strongly change (as in the present case, since n $\to \pi^*$ orbitals are very different from $\pi \to \pi^*$ orbitals) the state averaging does not lead to good transition energies. In this case, one should follow SA-CASSCF by MRCI/SORCI (see below) or alternatively use state specific procedures (optimize each state in turn and then calculate its correlation energy; such multireference methods are not yet available in a good form in ORCA).

5.1.7.8 CASSCF Natural Orbitals as Input for Coupled Cluster Calculations

Consider the possibility that you are not sure about the orbital occupancy of your system. Hence you carry out some CASSCF calculation for various states of the system in an effort to decide on the ground state. You can of course follow the CASSCF by MR-MP2 or MR-ACPF or SORCI calculations to get a true multireference result for the state ordering. Yet, in some cases you may also want to obtain a coupled cluster estimate for the state energy difference. Converging coupled cluster calculation on alternative states in a controlled manner is anything but trivial. Here a

feature of ORCA might be helpful. The best single configuration that resembles a given CASSCF state is built from the natural orbitals of this state. These orbitals are also ordered in the right way to be input into the MDCI program. The convergence to excited states is, of course, not without pitfalls and limitations as will become evident in the two examples below.

As a negative example consider first the following interesting dicarbene.

For this molecule we expect that we should use four active orbitals and four active electrons and that singlet, triplet and quintet states might be accessible. We start with a simple CASSCF(4,4) optimization on the lowest singlet state.

```
! SV(P) Conv TightSCF Opt

%casscf  nel 4 norb 4 end

* int 0 1
C 0 0 0   0 0 0
C 1 0 0   1.45 0 0
C 2 1 0   1.45 109.4712 0
H 1 2 3   1.1  109.4712 0
H 3 2 1   1.1  109.4712 0
H 2 1 3   1.1  109.4712 240
H 2 1 3   1.1  109.4712 120
*
```

Of course, one should also do optimizations on the the other two spin states (and with larger basis sets) but for the sake of the argument, we stick to the singlet structure. Next, the natural orbitals for each state are generated with the help of the MRCI module. To this end we run a state averaged CASSCF for the lowest four singlet, two triplets and the quintet and pass that information on to the MRCI module that does a CASCI only (e.g. no excitations):

```
! ano-pVDZ Conv TightSCF MRCI
%casscf  nel     4
         norb    4
         mult    1,3,5
         nroots  4,2,1
         end
%mrci    tsel       0
         tpre       0
         donatorbs  2
         densities  5,1
         newblock 1 * nroots 4 excitations none refs cas(4,4) end end
         newblock 3 * nroots 2 excitations none refs cas(4,4) end end
         newblock 5 * nroots 1 excitations none refs cas(4,4) end end
         end
* int 0 1
  C     0   0   0   0.000000    0.000    0.000
```

```
    C     1   0   0   1.494825      0.000      0.000
    C     2   1   0   1.494825    105.185      0.000
    H     1   2   3   1.083843    129.438      0.000
    H     3   2   1   1.083843    129.438      0.000
    H     2   1   3   1.094087    111.176    239.566
    H     2   1   3   1.094087    111.176    120.434
  *
```

This produces the files: **BaseName.b*m*_s*n*.nat** where 'm' is the number of the block (m=0,1,2 correspond to singlet, triplet and quintet respectively) and 'n' stands for the relevant state (n=0,1,2,3 for singlet, n=0,1 for triplet and n=0 for quintet).

These natural orbitals are then fed into unrestricted QCISD(T) calculations:

```
  ! UHF ano-pVDZ Conv TightSCF AOX-QCISD(T)
  ! moread noiter
  %moinp "DiCarbene-01b.b0_s0.nat"

  * int 0 1
   C     0   0   0   0.000000      0.000      0.000
   C     1   0   0   1.494825      0.000      0.000
   C     2   1   0   1.494825    105.185      0.000
   H     1   2   3   1.083843    129.438      0.000
   H     3   2   1   1.083843    129.438      0.000
   H     2   1   3   1.094087    111.176    239.566
   H     2   1   3   1.094087    111.176    120.434
  *
```

This produces the energies:

State	Energy (Eh)	Relative Energy (cm^{-1})
S0	-116,190891	0
S1	-116,067265	27132,7
S2	-116,067265	27132,7
S3	-116,067265	27132,7
T0	-116,155471	7773,9
T1	-116,114038	16867,4
Q0	-116,134658	12343,2

It is found that the calculations indeed converge to different states. The excited singlets all fall down to the same state that is approximately 27,000 cm^{-1} above the lowest solution. The triplets are distinct and the quintet is unique anyways. Inspection of the coupled cluster wavefunctions indicate that the singlet converged to the closed shell solution and the first doubly excited state respectively.

These energies can be compared with the genuine multireference calculation obtained from the SORCI method:

```
    ! ano-pVDZ Conv TightSCF SORCI

    %casscf  nel     4
             norb    4
             mult    1,3,5
             nroots  4,2,1
             end

    * int 0 1
      C    0   0   0   0.000000     0.000     0.000
      C    1   0   0   1.494825     0.000     0.000
      C    2   1   0   1.494825   105.185     0.000
      H    1   2   3   1.083843   129.438     0.000
      H    3   2   1   1.083843   129.438     0.000
      H    2   1   3   1.094087   111.176   239.566
      H    2   1   3   1.094087   111.176   120.434
    *
```

Which produces:

```
    State Mult Irrep Root Block    mEh        eV       1/cm
      0    1    -1    0    0      0.000     0.000       0.0
      1    3    -1    0    1      1.327     0.036     291.2
      2    5    -1    0    2     26.074     0.710    5722.6
      3    3    -1    1    1     41.078     1.118    9015.6
      4    1    -1    2    0     79.535     2.164   17455.9
      5    1    -1    1    0     84.362     2.296   18515.3
      6    1    -1    3    0     86.117     2.343   18900.5
```

With the description of the wavefunctions:
Singlets:

```
    STATE   0:  Energy=   -115.944889643 Eh RefWeight=  0.9224  0.00 eV      0.0 cm**-1
        0.3544  : h---h---[2020]
        0.3298  : h---h---[2002]
        0.1036  : h---h---[1111]
        0.0682  : h---h---[0220]
        0.0663  : h---h---[0202]
    STATE   1:  Energy=   -115.860527602 Eh RefWeight=  0.9141  2.30 eV  18515.3 cm**-1
        0.6784  : h---h---[2200]
        0.0631  : h---h---[2020]
        0.0705  : h---h---[2002]
        0.0874  : h---h---[1111]
        0.0040  : h---h---[0220]
        0.0052  : h---h---[0202]
        0.0055  : h---h---[0022]
    STATE   2:  Energy=   -115.865354536 Eh RefWeight=  0.8969  2.16 eV  17455.9 cm**-1
        0.7785  : h---h---[2110]
        0.0921  : h---h---[1201]
        0.0151  : h---h---[1021]
        0.0112  : h---h---[0112]
        0.0038  : h---h  6[2120]
        0.0049  : h---h---[2100]p14
```

```
   STATE   3: Energy=   -115.858772427 Eh RefWeight=  0.8988  2.34 eV  18900.5 cm**-1
        0.7572  : h---h---[2101]
        0.1092  : h---h---[1210]
        0.0224  : h---h---[1012]
        0.0099  : h---h---[0121]
Triplets:
STATE   0: Energy=   -115.943562624 Eh RefWeight=  0.9209  0.00 eV     0.0 cm**-1
        0.6645  : h---h---[2011]
        0.0671  : h---h---[1120]
        0.0647  : h---h---[1102]
        0.1246  : h---h---[0211]
STATE   1: Energy=   -115.903811398 Eh RefWeight=  0.9171  1.08 eV   8724.4 cm**-1
        0.6863  : h---h---[2110]
        0.1913  : h---h---[1201]
        0.0243  : h---h---[1021]
        0.0152  : h---h---[0112]
Quintets:
STATE   0: Energy=   -115.918815602 Eh RefWeight=  0.9264  0.00 eV     0.0 cm**-1
        0.9264  : h---h---[1111]
```

Thus, the singlet ground state is heavily multiconfigurational, the lowest triplet state is moderately multiconfigurational and the lowest quintet state is of course a single configuration. Interstingly, the lowest singlet, triplet and quintet do not form a regular spin ladder which might have been expected if one considers the system of being composed of two interacting S=1 systems. Rather, the lowest singlet and triplet states are close in energy while the lowest quintet is far away.

The energies are completely different from the QCISD(T) results. However, this is not unexpected based on the composition of these wavefunctions. These are the limitations of singlet reference methods. Nevertheless, this shows how such results can be obtained in principle.

As a more positive example, consider some ionized states of the water cation:

First the natural orbital generation:

```
   ! ano-pVDZ Conv TightSCF

%casscf  nel     7
         norb    6
         nroots  3
         end


%mrci    tsel       0
         tpre       0
         donatorbs  2
         densities  5,1
         newblock 2 *
            nroots 3
            excitations none
            refs cas(7,6)end
            end
```

```
          end

* int 1 2
  O     0   0   0   0.000000    0.000     0.000
  H     1   0   0   1.012277    0.000     0.000
  H     1   2   0   1.012177  109.288     0.000
*
```

Then the SORCI reference calculation:

```
! ano-pVDZ Conv TightSCF SORCI

%casscf  nel     7
         norb    6
         nroots  3
         end

* int 1 2
  O     0   0   0   0.000000    0.000     0.000
  H     1   0   0   1.012277    0.000     0.000
  H     1   2   0   1.012177  109.288     0.000
*
```

Then the three QCISD(T) calculations

```
! UHF ano-pVDZ Conv TightSCF QCISD(T)
! moread noiter
%moinp "H2O+-02.b0_s0.nat"

* int 1 2
  O     0   0   0   0.000000    0.000     0.000
  H     1   0   0   1.012277    0.000     0.000
  H     1   2   0   1.012177  109.288     0.000
*
```

we obtain the transition energies:

```
SORCI   QCISD(T) (in cm-1)
  D0      0     0.0
  D1  16269   18283
  D2  50403   50509
```

Thus, in this example the agreement between single- and multireference methods is good and the unrestricted QCISD(T) method is able to describe these excited doublet states. The natural orbitals have been a reliable way to guide the CC equations into the desired solutions. This will work in many cases.

5.1.8 N-Electron Valence State Perturbation Theory (NEVPT2)

ORCA has implemented the <u>strongly-contracted</u> version of the NEVPT2. Strongly-contracted NEVPT2 is an internally contracted multireference perturbation theory which applies to CASSCF type wavefunctions. Therefore NEVPT2 is suited for systems, where static and dynamic correlation are important. It is an efficient tool to treat multireference systems with many atoms ($\sim$2000 basis functions) and a large active space (14-16 active orbitals). Typically it takes less time than the respective CASSCF calculation. The NEVPT2 program is fully parallelized. For a list of available options and references to the literature see section 6.10.

Besides corrections to the correlation energy ORCA features UV- , IR-, CD- and MCD-spectra as well as EPR parameters for NEVPT2. These properties are computed using the "quasi-degenerate perturbation theory" that is described in section 6.9.2. The NEVPT2 corrections enter as "improved diagonal energies" in this formalism. Further improvement are possible with the quasi-degenerate NEVPT2 (QD-NEVPT2). Here, the effect of the perturbation treatment is maped onto the model space leading to an improved CAS-CI wavefunction.

As a simple example consider the groundstate of the nitrogen molecule N_2. After defining the computational details of our CASSCF calculation, we insert "! Nevpt2" as simple input or specify "Nevpt2 true" in the %casscf block.

```
!def2-svp NEVPT2
%casscf nel   6
  norb  6
  mult  1
end

* xyz 0 1
N 0.0 0.0 0.0
N 0.0 0.0 1.09768
*
```

For better control of the program flow it is advised to split the calculation into two parts. First converge the CASSCF calculation and then in a second step read the converged orbitals and execute the actual NEVPT2.

```
          ----------------------------------------------------------------
                          ORCA-CASSCF
          ----------------------------------------------------------------


          ...
          NEVPT2-SETTINGS:
            A NEVPT2 calculation will be performed on top of the CASSCF wavefunction!
          ...
          ----------------------------------------------------------------
                          < NEVPT2  >
          ----------------------------------------------------------------
          ...
             ==========================================================
                          NEVPT2 Results
             ==========================================================
```

```
    ***********************
     MULT 1, ROOT 0
    ***********************


   Class V0_ijab :          dE = -0.017748
   Class Vm1_iab :          dE = -0.023171
   Class Vm2_ab  :          dE = -0.042194
   Class V1_ija  :          dE = -0.006806
   Class V2_ij   :          dE = -0.005056
   Class V0_ia   :          dE = -0.054000
   Class Vm1_a   :          dE = -0.007091
   Class V1_i    :          dE = -0.001963


   ----------------------------------------------------------------
          Total Energy Correction : dE = -0.15802909
   ----------------------------------------------------------------
          Zero Order Energy       : E0 = -108.98888640
   ----------------------------------------------------------------
          Total Energy (E0+dE)    : E  = -109.14691549
   ----------------------------------------------------------------
```

Introducing dynamic correlation with NEVPT2 lowers the energy by 150mEh. ORCA also prints the contribution of each "excitation class V" to the NEVPT2 correction. We note that in the case of a single reference wavefunction the "V0_ij,ab" excitation class produces the exact MP2 correlation energy.

In chapter 5.1.7.6 the dissociation of the N_2 molecule has been investigated with the CASSCF method. Inserting "nevpt2 true" into the %casscf block we obtain the NEVPT2 correction as additional information.

```
    !def2-svp NEVPT2 veryslowconv
    %casscf nel   6
      norb  6
      mult  1
    end

    %paras
    R = 0.7, 2.5, 30
    end

    *xyz 0 1
    N 0.0 0.0 0.0
    N 0.0 0.0 {R}
    *
```

All of the options available in CASSCF can in principle be applied to NEVPT2.

NOTE
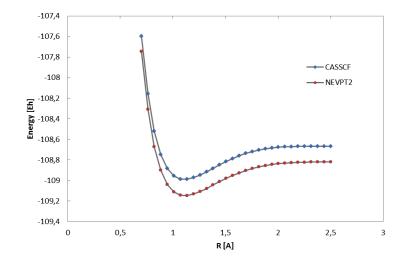
Figure 5.11: Potential Energy Surface of the N_2 molecule from CASSCF(6,6) and NEVPT2 calculations (def2-SVP).

- **NEVPT2 is implemented as a submodule of CASSCF and will inherit all settings from CASSCF(*! tightscf, !UseSym, !COSMO, !RIJCOSX . . .*).**

- NEVPT2 analytic gradients are not available, but numeriscal gradients are!

5.1.9 Full Configuration Interaction Energies and Properties

The MRCI and CASSCF modules may be used to produces energies and properties (g-tensors, zero-field splitting ...) at the level of full configuration interaction (FCI). Larger molcular system naturally require more computational resources and time. For the CASSCF module, the limit of still feasible calculations is by around 14 electrons in 14 orbitals using configuration state functions (CSF) for the CI expansion. Larger active spaces may be computed in the framework of the Density Matrix Renormalization Group (DMRG), which may be selected **CIStep** in the CASSCF module. These options and modules have their own sections in the manual with a detailed documentation on their usage and the available properties.

Aside of the mentioned options, ORCA interfaces a determinant based FCI toy-code, which is described here. Besides energies, the program is capable to compute dipole moments, g-tensors, hyperfine and quadrupole splittings in the framework of linear response. For response properties, it is assumed that the basis set does not depend on the given perturbation. To call the program prepare a simple SCF input and add the keyword "!FCI " on top.

```
!extremescf nofrozencore fci


*xyz 0 2
Li 0 0 0
*
```

By default, energies and the before mentioned properties are computed for the groundstate. For the g-tensor computation, the origin is chosen as center of electronic charge, while for dipols it is the center of mass. The default settings for may be edited in the FCI block:

```
    ETol                1e-10# Energy convergence thresh
    RTol                1e-8 # Residual convergence thresh
    RespTol            1e-8 # Convergence thresh for the response equations


    NRoots              1  # Number of states to be computed
    DoGTensor     true
    SOCG            true # SOC contribution to the g-tensor
    DoHFC          true
    SOCHFC        false # SOC contribution to the HFS
    DoQuad        true
    DoDipole       true
    PrintLevel      3 # PrintLevel
    ORIGIN         -3 # (see EPRMR-module)
```

The input and output resembles the ORCA standard format from the EPRNMR module and should thus be familiar. However, it is much more limited in the number of available options and the size of molecules that can be treated. Currently it can only be used **serial** and in combination with the **nofrozencore** option.

5.1.10 Scalar Relativistic SCF

Scalar relativistic all-electron calculations can be performed with a variety of relativistic approximations. However, these need to be combined with a suitable basis set since relativity changes the shapes of orbitals considerably. We have defined scalar relativistic contracted versions of the QZV, TZV and SV basis sets up to Hg for HF and DFT computations (but not yet for RI-MP2). They are automatically chosen if TZV or SV is used together with a scalar relativistic method. For other basis sets you have to take care of the recontraction yourself but note that this is an expert issue. All scalar relativistic models can be used for geometry optimization as well.

CAUTION:

- **For geometry optimizations we apply a one-center relativistic correction. This slightly changes the energies – so DO NOT MIX single-point calculations without the one-center approximation with geometry optimization energies that DO make use of this feature.The impact of the one-center approximation on the geometries is very small.**

5.1.10.1 Douglas-Kroll-Hess

ORCA has implemented the standard second-order DKH procedure that is normally satisfactory for all intents and purposes. The scalar relativistic DKH treatment is compatible with any of the SCF methods and will also be transferred over to the correlation treatments.

We rather strongly recommend the use of the SV, TZV and QZV basis sets with or without 'def2' and appropriate polarization functions. For these basis sets we have developed segmented relativistic all electron basis sets for almost the entire periodic table. The basis sets are tested and perform very well in an acceptably economic fashion.

The use of the code is very simple:

```
! UHF TZV DKH
```

NOTE: You should have the basis set and ZORA or DKH commands in the same input line!

5.1.10.2 ZORA and IORA

In addition to the DKH method the 0th order regular approximation (ZORA; pioneered by van Lenthe et al., see Ref. [81] and many follow up papers by the Amsterdam group) is implemented into ORCA in an approximate way (section 6.12) which facilitates the calculation of analytical gradients. Our ZORA implementation essentially follows van Wüllen [82] and solves the ZORA equations with a suitable model potential which is derived from accurate atomic ZORA calculations. At this point the elements up to atomic number 86 are available with more to come. The ZORA method is highly dependent on numerical integration and it is very important to pay attention to the subject of radial integration accuracy (*vide infra*)! If the relevant precautions are taken, the use of the ZORA or IORA methods is as easy as in the DKH case. For example:

```
! UHF TZVP ZORA
# for more detail use
%rel method ZORA          # or IORA
     modelpot 1,1,1,1
     modeldens rhoZORA
     velit  137.0359895  # speed of light used
     end
```

ATTENTION

- The scalar relativistic module has many options which allow you considerably finetune the calculations. Details are in section 6.12.

- The scalar relativistic treatment requires flexible basis sets, in particular in the core region. Only SV, TZV and QZV basis sets have been recontracted in the ZORA and DKH models (and the attached polarization functions of course). Alternatively, one choice that you have is to uncontract your basis set using the `Decontract` keyword but it is likely that you also need additional steep primitives.

- Scalar relativistic calculations may need larger integration grids in the radial part. Consider to use a higher `IntAcc` parameter or at least to increase the radial integration accuracy around the heavy atoms using `SpecialGridAtoms` and `SpecialGridIntAcc`

- The calculation of properties in relativistic treatments is not straightforward since the influence of the 'small component' in the Dirac equation is neglected in the calcuation of expectation values. ORCA takes these 'picture change' effects to some extent into account. Please refer to individual sections.

5.1.11 Efficient Calculations with Atomic Natural Orbitals

Atomic natural orbitals are a special class of basis sets. They are represented by the orthonormal set of orbitals that diagonalizes a spherically symmetric, correlated atomic density. The idea is to put as much information as possible into each basis functions such that one obtains the best possible result with the given number of basis functions. This is particularly important for correlated calculations where the number of primitives is less an issue than the number of basis functions.

Usually, ANO basis sets are 'generally contracted' which means that for any given angular momentum all primitives contribute to all basis functions. Since the concept of ANOs only makes sense if the underlying set of primitives is large, the calculations readily become very expensive unless special precaution is taken in the integral evaluation algorithms. ORCA features special algorithms for ANO basis sets together with accurate ANO basis sets for non-relativistic calculations. However, even then the integral evaluation is so expensive that efficiency can only be realized if all integrals are stored on disk and are re-used as needed.

In the first implementation, the use of ANOs is restricted to the built-in ANO basis sets (ano-pVnZ, saug-ano-pVnZ, aug-ano-pVnZ, n = D, T, Q, 5). These are built upon the cc-pV6Z primitives and hence, the calculations take significant time.

NOTE:

- There are no relativistic ANOs so far.

- Geometry optimizations with ANOs are discouraged; they will be *very* inefficient.

The use of ANOs is recommended in the following way:

```
! ano-pVTZ Conv TightSCF CCSD(T)
%maxcore 2000
* int 0 1
C  0 0 0    0   0   0
O  1 0 0  1.2   0   0
H  1 2 0  1.1 120   0
H  1 2 3  1.1 120 180
*
```

This yieds:

```
ano-pVTZ:
E(SCF) = -113.920388785
E(corr)=   -0.427730189
```

Compare to the cc-pVTZ value of:

```
cc-pVTZ:
E(SCF) = -113.911870901
E(corr)=   -0.421354947
```

Thus, the ANO-based SCF energy is ca. 8–9 mEh lower and the correlation energy almost 2 mEh lower than with the cc-basis set of the same size. Usually, the ANO results are much closer to the basis set limit than the cc-results. Also, ANO values extrapolate very well (see section 5.1.3.5)

Importantly, the integrals are all stored in this job. Depending on your system and your patience, this may be possible up to 300–500 basis functions. The ORCA correlation modules have been rewritten such that they deal efficiently with these stored integrals. Thus, we might as well have used ! MO-CCSD(T) or ! AO-CCSD(T), both of which would perform well.

Yet, the burden of generating and storing all four-index integrals quickly becomes rather heavy. Hence, the combination of ANO basis sets with the RI-JK technique is particularly powerful and efficient. For example:

```
! ano-pVTZ cc-pVTZ/JK RI-JK Conv TightSCF RI-CCSD(T)
```

For the SCF, this works very well and allows for much larger ANO based calculations to be done efficiently. Also, RI-MP2 can be done very efficiently in this way. However, for higher order correlation methods such as CCSD(T) the logical choice would be RI-CCSD(T) which is distinctly less efficient than the AO or MO based CCSD(T) (roughly a factor of two slower). Hence, ORCA implements a hybrid method where the RI approximation is used to generate all four index integrals. This is done via the 'RI-AO' keyword:

```
! ano-pVTZ cc-pVTZ/JK RI-AO Conv TightSCF AO-CCSD(T)
```

In this case either AO-CCSD(T) or MO-CCSD(T) would both work well. This does not solve the storage bottleneck with respect to the four index integrals of course. If this becomes a real issue, then RI-CCSD(T) is mandatory. The error in the total energy is less than 0.1 mEh in the present example.

NOTE:

- **With conventional RI calculations the use of a second fit basis set is not possible and inconsistent results will be obtained. Hence, stick to one auxiliary basis!**

5.2 Geometry Optimizations, Surface Scans, Transition States, MECPs

Finally, starting from version 2.2 ORCA supports geometry optimization using analytical gradients at the Hartree-Fock and DFT levels. From version 2.4.10 onward, redundant internal coordinates are available which lead to much more efficient structure optimizations. Starting from 2.4.30 the coordinate set-up has been improved and from version 2.5.30 onward, relaxed surface scans are available with ORCA. From 2.6.0 onward, transition state (TS) optimization is implemented in ORCA. In the DFT case particularly efficient optimizations result with nonhybrid functionals due to the use of the RI approximation (i.e. the analytic gradients for the RI-J method are also available). From 2.7.0 on, numerical gradients are available for all levels of theory, i.e. optimizations can be performed for all methods (although quite time-consuming). Additionally several approaches to construct an initial Hessian for TS optimization are available. Another interesting feature are MECP (Minimum Energy Crossing Point) optimizations.

5.2.1 Geometry Optimizations

Optimizations are fairly easy as in the following example:

```
! RKS B3LYP/G SV(P) TightSCF Opt
* int 0 1
    C  0 0 0 0.0000   0.000    0.00
    O  1 0 0 1.2029   0.000    0.00
    H  1 2 0 1.1075 122.016    0.00
    H  1 2 3 1.1075 122.016  180.00
*
```

an optimization with the RI method (the BP functional is recommend) would simply look like:

```
! RKS BP RI SV(P) SV/J TightSCF Opt
* int 0 1
    C  0 0 0 0.0000   0.000    0.00
    O  1 0 0 1.2029   0.000    0.00
    H  1 2 0 1.1075 122.016    0.00
    H  1 2 3 1.1075 122.016  180.00
*
```

5.2.2 Numerical Gradients

If the analytic gradient is not available, the numerical gradient can simply be requested via:

```
! NumGrad
```

as in the following example:

```
!RHF CCSD(T) TZVPP
!Opt NumGrad
* int 0 1
C 0 0 0  0   0   0
O 1 0 0  1.2 0   0
H 1 2 0  1.1 120 0
H 1 2 3  1.1 120 180
*
```

NOTE

- Be aware that the numerical gradient is quite expensive. The time for one gradient calculation is equal to 6 × (number of atoms) × (time for one single point calculation).

- The numerical gradient can be calculated in a multi-process run, using a maximum of three times the number of atoms (see section 2.2).

More details on various options, geometry convergence criteria and the like are found in section 6.13.

5.2.3 Some Notes and Tricks

NOTE

- `TightSCF` in the SCF part is set as default to avoid the buildup of too much numerical noise in the gradients.

TIP

- If you want to be on the safe side with DFT it is probably advisable to use the settings `!  Grid4 NoFinalGrid` although the defaults should also be o.k.

- In rare cases the redundant internal coordinate optimization fails. In this case, you may try to use `COPT` (optimization in Cartesian coordinates). If this optimization does not converge, you may try the desperate choice to use `ZOPT`, `GDIIS-COPT` or `GDIIS-ZOPT`. This will likely take many more steps to converge but should be stable.

- For optimizations in Cartesian coordinates the initial guess Hessian is constructed in internal coordinates and thus these optimizations should converge only slightly slower than those in internal coordinates. Nevertheless, if you observe a slow convergence behaviour , it may be a good idea to compute a Hessian initially (perhaps at a lower level of theory) and use `InHess read` in order to improve convergence.

- At the beginning of a TS optimization more information on the curvature of the PES is needed than a model Hessian can give. Nevertheless you do not need to calculate the full numerical Hessian when starting such a calculation. With ORCA we've made good experience with approximations to the exact Hessian. Here it is recommended to either directly combine the TS optimization with the results that are obtained with a relaxed surface scan or to use the Hybrid Hessian as the initial Hessian, depending on the nature of the TS mode. Note that these approximate Hessians do never replace the exact Hessian at the end of the optimization, which is always needed to verify the minimum or first order saddle point nature of the obtained structure.

5.2.4 Initial Hessian for Minimization

The convergence of a geometry optimization crucially depends on the quality of the initial Hessian. In the simplest case it is taken as a unit matrix (in redundant internal coordinates we use 0.5 for bonds, 0.2 for angles and 0.1 for dihedrals and improper torsions). However, simple model force-fields like the ones proposed by Schlegel, Lindh or Almlöf are available and lead to much better convergence. The different guess Hessians can be set via the `InHess` option which can be either `unit`, `Almloef`, `Lindh` or `Schlegel` in redundant internal coordinates. From version 2.5.30 on these model force-fields (built up in internal coordinates) can also be used in optimizations in cartesian coordinates. For minimizations we recommend the `Almloef` Hessian (which is also the default for minimizations), the `Lindh` and `Schlegel` Hessian yield a similar convergence behaviour. Of course the best Hessian is the exact one. `Read` may be used to input an exact Hessian or one that has been calculated at a lower level of theory (or a 'faster' level of theory). From version 2.5.30 on this option is also available in redundant internal coordinates. But we have to point out that the use of the exact Hessian as initial one is only of little help, since in these cases the convergence is usually only slightly faster, while at the same time much more time is spent in the calculation of the initial Hessian.

To sum it up: we advise to use one of the simple model force-fields for minimizations.

5.2.5 Coordinate Systems for Optimizations

The coordinate system for the optimization can be chosen by the **coordsys** (= **cartesian** or **redundant** or **redundant_old**) variable in the **%geom** Block. The default is the redundant internal coordinates (redundant_old is the coordinate set that was used as default redundant internal coordinates before version 2.4.30). If the optimization with **redundant** fails, **redundant_old** can still be used as an alternative, and in cases where the internal coordinate systems lead to problems, you can still try **cartesian**. If the optimization is then carried out in cartesian displacement coordinates with a simple model force-field Hessian, the convergence will be only slightly slower. With a unit matrix initial Hessian very slow convergence will result.

A job that starts from a semi-empirical Hessian is shown below:

```
# ---------------------------------------------------
# First calculate the frequencies at the input
# geometry which is deliberately chosen poorly
# ---------------------------------------------------
! RHF AM1 NumFreq
%base "FirstJob"
* int 0 1
 C 0 0 0  0   0 0
 O 1 0 0  1.3 0 0
 H 1 2 0  1.1 110 0
 H 1 2 3  1.1 110 180
*
$new_job
# ----------------------------------------------------------
# Now the real job
# ----------------------------------------------------------
! B3LYP SVP TightSCF PModel
! Opt
%base "SecondJob"
%geom GDIISMaxEq 20
      UseGDIIS false
      InHess      Read
      InHessName "FirstJob.hess"
        # this file must be either a .hess file from a
        # frequency run or a .opt file left over from a
        # previous geometry optimization
      end

* int 0 1
 C 0 0 0  0   0 0
 O 1 0 0  1.3 0 0
 H 1 2 0  1.1 110 0
 H 1 2 3  1.1 110 180
*
```

NOTE:

- The guess `PModel` is chosen for the second job since DFT or HF calculations cannot be started from semi-empirical orbitals

- GDIIS has been turned off and the number of gradients used in the quasi-Newton method has been enhanced. This is advisable if a good starting Hessian is available.

TIP

- For transition metal complexes MNDO, AM1 or PM3 Hessians are not available. You can use ZINDO/1 or NDDO/1 Hessians instead. They are of lower quality than MNDO, AM1 or PM3 for organic molecules but they are still far better than the standard unit matrix choice.

- If the quality of the initial semi-empirical Hessian is not sufficient you may use a "quick" RI-DFT job (e.g. `BP Def-1 NoFinalGrid`)

- In semi-empirical geometry optimizations on larger molecules or in general when the molecules become larger the redundant internal space may become large and the relaxation step may take a significant fraction of the total computing time. In this case you may revert to delocalized internal coordinates which only keep the non-redundant degrees of freedom and leads to more efficient update steps. It is slightly less robust though.

5.2.6 Constrained Optimizations

You can perform constrained optimizations which can, at times, be extremely helpful. This works as shown in the following example:

```
! RKS B3LYP/G SV(P) TightSCF Opt
%geom Constraints
      { B 0 1 1.25 C }
      { A 2 0 3 120.0 C }
      end
    end

* int 0 1
   C  0 0 0 0.0000   0.000    0.00
   O  1 0 0 1.2500   0.000    0.00
   H  1 2 0 1.1075 122.016    0.00
   H  1 2 3 1.1075 122.016  180.00
*
```

```
Constraining bond distances        :   { B N1 N2 value C }
Constraining bond angles           :   { A N1 N2 N1 value C }
Constraining dihedral angles       :   { D N1 N2 N3 N4 value C }
Constraining cartesian coordinates :   { C N1 C }
```

NOTE:

- "value" in the constraint input is optional. If you do not give a value, the present value in the structure is constrained. For cartesian constraints you can't give a value, but always the initial position is constrained.

- It is recommended to use a value not too far away from your initial structure.

- It is possible to constrain whole sets of coordinates:

```
all bond lengths where N1 is involved      : { B N1 * C}
all bond lengths                           : { B * * C}
all bond angles where N2 is the central atom: { A * N2 * C }
all bond angles                            : { A * * * C }
all dihedral angles with central bond N2-N3 : { D * N2 N3 * C }
all dihedral angles                        : { D * * * * C }
```

- If there are only a few coordinates that have to be optimized you can use the invertConstraints option:

```
%geom Constraints
        { B 0 1 C }
        end
      invertConstraints true # only the C-O distance is optimized
                             # does not affect Cartesian coordinates
      end
```

- In some cases it is advantageous to optimize only the positions of the hydrogen atoms and let the remaining molecule skeleton fixed:

```
%geom optimizehydrogens true
        end
```
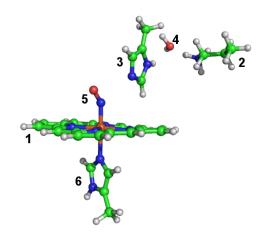
NOTE:

- In the special case of a fragment optimization (see next point) the optimizehydrogens keyword does not fix the heteroatoms, but ensures that all hydrogen positions are relaxed.

5.2.7 Constrained Optimizations for Molecular Clusters (Fragment Optimization)

If you want to study systems, which consist of several molecules (e.g. the active site of a protein) with constraints, then you can either use cartesian constraints (see above) or use ORCA's fragment constraint option. ORCA allows the user to define fragments in the system. For each fragment one can then choose separately whether it should be optimized or constrained. Furthermore it is possible to choose fragment pairs whose distance and orientation with respect to each other should be constrained. Here, the user can either define the atoms which make up the connection between the fragments, or the program chooses the atom pair automatically via a closest distance criterium. ORCA then chooses the respective constrained coordinates automatically. An example for this procedure is shown below.

The coordinates are taken from a crystal structure [PDB-code 2FRJ]. In our gas phase model we choose only a small part of the protein, which is important for its spectroscopic properties. Our selection consists of a heme-group (fragment 1), important residues around the reaction site (lysine (fragment 2) and histidine (fragment 3)), an important

water molecule (fragment 4), the NO-ligand (fragment 5) and part of a histidine (fragment 6) coordinated to the heme-iron. In this constrained optimization we want to maintain the position of the heteroatoms of the heme group. Since the protein backbone is missing, we have to constrain the orientation of lysine and histidine (fragments 2 and 3) side chains to the heme group. All other fragments (the ones which are directly bound to the heme-iron and the water molecule) are fully optimized internally and with respect to the other fragments. Since the crystal structure does not reliably resolve the hydrogen positions, we relax also the hydrogen positions of the heme group.

If you want to run this optimization: be aware that it will take some time!

```
! UKS BP86 RI SV(P) Opt TightSCF Grid4 NoFinalGrid
%geom
    ConstrainFragments { 1 } end # constrain all internal
                                 # coordinates of fragment 1
    ConnectFragments
      {1 2 C 12 28}  # connect the fragments via the atom pair 12/28 and 15/28 and
      {1 3 C 15 28} #  constrain the internal coordinates connecting
                    #  fragments 1/2 and 1/3
       {1 5 O}
       {1 6 O}
       {2 4 O}
       {3 4 O}
     end
     optimizeHydrogens true  # do not constrain any hydrogen position
   end
* xyz 1 2
Fe(1)    -0.847213    -1.548312    -1.216237 newgto "TZVP" end
 N(5)    -0.712253    -2.291076     0.352054 newgto "TZVP" end
 O(5)    -0.521243    -3.342329     0.855804 newgto "TZVP" end
 N(6)    -0.953604    -0.686422    -3.215231 newgto "TZVP" end
 N(3)    -0.338154    -0.678533     3.030265 newgto "TZVP" end
 N(3)    -0.868050     0.768738     4.605152 newgto "TZVP" end
 N(6)    -1.770675     0.099480    -5.112455 newgto "TZVP" end
 N(1)    -2.216029    -0.133298    -0.614782 newgto "TZVP" end
 N(1)    -2.371465    -2.775999    -1.706931 newgto "TZVP" end
 N(1)     0.489683    -2.865714    -1.944343 newgto "TZVP" end
 N(1)     0.690468    -0.243375    -0.860813 newgto "TZVP" end
 N(2)     1.284320     3.558259     6.254287
 C(2)     5.049207     2.620412     6.377683
```

C(2)	3.776069	3.471320	6.499073
C(2)	2.526618	2.691959	6.084652
C(3)	-0.599599	-0.564699	6.760567
C(3)	-0.526122	-0.400630	5.274274
C(3)	-0.194880	-1.277967	4.253789
C(3)	-0.746348	0.566081	3.234394
C(6)	0.292699	0.510431	-6.539061
C(6)	-0.388964	0.079551	-5.279555
C(6)	0.092848	-0.416283	-4.078708
C(6)	-2.067764	-0.368729	-3.863111
C(1)	-0.663232	1.693332	-0.100834
C(1)	-4.293109	-1.414165	-0.956846
C(1)	-1.066190	-4.647587	-2.644424
C(1)	2.597468	-1.667470	-1.451465
C(1)	-1.953033	1.169088	-0.235289
C(1)	-3.187993	1.886468	0.015415
C(1)	-4.209406	0.988964	-0.187584
C(1)	-3.589675	-0.259849	-0.590758
C(1)	-3.721903	-2.580894	-1.476315
C(1)	-4.480120	-3.742821	-1.900939
C(1)	-3.573258	-4.645939	-2.395341
C(1)	-2.264047	-4.035699	-2.263491
C(1)	0.211734	-4.103525	-2.488426
C(1)	1.439292	-4.787113	-2.850669
C(1)	2.470808	-3.954284	-2.499593
C(1)	1.869913	-2.761303	-1.932055
C(1)	2.037681	-0.489452	-0.943105
C(1)	2.779195	0.652885	-0.459645
C(1)	1.856237	1.597800	-0.084165
C(1)	0.535175	1.024425	-0.348298
O(4)	-1.208602	2.657534	6.962748
H(3)	-0.347830	-1.611062	7.033565
H(3)	-1.627274	-0.387020	7.166806
H(3)	0.121698	0.079621	7.324626
H(3)	0.134234	-2.323398	4.336203
H(3)	-1.286646	1.590976	5.066768
H(3)	-0.990234	1.312025	2.466155
H(4)	-2.043444	3.171674	7.047572
H(2)	1.364935	4.120133	7.126900
H(2)	0.354760	3.035674	6.348933
H(2)	1.194590	4.240746	5.475280
H(2)	2.545448	2.356268	5.027434
H(2)	2.371622	1.797317	6.723020
H(2)	3.874443	4.385720	5.867972
H(2)	3.657837	3.815973	7.554224
H(2)	5.217429	2.283681	5.331496
H(2)	5.001815	1.718797	7.026903
H(6)	-3.086380	-0.461543	-3.469767
H(6)	-2.456569	0.406212	-5.813597
H(6)	1.132150	-0.595619	-3.782287
H(6)	0.040799	1.559730	-6.816417
H(6)	0.026444	-0.139572	-7.404408
H(6)	1.392925	0.454387	-6.407850
H(1)	2.033677	2.608809	0.310182
H(1)	3.875944	0.716790	-0.424466
H(1)	3.695978	-1.736841	-1.485681

```
    H(1)     3.551716    -4.118236    -2.608239
    H(1)     1.487995    -5.784645    -3.308145
    H(1)    -1.133703    -5.654603    -3.084826
    H(1)    -3.758074    -5.644867    -2.813441
    H(1)    -5.572112    -3.838210    -1.826943
    H(1)    -0.580615     2.741869     0.231737
    H(1)    -3.255623     2.942818     0.312508
    H(1)    -5.292444     1.151326    -0.096157
    H(1)    -5.390011    -1.391441    -0.858996
    H(4)    -1.370815     1.780473     7.384747
    H(2)     5.936602     3.211249     6.686961
*
```

NOTE:

- You have to connect the fragments in such a way that the whole system is connected.

- You can divide a molecule into several fragments.

- Since the initial Hessian for the optimization is based upon the internal coordinates: Connect the fragments in a way that their real interaction is reflected.

- This option can be combined with the definition of constraints, scan coordinates and the optimizeHydrogens option (but: its meaning in this context is different to its meaning in a normal optimization run, relatively straightforward see section 6.13).

- Can be helpful in the location of complicated transition states (with relaxed surface scans).

5.2.8 Relaxed Surface Scans

A final thing that comes in really handy are relaxed surface scans, i.e. you can scan through one variable while all others are relaxed. It works as shown in the following example:

```
! RKS B3LYP/G SV(P) TightSCF Opt
%geom Scan
        B 0 1 = 1.35, 1.10, 12 # C-O distance that will be scanned
        end
      end

* int 0 1
    C  0 0 0 0.0000   0.000    0.00
    O  1 0 0 1.3500   0.000    0.00
    H  1 2 0 1.1075 122.016    0.00
    H  1 2 3 1.1075 122.016  180.00
*
```

In the example above the value of the bond length between C and O will be changed in 12 equidistant steps from 1.35 down to 1.10 Ångströms and at each point a constrained geometry optimization will be carried out.

NOTE:

- If you want to perform a geometry optimization at a series of values with non-equidistant steps you can give this series in square brackets, [ ]. The general syntax is as follows:

```
        B N1 N2 = initial-value, final-value, NSteps
or:
        B N1 N2 [value1 value2 value3 ... valueN]
```

- In addition to bond lengths you can also scan bond angles and dihedral angles:

```
        B N1 N2 = ...              # bond length
        A N1 N2 N3 = ...          # bond angle
        D N1 N2 N3 N4 = ...       # dihedral angle
```

TIP

- As in constrained geometry optimization it is possible to start the relaxed surface scan with a different scan parameter than the value present in your molecule. But keep in mind that this value should not be too far away from your initial structure.

A more challenging example is shown below. Here, the H-atom abstraction step from CH_4 to OH-radical is computed with a relaxed surface scan (*vide supra*). The job was run as follows:

```
    ! UKS B3LYP SV(P) TightSCF Opt SlowConv
    %geom scan B 1 0 = 2.0, 1.0, 15 end end
    * int 0 2
      C    0   0   0    0.000000    0.000    0.000
      H    1   0   0    1.999962    0.000    0.000
      H    1   2   0    1.095870  100.445    0.000
      H    1   2   3    1.095971   90.180  119.467
      H    1   2   3    1.095530   95.161  238.880
      O    2   1   3    0.984205  164.404   27.073
      H    6   2   1    0.972562  103.807   10.843
    *
```

It is obvious that the reaction is exothermic and passes through an early transition state in which the hydrogen jumps from the carbon to the oxygen. The structure at the maximum of the curve is probably a very good guess for the true transition state that might be located by a transition state finder.

You will probably find that such relaxed surface scans are incredibly useful but also time consuming. Even the simple job shown below required several hundred single point and gradient evaluations (convergence problems appear for the SCF close to the transition state and for the geometry once the reaction partners actually dissociate – this is to be expected). Yet, when you search for a transition state or you want to get insight into the shapes of the potential energy surfaces involved in a reaction it might be a good idea to use this feature. One possibility to ease the burden somewhat is to perform the relaxed surface scan with a "fast" method and a smaller basis set and then do single point calculations on all optimized geometries with a larger basis set and/or higher level of theory. At least you can hope that this should give a reasonable approximation to the desired surface at the higher level of theory – this is the case if the geometries at the lower level are reasonable.
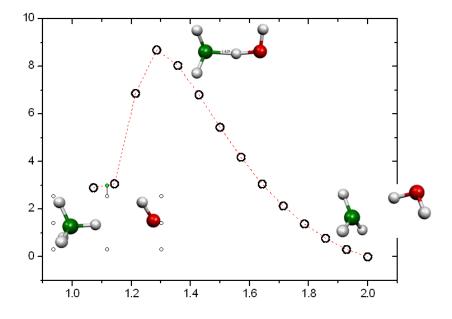
Figure 5.12: Relaxed surface scan for the H-atom abstraction from CH_4 by OH-radical (B3LYP/SV(P)).

5.2.9 Multiple XYZ File Scans

A different type of scan is implemented in ORCA in conjunction with relaxed surface scans. Such scans produce a series of structures that are typically calculated using some ground state method. Afterwards one may want to do additional or different calculations along the generated pathway such as excited state calculations or special property calculations. In this instance, the "multiple XYZ scan" feature is useful. If you request reading from a XYZ file via:

```
* xyzfile Charge Multiplicity FileName
```

this file could contain a number of structures. The format of the file is:

```
Number of atoms M
  Comment line
AtomName1 X Y Z
AtomName2 X Y Z
...
AtomNameM X Y Z
>
Number of atoms M
  Comment line
AtomName1 X Y Z
...
```

Thus, the structures are simply of the standard XYZ format, separated by a '>' sign. After the last structure no '>' should be given but a blank line instead. The program then automatically recognizes that a multiple XYZ scan run

is to be performed. Thus, single point calculations are performed on each structure in sequence and the results are collected at the end of the run in the same kind of trajectory.dat files as produced from trajectory calculations.

In order to aid in using this feature, the relaxed surface scans produce a file called MyJob.allxyz that is of the correct format to be re-read in a subsequent run.

5.2.10 Transition States

5.2.10.1 Introduction to Transition State Searches

If you provide a good estimate for the structure of the transition state (TS) structure, then you can find the respective transition state with the following keywords (in this example we take the structure with highest energy of the above relaxed surface scan):

```
! UKS B3LYP SV(P) TightSCF SlowConv NumFreq
! OptTS            # performs a TS optimization with the EF-algorithm
# Transition state: H-atom abstraction from CH4 to OH-radical
%geom Calc_Hess true # calculation of the exact Hessian
                     #  before the first optimization step
      Hybrid_Hess    # requests the calculation of a Hybrid Hessian
        [0 1 5 6]    #  for the atoms 0, 1, 5 and 6 instead of the
         end         #  exact Hessian. If Hybrid_Hess is not given,
                     #  then a full exact Hessian is calculated.
    end


    end
* int 0 2
    C  0 0 0 0.000000   0.000    0.000
    H  1 0 0 1.285714    0.000    0.000
    H  1 2 0 1.100174 107.375    0.000
    H  1 2 3 1.100975 103.353 119.612
    H  1 2 3 1.100756 105.481 238.889
    O  2 1 3 1.244156 169.257  17.024
    H  6 2 1 0.980342 100.836  10.515
*
```

NOTE:

- You need a good guess of the TS structure. Relaxed surface scans can help in almost all cases (see also example above).

- For TS optimization (in contrast to geometry optimization) an exact Hessian, a Hybrid Hessian or a modification of selected second derivatives is necessary.

- Check the eigenmodes of the optimized structure for the eigenmode with a single imaginary frequency. You can also visualize this eigenmode with `orca_pltvib` (section 5.9.3.5).

- If the Hessian is calculated during the TS optimization, it is stored as basename.001.hess, if it is recalculated several times, then the subsequently calculated Hessians are stored as basename.002.hess, basename.003.hess, . . .

- If you are using the Hybrid Hessian, then you have to check carefully at the beginning of the TS optimization (after the first three to five cycles) whether the algorithm is following the correct mode (see TIP below). If this is not the case you can use the same Hybrid Hessian again via the inhess read keyword and try to target a different mode (via the TS_Mode keyword, see below).

In the example above the TS mode is of local nature. In such a case you can directly combine the relaxed surface scan with the TS optimization with the

```
! ScanTS
```

command, as used in the following example:

```
! UKS B3LYP SV(P) TightSCF SlowConv
! ScanTS # perform a relaxed surface scan and TS optimization
        #  in one calculation
%geom scan B 1 0 = 2.0, 1.0, 15 end end
* int 0 2
  C     0   0   0   0.000000     0.000     0.000
  H     1   0   0   1.999962     0.000     0.000
  H     1   2   0   1.095870   100.445     0.000
  H     1   2   3   1.095971    90.180   119.467
  H     1   2   3   1.095530    95.161   238.880
  O     2   1   3   0.984205   164.404    27.073
  H     6   2   1   0.972562   103.807    10.843
*
```

NOTE:

- The algorithm performs the relaxed surface scan, aborts the Scan after the maximum is surmounted, chooses the optimized structure with highest energy, calculates the second derivative of the scanned coordinate and finally performs a TS optimization.

- If you do not want the scan to be aborted after the highest point has been reached but be carried out up to the last point, then you have to type:

```
%geom
      fullScan true # do not abort the scan with !ScanTS
   end
```

As transition state finder we implemented the quasi-Newton like hessian mode following algorithm. [83–91] This algorithm maximizes the energy with respect to one (usually the lowest) eigenmode and minimizes with respect to the remaining 3N-7(6) eigenmodes of the Hessian.

TIP

- You can check at an early stage if the optimization will lead to the "correct" transition state. After the first optimization step you find the following output for the redundant internal coordinates:

```
       ----------------------------------------------------------------------
                      Redundant Internal Coordinates
                         (Angstroem and degrees)
           Definition                   Value    dE/dq    Step   New-Value  frac.(TS mode)
       ----------------------------------------------------------------------
        1. B(H   1,C   0)               1.2857  0.013136  0.0286   1.3143        0.58
        2. B(H   2,C   0)               1.1002  0.014201 -0.0220   1.0782
        3. B(H   3,C   0)               1.1010  0.014753 -0.0230   1.0779
        4. B(H   4,C   0)               1.1008  0.014842 -0.0229   1.0779
        5. B(O   5,H   1)               1.2442 -0.015421 -0.0488   1.1954        0.80
        6. B(H   6,O   5)               0.9803  0.025828 -0.0289   0.9514
        7. A(H   1,C   0,H   2)        107.38  -0.001418  -0.88   106.49
        8. A(H   1,C   0,H   4)        105.48  -0.002209  -0.46   105.02
        9. A(H   1,C   0,H   3)        103.35  -0.003406   0.08   103.43
       10. A(H   2,C   0,H   4)        113.30   0.001833   0.35   113.65
       11. A(H   3,C   0,H   4)        113.38   0.002116   0.26   113.64
       12. A(H   2,C   0,H   3)        112.95   0.001923   0.45   113.40
       13. A(C   0,H   1,O   5)        169.26  -0.002089   4.30   173.56
       14. A(H   1,O   5,H   6)        100.84   0.003097  -1.41    99.43
       15. D(O   5,H   1,C   0,H   2)   17.02   0.000135   0.24    17.26
       16. D(O   5,H   1,C   0,H   4) -104.09  -0.000100   0.52  -103.57
       17. D(O   5,H   1,C   0,H   3)  136.64   0.000004   0.39   137.03
       18. D(H   6,O   5,H   1,C   0)   10.52   0.000078  -0.72     9.79
       ----------------------------------------------------------------------
```

Every Hessian eigenmode can be represented by a linear combination of the redundant internal coordinates. In the last column of this list the internal coordinates, that represent a big part of the mode which is followed uphill, are labelled. The numbers reflect their magnitude in the TS eigenvector (fraction of this internal coordinate in the linear combination of the eigenvector of the TS mode). Thus at this point you can already check whether your TS optimization is following the right mode (which is the case in our example, since we are interested in the abstraction of H_1 from C_0 by O_5).

- If you want the algorithm to follow a different mode than the one with lowest eigenvalue, you can either choose the number of the mode:

```
%geom
   TS_Mode {M 1} # {M 1} mode with second lowest eigenvalue
      end         #  (default: {M 0}, mode with lowest eigenvalue)
   end
```

or you can give an internal coordinate that should be strongly involved in this mode:

```
%geom
   TS_Mode {B 1 5} # bond between atoms 1 and 5,
      end           # you can also choose an angle: {A N1 N2 N1}
                    #  or a dihedral: {D N1 N2 N3 N4}
   end
```

5.2.10.2 Hessians for Transition State Calculations

For transition state (TS) optimization a simple initial Hessian, which is used for minimization, is not sufficient. In a TS optimization we are looking for a first order saddle point, and thus for a point on the PES where the curvature is negative in the direction of the TS mode (the TS mode is also called transition state vector, the only eigenvector of the Hessian at the TS geometry with a negative eigenvalue). Starting from an initial guess structure the algorithm used in the ORCA TS optimization has to climb uphill with respect to the TS mode, which means that the starting structure has to be near the TS and the initial Hessian has to account for the negative curvature of the PES at that point. The simple force-field Hessians can not account for this, since they only know harmonic potentials and thus positive curvature. The most straightforward option in this case would be (after having looked for a promising initial guess structure with the help of a relaxed surface scan) to calculate the exact Hessian before starting the TS optimization. With this Hessian (depending on the quality of the initial guess structure) we know the TS eigenvector with its negative eigenvalue and we have also calculated the exact force constants for all other eigenmodes (which should have positive force constants). The calculation of the exact Hessian is very time consuming, and one could ask if it is really necessary to calculate the full exact Hessian since the only special thing (compared to the simple force-field Hessians) that we need is the TS mode with a negative eigenvalue. Here ORCA provides two different possibilities to speed up the Hessian calculation, depending on the nature of the TS mode: the Hybrid Hessian and the calculation of the Hessian value of an internal coordinate. For both possibilities the initial Hessian is based on a force-field Hessian and only parts of it are calculated exactly. If the TS mode is of very local nature, which would be the case when e.g. cleaving or forming a bond, then the exactly calculated part of the Hessian can be the second derivative of only one internal coordinate, the one which is supposed to make up the TS mode (the formed or cleaved bond). If the TS mode is more complicated and more delocalized, as e.g. in a concerted proton transfer reaction, then the hybrid Hessian, a Hessian matrix in which the numerical second derivatives are calculated only for those atoms, which are involved in the TS mode (for more details, see section 6.13), should be sufficient. If you are dealing with more complicated cases where these two approaches do not succeed, then you still have the possibility to start the TS optimization with a full exact Hessian.

Numerical Frequency calculations are quite expensive. You can first calculate the Hessian at a lower level of theory or with a smaller basis set and use this Hessian as input for a subsequent TS optimization:

```
%geom inhess    Read       # this command comes with the following:
      InHessName "yourHessian.hess"
                           # filename of Hessian input file
   end
```

Another possibility to save computational time is to calculate exact Hessian values only for those atoms which are crucial for the TS optimization and to use approximate Hessian values for the rest. This option is very useful for big systems, where only a small part of the molecule changes its geometry during the transition and hence the information of the full exact Hessian is not necessary. With this option the coupling of the selected atoms are calculated exactly and the remaining Hessian matrix is filled up with a model initial Hessian:

```
%geom Calc_Hess true
      Hybrid_Hess [0 1 5 6] end # calculates a Hybrid Hessian
   end                          # with exact calculation for
                                # atoms 0, 1, 5 and 6
```

For some molecules the PES near the TS can be very far from ideal for a Newton-Raphson step. In such a case ORCA can recalculate the Hessian after a number of steps:

```
%geom
      Recalc_Hess  5 # calculate the Hessian at the beginning
   end               # and recalculate it after 5,10,15,... steps
```

Another solution in that case is to switch on the trust radius update, which reduces the step size if the Newton-Raphson steps behave unexpected and ensures bigger step size if the PES seems to be quite quadratic:

```
%geom
   Trust 0.3 # Trust <0 - use fixed trust radius.
             # Trust >0 - use trust radius update. I.e. 0.3 means:
             # start with trust radius 0.3 and use trust
   end       # radius update. (default -0.3 au)
```

5.2.10.3 Special Coordinates for Transition State Optimizations

- If you look for a TS of a breaking bond the respective internal coordinate might not be included in the list of redundant internal coordinates (but this would accelerate the convergence). In such a case (and of course in others) you can add coordinates to or remove them from the set of autogenerated redundant internal coordinates:

```
# add ( A ) or remove ( R ) internal coordinates
%geom
      modify_internal
         { B 10 0 A }     # add a bond between atoms 0 and 10
         { A 8 9 10 R }   # remove the angle defined
                          # by atoms 8, 9 and 10
         { D 7 8 9 10 R } # remove the dihedral angle defined
      end                 # by atoms 7, 8, 9 and 10
   end
```

5.2.11 MECP Optimization

There are reactions where the analysis of only one spin state of a system is not sufficient, but where the reactivity is determined by two or more different spin states (Two- or Multi-state reactivity). The analysis of such reactions reveals that the different PESs cross each other while moving from one stationary point to the other. In such a case you might want to use the ORCA optimizer to locate the point of lowest energy of the crossing surfaces (called the minimum energy crossing point, MECP).

As an example for such an analysis we show the MECP optimization of the quartet and sextet state of $[FeO]^+$.

```
!B3LYP TZVP Opt SurfCrossOpt SlowConv
%mecp
  Mult 4
 end
* xyz +1 6
Fe 0.000000    0.000000    1.000000
O  0.000000    0.000000    1.670000
*
```

NOTE:
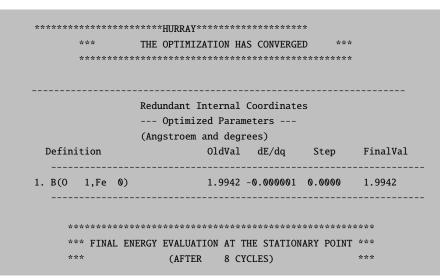
- For further options for the MECP calculation, see section 6.13.3.

TIP:

- You can often use a minimum or TS structure of one of the two spin states as initial guess for your MECP-optimization. If this doesn't work, you might try a scan to get a better initial guess.

The results of the MECP optimization are given in the following output. The distance where both surfaces cross is at 1.994 Å. In this simple example there is only one degree of freedom and we can also locate the MECP via a parameter scan. The results of the scan are given in Figure for comparison. Here we see that the crossing occurs at a Fe-O-distance of around 2 Å.

For systems with more than two atoms a scan is not sufficient any more and you have to use the MECP optimization.

```
*************************HURRAY*********************
      ***         THE OPTIMIZATION HAS CONVERGED     ***
      *************************************************


      ----------------------------------------------------------------
                  Redundant Internal Coordinates
                  --- Optimized Parameters ---
                  (Angstroem and degrees)
       Definition                OldVal   dE/dq    Step     FinalVal
       ----------------------------------------------------------------
      1. B(O   1,Fe  0)           1.9942 -0.000001  0.0000    1.9942
       ----------------------------------------------------------------


         *******************************************************
         *** FINAL ENERGY EVALUATION AT THE STATIONARY POINT ***
         ***                (AFTER    8 CYCLES)              ***
```
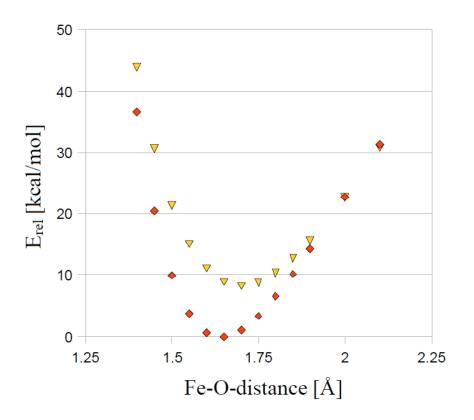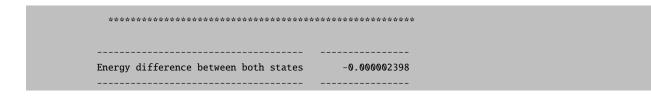
Figure 5.13: Parameter scan for the quartet and sextet state of $[FeO]^+$ (B3LYP/SV(P)).

```
          *********************************************************


          ---------------------------------------    ----------------
          Energy difference between both states         -0.000002398
          ---------------------------------------    ----------------
```

A more realistic example with more than one degree of freedom is the MECP optimization of a structure along the reaction path of the $CH_3O \leftrightarrow CH_2OH$ isomerization.

```
    !B3LYP SV SurfCrossOpt SurfCrossNumFreq
    %mecp Mult 1
      end
    *xyz +1 3
      C     0.000000     0.000000     0.000000
      H     0.000000     0.000000     1.300000
      H     1.026719     0.000000    -0.363000
      O    -0.879955     0.000000    -1.088889
      H    -0.119662    -0.866667     0.961546
    *
```

NOTE:

- To verify that a stationary point in a MECP optimization is a minimum, you have to use an adapted frequency analysis, called by `SurfCrossNumFreq` (see section 6.13.3).

5.2.12 **ORCA** as External Optimizer

ORCA can be used as external optimizer. For details on how to use only the optimizer of ORCA, have a look at section 6.13.5.

5.3 **Vibrational Frequencies**

Vibrational frequency calculations are available through analytical differentiation of the SCF energy as well as one- or two-sided numerical differentiation of analytical gradients, i.e. for Hartree-Fock and DFT models. For methods without analytical gradient a numerically calculated gradient can be used (Keyword NumGrad) for numerical frequencies. Please note, that this will be a very time consuming calculation.

The use of vibrational frequency calculations is fairly simple:

```
# any Hartree-Fock or DFT model can be used here
! RKS BP TZVP RI TZV/J
# --------------------------------------------------------
# These settings are advisable in order to minimize the
# numerical noise in the frequencies. You can/should also
# use larger integration grids, i.e. Grid4 NoFinalGrid
! TightSCF
# --------------------------------------------------------
# perform a geometry optimization first
! Opt
# --------------------------------------------------------
# Run an analytical or numerical frequency calculation afterwards
! AnFreq
# or
! Freq
# numerical:
! NumFreq
# -------------------------------------------------------
# details of the analytical frequency calculation
%freq  Hess2ElFlags 1,2,2,1   # use the RIJCOSX approximation
                              # (this is the new default for RIJCOSX Hessian)
       end
# --------------------------------------------------------
# --------------------------------------------------------
# details of the numerical frequency calculation
%freq  CentralDiff true   # use central-differences
                          # (this is the new default)
       Increment   0.005  # increment in bohr for the
                          # differentiation (default 0.005)
       end
# --------------------------------------------------------
```

```
    ! bohrs
    * xyz 0 1
     O          -1.396288    -0.075107     0.052125
     O           1.396289    -0.016261    -0.089970
     H          -1.775703     1.309756    -1.111179
     H           1.775687     0.140443     1.711854
    *
```

The user has full controll over approximations involved in the Hessian calculation. Hess2ElFlags(i1, i2, i3, i4), where ik (ik=0 to 2) adjust the use of the RI and COSX approximations. In case ik=0 no approximation in the 2-electron integrals are introduced. ik=1 means the RI approximation in the Coulomb part is in use. ik=2 corresponds to COSX algorithm in the HF exchange. i1 corresponds to explicit Fock matrix derivatives, i2 - Fock matrix depended on pseudo density, i3 - solution of the CP-SCF equations, i4 - explicit integral second derivative. In the introduced notation the RIJDX default flags are (1, 1, 1, 1), the RIJCOSX are (1, 2, 2, 1). Analytical frequency calculations are also implemented at MP2 level. Please note, that the Hess2ElFlags are ignored by the MP2 module. Furthermore, MP2 frequency calculations are very time-consuming and need a lot of disk space.

At the end of the frequency job you get an output like this:

```
    -----------------------
    VIBRATIONAL FREQUENCIES
    -----------------------


      0:         0.00 cm**-1
      1:         0.00 cm**-1
      2:         0.00 cm**-1
      3:         0.00 cm**-1
      4:         0.00 cm**-1
      5:         0.00 cm**-1
      6:       311.78 cm**-1
      7:       887.65 cm**-1
      8:      1225.38 cm**-1
      9:      1394.81 cm**-1
     10:      3624.88 cm**-1
     11:      3635.73 cm**-1
```

This output consists of the calculated vibrational frequencies, the vibrational modes and the thermochemical properties at 298.15 K. In the example above there are six frequencies which are identically zero. These frequencies correspond to the rotations and translations of the molecule. They have been projected out of the Hessian before the calculation of the frequencies and thus, the zero values do not tell you anything about the quality of the Hessian that has been diagonalized. The reliability of the calculated frequencies has to be judged by comparison of calculations with different convergence criteria, increments, integration grids etc. The numerical error in the frequencies may reach 50 cm^{-1} but should be considerably smaller in most cases. Significant negative frequencies of at least a few hundred cm^{-1} indicate saddle points of the energy hypersurface and prove that the optimization has not resulted in an energy minimum.

NOTE

- The calculation of frequencies puts rather high demands on both computer time and accuracy. In order to get reliable frequencies make sure that:

  - Your SCF is tightly converged. A convergence accuracy of at least 10^{-7} Eh in the total energy and 10^{-6} in the density is desirable.

  - Grids of at least `Grid4` or larger are recommended.

  - The use of two-sided (i.e. central) differences increases the computation time by a factor of two but gives more accurate and reliable results.

  - Small auxiliary basis sets like DGauss/J or DeMon/J have not resulted in fully converged frequencies in my hands (up to 40 cm^{-1} difference compared to frequencies calculated without RI). The larger Ahlrichs aux-basis sets SV/J and TZV/J are thought to give more reliable results.

  - Possibly, the convergence criteria of the geometry optimization need to be tightened in order to get fully converged results.

  - If you can afford it, decrease the numerical increment to 0.001 Bohr or so. This puts even higher demands on the convergence characteristics of the SCF calculation but should also give more accurate numerical second derivatives. If the increment is too small or too high inaccurate results are expected. The value 0.02 bohr used by many other workers has been found to be too large in my hands (residual frequencies of ca. 50 cm^{-1}).

Analytical frequencies calculation is a memory consuming step. To control memory consumption the `%maxcore` parameter must be set. For example `%maxcore 8192` - use 8 Gb of memory for the calculation. The user should provide the value according to the computer available memory. The batching based on `%maxcore` parameter will be introduced automatically to overcome probable memory shortage.

Numerical frequency calculations are restartable. If the job died for one reason or another you can simply continue from where it stopped as in the following example:

```
! RHF STO-3G
! NumFreq
%freq  Restart true  # restart an old calculation
                     # this requires a .hess file to be present
      end
* int 0 1
   C  0 0 0 0.0000   0 0
   C  1 0 0 1.2160   0 0
   H  1 2 0 1.083  180 0
   H  2 1 3 1.083  180 0
*
```

NOTE

- You must not change the level of theory, basis set or any other detail of the calculation. Any change will produce an inconsistent, essentially meaningless Hessian.

- The geometry at which the Hessian is calculated must be identical. If you followed a geometry optimization by a frequency run then you must restart the numerical frequency calculation from the optimized geometry.

- Numerical frequencies can be performed in multi-process mode. Please see section 2.2 ('Hints on the use of parallel ORCA') for more information.

- In multi-process mode the restart will take off from the locally calculated Hessians (`.procmyid.hess`, e.g. `.proc0.hess`, `.proc1.hess`) where `myid` is the process id of some local process. Please make sure that all these local Hessians get copied to your compute directory. If restart is set and no local files to be found, ORCA will try to restart on the global Hessian file.

- The Hessian file will contain a joblist, showing which displacements have been performed. You can recalculate a single (or more) displacements by changing the 1 ('done'-) entry to 0 ('to-be-done').

5.4 Reaction Energies from Compound Jobs

In order to make your life easier, we have implemented a mechanism that allows you to run several fragments of a reaction in one input file and obtain the energy difference between reactants, intermediates and products directly in the output file. The mechanism is such that you run a compound job and assign each fragment that you whish to include in the energy profile a name:

```
%id "Educt"
...
$new_job
%id "Educt"


...
$new_job
%id product
...
```

In this example, there are three fragments. The first two are educts and the third one is a product. Thus we look at a reaction A+B→C. However, more complicated transformations may easily be specified in an analogous way. The final energies are always referenced to the first fragment encountered in the file. It is of course your responsibility that the fragments make sense and that the chemical reactions that your are calculating are balanced with respect to nuclei and electrons.

Let us look at a very simple example, the reaction of H_2 and CO to give H_2CO:

```
! B3LYP SV(P)
%id "educt"
* xyz 0 1
H 0.000 0.000 0.000
H 0.75 0.000 0.000
*
$new_job
! B3LYP SV(P) PModel
%id "educt"
* xyz 0 1
```

```
C 0.000 0.000 0.000
O 1.13 0.000 0.000
*
$new_job
! B3LYP SV(P) PModel
%id "product"
* int 0 1
C 0 0 0 0 0 0
O 1 0 0 1.2 0 0
H 1 2 0 1.1 120 0
H 1 2 3 1.1 120 180
*
```

NOTE:

- You are not limited to "educt" and "product". You might as well include intermediates, transition states, minimum energy crossing points etc. in the list of fragments

If you do not have optimized geometries for each species you may want to run a geometry optimization at a lower level and obtain energies at a higher level. This can be accomplished in an automatic way as follows:

```
# ---------------
# Now fragment 1: H2 opt at BP/SV(P) then single point CCSD(T)
# ---------------
! BP SV(P) TightSCF Opt PModel
%base "H2-BP"
* xyz 0 1
H 0.000 0.000 0.000
H 0.75 0.000 0.000
*
$new_job
! RHF CCSD(T) TZVPP
%id "educt"
%base "H2-CCSDT"
* xyzfile 0 1 H2-BP.XYZ
# ---------------
# Now fragment 2: CO opt at BP/SV(P) then single point CCSD(T)
# ---------------
$new_job
! BP SV(P) TightSCF Opt PModel
%base "CO-BP"
* xyz 0 1
C 0.000 0.000 0.000
O 1.13 0.000 0.000
*
```

```
    $new_job
    ! RHF CCSD(T) TZVPP
    %id "educt"
    %base "CO-CCSDT"
    * xyzfile 0 1 CO-BP.XYZ
    # ---------------
    # Now fragment 3: H2CO opt at BP/SV(P) then single point CCSD(T)
    # ---------------
    $new_job
    ! BP SV(P) TightSCF Opt PModel
    %base "H2CO-BP"
    * int 0 1
    C 0 0 0 0 0 0
    O 1 0 0 1.2 0 0
    H 1 2 0 1.1 120 0
    H 1 2 3 1.1 120 180
    *
    $new_job
    ! RHF CCSD(T) TZVPP
    %id "product"
    %base "H2CO-CCSDT"
    * xyzfile 0 1 H2CO-BP.XYZ
```

NOTE:[10]

- For each new fragment, the guess has to be explicitly specified since otherwise the program will try to read it from the previous job and this is not a good idea since the number and identity of atoms does not match

- Note that "%id" is only given for the CCSD(T) jobs. This ensures that only these energies will occur later in the calculation of the reaction energies.

5.5 Excited States with CIS, CIS(D), ROCIS and TD-DFT

ORCA features a module to perform TD-DFT and single-excitation CI (CIS). The module works with either closed shell (RHF or RKS) or unrestricted (UHF or UKS) reference wavefunctions. For DFT models the module automatically chooses TD-DFT and for HF wavefunctions the CIS model. If the RI approximation is used in the SCF part it will also be used in the CIS/TD-DFT calculation. A detailed documentation is provided in section 6.14.

5.5.1 General Use

In its simplest form it is only necessary to provide the number of roots sought:

---

[10] This job produces a CCSD(T) reaction energy – not corrected for ZPE – of -3.7 kcal/mol; B3LYP gets -11.2 kcal/mol – a surprisingly large deviation - and B2PLYP predicts -6.1 kcal/mol which is certainly in the ballpark)

```
    ! RKS B3LYP/G SV(P) TightSCF


%tddft nroots    8
       maxdim   64
       triplets true
       end


* int 0 1
    C(1)  0 0 0 0.00 0.0     0.00
    O(2)  1 0 0 1.20 0.0     0.00
    H(3)  1 2 0 1.08 120     0.00
    H(3)  1 2 3 1.08 120   180.00
*
```

The MaxDim parameter controls the maximum number of expansion vectors in the iterative solution of the CI equations. It is wise to choose it as a multiple of the number of roots sought. The default is the smallest possible choice. The triplets parameter is only valid for closed shell references. If chosen as true the program will also determine the triplet excitation energies in addition to the singlets.

5.5.2 Use of TD-DFT for the Calculation of X-ray Absorption Spectra

In principle X-ray absorption spectra are 'normal' absorption spectra that are just taken in a special high-energy wavelength range. Due to the high energy of the radiation employed (several thousand eV), core-electrons rather than valence electrons are excited. This has two consequences: a) the method becomes element specific because the core-level energies divide rather cleanly into regions that are specific for a given element. b) the wavelength of the radiation is so short that higher-order terms in the expansion of the light-matter interaction become important. Most noticeably, quadrupole intensity becomes important.

X-ray absorption spectra can be generally divided into three regions: a) the pre-edge that corresponds to transitions of core electrons into low lying virtual orbitals that lead to bound states. b) the rising edge that corresponds to excitations to high-lying states that are barely bound, and c) the extended X-ray absorption fine structure region (EXAFS) that corresponds to electrons being ejected from the absorber atom and scattered at neighbouring atoms.

With the simple TD-DFT calculations described here, one focuses the attention on the pre-edge region. Neither the rising edge nor the EXAFS region are reasonably described with standard electronic structure methods and no comparison should be attempted. In addition, these calculations are restricted to K-edges as the calculation of L-edges is much more laborious and requires a detailed treatment of the core hole spin orbit coupling.

It is clearly hopeless to try to calculate enough states to cover all transitions from the valence to the pre-edge region. Hence, instead one hand-selects the appropriate donor core orbitals and only allows excitations out of these orbitals into the entire virtual space. This approximation was justified for example in DeBeer-George, Petrenko and Neese (**2008**), *J. Phys. Chem.*, 112, 12936. One should distinguish two situations: First, the core orbital in question may be well isolated and unambiguously defined. This is usually the case for metal 1s orbitals if there is only one metal of the given type in the molecule. Secondly, there may be several atoms of the same kind in the molecule and their core orbitals form the appropriate symmetry adapted linear combinations dictated by group theory. In this latter case special treatment is necessary: The sudden approximation dictates that the excitations occurs from a local core orbital. In previous versions of the program you had to manually localize the core holes. In the present version there is an automatic procedure that is described below.

A typical example is TiCl$_4$. If we want to calculate the titanium K-edge, the following input is appropriate:

```
! BP86 def2-TZVP(-f) def2-TZVP/J ZORA TightSCF
%maxcore 1024

%tddft   OrbWin[0] = 0,0,-1,-1
         NRoots    25
         MaxDim    150
         DoQuad    true
         end

* int 0 1
Ti 0 0 0  0 0 0
Cl 1 2 3 2.15 0 0
Cl 1 2 3 2.15 109.4712 0
Cl 1 2 3 2.15 109.4712 120
Cl 1 2 3 2.15 109.4712 240
*
```

NOTE:

- The absolute transition energies from such calculations are off by a few hundred electron volts due to the shortcomings of DFT. The shift is constant and very systematic for a given element. Hence, calibration is possible and has been done for a number of edges already. Calibration depends on the basis set!

- Quadrupole contributions (and magnetic dipole contributions) have been invoked with `DoQuad true`, which is essential for metal edges. For ligand edges, the contributions are much smaller.

- `OrbWin` is used to select the single donor orbital (in this case the metal 1s). The LUMO (45) and last orbital in the set (174) are selected automatically if '-1' is given. This is different from previous program versions where the numbers had to be given manually.

The output contains standard TD-DFT output but also:

```
-------------------------------------------------------------------------------------------------------
      COMBINED ELECTRIC DIPOLE + MAGNETIC DIPOLE + ELECTRIC QUADRUPOLE SPECTRUM (origin adjusted)
-------------------------------------------------------------------------------------------------------
State  Energy Wavelength   D2      M2      Q2       D2+M2+Q2      D2/TOT   M2/TOT   Q2/TOT
       (cm-1)  (nm)               (*1e6)  (*1e6)
-------------------------------------------------------------------------------------------------------
....
```

This section contains the relevant output since it combines electric dipole, electric quadrupole and magnetic dipole transition intensities into the final spectrum. Importantly, there is a gauge issue with the quadrupole intensity: the results depend on the where the origin is placed. We have proposed a minimization procedure that guarantees the fastest possible convergence of the multipole expansion:

DeBeer-George, S.; Petrenko, T.; Neese, F.: "*Time-dependent density functional calculations of ligand K-edge X-ray absorption spectra*", Inorg. Chim. Acta, (**2008**), 361, 965

The spectra are plotted by calling

```
orca_mapspc  MyOutput.out ABSQ -eV -x04890 -x14915 -w1.3
```

Now, let us turn to the Cl K-edge. Looking at the output of the first calculation, we have:

```
----------------
ORBITAL ENERGIES
----------------

  NO   OCC        E(Eh)          E(eV)
   0   2.0000   -180.182806     -4903.0234
   1   2.0000   -101.517692     -2762.4368
   2   2.0000   -101.517681     -2762.4365
   3   2.0000   -101.517442     -2762.4300
   4   2.0000   -101.517442     -2762.4300
   5   2.0000    -19.814042      -539.1675
   6   2.0000    -16.394212      -446.1092
   7   2.0000    -16.394210      -446.1091
   8   2.0000    -16.394203      -446.1089
   9   2.0000     -9.281040      -252.5499
  10   2.0000     -9.281011      -252.5491
  11   2.0000     -9.280892      -252.5459
  12   2.0000     -9.280892      -252.5459
  18   2.0000     -7.034911      -191.4297
....
```

And looking at the energy range or the orbital composition, we find that orbitals 1 through 4 are Cl 1s-orbitals. They all have the same energy since they are essentially non-interacting. Hence, we can localize them without invalidating the calculation. To this end, you can invoke the automatic localization for XAS which modifies the input to:

```
! BP86 def2-TZVP(-f) def2-TZVP/J ZORA TightSCF
%maxcore 1024

%tddft   XASLoc[0] = 1,4
         OrbWin[0] = 1,1,-1,-1
         NRoots    25
         MaxDim    150
         DoQuad   true
         end

* int 0 1
Ti 0 0 0  0 0 0
Cl 1 2 3 2.15 0 0
Cl 1 2 3 2.15 109.4712 0
Cl 1 2 3 2.15 109.4712 120
Cl 1 2 3 2.15 109.4712 240
*
```

- This localizes the orbitals 1 through 4 of operator 0 (the closed shell) and then allows excitations (arbitrarily) from core hole 1 only. You could choose any of the three other localized 1s orbitals instead without changing the result. You could even do all four core holes simultaneously (they produce identical spectra) in which case you have the entire ligand K-edge intensity and not just the one normalized to a single chlorine (this would be achieved with `OrbWin[0] = 1,4,-1,-1`).

- If you have a spin unrestricted calculation, you need to give the same `XASLoc` and `OrbWin` information for the spin-down orbitals as well.

Quite nice results have been obtained for a number of systems in this way. For example, check:

Ray, K.; DeBeer-George, S.; Solomon, E.I.; Wieghardt, K.; Neese, F. (**2007**) Description of the Ground State Covalencies of the Bis(dithiolato)Transition Metal Complexes Using X-ray Absorption Spectral and Time-Dependent-Density-Functional Studies, *Chem. Eur. Journal*,13, 2753.

5.5.3 Excited State Geometry Optimization

For CIS and TD-DFT the program can calculate analytic gradients. With the help of the `IRoot` keyword, a given state can be selected for geometry optimization. Note however, that if two states cross during the optimization it may fail to converge or fail to converge to the desired excited state!

```
! RHF SVP TightSCF Opt

%cis NRoots    1
     IRoot     1
     end

* int 0 1
   C(1)  0 0 0 0.00 0.0     0.00
   O(2)  1 0 0 1.20 0.0     0.00
   H(3)  1 2 0 1.08 120     0.00
   H(3)  1 2 3 1.08 120   180.00
*
```

Note that this example converges to a saddle point as can be verified through a numerical frequency calculation (which is also possible with CIS). The CIS relaxed density matrix is available from such gradient runs (`MyJob.cisp` when using the `KeepDens` keyword) and can be used for various types of analysis. The COSMO model is not yet implemented into the CIS gradient code. Note that the frozen core option is available starting from version 2.8.0.

5.5.4 Doubles Correction

For CIS (and also for perturbatively corrected double hybrid functionals) the program can calculate a doubles correction to the singles-only excited states. The theory is due to Head-Gordon and co-workers.

```
%cis dcorr n  # n=1,2,3,4 are four different algorithms that
              # lead to (essentially) the same result but differ
              # in the way the rate-limiting steps are handled
```

NOTE:

- CIS(D) is often a quite big improvement over CIS.

- The cost of the (D) correction is $O(N^5)$ and therefore comparable to RI-MP2. Since there are quite a few things more to be done for (D) compared to RI-MP2, expect the calculations to take longer. In the most elementary implementation the cost is about two times the time for RI-MP2 for each root.

- The (D) correction is compatible with the philosophy of the double hybrid density functionals and should be used it these functionals are combined with TD-DFT. The program takes this as the default but will not enforce it.

- In our implementation it is only implemented together with the RI approximation and therefore you need to supply an appropriate ('/C') fitting basis.

- The program will automatically put the RI-MP2 module into operation together with the (D) correction. This will result in the necessary integrals becoming available to the CIS module.

5.5.5 Restricted Open-shell CIS

In addition to the CIS/TD-DFT description of excited states, ORCA features the **orca_rocis** module to perform configuration interaction with single excitations calculations using a restricted open-shell Hartree–Fock (ROHF) reference. It can be used to calculate excitation energies, absorption intensities and CD intensities. In general, ROCIS calculations work on restricted open-shell HF reference functions but in this implementation it is possible to enter the calculations with RHF (only for closed shell molecules) or UHF reference functions as well. If the calculation starts with an UHF/UKS calculation, it will automatically produce the quasi-restricted orbitals which will then be used for the subsequent ROCIS calculations. Note that if the reference function is a RHF/RKS function the method produces the CIS results. The module is invoked by providing the number of roots sought in the **%rocis** block of the input file:

```
! SVP TightSCF

%rocis NRoots 2
       MaxDim 20
       end

* xyz -2 2
Cu  0.00    0.00    0.00
Cl  2.25    0.00    0.00
Cl -2.25    0.00    0.00
Cl  0.00    2.25    0.00
Cl  0.00   -2.25    0.00
*
```

In this example the `MaxDim` parameter is given in addition to the number of roots to be calculated. It controls the maximum dimension of the expansion space in the Davidson procedure that is used to solve the CI problem.

5.6 Excited States with EOM-CCSD

The methods described in the previous section are all based on the single excitation framework. For a more accurate treatment, double excitations should also be considered. The equation of motion (EOM) CCSD method (and the closely related family of linear response CC methods) provides an accurate way of describing excited states based on singles and doubles excitations within the coupled cluster framework. In this chapter, the typical usage of the EOM-CCSD routine will be described, along with a short list of its present capabilities. A detailed description will be given in 6.16.

5.6.1 General Use

The simplest way to call for an EOM calculation is via the usage of the EOM-CCSD keyword, together with a specification of the desired number of roots:

```
! RHF EOM-CCSD cc-pVDZ TightSCF

%mdci
  nroots 9
end

*xyz 0 1
  C     0.016227    -0.000000     0.000000
  O     1.236847     0.000000    -0.000000
  H    -0.576537     0.951580    -0.000000
  H    -0.576537    -0.951580    -0.000000
*
```

The above input will call the EOM routine with default settings. The main output is a list of excitation energies augmented with some further state specific data. For the above input the following output is obtained:

```
----------------
EOM-CCSD RESULTS
----------------

IROOT= 1:  0.147819 au     4.022 eV   32442.4 cm**-1
  Amplitude    Excitation
  -0.107929     4 ->   8
   0.665488     7 ->   8
  -0.104687     7 ->   8    6 ->   8
  Ground state amplitude:  0.000000
IROOT= 2:  0.314118 au     8.548 eV   68940.9 cm**-1
  Amplitude    Excitation
   0.671202     7 ->   9
  -0.135186     7 ->   9    6 ->   8
  Ground state amplitude: -0.000000
IROOT= 3:  0.343833 au     9.356 eV   75462.6 cm**-1
  Amplitude    Excitation
   0.670738     5 ->   8
```

```
      -0.112189      6 ->   8      5 ->    8
      Ground state amplitude:  0.000000
   IROOT= 4:  0.364190 au      9.910 eV   79930.5 cm**-1
      Amplitude     Excitation
       0.102899      4 ->  10
       0.484495      6 ->   8
      -0.438408      7 ->  10
      -0.166811      6 ->   8      6 ->    8
      Ground state amplitude: -0.037905
   IROOT= 5:  0.389389 au     10.596 eV   85461.1 cm**-1
      Amplitude     Excitation
      -0.646803      4 ->   8
      -0.122378      7 ->   8
      -0.171688      7 ->   8      6 ->    8
      Ground state amplitude: -0.000000
   IROOT= 6:  0.414583 au     11.281 eV   90990.4 cm**-1
      Amplitude     Excitation
       0.378458      6 ->   8
       0.537310      7 ->  10
      -0.124160      6 ->   8      6 ->    8
      -0.111795      7 ->  10      6 ->    8
      Ground state amplitude: -0.055334
   IROOT= 7:  0.423864 au     11.534 eV   93027.3 cm**-1
      Amplitude     Excitation
       0.673775      7 ->  11
      Ground state amplitude:  0.000000
   IROOT= 8:  0.444213 au     12.088 eV   97493.4 cm**-1
      Amplitude     Excitation
       0.664898      6 ->   9
      -0.130812      6 ->   9      6 ->    8
      Ground state amplitude: -0.000000
   IROOT= 9:  0.510512 au     13.892 eV  112044.3 cm**-1
      Amplitude     Excitation
       0.665729      6 ->  10
      -0.114405      6 ->  15
      -0.124717      6 ->  10      6 ->    8
      Ground state amplitude: -0.000000
```

5.6.2 Capabilities

At present, the EOM routine is capable to run excited state calculations for closed shell systems using an RHF reference function. It can be used for serial and parallel calculations. For the time being, the most useful information provided is the list of excitation energies. Wave function parameters are of little use in themselves, but they are essential for excited state density calculations and property evaluations. These latter functions will be available in future, along with many other additional capabilities.

5.7 Multireference Configuration Interaction and Pertubation Theory

5.7.1 Introductory Remarks

5.7.1.1 Introduction

ORCA contains a multireference correlation module (configuration interaction, MR-CI, and perturbation theory, MR-PT). Although there has been quite a bit of experience with it, this part of the program is still somewhat hard to use and requires patience and careful testing before the results should be accepted. While we try to make your life as easy as possible, you have to be aware that ultimately any meaningful multireference *ab initio* calculation requires more insight and planning from the user side than standard SCF or DFT calculation or single reference correlation approaches like MP2 – so don't be fainthearted! You should also be aware that with multireference methods it is very easy to let a large computer run for a long time and still to not produce a meaningful result – your insight is a key ingredient to a successful application! Below a few examples illustrate some basic uses of the `orca_mrci` module.

5.7.1.2 RI-approximation

First of all, it is important to understand that the default mode of the MR-CI module in its present implementation performs a full integral transformation from the AO to the MO basis. This becomes very laborious and extremely memory intensive beyond approximately 200 MOs that are included in the CI. Alternatively, one can construct molecular electron-electron repulsion integrals from the resolution of the identity (RI) approximation. *Thus a meaningful auxiliary basis set must be provided if this option is chosen*. We recommend the fitting bases developed by the TurboMole developers for MP2 calculations. These give accurate transition energies; however, the error in the total energies is somewhat higher and may be on the order of 1 mEh or so. Check `IntMode` to change the default mode for the integral transformation. Note that in either way, the individually selecting MRCI module requires to have all integrals in memory which sets a limit on the size of the molecule that can be studied.

5.7.1.3 Individual Selection

Secondly, it is important to understand that the MR-CI module is of the *individually selecting* type. Thus, only those excited configuration state functions (CSFs) which interact more strongly than a given threshold ($\mathbf{T_{sel}}$) with the 0th order approximations to the target states will be included in the variational procedure. The effect of the rejected CSFs is estimated using second order perturbation theory. The 0th order approximations to the target states are obtained from the diagonalization of the reference space configurations. A further approximation is to reduce the size of this reference space through another selection – all initial references which contribute less than a second threshold ($\mathbf{T_{pre}}$) to the 0th order states are rejected from the reference space.

5.7.1.4 Single excitations

One important aspect concerns the single excitations. If the reference orbitals come from a CASSCF calculation the matrix elements between the reference state and the single excitations vanishes and the singles will not be selected. However, they contribute to fourth and higher orders in perturbation theory and may be important for obtaining smooth potential energy surfaces and accurate molecular properties. Hence, the default mode of the MRCI module requires to include all of the single excitations via the flag `AllSingles =true`. This may lead to lengthy computations if the reference spaces becomes large!

5.7.1.5 Reference Spaces
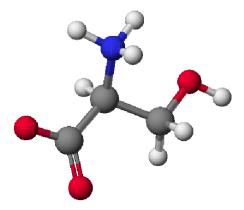
Third, the reference spaces in the MR-CI module can be of the complete active space (**CAS(n-electrons,m-orbitals)**) or restricted active space (**RAS**, explained later) type. It is important to understand that the program uses the orbitals around the HOMO-LUMO gap as provided by the user to build up the reference space! Thus, if the orbitals that you want to put in the active space are not coming "naturally" from your SCF calculation in the right place you have to reorder them using the "`moread`" and "`rotate`" features together with the `NoIter` directive. To select the most meaningful and economic reference space is the most important step in a multireference calculation. It *always* requires insight from the user side and also care and, perhaps, a little trial and error.

5.7.1.6 Size Consistency

Fourth, it is important to understand that CI type methods are *not* size consistent. Practically speaking the energy of the supermolecule A-B with noninteracting A and B fragments is not equal to the energies of isolated A and islotated B. There are approximate ways to account for this (**ACPF, AQCC** and **CEPA** methods) but the effect will be present in the energies, the more so the more electrons are included in the treatment. The same is *not* true for the perturbation theory based methods which are size consistent as long as the reference wavefunction is.

5.7.1.7 Performance

There are many flags that control the performance of the MR-CI program. Please refer to chapter 0 for a description of possible flags, thresholds and cut-offs. The most important thresholds are T_{sel} and T_{pre} and for SORCI also T_{nat}.

For some methods, like ACPF, it is possible to compare the performance of the MRCI module with the performance of the MDCI module. The MDCI module has been written to provide optimum performance if no approximations are introduced. The MRCI module has ben written more with the idea of flexibility rather than the idea of performance. Let us compare the performance of the two programs in a slightly nontrivial calculation – the zwitter-ionic form of serine. We compare the selecting MRCI approach with the approximation free MDCI module. The molecular size is such that still all four index integrals can be stored in memory.

Table 5.9: Comparison of the performance of the MRCI and MDCI modules for a single reference calculation with the bn-ANO-DZP basis set on the zwitter-ionic form of serine (14 atoms, 133 basis functions).

Module	Method	T_{sel}(Eh)	Time (sec)	Energy (Eh)
MRCI	ACPF	10^{-6}	3277	-397.943250

MDCI	ACPF	0	1530	-397.946429
MDCI	CCSD	0	2995	-397.934824
MDCI	CCSD(T)	0	5146	-397.974239

The selecting ACPF calculation selects about 15% of the possible double excitations and solves a secular problem of size $\approx$ 360,000 CSFs. The MDCI module ACPF calculation optimizes approximately 2.5 million wavefunction amplitudes — and this is not a large molecule or a large basis set! Despite the fact that the MDCI module makes no approximation, it runs twice as fast as the *selected* MRCI module and an estimated 50 times faster than the *unselected* MRCI module! This will become even more pronounced for the larger and more accurate basis sets that one should use in such calculations anyways. The error of the selection is on the order of 3 mEh or 2 kcal/mol in the total energy. One can hope that at least part of this error cancels upon taking energy differences.[11] The more rigorous CCSD calculation takes about a factor of two longer than the ACPF calculation which seems reasonable. The triples add another factor of roughly 2 in this example but this will increase for larger calculations since it has a steeper scaling with the system size. The ACPF energy is intermediate between CCSD and CCSD(T) which is typical — ACPF overshoots the effects of disconnected quadruples which partially compensates for the neglect of triples.

These timings will strongly depend on the system that you run the calculation on. Nevertheless, what you should take from this example are the message that if you can use the MDCI module, do it.

The MDCI module can avoid a full integral transformation for larger systems while the MRCI module can use selection and the RI approximation for larger systems. Both types of calculation will become very expensive very quickly! Approximate MDCI calculations are under development.

5.7.1.8 Symmetry

The MRCI program really takes advantage of symmetry adapted orbitals. In this case the MRCI matrix can be blocked according to irreducible representations and be diagonalized irrep by irrep. This is a big computational advantage and allows one to converge on specific excited states much more readily than if symmetry is not taken into account.

The syntax is relatively easy. If you specify:

```
newblock 1 *
        nroots 8
        refs cas(4,4) end
        end
```

Then the '*' indicates that this is to be repeated in each irrep of the point group. Thus, in C_{2v} the program would calculate 8 singlet roots in each of the four irreps of the C_{2v} point group thus leading to a total of 32 states.

Alternatively, you can calculate just a few roots in the desired irreps:

---

[11] Depending on whether one wants to take a pessimistic or an optimistic view one could either say that this result shows what can be achieved with a code that is dedicated to a single determinant reference. Alternatively one could (and perhaps should) complain about the high price one pays for the generality of the MRCI approach. In any case, the name of the game would be to develop MR approaches that are equally efficient than single reference approaches. This can probably not be achieved in an individually selecting fashion but is quite feasible – though complicated – using the concept of internal contraction. Presently only the MOLPRO program features an MRCI implementation that comes close to this goal while still not being applicable to larger molecules. The only MR method (other than the individually selecting approaches) that is applicable to larger molecules is the internally contracted second order perturbation theory (CASPT2) as implemented in the MOLCAS program.

```
newblock 1 0
      nroots 3
      refs cas(4,4) end
      end
    newblock 1 2
      nroots 5
      refs cas(4,4) end
      end
    newblock 3 1
      nroots 1
      refs cas(4,4) end
      end
```

In this example, we would calculate 3 singlet roots in the irrep '0' (which is A_1), then five roots in irrep '2' (which is B_1) and then 1 triplet root in irrep 1 (which is B_2).

Obviously, the results with and without symmetry will differ slightly. This is due to the fact that without symmetry the reference space will contain references that belong to 'wrong' symmetry but will carry with them excited configurations of 'right' symmetry. Hence, the calculation without use of symmetry will have more selected CSFs and hence a slightly lower energy. This appears to be unavoidable. However, the effects should not be very large for well designed reference spaces since the additional CSFs do not belong to the first order interacting space.

5.7.2 A Tutorial Type Example of a MR Calculation

Perhaps, the most important use of the MR-CI module is for the calculation of transition energies and optical spectra. Let us first calculate the first excited singlet and triplet state of the formaldehyde molecule using the MR-CI method together with the Davidson correction to approximately account for the effect of unlinked quadruple substitutions. We delibaratly choose a somewhat small basis set for this calculation which is already reasonable since we only look at a valence excited state and want to demonstrate the principle.

Suppose that we already know from a ground state calculation that the HOMO of H_2CO is an oxygen lone pair orbitals and the LUMO the π^* MO. Thus, we want to calculate the singlet and triplet $n \rightarrow \pi^*$ transitions and nothing else. Consequently, we only need to correlate two electrons in two orbitals suggesting a CAS(2,2) reference space.

```
    # A simple MRCI example
    ! def2-SVP def2-SVP/C UseSym

    %method frozencore fc_ewin
      end

    %mrci  ewin          -3,1000
           CIType           MRCI
           EUnselOpt      FullMP2
           DavidsonOpt Davidson1
           UseIVOs          true
```

```
        tsel            1e-6
        tpre            1e-2
        MaxMemInt        256
        MaxMemVec         32
        IntMode     FullTrafo
        AllSingles       true
        Solver           Diag
        # ground state 1A1
        NewBlock 1 0
          NRoots 1
          Excitations cisd
          Refs CAS(2,2) end
          End
        # HOMO LUMO transition 1A2
        NewBlock 1 1
          NRoots 1
          Excitations cisd
          Refs CAS(2,2) end
          End
        # HOMO LUMO triplet transition 3A2
        NewBlock 3 1
          NRoots 1
          Excitations cisd
          Refs CAS(2,2) end
          end
        end

  * int 0 1
   C    0   0   0   0.000000     0.000     0.000
   O    1   0   0   1.200371     0.000     0.000
   H    1   2   0   1.107372   121.941     0.000
   H    1   2   3   1.107372   121.941   180.000
  *
```

This input – which is much more than what is really required - needs some explanations: First of all, we choose a standard RHF calculation with the SVP basis set and we assign the SV/C fitting basis although it is not used in the SCF procedure at all. In the `%mrci` block all details of the MR-CI procedure are specified. First, EWin (`%method frozencore fc_ewin`) selects the MOs within the given orbital energy range to be included in the correlation treatment. The `CIType` variable selects the type of multireference treatment. Numerous choices are possible and MRCI is just the one selected for this application.

- **_NOTE: The CIType statement selects several default values for other variables. So it is a very good idea to place this statement at the beginning of the MR-CI block and possibly overwrite the program selected defaults <u>later</u>_**. If you place the `CIType` statement after one of the values which it selects by default your input will simply be overwritten!

The variables `EUnselOpt` and `DavidsonOpt` control the corrections to the MR-CI energies. `EUnselOpt` specifies the way in which the MR-CI energies are extrapolated to zero threshold T_{Sel}. Here we choose a full MR-MP2 calculation of the missing contributions to be done *after* the variational step, i.e. using the relaxed part of the reference wavefunction as a 0th order state for MR-PT. The `DavidsonOpt` controls the type of estimate made for the effect of higher substitutions. Again, multiple choices are possible but the most commonly used one (despite some real shortcomings) is certainly the choice `Davidson1`. The flag `UseIVOs` instructs the program to use "improved virtual orbitals". These are virtual orbitals obtained from a diagonalization of the Fock operator from which one electron has been removed in an averaged way from the valence orbitals. Thus, these orbitals "see" only a $N - 1$ electron potential (as required) and are not as diffuse as the standard virtual orbitals from Hartree-Fock calculations. If you input DFT orbitals in the MR-CI moldule (which is perfectly admittable and also recommened in some cases, for example for transition metal complexes) then it is recommended to turn that flag off since the DFT orbitals are already o.k. in this respect. The two thresholds `Tsel` and `Tpre` are already explained above and represent the selection criteria for the first order interacting space and the reference space respectively. `Tsel` is given in Hartree units and refers to the second order MR-MP2 energy contribution from a given excited CSF. 10^{-6} Eh is a pretty good value. Reliable results for transition energies start with $\approx 10^{-5}$; however, the total energy is converging pretty slowly with this parameter and this is one of the greatest drawbacks of individually selecting CI procedures! (see below). `Tpre` is dimensionless and refers to the *weight* of a given initial reference after diagonalization of the the given initial reference space (10^{-4} is a pretty good value and there is little need to go much lower. Aggressive values such as 10^{-2} only select the truly leading configurations for a given target state which can be time saving. Intermediate values are not really recommended). The parameters `MaxMemInt` and `MaxMemVec` tell the program how much memory (in MB) it is allowed to allocate for integrals and for trial and sigma-vectors respectively.

The flag `IntMode` tells the program to perform a full integral transformation. This is possible for small cases with less than, say, 100–200 MOs. In this case that it is possible it speeds up the calculations considerably. For larger molecules you *have to* set this flag to `RITrafo` which means that integrals are recomputed on the fly using the RI approximation which is more expensive but the only way to do the calculation. To switch between the possible modes use:

```
%mrci   IntMode FullTrafo # exact 4 index transformation
                RITrafo   # use auxiliary basis sets
```

For small molecules or if high accuracy in the total energies is required it is much better to use the exact four index transformation. The limitations are that you will run out of disk space or main memory with more than ca. 200–300 MOs.

The variable `Solver` can be `diag` (for Davidson type diagonalization) or `DIIS` for multirrot DIIS type treatments.

```
%mrci   Solver Diag   # Davidson solver
               DIIS   # Multiroot DIIS
```

For CI methods, the diag solver is usually preferable. For methods like ACPF that contain nonlinear terms, DIIS is imperative.

Next in the input comes the definition of what CI matrices are to be constructed and diagonalized. Each multiplicity defines a *block* of the CI matrix which is separately specified. Here we ask for two blocks – singlet and triplet. The general syntax is:

```
NewBlock Multiplicity Irrep
        NRoots 1            # Number of roots to determine
        Excitations cisd   # Type of excitations
        Refs CAS(NEl,NOrb) end  # Reference space def.
        end                # Finalize the block
```

Now that all input is understood let us look at the outcome of this calculation:

The first thing that happens after the SCF calculation is the preparation of the frozen core Fock matrix and the improved virtual orbitals by the program orca_ciprep. From the output the energies of the IVOs can be seen. In this case the LUMO comes down to –8.2 eV which is much more reasonable than the SCF value of +3.... eV. Concomitantly, the shape of this MO will be much more realistic and this important since this orbital is in the reference space!

```
        --------------------------------------------------------------------------
                                  ORCA CI-PREPARATION
        --------------------------------------------------------------------------

        One-Electron Matrix            ... Test-SYM-MRCI-H2CO.H.tmp
        GBW-File                       ... Test-SYM-MRCI-H2CO.gbw
        Improved virtual orbitals      ... Test-SYM-MRCI-H2CO.ivo
        First MO in the CI             ... 2
        Internal Fock matrix           ... Test-SYM-MRCI-H2CO.cif.tmp
        LastInternal Orbital           ... 6
        Integral package used          ... LIBINT
        Reading the GBW file           ... done
        Symmetry usage                 ... ON

        Reading the one-electron matrix    ... done
        Forming inactive density           ... done
        Forming averaged valence density   ...
        Scaling the occupied orbital occupation numbers
        First MO                       ... 2
        Last MO                        ... 7
        Number of electrons in the range   ...    12
        Scaling factor                 ... 0.917

        done
        Forming internal density           ... done
        Forming Fock matrix/matrices       ...
        Nuclear repulsion    ...     31.371502
        Core repulsion       ...     31.371502
        One-electron energy  ...   -114.942082
        Fock-energy          ...    -94.993430
        Final value          ...    -73.596255
        done
        Modifying virtual orbitals         ...
        Last occupied MO                   ...    7
        Total number of MOs                ...   38
        Number of virtual MOs              ...   30
        Doing diagonalization with symmetry
        The improved virtual eigenvalues:
            0: -0.2955 au   -8.041 eV    2- B1
```

```
    1: -0.0701 au   -1.908 eV    6- A1
    2: -0.0176 au   -0.480 eV    3- B2
    3:  0.0064 au    0.175 eV    7- A1
    4:  0.2922 au    7.951 eV    8- A1
    5:  0.2948 au    8.021 eV    3- B1
    6:  0.3836 au   10.439 eV    4- B2
    7:  0.4333 au   11.790 eV    9- A1
    8:  0.4824 au   13.128 eV    5- B2
    9:  0.5027 au   13.680 eV   10- A1
   10:  0.7219 au   19.643 eV   11- A1
   11:  0.8351 au   22.724 eV    4- B1
   12:  0.9372 au   25.501 eV    6- B2
   13:  1.0265 au   27.932 eV    1- A2
   14:  1.1141 au   30.317 eV   12- A1
   15:  1.2869 au   35.017 eV    5- B1
   16:  1.4605 au   39.743 eV    7- B2
...


done
Transforming integrals            ... done
Storing passive energy            ... done (   -73.59625452 Eh)
Transforming internal FI          ... done
            .... done with the Frozen Core Fock matrices
```

The next step is to transform the electron-electron repulsion integrals into the MO basis:

```
-----------------------------
PARTIAL COULOMB TRANSFORMATION
-----------------------------


Dimension of the basis      ...  38
Number of internal MOs      ...  36 (2-37)
Pair cutoff     ... 1.000e-11 Eh
Number of AO pairs included in the trafo   ...  741
Total Number of distinct AO pairs      ...  741
Memory devoted for trafo     ... 256.0 MB
Max. Number of MO pairs treated together   ... 45282
Max. Number of MOs treated per batch      ...  36
Number Format for Storage    ... Double (8 Byte)
Integral package used                ... LIBINT


 --->>> The Coulomb operators (i,j|mue,nue) will be calculated

Starting integral evaluation:
<ss|**>:    9404 b      1 skpd    0.023 s ( 0.002 ms/b)
<sp|**>:   10260 b      0 skpd    0.030 s ( 0.003 ms/b)
<sd|**>:    3420 b      0 skpd    0.016 s ( 0.005 ms/b)
<pp|**>:    3591 b      0 skpd    0.026 s ( 0.007 ms/b)
<pd|**>:    2052 b      0 skpd    0.025 s ( 0.012 ms/b)
<dd|**>:     513 b      0 skpd    0.009 s ( 0.017 ms/b)
Collecting buffer AOJ
    ... done with AO integral generation
Closing buffer AOJ ( 0.00 GB; CompressionRatio= 4.22)
Number of MO pairs included in the trafo   ...  666
    ... Now sorting integrals
```

```
    IBATCH = 1 of  2
    IBATCH = 2 of  2
    Closing buffer JAO ( 0.00 GB; CompressionRatio= 5.20)
    TOTAL TIME for half transformation ...    0.324 sec
    AO-integral generation        ...     0.118 sec
    Half transformation           ...     0.059 sec
    J-integral sorting            ...     0.146 sec
    Collecting buffer JAO


    -------------------
    FULL TRANSFORMATION
    -------------------


    Processing MO  10
    Processing MO  20
    Processing MO  30
    full transformation done
    Number of integrals made      ...       222111
    Number of integrals stored    ...        59070
    Timings:
    Time for first half transformation  ...    0.326 sec
    Time for second half transformation ...    0.160 sec
    Total time                    ...     0.516 sec
```

Total time ... 0.516 sec This will result in a few additional disk files required by orca_mrci. The program then tells you which multiplicities will be treated in this MRCI run:

```
    ------------------
    CI-BLOCK STRUCTURE
    ------------------


    Number of CI-blocks           ... 3


    ===========
    CI BLOCK 1
    ===========
    Multiplicity          ...   1
    Irrep                 ...   0
    Number of reference defs ...   1
      Reference   1: CAS(2,2)


    Excitation type       ... CISD
    Excitation flags for singles:
        1 1 1 1
    Excitation flags for doubles:
        1 1 1 / 1 1 1 / 1 1 1


    ===========
    CI BLOCK 2
    ===========
    Multiplicity          ...   1
    Irrep                 ...   1
    Number of reference defs ...   1
      Reference   1: CAS(2,2)
```

```
Excitation type          ... CISD
Excitation flags for singles:
    1 1 1 1
Excitation flags for doubles:
    1 1 1 / 1 1 1 / 1 1 1


===========
CI BLOCK 3
===========
Multiplicity             ...   3
Irrep                    ...   1
Number of reference defs ...   1
  Reference   1: CAS(2,2)

Excitation type          ... CISD
Excitation flags for singles:
    1 1 1 1
Excitation flags for doubles:
    1 1 1 / 1 1 1 / 1 1 1


          ----------------------------------------------------------------
          ------------------- ALL SETUP TASKS ACCOMPLISHED ------------------
          -------------------    (    1.512 sec)        ------------------
          ----------------------------------------------------------------
```

Now that all the setup tasks have been accomplished the MRCI calculation itself begins.

```
        ###############################################
        #                                             #
        #                 M R C I                     #
        #                                             #
        # TSel   =   1.000e-06 Eh                     #
        # TPre   =   1.000e-02                        #
        # TIntCut =   1.000e-10 Eh                    #
        # Extrapolation to unselected MR-CI by full MP2 #
        # DAVIDSON-1 Correction to full CI            #
        #                                             #
        ###############################################



---------------------
INTEGRAL ORGANIZATION
---------------------

Reading the one-Electron matrix         ... done
Reading the internal Fock matrix        ... assuming it to be equal to the one-electron matrix!!!
done
Preparing the integral list             ... done
Loading the full integral list          ... done
Making the simple integrals             ... done

              ****************************************
              *            CI-BLOCK  1               *
              ****************************************
```

```
    Configurations with insufficient # of SOMOs WILL be rejected
    Building a CAS(2,2) for multiplicity 1 and irrep=A1
    Reference Space:
     Initial Number of Configurations :      2
     Internal Orbitals :    2 -    6
     Active Orbitals   :    7 -    8
     External Orbitals :    9 -   37
    The number of CSFs in the reference is 2
    Calling MRPT_Selection with N(ref)=2
```

In the first step, the reference space is diagonalized. From this CI, the most important configurations are selected with `Tpre`:

```
    ------------------
    REFERENCE SPACE CI
    ------------------

     Pre-diagonalization threshold                 : 1.000e-02
    N(ref-CFG)=2 N(ref-CSF)=2

                        ****Iteration     0****
      Lowest Energy          :   -113.779221544551
      Maximum Energy change  :    113.779221544551 (vector 0)
      Maximum residual norm  :      0.000000000000

        *** CONVERGENCE OF RESIDUAL NORM REACHED ***
    Reference space selection using TPre= 1.00e-02

       ... found 1 reference configurations (1 CSFs)
           ... now redoing the reference space CI ...

    N(ref-CFG)=1 N(ref-CSF)=1

                        ****Iteration     0****
      Lowest Energy          :   -113.778810020485
      Maximum Energy change  :    113.778810020485 (vector 0)
      Maximum residual norm  :      0.000000000000

        *** CONVERGENCE OF RESIDUAL NORM REACHED ***
```

In this case, the CAS space only has 2 correctly symmetry adapted CSFs one of which (the closed shell determinant) is selected. In general, larger CAS spaces usually carry around a lot of unnecessary CSFs which are not needed for anything and then the selection is important to reduce the computational effort. The result of the second reference space CI is printed:

```
    ----------
    CI-RESULTS
    ----------

    The threshold for printing is 0.3 percent
    The weights of configurations will be printed. The weights are summed over
    all CSF's that belong to a given configuration before printing
```

```
      STATE   0:  Energy=   -113.778810020 Eh RefWeight=  1.0000  0.00 eV      0.0 cm**-1
            1.0000  : h---h---[20]
```

Energy is the total energy in Hartree. In the present case we can compare to the SCF energy `-113.778810021 Eh` and find that the reference space CI energy is identical, as it has to be since the lowest state coincides with the reference space. **RefWeight** gives the weight of the reference configurations in a CI state. This is 1.0 in the present case since there were only reference configurations. The number `1.000` is the weight of the following configuration in the CI vector. The description of the configuration `h---h---[20]p---p---` is understood as follows:[12] The occupation of the active orbitals is explicitly given in square brackets. Since the HOMO orbitals is number 7 from the SCF procedure, this refers to MOs 7 and 8 in the present example since we have two active orbitals. The `2` means doubly occupied, the `0` means empty. Any number (instead of `---`) appearing after an `h` gives the index of an internal orbital in which a hole is located. Simarly, any number after a `p` gives the index of an virtual (external) MO where a particle is located. Thus `h---h---[20]` is a closed shell configuration and it coincides with the SCF configuration—this was of course to be expected. The second root (in CI-Block 2) `h---h---[11]` by comparison refers to the configuration in which one electron has been promoted from the HOMO to the LUMO and is therefore the desired state that we wanted to calculate. Things are happy therefore and we can proceed to look at the output.

The next step is the generation of excited configurations and their selection based on `Tsel`:

```
      -----------------------------
      MR-PT SELECTION TSel=1.00e-006
      -----------------------------
      -----------------------------
      MR-PT SELECTION TSel= 1.00e-06
      -----------------------------


      Setting reference configurations WITH use of symmetry
      Building active patterns WITH use of symmetry

      Selection will be done from 1 spatial configurations
      Selection will make use of spatial symmetry
       ( 0) Refs        : Sel:      1CFGs/      1CSFs Gen:       1CFGs/      1CSFs
      Building active space densities           ... done
      Building active space Fock operators      ... done
       ( 1) (p,q)->(r,s): Sel:      1CFGs/      1CSFs Gen:       1CFGs/      1CSFs
       ( 2) (i,-)->(p,-): Sel:      1CFGs/      1CSFs Gen:       1CFGs/      1CSFs
       ( 3) (i,j)->(p,q): Sel:      8CFGs/      8CSFs Gen:       8CFGs/      8CSFs
       ( 4) (i,p)->(q,r): Sel:      0CFGs/      0CSFs Gen:       1CFGs/      1CSFs
       ( 5) (p,-)->(a,-): Sel:      8CFGs/      8CSFs Gen:       8CFGs/      8CSFs
       ( 6) (i,-)->(a,-): Sel:     52CFGs/     52CSFs Gen:      52CFGs/     52CSFs
       ( 7) (i,j)->(p,a): Sel:     95CFGs/    166CSFs Gen:      96CFGs/    167CSFs
       ( 8) (i,p)->(q,a): Sel:     21CFGs/     42CSFs Gen:      22CFGs/     44CSFs
       ( 9) (p,q)->(r,a): Sel:      3CFGs/      3CSFs Gen:       5CFGs/      5CSFs
      (10) (i,p)->(a,b): Sel:    555CFGs/   1082CSFs Gen:     584CFGs/   1139CSFs
      (11) (p,q)->(a,b): Sel:    124CFGs/    124CSFs Gen:     148CFGs/    148CSFs
      (12) (i,j)->(a,b): Sel:   1688CFGs/   2685CSFs Gen:    1887CFGs/   2947CSFs

      Selection results:
      Total number of generated configurations:      2814
```

---

[12]Note that for printing we always sum over all linearly independent spin couplings of a given spatial configuration and only print the summed up weight for the configuration rather than for each individual CSF of the configuration.

```
    Number of selected configurations     :      2557 ( 90.9%)
    Total number of generated CSFs        :      4522
    Number of selected CSFS               :      4173 ( 92.3%)


    The selected tree structure:
    Number of selected Internal Portions      :     11
    Number of selected Singly External Portions:     27
       average number of VMOs/Portion         :    6.39
       percentage of selected singly externals :  22.83
    Number of selected Doubly External Portions:     21
       average number of VMOs/Portion         :  107.59
       percentage of selected doubly externals :  27.76
```

Here, the program loops through classes of excitations. For each excitation it produces the excited configurations (CFGs) and from it the linearly independent spin functions (CSFs) which are possible within the configuration. It then calculates the interaction with the contracted 0th order roots and includes all CSFs belonging to a given CFG in the variational space if the largest second order perturbation energy is larger or equal to **Tsel**. In the present case $\approx 136,000$ CSFs are produced of which 25% are selected. For larger molecules and basis sets it is not uncommon to produce 10^9–10^{10} configurations and then there is no choice but to select a much smaller fraction than 20%. For your enjoyment, the program also prints the total energies of each state after selection:

```
    Diagonal second order perturbation results:
    State       E(tot)          E(0)+E(1)      E2(sel)      E2(unsel)
                 Eh                 Eh            Eh            Eh
    ----------------------------------------------------------------
       0    -114.108347273   -113.778810020   -0.329430    -0.000107
```

You can ignore this output if you want. In cases that the perturbation procedure is divergent (not that uncommon!) the total energies look strange—don't worry—the following variational calculation is still OK. The second order perturbation energy is here divided into a selected part **E2(sel)** and the part procedure by the unselected configurations **E2(unsel)**. Depending on the mode of **EUnselOpt** this value may already be used later as an estimate of the energetic contribution of the unselected CSFs. [13]

Now we have $\approx 4,200$ CSFs in the variational space of CI block 1 and proceed to diagonalize the Hamiltonian over these CSFs using a Davidson or DIIS type procedure:

```
    -----------------------
    DAVIDSON-DIAGONALIZATION
    -----------------------


    Dimension of the eigenvalue problem       ...    4173
    Number of roots to be determined          ...       1
    Maximum size of the expansion space       ...      15
    Convergence tolerance for the residual    ...    1.000e-06
    Convergence tolerance for the energies    ...    1.000e-06
    Orthogonality tolerance                   ...    1.000e-14
    Level Shift                               ...    0.000e+00
```

---

[13] In this case the maximum overlap of the 0th order states with the final CI vectors is computed and the perturbation energy is added to the "most similar root". This is of course a rather crude approximation and a better choice is to recomputed the second order energy of the unselected configurations rigorously as is done with EUnselOpt = FullMP2.

```
   Constructing the preconditioner                ... o.k.
   Building the initial guess                     ... o.k.
   Number of trial vectors determined            ...     2


                        ****Iteration    0****
     Size of expansion space: 2
     Lowest Energy        :  -113.854262408162
     Maximum Energy change :    113.854262408162 (vector 0)
     Maximum residual norm :      1.004640962238

                        ****Iteration    1****
     Size of expansion space: 3
     Lowest Energy        :  -114.076119460817
     Maximum Energy change :      0.221857052655 (vector 0)
     Maximum residual norm :      0.028974632398

                        ****Iteration    2****
     Size of expansion space: 4
     Lowest Energy        :  -114.085249547769
     Maximum Energy change :      0.009130086952 (vector 0)
     Maximum residual norm :      0.001957827970

                        ****Iteration    3****
     Size of expansion space: 5
     Lowest Energy        :  -114.086014164840
     Maximum Energy change :      0.000764617071 (vector 0)
     Maximum residual norm :      0.000167800384

                        ****Iteration    4****
     Size of expansion space: 6
     Lowest Energy        :  -114.086071121272
     Maximum Energy change :      0.000056956432 (vector 0)
     Maximum residual norm :      0.000011388989

                        ****Iteration    5****
     Size of expansion space: 7
     Lowest Energy        :  -114.086076153851
     Maximum Energy change :      0.000005032579 (vector 0)
     Maximum residual norm :      0.000001069291

                        ****Iteration    6****
     Size of expansion space: 8
     Lowest Energy        :  -114.086076506777
     Maximum Energy change :      0.000000352926 (vector 0)

         *** CONVERGENCE OF ENERGIES REACHED ***

   Storing the converged CI vectors              ... Test-SYM-MRCI-H2O.mrci.vec

                     *** DAVIDSON DONE ***
   Returned from DIAG section
```

The procedure converges on all roots simultaneously and finishes after six iterations which is reasonable. Now the program calculates the Davidson correction (**DavidsonOpt**) which is printed for each root.

```
    Davidson type correction:
    Root= 0 W=   0.912 E0=  -113.778810020 ECI=  -114.086076507 DE=-0.026914
```

Already in this small example the correction is pretty large, ca. 27 mEh for the ground state (and $\approx$ 36 mEh for the excited state, later in the output). Thus, a contribution of $\approx$ 9 mEh = 0.25 eV is obtained for the transition energy which is certainly significant. Unfortunately, the correction becomes unreliable as the reference space weight drops or the number of correlated electrons becomes large. Here 0.912 and 0.888 are still OK and the system is small enough to expect good results from the Davidson correction.

The next step is to estimate the correction for the unselected configurations:

```
    Unselected CSF estimate:
    Full relaxed MR-MP2 calculation          ...

    Selection will be done from 1 spatial configurations
    Selection will make use of spatial symmetry
    Selection will make use of spatial symmetry
    Selection will make use of spatial symmetry
    done
    Selected MR-MP2 energies                ...

     Root=  0  E(unsel)=     -0.000106951
```

In the present case this is below 1 mEh and also very similar for all three states such that it is not important for the transition energy.

```
    ----------
    CI-RESULTS
    ----------

    The threshold for printing is 0.3 percent
    The weights of configurations will be printed. The weights are summed over
    all CSF's that belong to a given configuration before printing

    STATE  0: Energy=   -114.113097002 Eh RefWeight=  0.9124  0.00 eV      0.0 cm**-1
        0.9124  : h---h---[20]
        0.0114  : h  6h  6[22]
```

The final ground state energy is `-114.113097002` which is an estimate of the full CI energy in this basis set. The leading configuration is still the closed shell configuration with a weight of $\approx$ 91%. However, a double excitation *outside* the reference space contributes some 1%. This is the excitation MO6,MO6 →LUMO,LUMO. This indicates that more accurate results are expected once MO6 is also included in the reference space (this is the HOMO-1). The excited state is dominated by the HOMO-LUMO transition (as desired) but a few other single- and double- excitations also show up in the final CI vector.

Now that all CI vectors are known we can order the states according to increasing energy and print (vertical) transition energies:

```
-------------------
TRANSITION ENERGIES
-------------------


The lowest energy is    -114.113097002 Eh


State Mult Irrep Root Block    mEh          eV        1/cm
  0    1    A1    0    0      0.000       0.000       0.0
  1    3    A2    0    2    134.073       3.648    29425.7
  2    1    A2    0    1    148.490       4.041    32589.8
```

This result is already pretty good and the transition energies are within ≈ 0.1 eV of their experimental gas phase values (≈ 3.50 and ≈ 4.00 eV) and may be compared to the CIS values of 3.8 and 4.6 eV which are considerably in error.

In the next step the densities and transition densities are evaluated and the absorption and CD spectra are calculated (in the dipole length formalism) for the spin-allowed transitions together with state dipole moments:

```
-----------------------------------------------------------------------------------------------
                                   ABSORPTION SPECTRUM
-----------------------------------------------------------------------------------------------
      States          Energy   Wavelength   fosc        T2        TX        TY        TZ
                      (cm-1)     (nm)                  (D**2)     (D)       (D)       (D)
-----------------------------------------------------------------------------------------------
  0( 0)-> 0( 1) 1    32589.8     306.8    0.000000000  0.00000  -0.00000   0.00000   0.00000


-------------------------------------------------------------------------------
                                   CD SPECTRUM
-------------------------------------------------------------------------------
      States          Energy   Wavelength   R*T        RX        RY        RZ
                      (cm-1)     (nm)     (1e40*sgs)   (au)      (au)      (au)
-------------------------------------------------------------------------------
  0( 0)-> 0( 1) 1    32589.8     306.8     0.00000   -0.00000  -0.00000   1.18711


-------------------------------------------------------------------------------
                              STATE DIPOLE MOMENTS
-------------------------------------------------------------------------------
  Root  Block        TX        TY        TZ        |T|
                   (Debye)   (Debye)   (Debye)   (Debye)
-------------------------------------------------------------------------------
    0    0        0.00000   -0.00000   2.33244    2.33244
    0    2        0.00000   -0.00000   1.45831    1.45831
    0    1        0.00000   -0.00000   1.58658    1.58658
```

Here the transition is symmetry forbidden and therefore has no oscillator strength. The state dipole moment for the ground state is 2.33 Debye which is somewhat lower than 2.87 Debye from the SCF calculation. Thus, the effect of correlation is to reduce the polarity consistent with the interpretation that the ionicity of the bonds which is always overestimated by HF theory is reduced by the correlation. Finally, you also get a detailed population analysis for each generated state density which may be compared to the corresponding SCF analysis in the preceding part of the output.

This concludes the initial example on the use of the MR-CI module. The module leaves several files on disk most of which are not yet needed but in the future will allow more analysis and restart and the like. The `.ivo` file is a standard `.gbw` type file and the orbitals therein can be used for visualization. This is important in order to figure

out the identity of the generated IVOs. Perhaps they are not the ones you wanted and then you need to re-run the MR-CI with the IVOs as input, `NoIter` and the IVO feature in the new run turned off! We could use the IVOs as input for a state averaged CASSCF calculation:

```
    ! moread UseSym KDIIS
    %moinp "Test-SYM-MRCI-H2CO.ivo"

    %casscf nel    2
            norb   2
            irrep  0,1,1
            mult   1,1,3
            nroots 1,1,1
            end
```

If we based a MR-ACPF calculation on this reference space we will find that the calculated transition energies are slightly poorer than in the MRCI+Q calculation. This is typical of approximate cluster methods that usually require somewhat larger reference spaces for accurate results. A similar result is obtained with SORCI.

```
    %mrci  CIType          SORCI
           tsel            1e-6
           tpre            1e-4
           tnat            1e-5
           AllSingles      true
           doNatOrbs       true
           IntMode     FullTrafo
           # ground state 1A1
           NewBlock 1 0
             NRoots 1
             Excitations cisd
             Refs CAS(2,2) end
             End
           # HOMO LUMO transition 1A2
           NewBlock 1 1
             NRoots 1
             Excitations cisd
             Refs CAS(2,2) end
             End
           # HOMO LUMO triplet transition 3A2
           NewBlock 3 1
             NRoots 1
             Excitations cisd
             Refs CAS(2,2) end
             end
```

This gives:

```
    State Mult Irrep Root Block     mEh          eV       1/cm
       0    1    A1    0    0      0.000      0.000       0.0
       1    3    A2    0    2    146.507      3.987   32154.5
       2    1    A2    0    1    161.801      4.403   35511.3
```

This is systematically 0.4 eV too high. But let us look at the approximate average natural orbital (AANOs) occupation numbers:

```
    -----------------------
    AVERAGE NATURAL ORBITALS
    -----------------------


    Trace of the density to be diagonalized = 12.000000
    Sum of eigenvalues = 12.000000
    Natural Orbital Occupation Numbers:
    N[  2] ( A1)=   1.99832583
    N[  3] ( A1)=   1.99760289
    N[  4] ( A1)=   1.99481021
    N[  5] ( B2)=   1.99044471
    N[  6] ( B1)=   1.95799339
    N[  7] ( B2)=   1.33003795
    N[  8] ( B1)=   0.70704982
    N[  9] ( B2)=   0.00988857
    N[ 10] ( A1)=   0.00448885
```

This shows that there is a low-occupancy orbital (MO6) that has not been part of the reference space. Thus, we try the same calculation again but now with one more active orbital and two more active electrons:

```
    ! moread
    %moinp "Test-SYM-MRCI-H2CO.gbw"

    %casscf nel 4
            norb 3
            irrep  0,1,1
            mult   1,1,3
            nroots 1,1,1
            end

    %mrci CIType          SORCI
          tsel            1e-6
          tpre            1e-4
          tnat            1e-5
          AllSingles      true
          doNatOrbs       true
          IntMode     FullTrafo
          # ground state 1A1
          NewBlock 1 0
            NRoots 1
```

```
        Excitations cisd
        Refs CAS(4,3) end
        End
      # HOMO LUMO transition 1A2
      NewBlock 1 1
        NRoots 1
        Excitations cisd
        Refs CAS(4,3) end
        End
      # HOMO LUMO triplet transition 3A2
      NewBlock 3 1
        NRoots 1
        Excitations cisd
        Refs CAS(4,3) end
        end
```

This gives:

```
    State Mult Irrep Root Block    mEh        eV      1/cm
      0    1    A1    0    0      0.000     0.000      0.0
      1    3    A2    0    2    137.652     3.746   30211.1
      2    1    A2    0    1    153.128     4.167   33607.7
```

Which is now fine since all essential physics has been in the reference space. Inspection of the occupation numbers show that there is no suspicious orbital any more. Note that this is still a much more compact calculation that the MRCI+Q.

Likewise, we get an accurate result from MRACPF with the extended reference space.

```
    State Mult Irrep Root Block    mEh        eV      1/cm
      0    1    A1    0    0      0.000     0.000      0.0
      1    3    A2    0    2    134.985     3.673   29625.8
      2    1    A2    0    1    148.330     4.036   32554.6
```

However, the SORCI calculation is *much* more compact. For larger molecules the difference becomes more and more pronounced and SORCI or even MRDDCI2 (with or without +Q) maybe the only feasible methods—if at all.

5.7.3 Excitation Energies between Different Multiplicities

As an example for a relatively accurate MRCI+Q calculation consider the following job which calculates the triplet-ground and as the first excited singlet states of O_2.

```
    ! ano-pVQZ RI-AO cc-pVQZ/JK VeryTightSCF NoPop Conv UseSym RI-MP2 PModel
    %mp2 density relaxed natorbs true end
    * xyz 0 3
    O  0 0 0
    O  0 0 1.2
    *


    $new_job
    ! ano-pVQZ RI-AO cc-pVQZ/JK VeryTightSCF NoPop Conv UseSym KDIIS
    ! moread
    %moinp "O2.mp2nat"
    %casscf nel      8
            norb     6
            irrep    1,0,1
            nroots   1,2,1
            mult     3,1,1
            trafostep    ri
            switchstep   nr
            end

    %mrci   citype mrci
            tsel 1e-7
            tpre 1e-5
            newblock 3 1 nroots 1 refs cas(8,6) end end
            newblock 1 0 nroots 2 refs cas(8,6) end end
            newblock 1 1 nroots 1 refs cas(8,6) end end
            end

    * xyz 0 3
    O  0 0 0
    O  0 0 1.2
    *
```

Note that the linear molecule is run in D_{2h}. This creates a slight problem as the CASSCF procedure necessarily breaks the symmetry of the $^1\Delta$ state.

```
    LOWEST ROOT (ROOT 1, MULT 3, IRREP B1g) =   -149.765310181 Eh -4075.321 eV

    STATE   ROOT MULT IRREP DE/a.u.    DE/eV    DE/cm**-1
       1:    0    1   Ag  0.033397    0.909    7329.8
       2:    0    1   B1g  0.033397    0.909    7329.8
       3:    1    1   Ag  0.062208    1.693    13653.2
```

The result of the MRCI+Q is:

```
             -------------------
             TRANSITION ENERGIES
             -------------------


     The lowest energy is    -150.176924817 Eh

     State Mult Irrep Root Block    mEh         eV       1/cm
       0    3  B1g    0    0      0.000      0.000       0.0
       1    1  B1g    0    2     36.995      1.007     8119.6
       2    1  Ag     0    1     38.046      1.035     8350.2
       3    1  Ag     1    1     62.793      1.709    13781.5
```

These excitation energies are accurate to within a few hundred wavenumbers. Note that the ≈ 200 wavenumber splitting in the degenerate ^{1}Delta state is due to the symmetry breaking of the CAS and the individual selection. Repeating the calculation with the MP2 natural orbitals gives an almost indistinguishable result and a ground state energy that is even lower than what was found with the CASSCF orbitals. Thus, such natural orbitals (that might often be easier to get) are a good substitute for CASSCF orbitals and at the same time the symmetry breaking due to the use of symmetry appears to be difficult to avoid.

```
             -------------------
             TRANSITION ENERGIES
             -------------------


     The lowest energy is    -150.177743426 Eh

     State Mult Irrep Root Block    mEh         eV       1/cm
       0    3  B1g    0    0      0.000      0.000       0.0
       1    1  B1g    0    2     37.369      1.017     8201.5
       2    1  Ag     0    1     38.237      1.040     8392.1
       3    1  Ag     1    1     62.731      1.707    13767.9
```

5.7.4 Correlation Energies

The logic we are following here is the following: CID minus SCF gives the effect of the doubles; going to CISD gives the effect of the singles; QCISD(=CCD) minus CID gives the effect of the disconnected quadruples. QCISD minus QCID gives simultaneously the effect of the singles and the disconnected triples. They are a bit difficult to separate but if one looks at the singles alone and compares with singles + disconnected triples, a fair estimate is probably obtained. Finally, QCISD(T) minus QCISD gives the effect of the connected triples. One could of course also use CCSD instead of QCISD but I felt that the higher powers of T_1 obscure the picture a little bit—but this is open to discussion of course.

First H_2O/TZVPP at its MP2/TZVPP equilibrium geometry ($T_{pre} = 10^{-6}$ and $T_{sel} = 10^{-9}$ Eh for the MRCI and MRACPF calculations):

Excitation class	Energy (Eh)	Delta-Energy (mEh)
None (RHF)	-76.0624	
Doubles (CID)	-76.3174	255
+Singles (CISD)	-76.3186	1
+Disconnected Quadruples (QCID)	-76.3282	11
+Disconnected Triples (QCISD)	-76.3298	2

+Connected Triples (QCISD(T))	**-76.3372**	7
CASSCF(8,6)	-76.1160	
CASSCF(8,6) + MRCI	-76.3264	210
CASSCF(8,6) + MRCI+Q	**-76.3359**	10
CASSCF(8,6) + MRACPF	**-76.3341**	218

One observes quite good agreement between single- and multireference approaches. In particular, the contribution of the disconnected triples and singles is very small. The estimate for the disconnected quadruples is fairly good from either the multireference Davidson correction or the ACPF and the agreement between CCSD(T) and these MR methods is 2-3 mEh in the total energy which is roughly within chemical accuracy.

In order to also have an open-shell molecule let us look at NH/TZVPP at its at 1.0 Ångström:

Excitation class	Energy (Eh)	Delta-Energy (mEh)
None (UHF)	-54.9835	
Doubles (CID)	-55.1333	150
+Singles (CISD)	-55.1344	1
+Disconnected Quadruples (QCID)	-55.1366	3
+Disconnected Triples (QCISD)	-55.1378	1
+Connected Triples (QCISD(T))	**-55.1414**	4
CASSCF(6,5)	-55.0004	
CASSCF(6,5) + MRCI	-55.1373	137
CASSCF(6,5) + MRCI+Q	**-55.1429**	6
CASSCF(6,5) + MRACPF	**-55.1413**	141

Again, the agreement is fairly good and show that both single- and multiple reference approaches converge to the same limit.

5.7.5 Thresholds

Now we choose the CO molecule (1.128 Ångström) with the SVP basis set and study the convergence of the results with respect to the selection threshold. Comparison to high level single-reference approaches is feasible (The SCF energy is `-112.645 946 Eh`).

5.7.5.1 Reference Values for Total Energies

The single-reference values are:

```
        BD: -112.938 48002
      CCSD: -112.939 79145
     QCISD: -112.941 95700
     BD(T): -112.950 17278
   CCSD(T): -112.950 63889
  QCISD(T): -112.951 37425
 MP4(SDTQ): -112.954 80113
```

The calculations without connected triples (BD, CCSD, QCISD) are about the best what can be achieved without explicitly considering triple excitations. The CCSD is probably the best in this class. As soon as connected triples are included the CCSD(T), QCISD(T) and BD(T) values are close and from experience they are also close to the full CI values which is then expected somewhere between –112.950 and –112.952 Eh.

5.7.5.2 Convergence of Single Reference Approaches with Respect to $\mathbf{T}_{sel}$

Next it is studied how these single reference methods converge with T_{sel}:

```
Closed-Shell ACPF:
   Tsel        Energy       (NCSF)    Energy        (NCSF)
   (Eh)      AllSingles=true      AllSingles=false
  TSel=0      -112.943 387 (5671)
  TSel=1e-14  -112.943 387 (2543)  -112.943 387 (2478)
  TSel=1e-10  -112.943 387 (2543)  -112.941 023 (2453)
  TSel=1e-08  -112.943 387 (2451)  -112.937 087 (2346)
  TSel=1e-06  -112.943 350 (2283)  -112.937 046 (2178)
  TSel=1e-05  -112.943 176 (1660)  -112.936 821 (1555)
  TSel=1e-04  -112.944 039 ( 782)  -112.938 381 ( 677)
```

It is clear that the convergence is erratic if the singles are not automatically included. This is the reason for making this the default from release 2.6.35 on. In the present case singles will only be selected due to round-off errors since by Brillouin's theorem the singles have zero-interaction with the ground state determinant. Thus, for individually selecting single-reference methods it is a good idea to automatically include all single-excitations in order to get converged results. The alternative would be a different singles selection procedure which has not yet been developed however. The selection of doubles appear to converge the total energies reasonably well. It is seen that the selection selects most CSFs between 10^{-5} and 10^{-7} Eh. Already a threshold of 10^{-6} Eh yields an error of less than 0.1 mEh which is negligible in relation to reaction energies and the like. Even 10^{-5} Eh gives an error of less than 0.1 kcal/mol.

5.7.5.3 Convergence of Multireference Approaches with Respect to $\mathbf{T}_{pre}$

We next turn to multireference treatments. Here we want to correlate all valence electrons in all valence orbitals and therefore a CAS(10,8) is the appropriate choice. We first ask for the converged value of T_{pre} by using $\mathrm{T}_{sel} = 10^{-14}$ and obtain for MRCI+Q:

```
TPre = 1e-1: -112.943 964
       1e-2: -112.952 963
       1e-3: -112.953 786
       1e-4: -112.954 019
       1e-5: -112.954 336
       1e-6: -112.954 416
       1e-7: -112.954 440
```

Thus, pretty good convergence is obtained for $\mathrm{T}_{pre} = 10^{-4} - 10^{-6}$. Hence 10^{-4} is the default.

To show a convenient input consider the following:

```
#
# Here we calculate the CO ground state correlation energy with several methods
#
! RHF aug-SVP aug-SV/C RI-MP2 CCSD(T)
%base "1"

%mp2   density relaxed
       donatorbs   true
       end

* int 0 1
 C     0    0    0    0.000000     0.000      0.000
 O     1    0    0    1.128        0.000      0.000
*


$new_job

! RHF aug-SVP MRACPF
! moread
%moinp "1.mp2nat"
# the CASSCF is done with MP2 natural orbitals which is a good idea and
# secondly we use a large level shift in order to help convergence
%casscf nel      10
         norb     8
         mult     1
         nroots   1
         shiftup  2
         shiftdn  2
         end

%mrci   tsel 1e-8
        tpre 1e-6
        end

* int 0 1
 C     0    0    0    0.000  0.000  0.000
 O     1    0    0    1.128  0.000  0.000
*
```

This job computes at the same time all of the below and demonstrates once more the agreement between consequent single- and multireference correlation methods

```
SCF          = -112.6523
RI-MP2       = -112.9591
```

```
    CCSD        = -112.9604
    CCSD(T)     = -112.9730
    CASSCF(10,8) = -112.7829
    MRACPF      = -112.9722
```

5.7.6 Energy Differences - Bond Breaking

For the calculation of energy differences we start again with the reference CCSD(T) calculation; this method is one of the few which can claim chemical accuracy in practical applications:

```
    Reference Total Energies for N2 at 1.0977 Angström with
    The SVP basis
      E(CCSD)   = -109.163 497
      E(CCSD(T))= -109.175 625
    Nitrogen Atom (4S), SVP basis, unrestricted
      E(CCSD)   =  -54.421 004
      E(CCSD(T))=  -54.421 7183
    Energy Difference:
    Delta-E(CCSD)   = -0.321 489 = 8.75 eV
    Delta-E(CCSD(T))= -0.332 188 = 9.04 eV
```

The basis set is of course not suitable for quantitative comparison to experimental values. However, this is not the point here in these calculations which are illustrative in nature. The SVP basis is just good enough to allow for a method assessment without leading to excessively expensive calculations.

This is now to be compared with the corresponding energy differences computed with some single-reference approaches. A typical input is (this is a somewhat old-fashioned example – in the present program version you would do a full valence CASSCF(10,8) or CASSCF(6,6) and invoke the MR-methods with a single keyword):

```
    ! RHF SVP TZVPP/C VeryTightSCF NoPop

    %base "1"

    * xyz 0 1
    N 0 0 0
    N 0 0 1.0977
    *
    %method frozencore fc_ewin
            end

    %mrci   EWin         -3,1000
            CIType       MRACPF2a
            Solver           DIIS
            IntMode      FullTrafo
```

```
         UseIVOs          true
         AllSingles       true
         TSel            1e-14
         TPre            1e-05
         TNat             0.0
         ETol            1e-10
         RTol            1e-10
         NewBlock 1 *
            NRoots 1
            Excitations CISD
            refs CAS(0,0) end
            end
         end

 $new_job


 ! ROHF SVP TZVPP/C VeryTightSCF NoPop PModel

 %base "2"

 * xyz 0 4
   N 0 0 0
 *
 %method frozencore fc_ewin
         end

 %mrci  EWin          -3,1000
        CIType        MRACPF2a
        IntMode       FullTrafo
        UseIVOs          true
        AllSingles       true
        TSel            1e-14
        TPre            1e-05
        TNat             0.0
        ETol            1e-10
        RTol            1e-10
        NewBlock 4 *
           NRoots 1
           Excitations CISD
           refs CAS(3,3) end
           end
        end
```

The results are:

```
     Single reference approaches:
      Method   N2-Molecule       N-Atom         Delta-E
      CISD+Q : -109.167 904     -54.422 769      8.77 eV
      ACPF   : -109.166 926     -54.421 783      8.80 eV
      ACPF2  : -109.166 751     -54.421 333      8.82 eV
      ACPF2a : -109.166 730     -54.421 186      8.83 eV
      CEPA1  : -109.159 721     -54.422 564      8.56 eV
      CEPA2  : -109.172 888     -54.422 732      8.91 eV
      CEPA3  : -109.161 034     -54.422 589      8.59 eV
      AQCC   : -109.160 574     -54.420 948      8.67 eV
      CEPA-0 : -109.174 924     -54.422 951      8.95 eV
```

With exception is CEPA1 and CEPA3, the results are OK. The reason for the poor performance of these methods is simply that the formalism implemented is only correct for closed shells – open shells require a different formalism which we do not have available in the MRCI module (but in the single reference MDCI module). Due to the simple approximations made in CEPA2 it should also be valid for open shells and the numerical results are in support of that.

Next we turn to the multireference methods and take a CAS(10,8) reference as for CO in order to correlate all valence electrons. [14]

```
     Multi reference approaches:
      Method      N2-Molecule        N-Atom          Delta-E
      MRCISD+Q: -109.180 089       -54.422 667       9.11 eV
      MRACPF  : -109.178 708       -54.421 685       9.12 eV
      MRACPF2 : -109.177 140       -54.421 236       9.11 eV
      MRAQCC  : -109.175 947       -54.420 851       9.10 eV
      SORCI   : -109.179 101       -54.422 703       9.08 eV
```

This test calculation pleasingly shows the high consistency of multireference approaches which all converge more or less to the same result which must be accurate.

5.7.7 Energy Differences - Spin Flipping

There are a number if interesting situations in which one is interested in a small energy difference which arises from two states of different multiplicity but same orbital configuration. This is the phenomenon met in diradicals or in magnetic coupling in transition metal complexes. As a primitive model for such cases one may consider the hypothetical molecule H-Ne-H in a linear configuration which will be used as a model in this section.

The reference value is obtained by a MR-ACPF calculation with all valence electrons active (again, this example is somewhat old fashioned – in the present program version you would do a CASSCF calculation followed by MR methods wit ha single keyword):

---

[14]Most of these results have been obtained with a slightly earlier version for which the MR energies are a little different from that what the present version gives. The energy differences will not be affected.

```
    ! ROHF SVP TZVPP/C VeryTightSCF NoPop
    %basis aux auto
            newauxgto h "TZVPP/C" end
            end
    * xyz 0 3
    H  0 0 0
    Ne 0 0 2.0
    H  0 0 4.0
    *
    %method frozencore fc_ewin
            end

    %mrci  EWin        -3,1000
            CIType     MRACPF2a
            IntMode    FullTrafo
            Solver        DIIS
            UseIVOs       true
            TSel            0
            TPre         1e-10
            ETol         1e-09
            RTol         1e-09
            DoDDCIMP2     true
            NewBlock 1 *
               NRoots 1
               Excitations CISD
               refs CAS(10,6) end
               end
            NewBlock 3 *
               NRoots 1
               Excitations CISD
               refs CAS(10,6) end
               end
            end
```

which gives the reference value 108 cm^{-1}. We now compare that to several other methods which only have the two "magnetic" orbitals (the 1s's on the hydrogens) in the active space:

```
    ... same as above
    %mrci  EWin        -10,1000
            CIType     MRDDCI3
            ... same as previously
            NewBlock 1 *
               NRoots 1
               refs CAS(2,2) end
```

```
              end
          NewBlock 3 *
              NRoots 1
              refs CAS(2,2) end
              end
          end
```

This gives the result:

```
    Method        S-T gap
    MR-CI+Q  :  98 cm-1
    MR-CI    :  93 cm-1
    MR-ACPF  :  98 cm-1
    MR-ACPF2 :  98 cm-1
    MR-ACPF2a:  97 cm-1
    MR-AQCC  :  95 cm-1
    SORCI    : 131 cm-1
    MR-DDCI2 :  85 cm-1
    MR-DDCI3 : 130 cm-1
```

All these methods give good results with SORCI leading to a somewhat larger error than the others. The (difference dedicated CI) DDCI2 method slightly underestimates the coupling which is characteristic of this method. It is nice in a way that DDCI3 gives the same result as SORCI since SORCI is supposed to approximate the DDCI3 (or better the IDDCI3) result which it obviously does.

This splitting can also be studied using broken symmetry HF and DFT methods as explained elsewhere in this manual:

```
    Method      S-T gap
     UHF     : 70 cm-1
     B3LYP/G : 240 cm-1
     BP86    : 354 cm-1
     PW91    : 234 cm-1
     PBE     : 234 cm-1
     PBE0    : 162 cm-1
     RPBE    : 242 cm-1
```

This confirms the usual notions; UHF underestimates the coupling and DFT overestimates it, less so for hybrid functionals than for GGA's. The BP86 is worse than PW91 or PBE. The PBE0 hybrid may be the best of the DFT methods. For some reason most of the DFT methods give the best results if the BS state is simply taken as an approximation for the true open shell singlet. This is, in our opinion, not backed up by theory but has been observed by other authors too.

Now let us study the dependence on T_{sel} as this is supposed to be critical (we use the DDCI3 method):

```
    Tsel      S-T gap
    1e-04      121
    1e-05      128
    1e-06      132
    1e-07      131
    1e-08      131
    1e-10      131
    1e-12      131
    0          131
```

The convergence is excellent once AllSingles are included.

5.7.8 Potential Energy Surfaces

Another situation where multireference approaches are necessary is when bond breaking is studied and one wants to calculate a full potential energy surface. Say we want to compute the potential energy surface of the CH molecule. First we have to figure out which states to include. Hence, let us first determine a significant number of roots for the full valence CASSCF reference state (we use a small basis set in order to make the job fast).

```
    ! bn-ANO-dzp VeryTightSCF NoPop Conv

%casscf nel       5  norb     5
        nroots    2  mult     2
        shiftup   2  shiftdn  2
        end

%mrci   CIType       MRCI
        NewBlock 2 *
          excitations none
          NRoots 15
          refs CAS(5,5) end
          end
        NewBlock 4 *
          excitations none
          NRoots 15
          refs CAS(5,5) end
          end
        end

* xyz 0 2
C  0 0 0
H  0 0 1.15
*
```

This yields:

```
-------------------
TRANSITION ENERGIES
-------------------

The lowest energy is    -38.308119994 Eh

State Mult Irrep Root Block    mEh          eV       1/cm
  0    2   -1    0    0      0.000       0.000       0.0
  1    2   -1    1    0      0.000       0.000       0.0
  2    4   -1    0    1     14.679       0.399    3221.6
  3    2   -1    2    0    126.464       3.441   27755.7
  4    2   -1    3    0    126.464       3.441   27755.7
  5    2   -1    4    0    132.689       3.611   29121.8
  6    2   -1    5    0    164.261       4.470   36051.2
  7    2   -1    6    0    305.087       8.302   66958.9
  8    2   -1    7    0    305.087       8.302   66958.9
  9    4   -1    1    1    328.911       8.950   72187.7
 10    4   -1    2    1    452.676      12.318   99350.8
 11    4   -1    3    1    452.676      12.318   99350.8
 12    2   -1    8    0    460.116      12.520  100983.9
 13    2   -1    9    0    463.438      12.611  101712.9
 14    2   -1   10    0    463.438      12.611  101712.9
...
```

Thus, if we want to focus on the low-lying states we should include five doublet and one quartet root. Now we run a second job with these roots and scan the internuclear distance.

```
! ano-pVDZ VeryTightSCF NoPop Conv MRCI+Q

%casscf nel    5
        norb   5
        nroots 5,1
        mult   2,4
        shiftup 2 shiftdn 2
        end


%paras R = 0.8,2.5,25
        end


* xyz 0 2
C  0 0 0
H  0 0 {R}
*
```

The surfaces obtianed in this run are shown in 5.14. You can nicely see the crossing of the $^2\Sigma$ and $^2\Delta$ states fairly close to the equilibrium distance and also the merging of the $^4\Sigma$ state with $^2\Pi$ and $^2\Sigma$ towards the asymptote that where C-H dossciates in a neutral C-atom in its ^3P ground state and a neutral hydrogen atom in its ^2S ground state. You can observe that once AllSingles is set to true (the default), the default settings of the MRCI module yield fairly smooth potential energy surfaces.
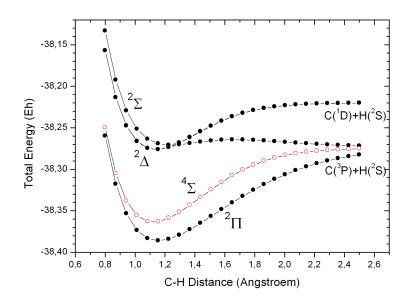
Figure 5.14: Potential energy surfaces for some low-lying states of CH using the MRCI+Q method

In many cases one will focus on the region around the minimum where the surface is nearly quadratic. In this case one can still perform a few (2,3,5,...) point polynomial fitting from which the important parameters can be determined. The numerical accuracy and the behavior with respect to T_{sel} has to be studied in these cases since the selection produces some noise in the procedure. We illustrate this with a calculation on the HF molecule:

```
! ano-pVDZ VeryTightSCF NoPop Conv MRCI+Q

%paras  R = 0.85,1.1,7
        end

%casscf nel     8
        norb    5
        nroots  1  mult    1
        shiftup 2.5 shiftdn 2.5 switchstep nr gtol    1e-5
        end

%mrci   tsel 1e-8
        tpre 1e-5
        end

* xyz 0 1
F  0 0 0
H  0 0 {R}
*
```

The output contains the result of a Morse fit:

```
    Morse-Fit Results:
     Re              =    0.94760 Angstroem
     we              =    3941.1 cm**-1
     wexe            =      97.2 cm**-1
```

Which may be compared with the CCSD(T) values calculated with the same basis set:

```
    Morse-Fit Results:
     Re              =    0.95070 Angstroem       =
     we              =    3958.3 cm**-1
     wexe            =      90.4 cm**-1
```

The agreement between MRCI+Q and CCSD(T) results is fairly good.

5.7.9 Multireference Systems - Ozone

The ozone molecule is a rather classical multireference system due to its diradical character. Let us look at the three highest occupied and lowest unoccupied MO (the next occupied MO is some 6 eV lower in energy and the next virtual MO some 10 eV higher in energy):

(a) MO-9   (b) MO-10   (c) MO 11(HOMO)   (d) MO 12(LUMO)
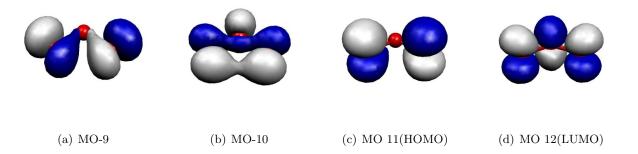
Figure 5.15: Frontier MOs of the Ozone Molecule.

These MOs are two σ lone pairs which are high in energy and then the symmetric and antisymmetric cominations of the oxygen π lone pairs. In particular, the LUMO is low lying and will lead to strong correlation effects since the $(HOMO)^2 \rightarrow (LUMO)^2$ excitation will show up with a large coefficient. Physically speaking this is testimony of the large diradical character of this molecule which is roughly represented by the structure $\uparrow$O-O-O$\downarrow$. Thus, the minimal active space to treat this molecule correctly is a CAS(2,2) space which includes the HOMO and the LUMO. We illustrate the calculation by looking at the RHF, MP2 MRACPF calculations of the two-dimensional potential energy surface along the O–O bond distance and the O-O-O angle (experimental values are 1.2717 Å and 116.78°).

```
    ! ano-pVDZ VeryTightSCF NoPop MRCI+Q Conv

    %paras R     = 1.20,1.40,21
           Theta = 100,150,21
           end

    %casscf nel    2
            norb   2
            mult   1
            nroots 1
            end

    %mrci   tsel   1e-8
            tpre   1e-5
            end

    * int 0 1
    O  0 0 0  0    0        0
    O  1 0 0 {R}   0        0
    O  1 2 0 {R}  {Theta}   0
    *
```

This is a slightly lengthy calculation due to the 441 energy evaluations required. RHF does not find any meaningful minimum within the range of examined geometries. MP2 is much better and comes close to the desired minimum but underestimates the O–O distance by some 0.03 Å. CCSD(T) gives a very good angle but a O–O distance that is too long. In fact, the largest doubles amplitude is ≈0.2 in these calculations (the HOMO–LUMO double excitation) which indicates a near degeneracy calculation that even CCSD(T) has problems to deal with. Already the CAS(2,2) calculation is in qualitative agreement with experiment and the MRCI+Q calculation then gives almost perfect agreement.

The difference between the CCSD(T) and MRCI+Q surfaces shows that the CCSD(T) is a bit lower than the MRCI+Q one suggesting that it treats more correlation. However, it does it in an unbalanced way and the important triples and quadruples that are brought in in the MRCI calculation through excitation one one- or two electrons from the HOMO–LUMO double excitation play an important role in balancing the MR calculation. These excitations are treated to all orders explicitly in the MRCI calculation but only approximately (quadruples as simultaneous pair excitations and triples perturbatively) in the coupled-cluster approach. Thus, despite the considerable robustness of CC theory in electronically difficult situations it is not applicable to genuine multireference problems.

This is a nice result despite the too small basis set used and shows how important it can be to go to a multireference treatment with a physically reasonable active space (even if is only 2×2) in order to get qualitatively and quantitatively correct results.

5.7.10 Size Consistency

Finally, we want to study the size consistency errors of the methods. For this we study two non-interacting HF molecules at the single reference level and compare to the energy of a single HF molecule. This should give a reasonably fair idea of the typical performance of each method (Energies in Eh):

(a) RHF

(b) CASSCF(2,2)

(c) MP2

(d) CCSD(T)

(e) MRCI+Q

(f) Difference CCSD(T)/MRCI+Q
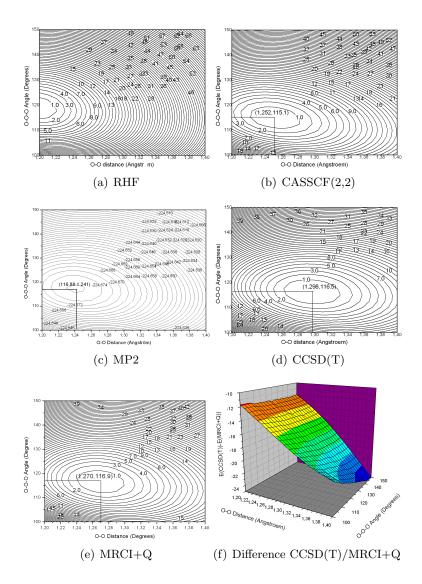
Figure 5.16: 2D potential energy surface for the O_3 molecule calculated with different methods.

	E(HF)	E(HF+HF)	|Difference|
CISD+Q	−100.138 475	−200.273 599	0.00335
ACPF	−100.137 050	−200.274 010	0.00000
ACPF2	−100.136 913	−200.273 823	0.00000
AQCC	−100.135 059	−200.269 792	0.00032

(most of these numbers were obtained with a slightly older version but will not change too much in the present version)

The results are roughly as expected – CISD+Q has a relatively large error, ACPF and ACPF/2 are perfect for this type of example; AQCC is not expected to be size consistent and is (only) about a factor of 10 better than CISD+Q in this respect. CEPA-0 is also size consistent.

5.7.11 Efficient MR-MP2 Calculations for Larger Molecules

Since version 2.7.0 ORCA avoids the full integral transformation for MR-MP2 which leads to significant savings in terms of time and memory. Thus, relatively large RI-MR-MP2 calculations can be done with fairly high efficiency. However, the program still uses an uncontracted first order wavfunction which means that for very large reference space, the calculations still become untractable.

Consider for example the rotation of the stilbene molecule around the central double bond
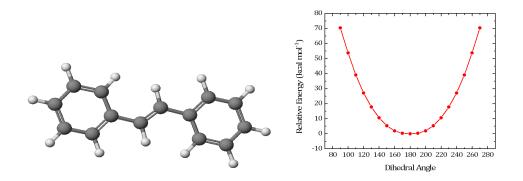
Figure 5.17: Rotation of stilbene around the central double bond using a CASSCF(2,2) reference and correlating the reference with MRMP2.

The input for this calculation is shown below. The calculation has more than 500 basis functions and still runs through in less than one hour per step (CASSCF-MRMP2). The program takes care of the reduced number of two-electron integrals relative to the parent MRCI method and hence can be applied to larger molecules as well. Note that we have taken a 'JK' fitting basis in order to fit the Coulomb and the dynamic correlation contributions both with sufficient accuracy. Thus, this example demonstrates that MRMP2 calculations for not too large reference spaces can be done efficiently with ORCA (as a minor detail note that the calculations were started at a dihedral angle of 90 degrees in order to make sure that the correct two orbitals are in the active space, namly the central carbon p-orbitals that would make up the pi-bond in the coplanar structure).

```
#
# Stilbene rotation using MRMP2
#
! def2-TZVP def2-TZVP/JK RIJCOSX RI-MRMP2

%casscf nel          2
        norb         2
        trafostep   rimo
        switchstep  diis
        shiftup      0.5
        shiftdn      0.5
        end


%mrci   maxmemint 2000
        tsel 1e-8
        end



%paras  DIHED = 90,270, 19
```

```
        end

  * int 0 1
  C     0   0   0   0.000000     0.000      0.000
  C     1   0   0   1.343827     0.000      0.000
  C     2   1   0   1.490606   125.126      0.000
  C     1   2   3   1.489535   125.829    {DIHED}
  C     4   1   2   1.400473   118.696    180.000
  C     4   1   2   1.400488   122.999      0.000
  C     6   4   1   1.395945   120.752    180.000
  C     5   4   1   1.394580   121.061    180.000
  C     8   5   4   1.392286   120.004      0.000
  C     3   2   1   1.400587   118.959    180.000
  C     3   2   1   1.401106   122.779      0.000
  C    11   3   2   1.395422   120.840    180.001
  C    12  11   3   1.392546   120.181      0.000
  C    13  12  11   1.392464   119.663      0.000
  H     1   2   3   1.099419   118.266      0.000
  H     2   1   3   1.100264   118.477    179.999
  H     5   4   1   1.102119   119.965      0.000
  H     6   4   1   1.100393   121.065      0.000
  H     7   6   4   1.102835   119.956    180.000
  H     8   5   4   1.102774   119.989    180.000
  H     9   8   5   1.102847   120.145    180.000
  H    10   3   2   1.102271   120.003      0.000
  H    11   3   2   1.100185   121.130      0.000
  H    12  11   3   1.103001   119.889    180.000
  H    13  12  11   1.102704   120.113    180.000
  H    14  13  12   1.102746   119.941    180.000
  *
```

5.8 Solvation

In a joint effort with COSMOlogic, the conductor like screening model (COSMO) was implemented throughout ORCA. Thus, the following calculations can be used to carry out calculations in a polarizable continuum using a realistic Van-der-Waals cavity

- Energies of molecules in solution with a finite dielectric constant ε using any HF or DFT method.

- Optimization of molecular structures in solution using any HF or DFT method using analytic gradients.

- Calculation of solvent effects on response properties like polarizabilities through coupled-perturbed SCF theory. For magnetic response properties such as the g-tensor the COSMO response vanishes.

- Calculations of solvent shifts on transition energies using the time-dependent DFT or CIS method. Here one needs to know the refractive index of the solvent in addition to the dielectric constant.

- First order perturbation estimate of solvent effects on state and transition energies in multireference perturbation and configuration-interaction calculations.

As a simple example let us look at the solvent effect on the transition energy of the $n \to \pi^*$ transition in formaldehyde. We first do a normal CIS calculation:

```
   ! RHF TZVP

   %cis   nroots 1
          end

   * int 0 1
    C    0   0   0   0.000000      0.000      0.000
    O    1   0   0   1.200371      0.000      0.000
    H    1   2   0   1.107372    121.941      0.000
    H    1   2   3   1.107372    121.941    180.000
    *
```

yielding a transition energy of 4.582 eV. Now we repeat the same calculation but with the COSMO model enabled (which is fairly simple; nothing except the dielectric constant and the refractive index needs to be input; there are technical parameters which can be defined by the user but this is not necessary in most applications; all modules will automatically recognize the presence of the COSMO terms automatically).

```
   ! RHF TZVP

   %cosmo epsilon 80
          refrac 1.33
          end

   %cis   nroots 1
          end

   * int 0 1
    C    0   0   0   0.000000      0.000      0.000
    O    1   0   0   1.200371      0.000      0.000
    H    1   2   0   1.107372    121.941      0.000
    H    1   2   3   1.107372    121.941    180.000
    *
```

This calculation yields:

```
   ------------------------
   CALCULATED SOLVENT SHIFTS
   ------------------------

    State Shift(Eh) Shift(eV) Shift(cm**-1) Shift(nm) ECI(eV)  ECI+SHIFT(eV)
   ------------------------------------------------------------------------
      0: -0.0026760 -0.073      -587.3         3.7      4.948      4.875
```

Note that there are two different types of shift: the first is the difference between the CIS calculation in the gas phase and the one using the solvated orbitals. This is the difference $4.984 - 4.582$ eV = 0.402 eV. This term is called the 'slow' term and represents the interaction of the difference density with the frozen screening charges of the ground

state. The second term depends on the instantaneous polarization and is called the 'fast' term. It is given by the extra -0.073 eV printed above thus yielding a total solvent shift of $0.402 - 0.073$ eV $= 0.329$ eV and a final estimate of the transition energy of 4.875 eV in solution (this is clearly not an accurate value since it is too high as is always found with CIS; the calculated shift is also somewhat higher compared to accurate MR-CI calculations which give 0.25 eV but it is in the right ballpark; larger basis sets yield also better results here).

5.9 Calculation of Properties

5.9.1 Population Analysis and Related Things
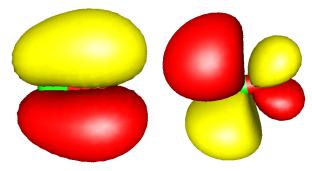
Atomic populations and the like are not really a molecular property since they are not observable. They are nevertheless highly useful for chemical interpretation purposes. ORCA lets you obtain very detailed information about the calculated molecular orbitals. Mulliken, Löwdin and Mayer population analysis can be performed and many useful details can be printed. However – it is also easy to get lost in the output from such a calculation since you may not be interested in all these details. In this case ORCA lets you turn most features off. The default is to perform a rather large amount of population analysis.
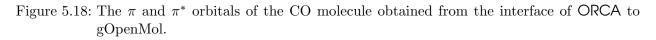
```
! HF SVP Mulliken Loewdin Mayer PrintMOs
* xyz 0 1
  C  0  0  0
  O  0  0  1.13
*
```

The Mulliken, Löwdin and Mayer analysis tools should be self-explanatory. If you choose 'PrintMOs' you will *not* get a printout of the raw MO expansion coefficients but instead a reduced orbital population where the percentage contributions per basis function type on each atom are listed. This is highly useful in figuring out the character of the MOs. You can, however, also request a printout of the MO coefficients themselves via the output block (section 6.22).

In many cases it is not so interesting to look at the MO coefficients but you want to get a full three dimensional picture of MO's, electron densities and spin densities. This is relatively easily accomplished with ORCA through the interface to the gOpenMol and Molekel packages (see section 6.23 for details).

The following example:

```
# test populations
! HF SVP XYZFile
%plots Format gOpenMol_bin
        MO("CO-4.plt",4,0);
        MO("CO-8.plt",8,0);
        end
* xyz 0 1
  C  0  0  0
  O  0  0  1.13
*
```

produces (after running it through gOpenMol, section 6.23.2) the following output:

Figure 5.18: The π and π^* orbitals of the CO molecule obtained from the interface of ORCA to gOpenMol.

which are the textbook like π and π^* orbitals of CO respectively. The format gOpenMol_bin is the most easy to use. The alternative format gOpenMol_ascii would require you to use the gOpenMol conversion utitlity. You can also plot spin densities, electron densities and natural orbitals. See section 6.23.2 for full details. The command `MO("CO-4.plt",4,0);` is to be understood as follows: there is an MO to be evaluated on a grid and the output is stored in the file CI-4.plt. It is MO four of operator 0. Operator zero is the closed shell RHF operator. For UHF wavefunctions operator 0 is that for spin-up and operator 1 that for spin-down. For ROHF you should also use operator 0. There are also some alternative output formats including simple ascii files that you can convert yourself to the desired format.

In order to use the interface to `Molekel` (see http://www.cscs.ch/molekel/) you have to choose the format `Cube` or `Gaussian_Cube` which can be read directly by molekel. Since the cube-files are ASCII files you can also transfer them between platforms.

```
# test populations
! HF SVP XYZFile
%plots Format Cube
        MO("CO-4.cube",4,0);
        MO("CO-8.cube",8,0);
        end
* xyz 0 1
  C  0  0  0
  O  0  0  1.13
*
```

You can now start `Molekel` and load (via a right mouse click) the XYZ file (or also directly the .cube file). Then go to the surface menu, select "gaussian-cube" format and load the surface. For orbitals click the "both signs" button and select a countour value in the "cutoff" field. The click "create surface". The colour schemes etc. can be adjusted at will – try it! It's easy and produces nice pictures. Create files via the "snapshot" feature of `Molekel`. Other programs can certainly also deal with Gaussian-Cube files. If you know about another nice freeware program – please let me know![15]

---

[15] The `Molekel` developers ask for the following citation – please do as they ask:

MOLEKEL 4.2,P.Flukiger,H.P.Luthi,S.Portmann,J.Weber,Swiss Center for Scientific Computing,Manno (Switzerland),2000-2002.

Stefan Portmann &Hans Peter Luthi.MOLEKEL:An Interactiv Molecular Graphics Tool. CHIMIA (2000), 54, 766-770.
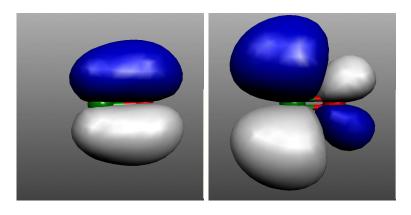
Figure 5.19: The π and π^*-MOs of CO as visualized by Molekel.

Another thing that may in some situations be quite helpful is the visualization of the electronic structure in terms of localized molecular orbitals. As unitary transformations among the occupied orbitals do not change the total wavefunction such transformations can be applied to the canonical SCF orbitals with no change of the physical content of the SCF wavefunction. The localized orbitals correspond more closely to the pictures of orbitals that chemists often enjoy to think about. Localized orbitals according to the Pipek-Mezey (population-localization) scheme are quite easy to compute. For example, the following run reproduces the calculations reported by Pipek and Mezey in their original paper for the N_2O_4 molecule. In the output you will find that the localized set of MO's consists of 6 core like orbitals (one for each N and one for each O), two distinct lone pairs on each oxygen, a σ- and a π-bonding orbital for each N-O bond and one N-N σ-bonding orbital which corresponds in a nice way to the dominant resonance structure that one would draw for this molecule. You will also find a file with the extension .loc in the directory where you run the calculation. This is a standard GBW file that you can use for plotting or as input for another calculation (warning! The localized orbitals have no well defined orbital energy. If you do use them as input for another calculation use GuessMode=CMatrix in the [SCF] block).

```
#----------------------------------------
# Localized MO's for the N2O4 molecule
#----------------------------------------
! HF STO-3G Bohrs

%loc
     LocMet  PipekMezey # localization method. Choices:
                        # PipekMezey (=PM)
                        # FosterBoys (=FB)
     T_Core  -1000 # cutoff for core orbitals
     Tol     1e-8  # conv. Tolerance (default=1e-6)
     MaxIter 20    # max. no of iterations (def. 128)
     end

* xyz 0 1
  N  0.000000  -1.653532 0.000000
  N  0.000000   1.653532 0.000000
  O -2.050381  -2.530377 0.000000
  O  2.050381  -2.530377 0.000000
  O -2.050381   2.530377 0.000000
  O  2.050381   2.530377 0.000000
*
```

If you have access to a version of the **gennbo** program from Weinhold's group[16] you can also request natural population analysis and natural bond orbital analysis. The interface is very elementary and is invoked through the keywords `NPA` and `NBO` respectively

```
# ---------------------------------------
# Test the interface to the gennbo program
# ------------------------------------------
! HF SVP NPA XYZFile
* xyz 0 1
  C  0  0  0
  O  0  0  1.13
*
```

If you choose simple `NPA` then you will only obtain a natural population analysis. Choosing instead `NBO` the natural bond orbital analysis will also be carried out. The program will leave a file `jobname.47` on disk. This is a valid input file for the gennbo program which can be edited to use all of the features of the **gennbo** program in the stand-alone mode. Please refer to the NBO manual for further details.

5.9.2 Absorption and Fluorescence Bandshapes

Bandshape calculations are nontrivial but can be achieved with ORCA using the procedures described in section 6.18. Starting from version 2.80, analytical TD-DFT gradients are available which make these calculations quite fast and applicable without expert knowledge to larger molecules.

In a nutshell, let us look into the H_2CO molecule. First we generate some Hessian (e.g. BP86/SV(P)). Then we run the job that makes the input for the orca_asa program. For example, let us calculate the five lowest excited states:

```
#
! aug-cc-pVDZ BHandHLYP TightSCF NMGrad

%tddft nroots    5
        end

# this is ASA specific input
%rr    states 1,2,3,4,5
        HessName  "Test-ASA-H2CO-freq.hess"
        ASAInput  true
        end

* int 0 1
C 0 0 0  0   0     0
O 1 0 0  1.2 0     0
H 1 2 0  1.1 120   0
H 1 2 3  1.1 120 180
*
```

NOTE:

- Functionals with somewhat more HF exchange produce beter results and are not as prone to 'ghost states' as GGA functional unfortunately are!

---

[16] Information about the NBO program can be found at http://www.chem.wisc.edu/~nbo5

- Calculations can be greatly speeded up by the RI or RIJCOSX approximations!

- Analytic gradients for the (D) correction and hence for double hybrid functionals are NOT available

The ORCA run will produce a file `Test-ASA-H2CO.asa.inp` that is an input file for the program that generates the various spectra. It is an ASCII file that is very similar in appearance to an ORCA input file:

```
#
# ASA input
#
%sim     model IMDHO
         method Heller

         AbsRange     25000.0,  100000.0
         NAbsPoints   1024

         FlRange      25000.0,  100000.0
         NFlPoints    1024

         RRPRange     5000.0,  100000.0
         NRRPPoints   1024

         RRSRange         0.0,    4000.0
         NRRSPoints   4000

         # Excitation energies (cm**-1) for which rR spectra will
         # be calculated. Here we choose all allowed transitions
         # and the position of the 0-0 band
         RRSE 58960, 66884, 66602

         # full width half maximum of Raman bands in rR spectra
         # (cm**-1):
         RRS_FWHM 10.0

         AbsScaleMode Ext
         FlScaleMode  Rel
         # RamanOrder=1 means only fundamentals. For 2 combination
         # bands and first overtones are also considered, for 3
         # one has second overtones etc.
         RamanOrder   1

         # E0 means the adiabatic excitation energy
         # EV would mean the vertical one. sprints vertical
         # excitations in the TD-DFT output but for the input into
         # the ASA program the adiabatic excitation energies are
         # estimated. A rigorous calculation would of course in-
         # volve excited state geometry optimization
         EnInput E0

         CAR   0.800
         end

   # These are the calculated electronic states and transition moments
   # Note that this is in the Franck-Condon approximation and thus
   # the transition moments have been calculated vertically
   $el_states
```

```
5
    1 32200.79   100.00    0.00    -0.0000    0.0000   -0.0000
    2 58960.05   100.00    0.00     0.0000   -0.4219    0.0000
    3 66884.30   100.00    0.00    -0.0000    0.4405    0.0000
    4 66602.64   100.00    0.00    -0.5217   -0.0000    0.0000
    5 72245.42   100.00    0.00     0.0000    0.0000    0.0000

# These are the calculated vibrational frequencies for the totally
# symmetric modes. These are the only ones tha contribute. They
# correspond to x, H-C-H bending, C=O stretching and C-H stretching
# respectively
$vib_freq_gs
3
    1      1462.948534
    2      1759.538581
    3      2812.815170

# These are the calculated dimensional displacements for all
# electronic states along all of the totally symmetric modes.
$sdnc
3 5
                   1          2          3          4          5
       1      -0.326244   0.241082  -0.132239   0.559635   0.292190
       2      -1.356209   0.529823   0.438703   0.416161   0.602301
       3      -0.183845   0.418242   0.267520   0.278880   0.231340
```

Before the orca_asa program can be invoked this file must be edited. We turn the `NAbsPoints` variables and spectral ranges to the desired values and then invoke **orca_asa**:

```
orca_asa Test-ASA-H2CO.asa.inp
```
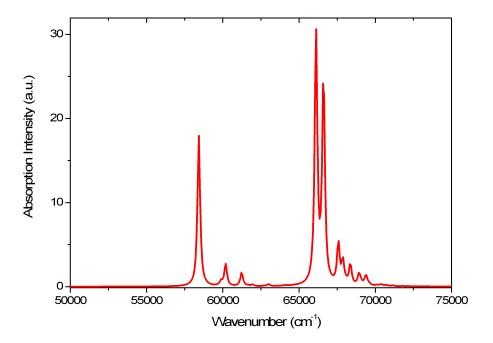
This produces the output:

```
                         ******************
                         * O R C A   A S A *
                         ******************


               --- A program for analysis of electronic spectra ---


     Reading file: Test-ASA-H2CO.asa.inp ... done


          ****************************************************************
          *          GENERAL CHARACTERISTICS OF ELECTRONIC SPECTRA          *
          ****************************************************************


     ----------------------------------------------------------------------
        State      E0          EV        fosc    Stokes shift  Effective Stokes shift
             (cm**-1)    (cm**-1)              (cm**-1)         (cm**-1)
     ----------------------------------------------------------------------
          1:     30457.24   32200.79   0.000000     0.00             0.00
          2:     58424.56   58960.05   0.031879     0.00             0.00
```

```
   3:     66601.54   66884.30   0.039422       0.00              0.00
   4:     66111.80   66602.64   0.055063       0.00              0.00
   5:     71788.55   72245.42   0.000000       0.00              0.00
```
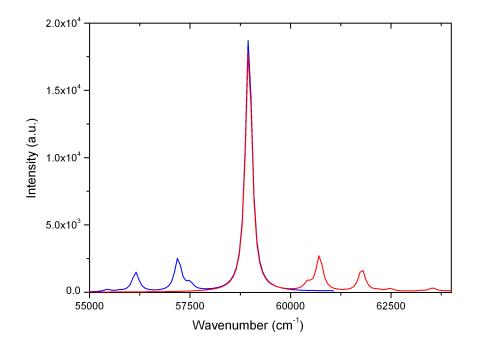
```
----------------------------------------------------------------------------------
                        BROADENING PARAMETETRS (cm**-1)
----------------------------------------------------------------------------------

            Intrinsic                          Effective
   State  -------------------------   -----------------------------------------------

                                               Sigma                    FWHM
                                      -------------------------   -------------------------
            Gamma    Sigma    FWHM
                                        0K     77K    298.15K      0K     77K    298.15K
        ----------------------------------------------------------------------------------
     1:   100.00    0.00    200.00     0.00    0.00    0.00     200.00  200.00  200.00
     2:   100.00    0.00    200.00     0.00    0.00    0.00     200.00  200.00  200.00
     3:   100.00    0.00    200.00     0.00    0.00    0.00     200.00  200.00  200.00
     4:   100.00    0.00    200.00     0.00    0.00    0.00     200.00  200.00  200.00
     5:   100.00    0.00    200.00     0.00    0.00    0.00     200.00  200.00  200.00
```

```
Calculating absorption spectrum   ...
The maximum number of grid points ... 5840
Time for absorption            ...      9.569 sec (=   0.159 min)
Writing file: Test-ASA-H2CO.asa.abs.dat ... done
Writing file: Test-ASA-H2CO.asa.abs.as.dat ... done

Generating vibrational states up to the  1-th(st) order  ...   done
Total number of vibrational states                       ...      3

Calculating rR profiles for all vibrational states up to the 1-th order
State 1 ...
The maximum number of grid points ... 6820
Resonance Raman profile is done
State 2 ...
The maximum number of grid points ... 6820
Resonance Raman profile is done
State 3 ...
The maximum number of grid points ... 6820
Resonance Raman profile is done
Writing file: Test-ASA-H2CO.asa.o1.dat... done
Writing file: Test-ASA-H2CO.asa.o1.info... done

Calculating rR spectra involving vibrational states up to the 1-th(st) order
State    1 ... done
State    2 ... done
State    3 ... done

Writing file: Test-ASA-H2CO.asa.o1.rrs.58960.dat      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.58960.stk      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.66884.dat      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.66884.stk      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.66602.dat      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.66602.stk      ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.as.58960.dat   ... done
Writing file: Test-ASA-H2CO.asa.o1.rrs.as.58960.stk   ... done
```

```
     Writing file: Test-ASA-H2CO.asa.o1.rrs.as.66884.dat    ... done
     Writing file: Test-ASA-H2CO.asa.o1.rrs.as.66884.stk    ... done
     Writing file: Test-ASA-H2CO.asa.o1.rrs.as.66602.dat    ... done
     Writing file: Test-ASA-H2CO.asa.o1.rrs.as.66602.stk    ... done
     Writing file: Test-ASA-H2CO.asa.o1.rrs.all.xyz.dat     ... done

     TOTAL RUN TIME: 0 days 0 hours 1 minutes 17 seconds 850 msec
```
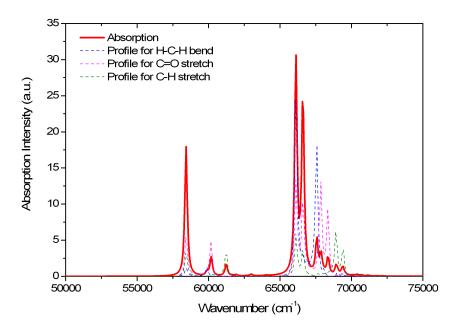
The vibrationally resolved absorption spectrum looks like:
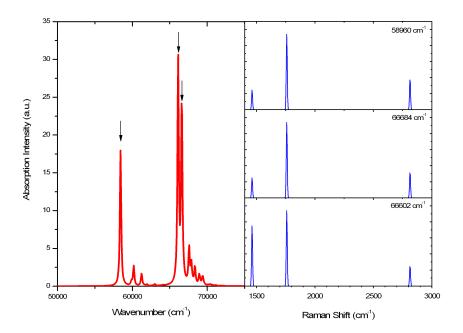
The fluorescence spectrum of the lowest energy peak (in this case S2 which is not very realistic but for illustrative purposes it might be enough):

The Resonance Raman excitation profiles of the three totally symmetric vibrational modes can be obtained as well:

The dominant enhancement occurs under the main peaks for the C=O stretching vibration which might noth be a big surprise. Higher energy excitations do enhance the C-H vibrations particularly strongly. The resonance Raman spectra taken at the vertical excitation energies are also calculated:

In this particular example, the dominant mode is the C=O stretching and the spectra look similar for all excitation wavelength. However, in 'real life' where one has electronically excited state of different nature the rR spectra also dramatically change and are then powerful fingerprints of the electronic excitation being studied – even if the vibrational structure of the absorption band is not resolved (which is usually the case for larger molecules).

This is a cursory example of how to use the orca_asa program. It is much more powerful than described in this section. Please refer to section 6.18 for a full description of features. The `orca_asa` program can also be interfaced to other electronic structure codes that deliver excited state gradients and can be used to fit experimental data. It is thus a tool for experimentalists and theoreticians at the same time!

5.9.3 IR/Raman Spectra, Vibrational Modes and Isotope Shifts

5.9.3.1 IR Spectra

IR spectral intensities are calculated automatically in frequency runs. Thus, there is nothing to control by the user. Consider the following job[17]:

```
! RHF STO-3G TightSCF SmallPrint
! Opt NumFreq
* xyz 0 1
  C  0.000000    0.000000   -0.533905
  O  0.000000    0.000000    0.682807
  H  0.000000    0.926563   -1.129511
```

[17] This and the following jobs in this section are of very poor quality due to the insufficient quality STO-3G basis set (see section 7.1). However, these jobs execute very rapidly and are therefore adequate for demonstration purposes.

```
    H  0.000000    -0.926563    -1.129511
    *
```

which gives you the following output:

```
    -----------
    IR SPECTRUM
    -----------

    Mode     freq (cm**-1)   T**2        TX          TY          TZ
    -------------------------------------------------------------------
      6:       1278.77      6.157280  ( -2.481387   -0.000010   -0.000287)
      7:       1395.78     29.682711  ( -0.000003   -5.448182   -0.004815)
      8:       1765.08      4.180497  (  0.000537   -0.022003    2.044508)
      9:       2100.53      8.550050  (  0.000080    0.011990    2.924022)
     10:       3498.72      1.843422  (  0.001027   -0.178764   -1.345907)
     11:       3646.23     19.778426  (  0.000035    4.446922   -0.057577)
```

The "Mode" indicates the number of the vibration, then the vibrational frequency follows. The value T**2 is the square of the change of the dipole moment along a given vibrational mode in KM/mole. This number is directly proportional to the intensity of a given fundamental in an IR spectrum and is what is plotted by **orca mapspc**.

If you want to obtain a plot of the spectrum then call the small utility program like this:

```
    orca_mapspc Test-NumFreq-H2CO.out ir -w50
```

The options to the program **orca mapspc** are:

```
    -wvalue : a value for the linewidth (gaussian shape, fwhm)
    -x0value : start value of the spectrum in cm**-1
    -x1value : end value of the spectrum in cm**-1
    -nvalue : number of points to use
```

You get a file `Test-NumFreq-H2CO.out.ir.dat` which contains a listing of intensity versus wavenumber which can be used in any graphics program for plotting. For example:
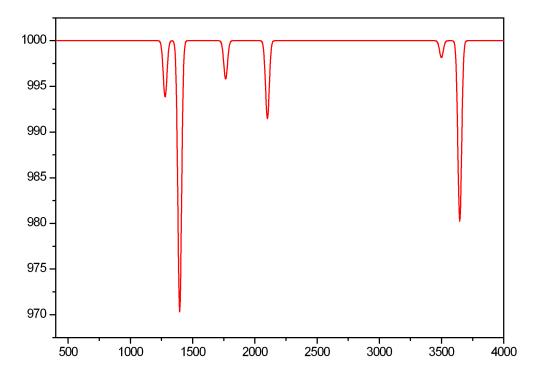
Figure 5.20: The predicted IR spectrum of the H$_2$CO molecule using the numerical frequency routine of ORCA and the tool orca_mapspc to create the spectrum.

5.9.3.2 Raman Spectra

In order to predict the Raman spectrum of a compound one has to know the derivatives of the polarizability with respect to the normal modes. Thus, if a frequency run is combined with a polarizability calculation the Raman spectrum will be automatically calculated too.

Consider the following example:

```
! RHF STO-3G TightSCF SmallPrint
! Opt NumFreq
#
# ... turning on the polarizability calculation
#     together with NumFreq automatically gives
#     the Raman spectrum
#
%elprop Polar 1
        end
* xyz 0 1
  c  0.000000    0.000000   -0.533905
  o  0.000000    0.000000    0.682807
  h  0.000000    0.926563   -1.129511
  h  0.000000   -0.926563   -1.129511
*
```

The output consists of the Raman activity (in A^4/AMU) and the Raman depolarization ratios:

```
 --------------
RAMAN SPECTRUM
 --------------


  Mode    freq (cm**-1)   Activity   Depolarization
 ---------------------------------------------------
    6:       1278.77      0.007349      0.749649
    7:       1395.78      3.020010      0.749997
    8:       1765.08     16.366586      0.708084
    9:       2100.53      6.696490      0.075444
   10:       3498.72     38.650431      0.186962
   11:       3646.23     24.528483      0.748312
```

The polarizability derivatives and the Raman intensities will also be added to the `.hess` file. This allows the effect of isotope substitutions on the Raman intensities to be calculated.

As with IR spectra you can get a plot of the Raman spectrum using:

```
orca_mapspc Test-NumFreq-H2CO.out raman -w50
```
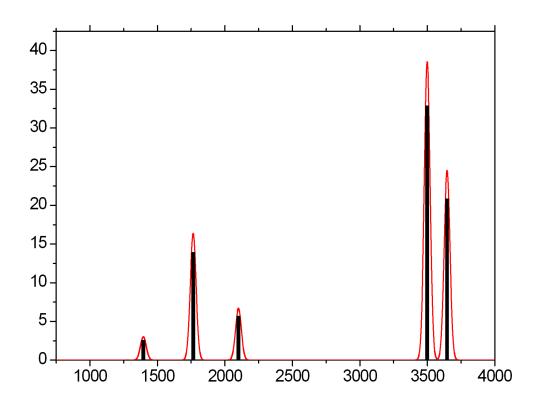

Figure 5.21: Calculated Raman spectrum for H_2CO at the STO-3G level using the numerical frequency routine of ORCA and the tool orca_mapspc to create the spectrum.

NOTE:

- The Raman module will only work if the polarizabilities are calculated analytically. This means that only those wavefunction models for which the analytical derivatives w.r.t. to external fields are available can be used (presently Hartree-Fock and DFT but NOT MP2 or CASSCF).

- Raman calculations take significantly longer than IR calculations due to the extra effort of calculating the polarizabilities at all displaced geometries. Since the latter step is computationally as expensive as the solution of the SCF equations you have to accept an increase in computer time by a factor of ≈ 2.

5.9.3.3 Resonance Raman Spectra

Resonance Raman spectra and excitation profiles can be predicted or fitted using the procedures described in section 6.18.

5.9.3.4 NRVS Spectra

If you happen to have iron in your molecule and you want to calculate the nuclear resonant vibrational scattering spectrum you simply have to run `orca_vib` program on your `.hess` file and you will get an output that can be used together with `orca_mapspc` program for vizualisation.

```
orca_vib MyJob.hess >MyJob.vib.out
orca_mapspc MyJob.vib.out NRVS
```

The calculations are usually pretty good in conjunction with DFT frequency calculations. For example, take the ferric-azide complex from the second reference listed below. As for the calculation of resonance Raman spectra described in section 6.18 the DFT values are usually excellent starting points for least-square refinements.

The reference for the theory and implementation are

- Petrenko T.; Sturhahn, W.; Neese, F. (**2008**) First principles calculation of Nuclear Resonance Vibrational Spectra, *Hyperfine interactions*, <u>175</u>, 165-174

- DeBeer-George, S.; Petrenko, T.; Aliaga-Alcade, N.; Bill, E.; Mienert, B.; Sturhan, W.; Ming, Y.; Wieghardt, K.; Neese, F. (**2007**) Characterization of a Genuine Iron(V)Nitrido Species by Nuclear Resonant Vibrational Spectroscopy Coupled to Density Functional Calculations, *J. Am. Chem. Soc.*, <u>129</u>, 11053-11060.

Here we illustrate the procedure for getting such plots using a $Fe(SH)_4^{1-}$ model complex as an example. One first optimized and computes the vibrations of the complex given in one step with the following input (calculations are usually pretty good in conjunction with DFT frequency calculations and the BP86 functional).

```
! UKS BP86  def2-TZVP def2-TZVP/J  TightSCF SmallPrint
! Opt NumFreq
*xyz -1  6
Fe    -0.115452     0.019090    -0.059506
S     -0.115452     1.781846     1.465006
S     -0.115452    -1.743665     1.462801
S     -1.908178    -0.072782    -1.518702
S      1.560523     0.154286    -1.656664
H      0.410700     2.760449     0.687716
H     -0.674147    -2.708278     0.690223
```
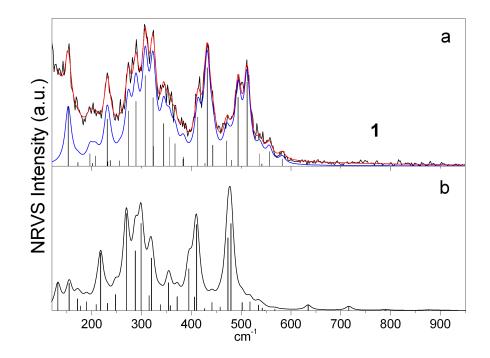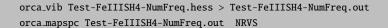
Figure 5.22: Experimental (a, black curve), fitted (a, red) and simulated (b) NRVS spectrum of the Fe(III)-azide complex obtained at the BP86/TZVP level (T = 20 K). Bar graphs represent the corresponding intensities of the individual vibrational transitions. The blue curve represents the fitted spectrum with a background line removed.

```
    H       -2.905212       0.345589      -0.699907
    H        2.647892      -0.211681      -0.932926
    *
```

From this calculations we get numerous files from which the Hessian file is of importance here. Now we run the `orca_vib` program on the `.hess` file and get an output that can be used further with `orca_mapspc` program to prepare raw data for visualizations:

```
    orca_vib Test-FeIIISH4-NumFreq.hess > Test-FeIIISH4-NumFreq.out
    orca_mapspc Test-FeIIISH4-NumFreq.out   NRVS
```

The latter command creates a file `Test-FeIIISH4-NumFreq.nrvs.dat` which can be used directly for visualization. The text-file contains data in xy-format which allows the NRVS intensity (y, in atomic units) to be plotted as a function of the phonon enrgy (x, in cm^{-1}).

From the given run we obtain the NRVS plot below in which we compare with the theoretical IR spectrum on the same scale. NRVS reports the Doppler broadening of the Moessbauer signal due to resonant scattering of phonons (vibrations) dominated by the Fe nuclei movements. This are a valuable addition to IR spectra where the corresponding vibrations might have very small intensity.
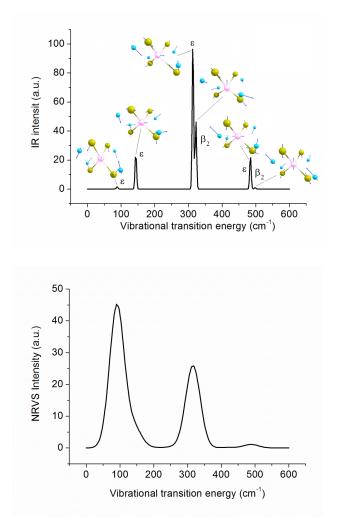
Figure 5.23: Theoretical IR spectrum with the shapes of vibrations dominating the IR intensity and NRVS scattering

5.9.3.5 Animation of Vibrational Modes

In order to animate vibrational modes and to create 'arrow-pictures' you have to use the small utility program orca_pltvib. This program uses an ORCA output file and creates a series of files that can be used together with any visualization program (here: ChemCraft).

For example:

```
# NAME $=$ Test-NumFreq-H2CO.inp
! RHF STO-3G TightSCF SmallPrint
! Opt NumFreq
*xyz 0 1
C 0.000000 0.000000 -0.533905
O 0.000000 0.000000 0.682807
H 0.000000 0.926563 -1.129511
```

```
    H 0.000000 -0.926563 -1.129511
    *
```

From this we get vibrations and transition probabilities (Test-NumFreq-H2CO.out)

```
    Mode freq (cm**-1) T**2 TX TY TZ
    ------------------------------------------------------------------
    6: 1284.36 6.411244 ( -2.532043 -0.000000 -0.000000)
    7: 1397.40 29.590615 ( 0.000000 -5.439726 0.000000)
    8: 1766.60 4.188394 ( -0.000000 -0.000000 2.046557)
    9: 2099.20 8.429963 ( -0.000000 -0.000000 2.903440)
    10: 3499.11 1.792263 ( -0.000000 -0.000000 -1.338754)
    11: 3645.24 19.984096 ( 0.000000 -4.470357 0.000000)
```
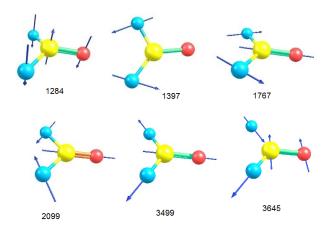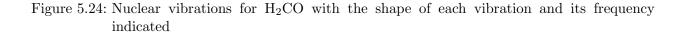
The Test-NumFreq-H2CO.out file can be directly opened with ChemCraft which allows immediate observation of any vibrations and preparing plots as shown.

Figure 5.24: Nuclear vibrations for H_2CO with the shape of each vibration and its frequency indicated

We can infer for this example, that say the vibration 1397 cm^{-1} is a kind of wagging motion of the hydrogen atoms.

It might be that you can prefer to animate vibrations with the (free) program gOpenMol package; there is a small utility program orca_pltvib. This program uses an ORCA output file and creates a series of files that can be used together with gOpenMol. You can execute orca_pltvib in the following way:

Use:

```
    orca_pltvib Test-NumFreq-H2CO.out [list of vibrations or all]
```

For example, let us see what the strong mode at 1397 cm^{-1} corresponds to:

```
orca_pltvib Test-NumFreq-H2CO.out 7
```

you will get a file `Test-NumFreq-H2CO.out.v007.xyz`. Then start up the gOpenMol program and read this file as a `Import->coords` in Xmol format. After this go to the `Trajectory->Main` menu and import the file again (again in Xmol format). Now you are able to animate the vibration. In order to create a printable picture press `Dismiss` and then type `lulVectorDemo {4 0.1 black}` into the gOpenMol command line window. What you get is:
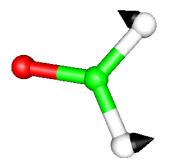
Figure 5.25: The 1395 cm^{-1} mode of the H$_2$CO molecule as obtained from the interface of ORCA to gOpenMol and the orca_pltvib tool to create the animation file.

which indicates that the vibration is a kind of wagging motion of the hydrogens. (I'm sure that you can get nicer arrows with some playing around with gOpenMol). At the gOpenMol homepage you can find a very nice tutorial to teach you some essential visualization tricks.

5.9.3.6 Isotope Shifts

Suppose you have calculated a hessian as in the example discussed above and that you want to predict the effect of substitution with ^{18}O. It would be very bad pratice to recalculate the hessian to do this since the calculation is expensive and the hessian itself is independent of the masses. In this case you can use the small utility program `orca_vib`. First of all you have to look at the `.hess` file and edit the masses given there by hand. For the example given above the `.hess` file looks like:

```
$orca_hessian_file
.....................
$hessian
12
... the cartesian hessian in Eh/bohr**"

$vibrational_frequencies
12


...the vibrational frequencies (in cm-1) as in the output

$normal_modes
12 12
... the vibrational normal modes in Cartesian displacements
#
# The atoms: label  mass x  y  z
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
    #   Here we have changes 15.999 for oxygen into
    #   18.0  in order to see the oxygen 18 effects
    # !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    $atoms
    4
     C     12.0110      0.000000      0.000000     -1.149571
     O     18.0000     -0.000000     -0.000000      1.149695
     H      1.0080     -0.000000      1.750696     -2.275041
     H      1.0080     -0.000000     -1.750696     -2.275041


    $actual_temperature
       0.000000


    $dipole_derivatives
    12


    ... the dipole derivatives (Cartesian displacements)
    #
    # The IR spectrum
    #  wavenumber T**2 TX TY  TY
    #
    $ir_spectrum
    12
    ... the IR intensities
```

if you now call:

```
    orca_vib Test-NumFreq-H2CO.hess
```

you get then printed to standard out the IR spectrum, the vibrational frequencies, the modes, etc. Let us compare the output of this calculation with the original frequency calculation:

```
    H2C16O        H2CO18O          Shift
       6:         1284.36    1282.82 cm**-1      -1.54
       7:         1397.40    1391.74 cm**-1      -5.66
       8:         1766.60    1751.62 cm**-1     -14.98
       9:         2099.20    2061.49 cm**-1     -37.71
      10:         3499.11    3499.02 cm**-1      -0.09
      11:         3645.24    3645.24 cm**-1      -0.00
```

The calculated isotope shifts greatly aid in the identification of vibrations, the interpretation of experiments and in the judgement of the reliability of the calculated vibrational normal modes. A different way of analyzing these isotope shifts is to plot the two predicted spectra and then subtract them from another. This will give you derivative shaped peaks with a zero crossing at the position of the isotope sensitive modes.

5.9.4 Thermochemistry

The second thing that you get automatically as the result of a frequency calculation is a standard thermochemical analysis based on ideal gas statistical mechanics. This can be used to study heats of formation, dissociation energies and similar thermochemical properties.

For example let us calculate a number for the oxygen-oygen dissociation energy in the H_2O_2 molecule. First run the following job:

```
#
# Calculate a value for the O-O bond strength in H2O2
#
! RKS B3LYP TZVP TightSCF SmallPrint Grid4 NoFinalgrid
! Opt NumFreq
! bohrs
* xyz 0 1
 O          -1.396288    -0.075107     0.052125
 O           1.396289    -0.016261    -0.089970
 H          -1.775703     1.309756    -1.111179
 H           1.775687     0.140443     1.711854
*
#
# Now the OH radical job
#
$new_job
! UKS B3LYP TZVP TightSCF SmallPrint Grid4 NoFinalgrid
! Opt NumFreq PModel
! bohrs
* xyz 0 2
 O          -1.396288    -0.075107     0.052125
 H          -1.775703     1.309756    -1.111179
*
```

The first job gives you the following output following the frequency calculation:

```
--------------------------
THERMOCHEMISTRY AT 298.15K
--------------------------


Temperature         ... 298.15 K
Pressure            ... 1.00 atm
Total Mass          ... 34.01 AMU


Throughout the following assumptions are being made:
  (1) The electronic state is orbitally nondegenerate
  (2) There are no thermally accessible electronically excited states
  (3) Hindered rotations indicated by low frequency modes are not
      treated as such but are treated as vibrations and this may
      case some error
  (4) All equations used are the standard statistical mechanics
      equations for an ideal gas
  (5) All vibrations are strictly harmonic


  ------------
```

```
    INNER ENERGY
    ------------


    The inner energy is: U= E(el) + E(ZPE) + E(vib) + E(rot) + E(trans)
        E(el)  - is the total energy from the electronic structure calculation
                 = E(kin-el) + E(nuc-el) + E(el-el) + E(nuc-nuc)
        E(ZPE) - the zero temperature vibrational energy from the frequency calculation
        E(vib) - the finite temperature correction to E(ZPE) due to population
                 of excited vibrational states
        E(rot) - is the rotational thermal energy
        E(trans)- is the translational thermal energy

    Summary of contributions to the inner energy U:
    Electronic energy               ...    -151.54452527 Eh
    Zero point energy               ...       0.02624039 Eh      16.47 kcal/mol
    Thermal vibrational correction  ...       0.00044174 Eh       0.28 kcal/mol
    Thermal rotational correction   ...       0.00141627 Eh       0.89 kcal/mol
    Thermal translational correction ...      0.00141627 Eh       0.89 kcal/mol
    ----------------------------------------------------------------------
    Total thermal energy                    -151.51501060 Eh


    Summary of corrections to the electronic energy:
    (perhaps to be used in another calculation)
    Total thermal correction                0.00327428 Eh       2.05 kcal/mol
    Non-thermal (ZPE) correction            0.02624039 Eh      16.47 kcal/mol
    ----------------------------------------------------------------------
    Total correction                        0.02951468 Eh      18.52 kcal/mol


    --------
    ENTHALPY
    --------


    The enthalpy is H = U + kB*T
                   kB is Boltzmann's constant
    Total free energy               ...    -151.51501060 Eh
    Thermal Enthalpy correction     ...       0.00094421 Eh       0.59 kcal/mol
    -----------------------------------------------------------------------
    Total Enthalpy                  ...    -151.51406639 Eh


    -------
    ENTROPY
    -------


    The entropy contributions are T*S = T*(S(el)+S(vib)+S(rot)+S(trans))
        S(el)  - electronic entropy
        S(vib) - vibrational entropy
        S(rot) - rotational entropy
        S(trans)- translational entropy
    The entropies will be listed as multiplied by the temperature to get
    units of energy

    Electronic entropy              ...       0.00000000 Eh       0.00 kcal/mol
    Vibrational entropy             ...       0.00066248 Eh       0.42 kcal/mol
```

```
   Rotational entropy              ...      0.00752687 Eh      4.72 kcal/mol
   Translational entropy           ...      0.01734394 Eh     10.88 kcal/mol
   ----------------------------------------------------------------------
   Final entropy term              ...      0.02553329 Eh     16.02 kcal/mol



   -------------------
   GIBBS FREE ENTHALPY
   -------------------


   The Gibbs free enthalpy is G = H - T*S

   Total enthalpy                  ...   -151.51406639 Eh
   Total entropy correction        ...     -0.02553329 Eh    -16.02 kcal/mol
   ----------------------------------------------------------------------
   Final Gibbs free enthalpy       ...   -151.53959968 Eh

   For completeness - the Gibbs free enthalpy minus the electronic energy
   G-E(el)                         ...      0.00492559 Eh      3.09 kcal/mol
```

And similarly for the OH-radical job.

```
   ------------
   INNER ENERGY
   ------------



   Summary of contributions to the inner energy U:
   Electronic energy               ...    -75.73288320 Eh
   Zero point energy               ...      0.00838485 Eh      5.26 kcal/mol
   Thermal vibrational correction  ...      0.00000000 Eh      0.00 kcal/mol
   Thermal rotational correction   ...      0.00094418 Eh      0.59 kcal/mol
   Thermal translational correction ...     0.00141627 Eh      0.89 kcal/mol
   ----------------------------------------------------------------------
   Total thermal energy                   -75.72213790 Eh



   Summary of corrections to the electronic energy:
   (perhaps to be used in another calculation)
   Total thermal correction                0.00236045 Eh      1.48 kcal/mol
   Non-thermal (ZPE) correction            0.00838485 Eh      5.26 kcal/mol
   ----------------------------------------------------------------------
   Total correction                        0.01074531 Eh      6.74 kcal/mol



   --------
   ENTHALPY
   --------


   The enthalpy is H = U + kB*T
                kB is Boltzmann's constant
   Total free energy               ...    -75.72213790 Eh
   Thermal Enthalpy correction     ...      0.00094421 Eh      0.59 kcal/mol
   ----------------------------------------------------------------------
   Total Enthalpy                  ...    -75.72119369 Eh
```

```
    -------
    ENTROPY
    -------


    Electronic entropy          ...      0.00065446 Eh      0.41 kcal/mol
    Vibrational entropy         ...      0.00000000 Eh      0.00 kcal/mol
    Rotational entropy          ...      0.00256479 Eh      1.61 kcal/mol
    Translational entropy       ...      0.01636225 Eh     10.27 kcal/mol
    ----------------------------------------------------------------------
    Final entropy term          ...      0.01958150 Eh     12.29 kcal/mol




    -------------------
    GIBBS FREE ENTHALPY
    -------------------


    The Gibbs free enthalpy is G = H - T*S

    Total enthalpy              ...     -75.72119369 Eh
    Total entropy correction    ...      -0.01958150 Eh    -12.29 kcal/mol
    ----------------------------------------------------------------------
    Final Gibbs free enthalpy   ...     -75.74077518 Eh

    For completeness - the Gibbs free enthalpy minus the electronic energy
    G-E(el)                     ...      -0.00789198 Eh     -4.95 kcal/mol
```

Let us calculate the free energy change for the reaction:

```
    H2O2  ->   2 OH
```

The individual energy terms are: 2OH(Eh) - H2O(Eh) kcal/mol

Electronic Energy: (-151.465766)-(-151.544525) 49.42

Zero-point Energy: (0.016770)-(0.026240) -5.94

Thermal Correction(translation/rotation): (0.00472)-(0.002832) 1.18

Thermal Enthalpy Correction: (0.001888)-(0.000944) 0.59

Entropy: -(0.039163)-(-0.025533) -8.55

Final G 36.70

Thus, both the zero-point energy and the entropy terms both contribute significantly to the total free energy change of the reaction. The entropy term is favoring the reaction due to the emergence of new translational and rotational degrees of freedom. The zero-point correction is also favoring the reaction since the zero-point vibrational energy of the O-O bond is lost. The thermal correction and the enthalpy correction are both small.

TIP:

- You can run the thermochemistry calculations at several user defined temperatures by providing the program with a list of temperatures:

```
%freq  Temp  290, 295,300
        end
```

5.9.5 Electrical Properties

A few basic electric properties can be calculated in ORCA although this has never been a focal point of development. The properties can be accessed straightforwardly through the %elprop block:

```
! RKS B3LYP SVP SmallPrint TightSCF Grid4

%elprop Dipole      true
        Quadrupole True
        Polar      1   # analytic polarizability through CP-SCF
                       # possible for SCF runs (HF and DFT)
                       # = 3: fully numeric
        end
* int 0 1
   C  0  0  0  0       0           0
   H  1  0  0  1.09  109.4712      0
   H  1  2  0  1.09  109.4712      0
   H  1  2  3  1.09  109.4712  120
   H  1  2  3  1.09  109.4712  240
*
```

The polarizability is calculated analytically through solution of the coupled-perturbed SCF equations for HF and DFT runs. For MP2 one can differentiate the analytical dipole moment calculated with relaxed densities. For other correlation methods only a fully numeric approach is possible.

```
-------------------------
THE POLARIZABILITY TENSOR
-------------------------


The raw cartesian tensor (atomic units):
    12.85800     -0.00001     -0.00000
    -0.00001     12.85797     -0.00000
    -0.00000     -0.00000     12.85794
diagonalized tensor:
    12.85794     12.85797     12.85801


     0.00022     -0.24634      0.96918
     0.00063     -0.96918     -0.24634
     1.00000      0.00067     -0.00005


Isotropic polarizability :  12.85797
```

As expected the polarizability tensor is isotropic.

The following jobs demonstrate the numeric and analytic calculations of the polarizability:

```
# -------------------------------------------
# Numerical calculation of the polarizability
# -------------------------------------------
! UKS B3LYP/G SVP VeryTightSCF
%elprop  Polar        3
         EField   1e-5
         end
* int 1 2
   C  0  0  0  0      0  0
   O  1  0  0  1.1105 0  0
*


$new_job
# -------------------------------------------
# Analytical calculation of the polarizability
# -------------------------------------------
! UKS B3LYP/G SVP VeryTightSCF
%elprop  Polar      1
         Tol     1e-7
         end
* int 1 2
   C  0  0  0  0      0  0
   O  1  0  0  1.1105 0  0
*
```

Here the polarizability of CO^+ is calculated twice – first numerically using a finite field increment of 10^{-5} au and then analytically using the CP-SCF method. In general the analytical method is much more efficient, especially for increasing molecular sizes.

At the MP2 level you need to differentiate the dipole moment numerically in order to obtain the polarizability tensor. This will in general require tight SCF converge in order to not get too much numerical noise in the second derivative. Also, you should experiment with the finite field increment in the numerical differentiation process. For example consider the following simple job:

```
! RHF MP2 SVP VeryTightSCF
%elprop  Polar   2
         EField  1e-4
         end
* int 0 1
   C  0  0  0  0      0  0
   O  1  0  0  1.130  0  0
*
```

In a similar way, polarizability calculations are possible with CASSCF. For other correlation methods, where not even response densities are available, only a fully numeric approach (Polar=3) is possible and requires obnoxiously tight convergence.

Note that polarizability calculations have higher demands on basis sets. A rather nice basis set for this property is the Sadlej one (see 6.3.1).

5.9.6 NMR Chemical Shifts

NMR chemical shift at the SCF level (HF and DFT) can be obtained from the EPR/NMR module of ORCA. The program supports two meaningful choices of origin: the first and recommended one is the use of the IGLO procedure; the second (less recommended) is to choose the "parent" nucleus as the origin.

The use of the chemical shift module is relatively simple:

```
! RHF VWN5 IGLOII SmallPrint TightSCF Grid4
* int 0 1
   C  0  0  0  0       0         0
   H  1  0  0  1.09  109.4712    0
   H  1  2  0  1.09  109.4712    0
   H  1  2  3  1.09  109.4712  120
   H  1  2  3  1.09  109.4712  240
*
%eprnmr  ori IGLO  # alternative OwnNuc
         LocMet FB # localization method for IGLO
                   # FB=Foster-Boys PM=Pipek-Mezey (default)
         Nuclei = all C { shift }
         Nuclei = all H { shift }
         end
```

The output looks like the following. It contains detailed information about the orientation of the tensor, the eigenvalues, its isotropic part etc.

```
    ---------------
    CHEMICAL SHIFTS
    ---------------


    --------------
    Nucleus   0C :
    --------------
    Raw-matrix :
          0.0001964   -0.0000000    0.0000000
         -0.0000000    0.0001964   -0.0000000
         -0.0000000    0.0000000    0.0001964

    Diagonalized sT*s matrix:
    sDSO    0.0002562    0.0002562    0.0002562  iso=   0.0002562
    sPSO   -0.0000598   -0.0000598   -0.0000598  iso=  -0.0000598
            ----------   ----------   ----------
```

```
     Total    0.0001964      0.0001964      0.0001964  iso=   0.0001964


     Orientation:
       X    -0.0155890    -0.4861104     0.8737584
       Y    -0.0233997    -0.8734479    -0.4863551
       Z     0.9996046    -0.0280275     0.0022413


     --------------
     Nucleus   1H :
     --------------
     Raw-matrix :
             0.0000379      0.0000000      0.0000000
             0.0000000      0.0000273     -0.0000000
             0.0000000      0.0000000      0.0000273


     Diagonalized sT*s matrix:
     sDSO   0.0000255      0.0000255      0.0000379  iso=   0.0000297
     sPSO   0.0000018      0.0000018     -0.0000000  iso=   0.0000012
            ----------     ----------     ----------
     Total   0.0000273      0.0000273      0.0000379  iso=   0.0000308


     Orientation:
       X     0.0000001      0.0000060     1.0000000
       Y    -0.0747638     -0.9972013     0.0000060
       Z     0.9972013     -0.0747638     0.0000004
```

Thus, the absolute, isotropic shielding for the ^{1}H nuclei is predicted to be 30.8 ppm and for ^{13}C it is 196.4 ppm. NMR calculations with IGLO are perhaps best done with the specifically designed IGLO-II or IGLO-III basis sets. The EPR-II and EPR-III basis sets are also appropriate. However, since IGLO converges rather nicely to the basis set limit concerns about the basis set only arise if it is really too small, e.g. minimal basis sets or basis sets without polarization functions.[18]

In order to compare to experiment it is common practice to choose a standard molecule for the type of nucleus of interest (for example either CH_4 or TMS for proton shifts) and then obtain the chemical shift by subtraction of the reference value from the computed value. It is of course important that the reference and target calculations have been done with the same basis set, functional choice of origin and localization procedure. Otherwise the values are simply not compatible with each other.

5.9.7 Hyperfine and Quadrupole Couplings

Hyperfine and quadrupole couplings can be obtained from the EPR/NMR module of ORCA. Since there may be several nuclei that you might be interested in the input is relatively sophisticated.

An example how to calculate the hyperfine and field gradient tensors for the CN radical is given below:

---

[18] However, basis sets like SVP may be a little too small. For example, the ^{13}C shift for CH_4 is predicted to be 239.9 ppm which is a large error compared to IGLO-II (196.6 ppm); IGLO-III gives 192 ppm. The much larger aug-cc-pVTZ leads to a predicted ^{13}C shielding of 214.3 ppm thus showing that there still is a fair bit of oscillation around the basis set limit and that therefore it is important to properly standardize.

```
   ! UKS PBE0 DZP SmallPrint TightSCF Grid5
   * int 0 2
     C  0  0  0  0     0  0
     N  1  0  0  1.170 0  0
   *
   %eprnmr  Nuclei = all C { aiso, adip }
            Nuclei = all N { aiso, adip, fgrad }
            end
```

In this example the hyperfine tensor is calculated for all carbon atoms and all nitrogen atoms which, of course, happen to be a single atom in this specific case. In general you can also give the numbers of certain atoms instead of 'all' in order to avoid undesired calculations. **A more subtle point is that you have to specify the Nuclei statement** *after* **the definition of the atomic coordinates or the program will not figure out what is meant by 'all'.**

The output looks like the following. It contains similar detailed information about the individual contributions to the hyperfine couplings, its orientation, its eigenvalues, the isotropic part and (if requested) also the quadrupole coupling tensor.

```
   ----------------------------------------
   ELECTRIC AND MAGNETIC HYPERFINE STRUCTURE
   ----------------------------------------


   -----------------------------------------------------------
   Nucleus   0C : I= 0.5 P=134.1900 MHz/au**3 Q=  0.0000 barn
   -----------------------------------------------------------
   Raw HFC matrix (all values in MHz):
           696.2146      -0.0000      -0.0000
            -0.0000     542.9631      -0.0000
            -0.0000      -0.0000     542.9631

   A(FC)    594.0469     594.0469     594.0469
   A(SD)    -51.0838     -51.0838     102.1677
            ----------   ----------   ----------
   A(Tot)   542.9631     542.9631     696.2146 A(iso)=   594.0469
   Orientation:
    X     -0.0000000   -0.0000000   -1.0000000
    Y     -0.9999951    0.0031335    0.0000000
    Z     -0.0031335   -0.9999951    0.0000000


   -----------------------------------------------------------
   Nucleus   1N : I= 1.0 P= 38.5677 MHz/au**3 Q=  0.0193 barn
   -----------------------------------------------------------
   Raw HFC matrix (all values in MHz):
            13.1112       0.0000       0.0000
             0.0000     -45.8793       0.0000
             0.0000       0.0000     -45.8793

   A(FC)    -26.2158     -26.2158     -26.2158
   A(SD)     39.3270     -19.6635     -19.6635
            ----------   ----------   ----------
   A(Tot)    13.1112     -45.8793     -45.8793 A(iso)=   -26.2158
```

```
     Orientation:
      X     -1.0000000     0.0000000     0.0000000
      Y     -0.0000000    -0.9999998     0.0005676
      Z     -0.0000000    -0.0005676    -0.9999998


      Raw EFG matrix (all values in a.u.**-3):
              -0.1952        0.0000        0.0000
               0.0000        0.0976        0.0000
               0.0000        0.0000        0.0976


     V(El)      0.6527        0.6527       -1.3054
     V(Nuc)    -0.5551       -0.5551        1.1102
               ----------    ----------    ----------
     V(Tot)     0.0976        0.0976       -0.1952
     Orientation:
      X     -0.0000001     0.0000001     1.0000000
      Y      0.0069379     0.9999759    -0.0000001
      Z     -0.9999759     0.0069379    -0.0000001


     Quadrupole tensor eigenvalues (in MHz;Q= 0.0193 I=  1.0)
      e**2qQ            =     -0.886 MHz
      e**2qQ/(4I*(2I-1))=     -0.222 MHz
      eta               =      0.000
      NOTE: the diagonal representation of the SH term
         I*Q*I = e**2qQ/(4I(2I-1))*[-(1-eta),-(1+eta),2]
```

Another point for hyperfine calculations is that you should normally use basis sets that have more flexibility in the core region. In the present example a double zeta basis set was used. For accurate calculations you need more flexible basis sets. There are several dedicated basis set for hyperfine calculations: (a) the EPR-II basis of Barone and co-workers. It is only available for a few light atoms (H, B, C, N, O, F). It is essentially of double-zeta plus polarization quality with added flexibility in the core region and should give reasonable results (b) The IGLO-II and IGLO-III bases of Kutzelnigg and co-workers. They are fairly accurate but also only available for some first and second row elements (c) the CP basis for first row transition metals which is accurate as well. (d) General purpose HF-limit basis sets are the uncontracted Partridge basis sets. They will probably be too expensive for routine use but are useful for calibration purposes.

For other elements ORCA does not yet have dedicated default basis sets for this situation it is very likely that you have to tailor the basis set to your needs. If you use the statement `Print[p_basis] 2` in the `output` block the program will print the actual basis set in input format (for the basis block). You can then add or remove primitives, uncontract core bases etc. For example here is a printout of the carbon basis DZP in input format:

```
 # Basis set for element : C
   NewGTO 6
   s 5
     1   3623.8613000000        0.0022633312
     2    544.0462100000        0.0173452633
     3    123.7433800000        0.0860412011
     4     34.7632090000        0.3022227208
     5     10.9333330000        0.6898436475
   s 1
```

```
        1        3.5744765000         1.0000000000
    s  1
        1        0.5748324500         1.0000000000
    s  1
        1        0.1730364000         1.0000000000
    p  3
        1        9.4432819000         0.0570590790
        2        2.0017986000         0.3134587330
        3        0.5462971800         0.7599881644
    p  1
        1        0.1520268400         1.0000000000
    d  1
        1        0.8000000000         1.0000000000
    end;
```

The 's 5', for example, stands for the angular momentum and the number of primitives in the first basis function. Then there follow five lines that have the number of the primitive, the exponent and the contraction coefficient (unnormalized) in it. Remember also that when you add very steep functions you *must* increase the size of the integration grid if you do DFT calculations! *If you do not do that your results will be inaccurate.* You can increase the radial grid size by using `IntAcc` in the `Method` block or for individual atoms (section 6.2.2.2 explains how to do this in detail). In the present example the changes caused by larger basis sets in the core region and more accurate integration are relatively modest – on the order of 3%, which is, however, still significant if you are a little puristic.

The program can also calculate the spin-orbit coupling contribution to the hyperfine coupling tensor as described in section 6.21.3

If also EPR g-tensor or D-tensor calculations (see next section) are carried out in the same job, ORCA automatically prints the orientation between the hyperfine/quadrupole couplings and the molecular g- or D-tensor. For more information on this see section 6.24.10.

5.9.8 The EPR g-Tensor and the Zero-Field Splitting Tensor

The EPR g-tensor is a property that can be well calculated at the SCF level with ORCA through solution of the coupled-perturbed SCF equations. Consider the following multi-job input that computes the g-tensor at three different levels of theory:

```
! UHF HF SVP TightSCF SOMF(1X)
%method frozencore fc_ewin end
%eprnmr gtensor 1 ewin -100,100 ori -3
        end
* int 1 2
   O  0  0  0  0       0  0
   H  1  0  0  1.1056 0  0
   H  1  2  0  1.1056 109.62 0
*
```

```
$new_job
! UKS LSD Def-2 RI SmallPrint PModel SOMF(1X)
%method frozencore fc_ewin end
%eprnmr gtensor 1 ewin -100,100 ori -3
        end
* int 1 2
   O  0  0  0  0      0  0
   H  1  0  0  1.1056 0  0
   H  1  2  0  1.1056 109.62 0
*


$new_job
! UKS BP Def-2 RI SmallPrint PModel SOMF(1X)
%method frozencore fc_ewin end
%eprnmr gtensor 1 ori -3
        end
* int 1 2
   O  0  0  0  0      0  0
   H  1  0  0  1.1056 0  0
   H  1  2  0  1.1056 109.62 0
*
```

The statement `ori -3` means that the origin for this gauge dependent property is to be taken at the center of electronic charge. The `SOMF(1X)` defines the details of the spin-orbit-coupling operator and are explained in section 6.21.2 (in the present case an accurate variant is specified) Other choices and additional variables exist and are explained in detail in section 6.21.2.

The output looks like the following. It contains information on the contributions to the g-tensor (relativistic mass correction, diamagnetic spin-orbit term (= gauge-correction), paramagnetic spin-orbit term (= OZ/SOC)), the isotropic g-value and the orientation of the total tensor.

```
-------------------
ELECTRONIC G-MATRIX
-------------------

 Diagonalized gT*g matrix (sqrt of eigenvalues taken):
          2.0101862    -0.0031528     0.0000000
         -0.0031529     2.0079392     0.0000000
          0.0000000     0.0000000     2.0021428


 gel       2.0023193    2.0023193    2.0023193
 gRMC     -0.0003182   -0.0003182   -0.0003182
 gDSO(tot) 0.0000725    0.0001471    0.0001485
 gPSO(tot) 0.0000692    0.0035675    0.0102602
          ----------   ----------   ----------
 g(tot)    2.0021428    2.0057157    2.0124097 iso= 2.0067561
 Delta-g  -0.0001765    0.0033964    0.0100905 iso= 0.0044368
 Orientation:
```

```
    X          -0.0000000   -0.5763356    0.8172131
    Y          -0.0000000   -0.8172131   -0.5763356
    Z           1.0000000   -0.0000000   -0.0000000
```

If the total spin of the system is $S > 1/2$ then the zero-field-splitting tensor can also be calculated and printed. For example consider the following job on a hypothetical Mn(III)-complex.

```
#
# Test the ZFS program
#
! UKS BP86 RI Def-2 NoFinalGrid SOMF(1X)

%eprnmr gTensor   1  # do the g-tensor
        Ori      -3  # choice of origin
        DTensor  ssandso  # SS and SO parts
                    # SS  = only SS part
                    # SO  = only SOC part
                    # SSandSO = both parts
        DSOC     CP  # coupled perturbed (default)
                    # = QRO  (quasi-restricted method)
                    # = PK   (Pederson-Khanna method)
        DSS      DIRECT #  directly use the canonical orbitals for the spin-density
                    # = uno    use spin-density from UNOs
        end

* int  1 5
Mn 0 0 0  0 0 0
 O 1 0 0  2.05   0   0
 O 1 2 0  2.05  90   0
 O 1 2 3  2.05  90 180
 O 1 2 3  2.05 180   0
 F 1 2 3  1.90  90  90
 F 1 2 3  1.90  90 270
 H 2 1 6  1.00 127   0
 H 2 1 6  1.00 127 180
 H 3 1 6  1.00 127   0
 H 3 1 6  1.00 127 180
 H 4 1 6  1.00 127   0
 H 4 1 6  1.00 127 180
 H 5 1 6  1.00 127   0
 H 5 1 6  1.00 127 180
*
```

The output documents the individual contributions to the D-tensor which also contains (unlike the g-tensor) contributions from spin-flip terms.

Some explanation must be provided:

- The present implementation in ORCA is valid for HF, DFT and hybrid DFT. An approximation to DSS can be obtained from CASSCF.

- There are four different variants of the SOC-contribution, which shows that this is a difficult property. We will briefly discuss the various choices.

- The QRO method is fully documented in (Neese, F. (**2006**) Importance of Direct Spin-Spin Coupling and Spin-Flip Excitations for the Zero-Field Splittings of Transition Metal Complexes: A Case Study, *J. Am. Chem. Soc.*, 128, 10213-10222) and this paper should be cited. It is based on the theory developed earlier (Neese, F.; Solomon, E.I. (**1998**) Calculation of Zero-Field Splittings, *g*-values and the Relativistic Nephelauxetic Effect in Transition Metal Complexes. Application to High Spin Ferric Complexes. *Inorg. Chem.*, 37, 6568-6582). The QRO method is reasonable but somewhat simplistic and is superseded by the CP method described below.

- The Pederson-Khanna model was brought forward in 1999 from qualitative reasoning and is documented in: Pederson, M. R.; Khanna, S. N., *Phys. Rev. B.* **1999,** 60, (13), 9566. It also contains incorrect prefactors for the spin-flip terms. We have nevertheless implemented the method for comparison. In the original form it is only valid for local functionals. In ORCA it is extended to hybrid functionals and HF.

- The coupled-perturbed method is a generalization of the DFT method for ZFSs it uses revised prefactors for the spin-flip terms and solves a set of coupled-perturbed equations for the SOC perturbation. Therefore it is valid for hybrid functionals. It is described in detail in Neese, F. (**2007**) Calculation of the Zero-Field Splitting Tensor Using Hybrid Density Functional and Hartree-Fock Theory. *J. Chem. Phys.*, 127, 164112.

- The DSS part is an expectation value that involves the spin-density of the system. In detailed calibration work (Sinnecker, S.; Neese, F. (**2006**) *J. Phys. Chem. A*, 110, 12267-12275) it was found that the spin-unrestricted DFT methods behave somewhat erratically and that much more accurate values were obtained from open-shell spin-restricted DFT. Therefore the 'UNO' option allows the calculation of the SS term with a restricted spin-density obtained from the singly occupied unrestricted natural orbitals.

- The DSS part contains an erratic selft-interaction term for UKS/UHF wavefunction and canonical orbitals. Thus, `UNO` is recommended for these types of calculations. The issue is reported in the paper: Riplinger, C.; Neese, F.; et al. (**2009**) *J. Am. Chem. Soc.*, 131, 10092-10106. If the option `DIRECT` is used nevertheless, ORCA will print a warning in the respective part of the output.

5.9.9 Mössbauer Parameters

^{57}Fe Mössbauer spectroscopy probes the transitions of the nucleus between the $I = \frac{1}{2}$ ground state and the $I = \frac{3}{2}$ excited state at 14.4 keV above the ground state. The important features of the Mössbauer spectrum are the isomer shift (δ) and the quadrupole splitting (ΔE_Q). An idealized spectrum is shown in Figure 5.26.

The isomer shift measures the shift in the energy of the γ-ray absorption relative to a standard, usually Fe foil. The isomer shift is sensitive to the electron density at the nucleus, and indirectly probes changes in the bonding of the valence orbitals due to variations in covalency and 3d shielding. Thus, it can be used to probe oxidation and spin states, and the coordination environment of the iron.

The quadrupole splitting arises from the interaction of the nuclear quadrupole moment of the excited state with the electric field gradient at the nucleus. The former is related to the non-spherical charge distribution in the excited state. As such it is extremely sensitive to the coordination environment and the geometry of the complex.
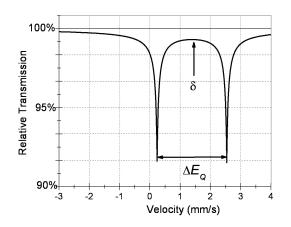
Figure 5.26: A typical Mössbauer spectrum showing both the isomer shift, δ, and the quadrupole splitting, ΔE_{Q}.

Both the isomer shift and quadrupole splitting can be successfully predicted using DFT methods. The isomer shift is directly related to the s electron density at the nucleus and can be calculated using the formula

$$\delta = \alpha(\rho_0 - C) + \beta \tag{5.10}$$

where α is a constant that depends on the change in the distribution of the nuclear charge upon absorption, and ρ_0 is the electron density at the nucleus [92]. The constants α and β are usually determined via linear regression analysis of the experimental isomer shifts versus the theoretically calculated electron density for a series of iron compounds with various oxidation and spin states. Since the electron density depends on the functional and basis set employed, fitting must be carried out for each combination used. A recent compilation of calibration constants (α, β and C) for various methods is given in Roemelt, M.; Ye, S.; Neese, F. *Inorg. Chem.* **2009**, <u>48</u>, 784–785. Usually an accuracy of better than 0.10 mm s^{-1} can be achieved for DFT with reasonably sized basis sets.

The quadrupole splitting is proportional to the largest component of the electric field gradient (EFG) tensor at the iron nucleus and can be calculated using the formula:

$$\Delta E_{\mathrm{Q}} = \frac{1}{2} eQV_{zz} \left(1 + \frac{\eta^2}{3}\right)^{\frac{1}{2}} \tag{5.11}$$

where e is the electrical charge of an electron and Q is the nuclear quadrupole moment of Fe (approximately 0.16 barns). V_{xx}, V_{yy} and V_{zz} are the electric field gradient tensors and η, defined as

$$\eta = \left|\frac{V_{xx} - V_{yy}}{V_{zz}}\right| \tag{5.12}$$

is the asymmetry parameter in a coordinate system chosen such that $|V_{zz}| \geq |V_{yy}| \geq |V_{xx}|$.

An example of how to calculate the electron density and quadrupole splitting of an iron center is as follows:

```
    %eprnmr
       nuclei = all Fe {fgrad, rho}
    end
```

If the core properties basis set CP(PPP) is employed, one has to increase the radial integration accuracy for the iron atom, for example:

```
    %basis newgto Fe "CP(PPP)" end
    end

    %method
      SpecialGridAtoms 26
      SpecialGridIntAcc 7
    end
```

The output file should contain the following lines, where you obtain the calculated quadrupole splitting directly and the RHO(0) value (the electron density at the iron nucleus). To obtain the isomer shift one has to insert the RHO(0) value into the appropriate linear equation (Eq. 5.10).

```
    Moessbauer quadrupole splitting parameter (proper coordinate system)
    e**2qQ   = -0.406 MHz = -0.035 mm/s
    eta      = 0.871
    Delta-EQ=(1/2e**2qQ*sqrt(1+1/3*eta**2)) = -0.227 MHz = -0.020 mm/s
    RHO(0)=  11581.352209571 a.u.**-3   # the electron density at the Fe nucleus.
```

5.9.10 Broken-Symmetry Wavefunctions and Exchange Couplings

A popular way to estimate the phenomenological parameter J_{AB} that enters the Heisenberg–Dirac–van Vleck Hamiltonian which parameterizes the interaction between two spin systems is the 'broken-symmetry' formalism. The phenomenological Hamiltonian is:

$$H_{HDvV} = -2J_{AB}\vec{S}_A\vec{S}_B \tag{5.13}$$

It is easy to show, that such a Hamiltonian leads to a 'ladder' of spin states from $S = S_A + S_B$ down to $S = |S_A - S_B|$. If the parameter J_{AB} is positive the systems 'A' and 'B' are said to be *ferromagnetically* coupled because the highest spin-state is lowest in energy while in the case that J_{AB} is negative the coupling is *antiferromagnetic* and the lowest spin state is lowest in energy.

In the broken symmetry formalism one tries to obtain a wavefunction that breaks spatial (and spin) symmetry. It may be thought of as a 'poor man's MC-SCF' that simulates a multideterminantal character within a single determinant framework. Much could be said about the theoretical advantages, disadvantages, problems and assumptions that underly this approach. Here, we only want to show how this formalism is applied within ORCA.

For N_A unpaired electrons localized on 'site A' and N_B unpaired electrons localized on a 'site B' one can calculate the parameter J_{AB} from two separate spin-unrestricted SCF calculations: (a) the calculation for the high-spin state with $S = \frac{(N_A + N_B)}{2}$ and (b) the 'broken symmetry' calculation with $M_S = \frac{(N_A - N_B)}{2}$ that features N_A spin-up electrons

that are quasi-localized on 'site A' and N_B spin-down electrons that are quasi-localized on 'site B'. Several formalisms exist to extract J_{AB}:

$$J_{AB} = -\frac{(E_{HS} - E_{BS})}{(S_A + S_B)^2} \tag{5.14}$$

$$J_{AB} = -\frac{(E_{HS} - E_{BS})}{(S_A + S_B)(S_A + S_B + 1)} \tag{5.15}$$

$$J_{AB} = -\frac{(E_{HS} - E_{BS})}{\langle S^2 \rangle_{HS} - \langle S^2 \rangle_{BS}} \tag{5.16}$$

We prefer the last definition (Eq. 5.16) because it is approximately valid over the whole coupling strength regime while the first equation implies the weak coupling limit and the second the strong coupling limit.

In order to apply the broken symmetry formalism use:

```
%scf BrokenSym NA,NB
      end
```

The program will then go through a number of steps. Essentially it computes an energy and wavefunction for the high-spin state, localizes the orbitals and reconverges to the broken symmetry state.

CAUTION: Make sure that in your input coordinates 'site A' is the site that contains the larger number of unpaired electrons!

Most often the formalism is applied to spin coupling in transition metal complexes or metal-radical coupling or to the calculation of the potential energy surfaces in the case of homolytic bond cleavage. In general, hybrid DFT methods appear to give reasonable semiquantitative results for the experimentally observed splittings.

As an example consider the following simple calculation of the singlet–triplet splitting in a stretched Li_2 molecule:

```
#
# Example of a broken symmetry calculation
#
! UKS B3LYP/G SVP TightSCF Grid4 NoFinalGrid
%scf BrokenSym 1,1
      end
* xyz 0 3
  Li  0  0  0
  Li  0  0  4
*
```

There is a second mechanism for generating broken-symmetry solutions in ORCA. This mechanism uses the individual spin-densities and is invoked with the keywords `FlipSpin` and `FinalMs`. The strategy is to exchange the alpha and beta spin blocks of the density on certain user-defined centers after converging the high-spin wavefunction. With this method arbitrary spin topologies should be accessible. The use is simple:

```
     #
     # Example of a broken symmetry calculation using the "FlipSpin" feature
     #
     ! UKS B3LYP/G SVP TightSCF Grid4 NoFinalGrid
     %scf
          FlipSpin 1
            # Flip spin is a vector and you can give a list of atoms
            # on which you want to have the spin flipped. For example
            #  FlipSpin 17,38,56
            # REMEMBER: counting starts at atom 0!
          FinalMs  0
            # The desired Ms value of the broken symmetry determinant.
            # This value MUST be given since the program cannot determine it by itself.
          end
     * xyz 0 3
       Li  0  0  0
       Li  0  0  4
     *
```

5.10 Molecular Dynamics Simulations

Since ORCA version 2.8, a molecular dynamics program is available. It allows to run direct molecular dynamics also known as "on-the-fly" molecular dynamics. Any method in ORCA can be used with either analytical or numerical gradients, which are automatically requested. The obtained gradients are inserted in the classical Newton's equations of motions and solved numerically using one of three variants of the Verlet algorithm (Velocity Verlet [93], Verlet [94] and Leapfrog [95]). The keyword to select the algorithm is MD_Algorithm and the default is the Velocity Verlet. In addition, an implicit solvent model can be included using COSMO.

5.10.1 Basic Input

In order to request a molecular dynamics simulation with default values just an MD keyword is required on the simple input line:

```
     ! HF SVP MD
     * xyz 0 1
         H  0.00 0.00 0.00
         H  0.00 0.00 1.00
     *
```

A modification of the default settings can be done in the MD block:

```
     %md    MD_Algorithm  0    # Velocity_Verlet(Default)
                          1    # Verlet
                          2    # Leapfrog


            DT        10.0     # time step in atomic time units
            MAXSTEP   100.0    # maximum number of iterations


            INIT_VELOCITIES    # initial velocities
                      ZERO     # All velocities are set to 0
                      READ     # External file provides the velocities


            THERMO       0     # No thermostat
                         1     # Scaling velocities
```

5.11 QM/MM Calculations

ORCA is easy to interface as a QM engine in pretty much any QM/MM environment, as it will accept a set of point charges from an external file (see section 6.1.3) and it will return, in a transparent format, all the required information for computing energies and gradients to the driving program. In our research group we have experience with three different QM/MM environments: Gromacs, pDynamo and ChemShell. In the following each of the interfaces are described. Is beyond the scope of the manual to be exhaustive, and only minimal working examples are going to be presented. These will cover mainly the technical aspects with respect to the QM part of the QM/MM calculation. For the initial preparation of the system the user is referred to the documentation of the driving program.

5.11.1 ORCA and Gromacs

In cooperation with the developers of Gromacs software package we developed an interface to ORCA. The interface is available starting with Gromacs version 4.0.4.

As mentioned above, the initial setup has to be done with the Gromacs. In the QM/MM calculation Gromacs is will call ORCA to get the energy and the gradients of the QM atoms. The interface can be used to perform all job types allowed by the Gromacs software package, e.g. optimizations, molecular dynamics, free energy. In addition, for geometry optimizations we have implemented a microiterative scheme that can be requested by setting the keyword bOpt = yes in the Gromacs .mdp file. This will cause ORCA to perform a QM geometry optimization every time it is called by Gromacs. During this optimization ORCA will also compute the Lennard-Jones interaction between the QM and MM atoms, and freeze the boundary atoms. This microiterative scheme can also be used to perform a transition state optimization. If you are looking for a transition state and have a good initial guess for the structure, you can carry out an optimization of the MM system and at the same time perform a transition state optimization of the QM system with ORCA via the microiterative scheme. Since it is expensive to calculate the Hessian for each microiterative microiterative step, the user can tell ORCA to use the (updated) Hessian matrix of the previous step via read_temp_Hess in the ORCA input.

In order to allow full flexibility to the user, the information for the QM run are provided in a separate plain text file, called GromacsBasename.ORCAINFO. When Gromacs writes the input for the ORCA calculation, it will merge this file with the information on the coordinates, point charges, Lennard-Jones coefficients and type of calculation (EnGrad, Opt, TSOpt).

Below is a typical example of an input file created by Gromacs, where for each Gromacs optimization step, a full optimization of the QM-part will be performed by ORCA, instead of just doing the energy and gradient calculation.

```
# Optimization step in the Lennard-Jones and point charges field of the MM atoms
! QMMMOpt

# file containing the Lennard Jones coefficients for the Lennard-Jones interaction
%LJCoefficients "temp.LJ"
# file containing the point charges for the electrostatic interaction
%pointcharges "temp.pc"

%geom
    # calculate the exact Hessian before the first optimization step
    Calc_Hess true
    # in case of a TS optimization the updated Hessian of the previous
    # TS optimization run is read instead of calculating a new one
    read_temp_Hess true
  end
```

NOTES:

- If you are using b0pt or bTS you have to take care of additional terms over the boundary. Either you can zero out the dihedral terms of the Q2-Q1-M1-M2 configurations, or you can fix the respective Q2 atoms.

- If you want to use the ORCA constraints, you have to also put them in the Gromacs part of the calculation.

- If there are no bonds between QM and MM system, the b0pt optimization of the QM system might have convergence problems. This is the case if the step size is too large and thus QM atoms come too close to MM atoms. The convergence problems can be circumvented by adding extra coordinates and Hessian diagonal values for Cartesian coordinates of selected QM atoms to the set of internal coordinates. This should be done for only few atoms that are in the boundary region.

```
%geom
    # add the Cartesian position of atoms 2 and 4 to the
    # set of internal coordinates with a diagonal Hessian value of 0.1
    Hess_Internal
    {C 2 D 0.1}
    {C 4 D 0.1}
    end
end
```

ORCA and pDynamo

The interface with the pDynamo library is briefly documented here. It uses the same plain text files to exchange information between the pDynamo library and ORCA. In order to have simiar functionality as presented above, we have extended the interface to generate more complex ORCA input files by accepting verbatim blocks of text. We have also implemented in pDynamo the capability of writing files containing Lennard-Jones parameters.

A simple example which calculates the QM/MM energy for a small designed peptide in which the side chain of one amino acid is treated QM is ilustrated below. For this example, a complete geometry optimization is going to be performed during the ORCA call.

```python
import os
import pCore
import pBabel
import pMolecule
import pMoleculeScripts

# Read the initial coordinates from the .pdb file.
system = pMoleculeScripts.PDBFile_ToSystem(
        "1UAO.pdb", modelNumber=1, useComponentLibrary=True)

# Instantiate the required models.
mmmodel = pMolecule.MMModelOPLS("protein")
nbmodel = pMolecule.NBModelORCA()
qcmodel = pMolecule.QCModelORCA(
        command=os.getenv("ORCA_COMMAND"),
        deleteJobFiles=False, header="bp86 def2-svp qmmmopt/pdynamo",
        job="chignolin", run=True)

# Assign the models to the system.
system.DefineMMModel(mmmodel)
system.DefineQCModel(
        qcmodel, qcSelection=pCore.Selection([35, 36, 37, 34, 40, 41]))
system.DefineNBModel(nbmodel)
system.electronicState = pMolecule.ElectronicState(
        charge=-1, multiplicity=1)

# Print a summary and calculate the energy.
system.Summary()
system.Energy()
```

After the execution of the above Python program, a series of files are going to be created `chignolin.inp`, `chignolin.pc`, `chignolin.lj` and ORCA is going to be called. The generated ORCA input file is listed below.

```
! bp86 def2-svp qmmmopt/pdynamo
% geom
    constraints
        {C 0 C}
        {C 1 C}
        end
    end
```

```
% pointcharges "chignolin.pc"
% ljcoefficients "chignolin.lj"
* xyz -1 1
H                 -1.0637532468        1.1350324675         2.4244220779
C                 -0.5230000000        0.6870000000         3.2490000000
C                  0.4180000000        1.7240000000         3.8660000000
O                 -0.0690000000        2.7620000000         4.2830000000
O                  1.6090000000        1.4630000000         3.9110000000
H                 -1.2240000000        0.3460000000         3.9970000000
H                  0.0550000000       -0.1510000000         2.8890000000
*
```

There are few points that have to be raised here. Because the keyword `qmmm/pdynamo` was specified in the `header` variable, the pDynamo library will automatically add the `constraint` block in the ORCA input, which will freeze the link atoms and the QM atoms to which they are bound. It will also generate the `chignolin.lj` file containing all the Lennard-Jones parameters. The important parts of this file, which is somewhat different than the one generated by Gromacs, are listed next.

```
# number of atoms  combination rule
138 0
#      x           y           z          sigma       epsilon   id
    -6.778000   -1.424000    4.200000    3.250000    0.711280   -1
    -6.878000   -0.708000    2.896000    3.500000    0.276144   -1
    -5.557000   -0.840000    2.138000    3.750000    0.439320   -1
...
     0.433000    0.826000    0.502000    2.960000    0.878640   -1
    -0.523000    0.687000    3.249000    3.500000    0.276144    1
     0.418000    1.724000    3.866000    3.750000    0.439320    2
    -0.069000    2.762000    4.283000    2.960000    0.878640    3
     1.609000    1.463000    3.911000    2.960000    0.878640    4
    -2.259000   -0.588000    1.846000    0.000000    0.000000   -1
    -1.795000    2.207000    2.427000    2.500000    0.125520   -1
    -1.224000    0.346000    3.997000    2.500000    0.125520    5
     0.055000   -0.151000    2.889000    2.500000    0.125520    6
    -0.311000    2.922000    0.557000    3.250000    0.711280   -1
...
    -1.387000   -2.946000    5.106000    2.500000    0.125520   -1
# number of special pairs
22
#        atom1       atom2    factor
          34          32     0.000000
          35          39     0.500000
          40          31     0.000000
          41          30     0.500000
          41          32     0.500000
```

```
           36            31        0.500000
           40            32        0.500000
           40            39        0.500000
           34            31        0.000000
           35            30        0.500000
           34            11        0.500000
           34            38        0.500000
           41            31        0.000000
           37            31        0.500000
           34            33        0.500000
           34            39        0.000000
           40            30        0.500000
           41            39        0.500000
           34            30        0.000000
           35            31        0.000000
           34            42        0.500000
           35            32        0.500000
```

The second number on the first line refers to the type of combination rule used to calculate the Lennard-Jones interaction. It is 0 if a geometric average is used (OPLS force field), or 1 for the Lorentz-Bertelot rules (AMBER force field). The `id` on the last column is -1 for MM atoms and is equal to the atom number for the QM atoms. In this case the hydrogen link atom is atom 0. The last block of the file is composed of atom pairs and a special factor by which their Lennard-Jones interaction is scaled. In general this factor is equal to 1, but for atoms one or two bonds apart is zero, while for atoms three bonds apart depends on the type of force field, and in this case is 0.5.

After successful completion of the ORCA optimization run, the information will be relayed back the pDynamo library, which will report the total QM/MM energy of the system. At this point the type QM/MM of calculation is limited only by the capabilities of the pDynamo library, which are quite extensive.

6 Detailed Documentation

6.1 More on Coordinate Input

We will now enter the detailed discussion of the features of ORCA. Note that some examples are still written in the 'old syntax' but that there is no need for the user to adopt that old syntax. The new one works as well.

6.1.1 Fragment Specification

The atoms in the molecule can be assigned to certain *fragments*. Presently this doesn't do much but to help to organize the output in the population analysis section (and can therefore be very useful). To assign a given atom to a given fragment put a **(n)** directly after the atomic symbol. Fragment enumeration starts with fragment 1!

```
%coords
 CTyp   xyz  # the type of coordinates xyz or internal
 Charge -2   # the total charge of the molecule
 Mult   2    # the multiplicity = 2S+1
 coords
    Cu(1) 0    0    0
    Cl(2) 2.25 0    0
    Cl(2) -2.25 0    0
    Cl(2) 0    2.25 0
    Cl(2) 0    -2.25 0
 end
end
```

In this example the fragment feature is used to divide the molecule into a 'metal' and a 'ligand' fragment and consequently the program will print the metal and ligand characters contained in each MO in the population analysis section.

6.1.2 Defining Geometry Parameters and Scanning Potential Energy Surfaces

ORCA lets you define the coordinates of all atoms as functions of user defined geometry parameters. By giving not only a value but a range of values (or a list of values) to this parameters potential energy surfaces can be scanned. In this case the variable **RunTyp** is automatically changed to **Scan**. The format for the parameter specification is straightforward:

```
    %coords
     CTyp    internal
     Charge 0
     Mult    1
     pardef
      rCH  = 1.09;    # a C-H distance
      ACOH = 120.0;   # a C-O-H angle
      rCO  = 1.35, 1.10, 26; # a C-O distance that will be scanned
     end
     coords
        C  0  0  0   0      0          0
        O  1  0  0  {rCO}   0          0
        H  1  2  0  {rCH}  {ACOH}      0
        H  1  2  3  {rCH}  {ACOH}    180
     end
    end
```

In the example above the geometry of formaldehyde is defined in internal coordinates (the geometry functions work exactly the same way with cartesian coordinates). Each geometrical parameter can be assigned as a function of the geometry parameters by enclosing the function braces, "{}". For example a function may look like {0.5*cos(Theta)*rML+R}. Note that all trigonometric functions expect their arguments to be in degrees and not radians. The geometry parameters are expected to be defined such that the lengths come out in Ångströms and the angles in degrees. *After* evaluating the functions the coordinates will be converted to atomic units. In the example above the variable rCO was defined as a 'Scan parameter'. Its value will be changed in 26 steps from 1.3 down to 1.1 and at each point a single point calculation will be done. At the end of the run the program will summarize the total energy at each point. This information can then be copied into the spreadsheet of a graphics program and the potential energy surface can be plotted. Up to three parameters can be scan parameters. In this way grids or cubes of energy (or property) values as a function of geometry can be constructed.

If you want to define a parameter at a series of values rather than evenly spaced intervals, the following syntax is to be used:

```
    %coords
     CTyp    internal
     Charge 0
     Mult    1
     pardef
      rCH = 1.09;    # a C-H distance
      ACOH= 120.0;  # a C-O-H angle
      rCO [1.3 1.25 1.22 1.20 1.18 1.15 1.10];  # a C-O distance that will be scanned
     end
     coords
        C  0  0  0   0      0          0
        O  1  0  0  {rCO}   0          0
        H  1  2  0  {rCH}  {ACOH}      0
```

```
     H  1  2  3  {rCH}  {ACOH}  180
   end
 end
```

In this example the C-O distance is changed in seven non-equidistant steps. This can be used in order to provide more points close to a minimum or maximum and fewer points at less interesting parts of the surface.

A special feature has also been implemented into ORCA - the parameters themselves can be made functions of the other parameters as in the following (nonsense) example:

```
%coords
 CTyp   internal
 Charge 0
 Mult   1
 pardef
  rCOHalf= 0.6;
  rCO    = { 2.0*rCOHalf };
 end
 coords
    C  0  0  0   0     0          0
    O  1  0  0  {rCO}  0          0
    O  1  0  0  {rCO}  180        0
 end
end
```

In this example the parameter `rCO` is computed from the parameter `rCOHalf`. In general the geometry is computed (assuming a `Scan` calculation) by (a) incrementing the value of the parameter to be scanned, (b) evaluating the functions that assign values to parameters and (c) evaluating functions that assign values to geometrical variables. Although it is not mandatory it is good practice to *first* define the static or scan-parameters and then define the parameters that are functions of these parameters.

Finally ORCA has some special features that may help to reduce the computational effort for surface scans:

```
%method
 SwitchToSOSCF true
                   # switches the converger to SOSCF
                   # after the first point. SOSCF may
                   # converge better than DIIS if the
                   # starting orbitals are good.
                   # default = false
 ReducePrint true # reduce printout after the first
                   # point default=true
 # The initial guess can be changed after the first
 # point. The default is MORead. The MO's of the pre-
 # vious point will in many cases be a very good guess
```

```
        # for the next point. In some cases however, you may
        # want to be more conservative and use a general guess.
        ScanGuess   OneElec  # the one electron matrix
                    Hueckel  # the extended Hueckel guess
                    PAtom    # the PAtom guess
                    PModel   # the PAtom guess
                    MORead   # MO's of the prev. point
        end
```

NOTE:

- You can scan along normal modes of a hessian using the `NMScan` feature as described in section 6.14.8.9.

- The surface scan options are also supported in conjunction with TD-DFT/CIS or MR-CI calculations (see section 6.14.8.7).

6.1.3 Inclusion of Point Charges

In some situations it is desirable to add point charges to the system. In ORCA there are two mechanisms to add point-charges. If you only want to add a few point charges you can "mask" them as atoms as in the following (nonsense) input:

```
    # A water dimer
    ! BP86 def2-SVP

    * xyz 0 1
    O           1.4190    0.0000    0.0597
    H           1.6119    0.0000   -0.8763
    H           0.4450    0.0000    0.0898
    Q  -0.834  -1.3130    0.0000   -0.0310
    Q   0.417  -1.8700    0.7570    0.1651
    Q   0.417  -1.8700   -0.7570    0.1651
    *
```

Here the "Q"'s define the atoms as point charges. The next four numbers are the magnitude of the point charge and its position. The program will then treat the point charges as atoms with no basis functions and nuclear charges equal to the Q'values.

If you have thousands of point charges to treat, as in a QM/MM calculation, it is more convenient, and actually necessary, to read the point charges from an external file as in the following example:

```
    # A water dimer
    ! BP86 def2-SVP

    % pointcharges "pointcharges.pc"
```

```
   * xyz 0 1
   O    1.4190    0.0000    0.0597
   H    1.6119    0.0000   -0.8763
   H    0.4450    0.0000    0.0898
   *
```

The program will now read the file "pointcharges.pc" that contains the point-charge information and then call the module **orca_pc** which adds the point charge contribution to the one-electron matrix and the nuclear repulsion. The file "pointcharges.pc" is a simple ASCII file in the following format:

```
   3
-0.834   -1.3130    0.0000   -0.0310
 0.417   -1.8700    0.7570    0.1651
 0.417   -1.8700   -0.7570    0.1651
```

The first line gives the number of point charges. Each conseqcutive line gives the magnitude of the point charge (in atomic units) and its position (in Angström units!).

ORCA can also correctly do QM/MM calculations with external point charges for semi-empirical calculations. In this case, the point-charge-atom interactions are estimated as in the semi-empirical methods themselves, namely using the ZDO approximation and employing the appropriate GAMMA-integrals that are also used in the ZDO Hamiltonian itself.

6.2 Choice of Computational Model

6.2.1 Features Common to All Calculations

The computational model is specified in the block **%method**. The following choices exist:

```
   %method
     Method  HFGTO # Hartree-Fock with GTOs (synonym HF)
             DFGTO # Density Functional with GTOs (syn. DFT)
             MP2   # Second order Moeller-Plesset
             CNDO  # complete neglect of differential overlap
             INDO  # intermediate neglect of d. o.
             NDDO  # neglect of diatomic d. o.
       end
```

In the case of Hartree-Fock calculations [96] nothing else is required in this block. Density functional calculations [97,98] need slightly more attention.

The **RunType** (=type of calculation to be performed) is chosen as follows:

```
    %method
      RunTyp  Energy   # single point calc. (default)
              Gradient # single point energy and gradient
              Opt      # Geometry optimization
              MD       # Molecular dynamics
              Scan     # scan of geometric parameters
      end
```

You can tell the main program the explicit names and positions of the other modules. In this way you could in principle also interface your own programs to ORCA as long as they respect the input/output conventions used in ORCA (which are, however, reasonably complicated).

```
    %method
      #*** the name of the SCF program
      ProgSCF    "MySCFProg.exe"
      #*** the name of the GTO integral program
      ProgGTOInt "MyGTOIntProg.exe"
      #**** the name of the MP2 module
      ProgMP2    "MyProgMP2.exe"
      #*** the name of the plot program
      ProgPlot   "MyPlotProgram.exe"
      #*** the name of the SCF gradient program
      ProgSCFGrad "MySCFGradientProg.exe"
      #*** the name of the geometry relaxation program
      ProgGStep  "MyProgGStep.exe"
      #*** the name of the molecular dynamics program
      ProgMD     "MyProgMD.exe"
      #*** the name of the moment integral program
      ProgMom    "MyProgMom.exe"
      # *** the name of the EPR/NMR module
      ProgEPRNMR   "MyProgEPRNMR.exe"
      #*** the name of the CP-SCF program
      ProgCPSCF   "MyProgCPSCF.exe"
      # *** the name of the CI-singles and TD-DFT module
      ProgCIS    "MyProgCIS.exe"
      # *** the name of the Relativistics module
      ProgREL    "MyProgREL.exe"
      end
```

For example, if the executables are all located in the same run directory (and the PATH variable contains this directory!) use:

```
    %method
      ProgSCF    "orca_scf.exe"
```

```
    ProgGTOInt  "orca_gtoint.exe"
    ProgMP2     "orca_mp2.exe"
    ProgPlot    "orca_plot.exe"
    ProgSCFGrad "orca_scfgrad.exe"
    ProgGStep   "orca_gstep.exe"
    ProgMD      "orca_md.exe"
    ProgMom     "orca_mom.exe"
    ProgCPSCF   "orca_cpscf.exe"
    ProgEPRNMR  "orca_eprnmr.exe"
    ProgCIS     "orca_cis.exe"
    ProgRel     "orca_rel.exe"
    ProgMDCI    "orca_mdci.exe"
    end
```

6.2.2 Density Functional Calculations

6.2.2.1 Choice of Functional

Basic Choice of Density Functional. If you are doing a DFT calculation [97,98], the following choices for local and gradient corrected density functionals are available:

```
[METHOD]
  # Choices for 'Functional'. References to the individual
  # exchange and correlation parts of the functionals are
  # given below
  Functional
            #****************************************
            # Local functionals
            #****************************************
             HFS    # Hartree-Fock Slater
             XAlpha # The famous old Slater Xa theory
             LSD    # Local spin density (VWN-5A form)
             VWN5   # Local spin density (VWN-5)
             VWN3   # Local spin density (VWN-3)
             PWLDA  # Local spin density (PW-LDA)
            #****************************************
            # 'Pure' GGA  functionals
            #****************************************
             BNULL  # Becke '88 exchange, no corr.
             BVWN   # Becke '88 exchane, VWN-5 corr.
             BP     # Becke X-Perdew 86 correlation
             PW91   # Perdew-Wang GGA-II '91 func.
             mPWPW  # Modified PW with PW correlation
             mPWLYP # same with LYP correlation
```

```
           BLYP   # Becke X with LYP correlation
           GP     # Gill '96 X, Perdew '86 corr.
           GLYP   # Gill '96 X with LYP correlation
           PBE    # Perdew-Burke-Ernzerhof
           revPBE # Revised PBE (exchange scaling)
           RPBE   # Revised PBE (functional form of X)
           PWP    # PW91 exchange + P86 correlation
           OLYP   # the optimized exchange and LYP
           OPBE   # the optimized exchange and PBE
           XLYP   # the Xu/Goddard exchange and LYP
           B97-D  # Grimme's GGA including D2 dispersion correction
           B97-D3 # Grimme's GGA including D3 dispersion correction
        #**************************************
        # Meta GGA  functionals
        #**************************************
          TPSS   # the TPPS functional
        #**************************************
        # Hybrid  functionals
        #**************************************
          B1LYP       # One parameter Hybrid of BLYP
          B3LYP       # Three parameter Hybrid of BLYP
          B1P         # Analogous with Perdew exchange
          B3P         # Analogous with Perdew exchange
          G1LYP       # 1 par. analog with Gill 96 X
          G3LYP       # 3 par. analog with Gill 96 X
          G1P         # similar with P correlation
          G3P         # similar with P correlation
          PBE0        # 1 parameter version of PBE
          PWP1        # 1 parameter version of PWP
          mPW1PW      # 1 parameter version of mPWPW
          mPW1LYP     # 2 parameter version of mPWLYP
          PW91_0      # 1 parameter version of PW91
          O3LYP       # 3 parameter version of OLYP
          X3LYP       # 3 parameter version of XLYP
          PW6B95      # Hybrid functional by Truhlar [93]
          B97         # Becke's original hybrid
          BHANDHLYP   # Half-and-half Becke hybrid functional
        #**************************************
        # Meta Hybrid  functionals
        #**************************************
          TPSSh # hybrid version of TPSS with 10% HF exchange
          TPSS0 # hybrid version of TPSS with 25% HF exchange
        #**************************************
        # Double Hybrid  functionals (mix in MP2)
        #**************************************
```

```
              B2PLYP   # Grimme's 2006 double hybrid
              mPW2PLYP # Schwabe/Grimme improved double hybrid
      PWPB95   # New Grimme double hybrid [94]
        end
```

Note that `Functional` is a *compound key*. It does nothing else but to choose specific values for the variables `Exchange`, `Correlation` and `ACM` described below. You can explicitly give these variables instead or in addition to `Functional`. However, make sure that you specify these variables *after* you have assigned a value to `Functional` or the values of `Exchange`, `Correlation` and `ACM` will be reset to the values chosen by `Functional`.

Empirical Parameters in Density Functionals. Some of the functionals incorporate empirical parameters that can be changed to improve agreement with experiment. In ORCA there is some freedom to change functional parameters. Currently there are three parameters that can be changed (other than the parameters used in the hybrid functionals). They are all part of exchange functionals. The first of these parameters is α of Slaters X_α method. Theoretically it has a value of 2/3 and this is used in the HFS and LSD functionals. However, exchange is about 10% underestimated by this approximation (a very large number!) and a value around 0.70-0.75 seems to be better for molecules. The second parameter is the parameter β for Becke's gradient corrected exchange functional. Becke has determined the value 0.0042 by fitting the exchange energies for rare gas atoms. There is some evidence that with smaller basis sets for molecules a slightly smaller value such as 0.0039 gives improved results. The final parameter is the value κ occuring in the `PBE` exchange functional. It has been given the value 0.804 by Perdew et al. in order to satisfy the Lieb-Oxford bound. Subsequently other workers have argued that a larger value for this parameter (around 1.2) gives better energetics and this is explored in the `revPBE` functional. Note that it also has been shown that while `revPBE` gives slightly better energetics it also gives slightly poorer geometries.

```
    %method
      XAlpha  0.75   # Slater's alpha parameter (default 2/3)
      XBeta   0.0039 # Becke's beta parameter
                     # (default 0.0042)
      XKappa  0.804  # PBE kappa parameter (default 0.804)
      end
```

Specifying Exchange and Correlation approximations individually. The following variables are available for specifying the exchange and correlation approximations individually and to construct user defined hybrid or "extended" hybrid functionals:

```
    %method
      Exchange    X_NOX       # no exchange
                  X_SLATER    # Slaters local exchange [18]
                  X_B88       # Becke 88 gradient exchange [19]
                  X_G96       # Gill 96 gradient exchange [20]
                  X_PW91      # Perdew-Wang 91 gradient X [21]
                  X_mPW       # Adamo-Barone modification of PW
                                  (JCP 1997, 108, 664)
                  X_PBE       # PBE exchange [22,24]
                  X_RPBE      # Revised PBE [23]
```

```
                 X_OPTX       # optimized exchange
                                 (CPL 2001, 341 319-328)
                 X_X          # Xu/Goddard PNAS, 2004, 101, 2673-2677
                 X_TPSS       # TPSS meta GGA exchange
                 X_B97D       # Grimme's modified exchange for the B97-D GGA
                              # (JCC 2006, 27, 1787)
                 X_B97BECKE   # Becke's original exchange for the B97 hybrid
                              # (JCP 1997, 107, 8554)
     Correlation C_NOC        # no correlation
                 C_VWN5       # Local VWN-V parameters [25]
                 C_VWN3       # Local VWN-III parameters [25]
                 C_PWLDA      # Local PW '91 [26]
                 C_P86        # Perdew '86 correlation [27]
                 C_PW91       # Perdew-Wang '91 correlation [27]
                 C_PBE        # PBE correlation [22]
                 C_LYP        # LYP correlation [28]
                 C_TPSS       # TPSS meta-GGA correlation
                 C_B97D       # Grimme's modified correlation for the B97-D GGA
                              # (JCC 2006, 27, 1787)
                 C_B97BECKE   # Becke's original correlation for the B97 hybrid
                              # (JCP 1997, 107, 8554)
     # for hybrid functionals. Reference, Becke [29]
     ACM   ACM-A, ACM-B, ACM-C
           # ACM-A: fraction of HF-exchange in hybrid DFT
           # ACM-B: scaling of GGA part of DFT exchange
           # ACM-C: scaling of GGA part of DFT correlation
     # "extended" hybrid functional
     ScalLDAC 1.0            # scaling of the LDA correlation part
     ScalMP2C 0.0            # fraction of MP2 correlation mixed into
                            # the density functional
     end
```

Hybrid Density Functionals. The hybrid DFT's [99, 100] are invoked by choosing a nonzero value for the variable ACM. (ACM stands for 'adiabatic connection model'). Specifically, these functionals have the following form:

$$E_{XC} = aE_{HF}^X + (1-a) E_{LSD}^X + bE_{GGA}^X + E_{LSD}^C + cE_{GGA}^C \tag{6.1}$$

Here, E_{XC} is the total exchange/correlation energy, E_{HF}^X is the Hartree-Fock exchange, E_{LSD}^X is the local (Slater) exchange, E_{GGA}^X is the gradient correction to the exchange, E_{LSD}^C is the local part of the correlation energy and E_{GGA}^C is the gradient correction to the correlation energy. This brings use to a little bit awkward subject: several hybrid functionals with the same name give different values in different programs. The reason for this is that they either choose slightly different default values for the parameters a,b and c and or they differ in the way they treat the local part of the correlation energy. Different parameterizations exist. The most popular is due to Vosko, Wilk and Nusair (VWN, [101]). However, VWN in their classic paper give two sets of parameters - one in the main body (parameterization of RPA results; known as VWN-III) and one in their table 5 of correlation energies (parameterization of the Ceperly/Alder Monte Carlo results; known as VWN-V). Some programs choose one set, others the other. In

addition a slightly better fit to the uniform electron gas has been produced by Perdew and Wang [102]. The results from this fit are very similar to what the parameters `VWN5` produce (the fit to the Ceperly Alder results) whereas `VWN3` (fit to the RPA results) produces quite different values. To be short - in ORCA almost all functionals choose `PWLDA` as the underlying LDA functional. A special situation arises if `LYP` is the correlation functional [103]. `LYP` itself is *not* a correction to the correlation but includes the full correlation. It is therefore used in the `B3LYP` method as:

$$E_{B3LYP}^C = E_{LSD}^C + c \left(E_{LYP}^C - E_{LSD}^C \right) \tag{6.2}$$

In ORCA `VWN5` is chosen for the local correlation part. This choice is consistent with the TurboMole program [104–106] but not with the Gaussian program [107]. However, the user has full control over this point. You can choose the underlying local part of any correlation functional with the variable `LDAOpt`:

```
%method
   LDAOpt  C_PWLDA
           C_VWN5
           C_VWN3
   end
```

Specifying `C_VWN3` for `LDAOpt` together with `Functional=B3LYP` should give results very close to the B3LYP functional as implemented in the Gaussian series of programs[1].

In particular for the popular B3LYP functional the following aliases are defined in order to facilitate comparisons with other major electronic structure packages:

```
%method
   Functional B3LYP     # consistent with TurboMole
              B3LYP_TM  # consistent with TurboMole
                        # = Functional= B3LYP
                        #   LDAOpt    = C_VWN5;
              B3LYP_G   # consistent with Gaussian
                        # = Functional= B3LYP
                        #   LDAOpt    = C_VWN3;
   end
```

One Parameter Hybrid Density Functionals. A few words on the one parameter hybrid methods appears in order. Through the underlying LDA dependence of the three parameter hybrids different programs give different answers because they differ in the underlying LDA. On the other hand, it has recently been argued from theoretical reasoning that the optimal mixing ratio for DFT and HF exchange is 0.25 [109]. Furthermore numerical calculations have shown that the results of using this fixed ratio and *not* scaling the GGA correlation or exchange are as good as the original three parameter hybrids [110]. I personally sympathize with these ideas because they are based on theory and they remove some arbitrariness from the hybrid procedures. Also the slightly higher HF-exchange (0.25 in favor of 0.20 used in the original three parameter hybrids) is, I believe, in the right direction. Thus the one parameter hybrids have the simple form:

---

[1]There is some evidence that the version used in the Gaussian program gives miniscule better results in molecular applications then the TurboMole variant but the differences are very small [108]

$$E_{XC} = E_X^{DFT} + a' \left(E_X^{HF} - E_X^{DFT} \right) + E_C^{DFT} \tag{6.3}$$

with $a^t = \frac{1}{4}$ which is the same as putting: $a = a^t$, $b = 1 - a^t$ and $c = 1$ in the three parameter hybrids and this is how it is implemented. The one parameter hybrid `PBE0` has been advertised as a hybrid functional of overall well balanced accuracy [111].

Extended 'double' hybrid functionals. In addition to mixing the HF-exchange into a given DF, Grimme (Grimme, S.; JCP (2006), 124, 034108) has proposed to mix in a fraction of the MP2 correlation energy calculated with hybrid DFT orbitals. Such functionals may be refered to as "extended" hybrid functionals. Grimme's expression is:

$$E_{XC} = a E_X^{HF} + (1 - a) E_X^{DFT} + (1 - c) E_C^{DFT} + c E_C^{MP2} \tag{6.4}$$

Such functionals can be user-defined in ORCA as follows:

```
%method
    ScalHFX  = a
    ScalDFX  = 1-a
    ScalGGAC = 1-c
    ScalLDAC = 1-c
    ScalMP2C = c
    end
```

Grimme recommends the B88 exchange functional, the LYP correlation functional and the parameters a=0.53 and c=0.27. This gives the B2PLYP functional which appears to be a fair bit better than B3LYP based on Grimme's detailed evaluation study.

Presently, this methodology covers single points and analytic gradients (hence all forms of geometry optimization, relaxed scans, transition state searches and numerical frequencies are supported). Note that you need to choose `%mp2 density relaxed end` in order to get the correct response density which is consistent with first order properties as analytic derivatives. By default this density is not calculated since its construction adds significant overhead to the calculation. Therefore you have to specifically request it if you want to look at the consistent density. You can also choose `%mp2 density unrelaxed end` which would give you the unrelaxed (expectation value like) density of the method at considerably less computational cost. However, this is not recommended since the changes to the relaxed density are considerable in our experience and the unrelaxed density has a much weaker theoretical status than its relaxed counterpart.

Range-separated hybrid functionals. ORCA supports functionals based on the error function splitting of the two-electron operator used for exchange as first realized by Hirao and coworkers [76]:

$$r_{12}^{-1} = \underbrace{\text{erfc}(\mu \cdot r_{12}) \cdot r_{12}^{-1}}_{\text{SR}} + \underbrace{\text{erf}(\mu \cdot r_{12}) \cdot r_{12}^{-1}}_{\text{LR}} \tag{6.5}$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$ and $\text{erfc}(x) = 1 - \text{erf}(x)$. Note that the splitting is only applied to exchange; one-electron parts of the Hamiltonian, the electron-electron Coulomb interaction and the approximation for the DFT correlation are not affected. Later, Handy and coworkers generalized the ansatz to: [77]

$$r_{12}^{-1} = \underbrace{\frac{1 - [\alpha + \beta \cdot \text{erf}(\mu \cdot r_{12})]}{r_{12}^{-1}}}_{\text{SR}} + \underbrace{\frac{\alpha + \beta \cdot \text{erf}(\mu \cdot r_{12})}{r_{12}^{-1}}}_{\text{LR}} \tag{6.6}$$

The splitting has been described in graphical form (according to Handy and coworkers), along the terminology of
ORCA, in Figure 6.1:
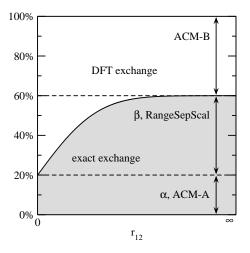
Figure 6.1: Graphical description of the Range-Separation ansatz. The gray area corresponds to
Hartree-Fock exchange. α and β follow Handy's terminology. [77]

The splitting has been used to define the ωB97 family of functionals where the short-range part (SR) is described by
DFT exchange and the long-range part by exact exchange/Hartree-Fock exchange. The same is true for CAM-B3LYP
and LC-BLYP. It is possible to use a fixed amount of Hartree-Fock exchange (EXX) and/or a fixed amout of DFT
exchange in this ansatz.

Functional	Keyword	fixed EXX	variable part	μ/bohr^{-1}	fixed DFT-X	Reference
ωB97	WB97	—	100%	0.40	—	[79]
ωB97X	WB97X	15.7706%	84.2294%	0.30	—	[79]
ωB97X-D3	WB97X-D3	19.5728%	80.4272%	0.25	—	[80]
CAM-B3LYP	CAM-B3LYP	19%	46%	0.33	35%	[77]
LC-BLYP	LC-BLYP	—	100%	0.33	—	[78]

The currently available speed-up options RIJONX and RIJCOSX. Otherwise, integral-direct single-point calculations
and TDDFT are the only supported job types thus far. In principle, it is possible to change the amount of fixed
Hartree-Fock exchange (ACM-A) and the amount of variable exchange (RangeSepScal) and μ, though this is not
recommended. The amount of fixed DFT Exchange (ACM-B) can only be changed for CAM-B3LYP and LC-BLYP. In
other words, ACM-B is ignored by the ωB97 approaches, because no corresponding μ-independent exchange functional
has been defined.

```
   ! RKS CAM-B3LYP SVP RIJONX def2-SVP/J

%method
  RangeSepEXX  True  # must be set
  RangeSepMu   0.25  # should not be set to 0.0 or below
  RangeSepScal 0.7   # should sum to 1 with ACM-A and ACM-B
  ACM 0.2, 0.1, 1.0  # ACM-A, ACM-B, ACM-C
end
```

```
* xyz 0 1
H 0.0 0.0 0.0
H 0.0 0.0 0.7
*
```

Note: For information on the ACM formalism, see preceding section called "Specifying Exchange and Correlation approximations individually". While it is technically possible to choose an exchange functional that has no μ-dependence, this makes conceptually no sense.

6.2.2.2 Choice of Integration Grid

The next subject of importance for the DFT calculation is the size of the integration grid for numerical integration. Three-dimensional numerical quadrature is necessary due to the complicated analytical form of the exchange-correlation potential (and energy). The expressions for these quantities are so complicated that there is no hope of finding analytical solutions to the required integrals and some numerical approximation, commonly three-dimensional numerical integration, is necessary. The numerical integration [112–117] is a major step in any DFT calculation and if the RI approximation [115–121] is used for the Coulomb part it usually dominates the total wall clock time.

Accuracy. The good news about the numerical integration is that the effort only increases approximately linearly with the molecular size such as to make calculations on relatively large molecules not prohibitive [114, 122–124]. The bad news about numerical integration is that it is not trivial to make it highly accurate. Especially if you choose relatively small grids it is advisable to check that the results are converged (say to at least ≈1 mEh) by reconverging the calculation with a larger grid (`Grid`) and higher radial integral accuracy (`IntAcc`). As a rule of thumb the error in the numerically integrated total electron density is on the same order of magnitude as the error in the numerically integrated XC energy [113][2]. Therefore the program prints this information each iteration (`Num. Int. El. : ...`).

Rotational Invariance. Through the very fact that a grid is used, the total energy depends on the molecular orientation which is of course unphysical [125, 126]. Unless the integration grid is large enough to make the dependence negligible it is therefore not a good idea to calculate closely related molecules in grossly different orientations. This unpleasant feature arises because ORCA does not yet have a 'standard orientation' for molecules like other programs. Note that other programs also have that rotational variance problem but they hide it through employment of a standard orientation. Grid free methods to integrate the XC part are known [127–129] and they remove these problems. However, they will probably not be implemented in the near future into ORCA because they have a less favorable computational scaling than the numerical integration, requiring a large additional $O(N^3)$ diagonalization at each iteration.

Choice of Grid. Several default grids are available in ORCA. They are chosen by specifying the variable `Grid`.

```
%method
   Grid 0  # product grid
        1  # Lebedev 50  (not recommended)
        2  # Lebedev 110 points (default for SCF iterations)
        3  # Lebedev 194 points (more accurate)
        4  # Lebedev 302 points (default for FinalGrid)
```

---

[2]Actually, the XC energy and potential are somewhat smoother than the density itself leading to the expectation that the error in the density is a pessimistic estimate of the error in the XC energy.

```
        5  # Lebedev 434 points (large)
        6  # Lebedev 590 points (larger)
        7  # Lebedev 770 points (very large)
    end
```

Like `Functional`, `Grid` is merely a compound variable that affects several other variables that control the details of grid design. Any of these variables can be specified *after* `Grid` in order to change the default values. In general the grid consists of a series of radial shells around each atom and an angular grid for each radial shell.

Specifying Grid parameters Individually. A particular radial integration method[3] is determined by specifying `RadialGrid`. In my hands, the default, `GaussChebyshev` [113, 130], is the most efficient and it is not recommended to change to `EulerMcLaurin` [114, 122–124] integration. The number of radial points is determined by the parameter `IntAcc`. From `IntAcc` the number of radial shells for a given atom is computed as (taken from Krack and Köster [130] $\varepsilon = 10^{-IntAcc}$):

$$n_{rad}(A) = -5\left(3\log\varepsilon - n_A + 8\right) \tag{6.7}$$

where n_A is the row of the periodic table of element A. The defaults for `IntAcc` range from $\approx$3.7 to $\approx$5.5.[4] It is not recommended to go below 3.5. `AngularGrid` specifies the *largest* angular grid to be used (see below).

If `SimpleGrid` is chosen a product integration is performed over the angular variables with a Gauss-Legendre integration for θ and equally spaced points for φ [114, 122–124]. This integration is useful because it is open ended - the number of points can be increased *ad infinitum*. It is however, difficult to imagine situations where the largest Lebedev grid is still too small. The Lebedev grids [131–137] are generally more efficient (theoretically by a factor 1.5). For the `SimpleGrid` the parameter `NThetaMax` is the number to multiply n_{rad} with to obtain the maximum number of θ points for a given atom. The number of φ points is automatically chosen for a given θ based on a recommendation by Treutler and Ahlrichs and avoids crowding of points near the poles [113].

Finally **BFCut** and **WeightCut** are thresholds. If a gridpoint has a weight smaller than `WeightCut` it is simply neglected (Becke scheme). If the weight scheme is chosen according to `Weight_AtomicXC`, the grid point is also neglected if it gives a contribution of <WeightCut to the promolecular density which is constructed from a superposition of spherically symmetric atomic densities. `BFCut` determines when to neglect the contribution of a given basis function to a given gridpoint or the contribution of a given basis function product to the total density at a given gridpoint. There was extensive experimentation with these values and they should probably not be changed. In particular, the results may be quite sensitive to `WeightCut` (larger values are not recommended) but should be quite stable with `BFCut` (smaller values should not change the energy appreciably but may significantly raise the computation time).

```
    %method
      RadialGrid    GaussChebyshev  # (default)
                    EulerMcLaurin
      AngularGrid   Lebedev50  # =1
                    Lebedev110 # =2
                    Lebedev194 # =3 (default)
```

---

[3]For the experts - ORCA always uses the mapping M3 and also the atomic size adjustments of Treutler and Ahlrichs in the radial integration.

[4]In general it is usual the angular integration that limits the accuracy. Thus it is a questionable strategy to choose a small angular grid and then increase `IntAcc` to a large value.

```
                    Lebedev302 # =4
                    Lebedev434 # =5
                    Lebedev590 # =6
                    Lebedev770 # =7
                    SimpleGrid # =0
    IntAcc        5.0  # determines no. of radial points
    NThetaMax     0.7  # only for AngularGrid=0
    GridPruning   0    # no Pruning
                  1    # Grid pruning algorithm 1
                  2    # Grid pruning algorithm 2 (no longer there)
                  3    # Grid pruning algorithm 3 (default)
                  4    # Adaptive grid pruning (no longer there)
    WeightScheme Weight_Becke # The Becke weight scheme
                 Weight_AtomicXC # default. Choose weights from
                              # superposition of atomic
                              # exchange densities. Good choice
                              # for DFT!
    HGridReduced true  # Reduce grids for H and He by one
                       # unit (recommended)
    BFCut         1e-10 # basis fcn. cut. Is adjusted according to
                        # convergence tolerances
    WeightCut     1e-14 # grid weight cut. default: 1e-14
    end
```

Grid pruning. Like most other programs ORCA 'prunes' the angular grids [114, 122–124]. It is well known that the radial shells close to the nucleus need less angular grid points to achieve high accuracy while those in the bonding region need larger angular grids. Taking this into account one can save large numbers of grid points (and therefore computation time) by using smaller angular grids close to the nuclei. There was some experimentation with this feature and the defaults should probably not be changed.

- **GridPruning**=3 Here the radial integration is divided into five domains that defined through the numbers γ_1 to γ_4 [114, 122–124]. There are spheres defined by $\gamma_i R_{max}$, where $R_{\max}$ is Clementi's radius of the outermost valence orbital [138]. The angular grids are used in order n$_g$-3, n$_g$-2, n$_g$-1, n$_g$, n$_g$-1 where n$_g$ is the size of the largest angular grid. For example for **AngularGrid**=4 the Lebedev grids with 50, 110, 194, 302, 194 points are used in the domains one to five. In my experience this leads to significant reduction in the number of points as well as small errors (on the order of some μE_h) relative to the unpruned grids of the same size.

Default Grids Having specified the individual parameters, the default grids are characterized by the following settings:

Grid[c]	AngularGrid	IntAcc	H,He[d]	Li-Ne[d]	Na-Ar[d]	K-Kr[d]
0	SimpleGrid	4.34[a]	30	35	40	45
1	Lebedev50	4.34	30	35	40	45
2	Lebedev110	4.34	30	35	40	45
3	Lebedev194	4.34	30	35	40	45
4	Lebedev302	4.67	30	35	40	45
5	Lebedev434	5.01	35	40	45	50

| 6 | Lebedev590 | 5.34 | 40 | 45 | 50 | 55 |
| 7 | Lebedev770 | 5.67 | 45 | 50 | 55 | 60 |

[a] NThetaMax=0.4 for Grid 0.

[b] The choice of IntAcc was such that there is no further improvement in the accuracy with more radial points

[c] for all grids GridPruning=3, RadialGrid=GaussChebyshev, HGridReduce=true (except grid 0)

[d] Number of radial points

If no value for `Grid` is specified in the input, `Grid`=2 is used for the SCF iterations and `Grid`=4 for the final energy (*vide infra*).

Multigrid feature. The TurboMole developers [104–106] have advocated a method where the SCF iterations are done with a small grid and gradients and final energies are evaluated on a larger, more accurate grid [113]. Overall this gives a substantial improvement in speed with no significant loss in accuracy as long as the smaller grid for the SCF iterations is large enough such that it does not produce significant errors in the final density.

Using the multigrid feature in ORCA is essentially very easy:

```
%method
   FinalGrid 4   # grid to use for the final energy
                 # evaluation
   end
```

The grid parameters of the final grid can also be individually specified:

```
%method
    RadialGrid_fin    GaussChebyshev
    AngularGrid_fin   Lebedev434
    IntAcc_fin        5.0
    NThetaMax_fin     0.7
    GridPruning_fin   0
    WeightScheme_fin  Weight_Becke
    HGridReduced_fin  true
    end
```

By default the multigrid feature *is* used and the final grid defaults to grid 4. If you do not want to use the multigrid feature you have to specify:

```
%method
   UseFinalGrid false
   End
Or
! NoFinalGrid
```

CAUTION:

- if you use a large grid (i.e. `Grid`=6) and forget to turn off the multigrid feature the energies you get are only as accurate as `Grid 4`.

Post-SCF-GGA Option. For many molecules the cost of evaluating the XC energy and potential on a three dimensional grid will dominate the overall computational cost. At the GGA level (functionals other than the local density approximation, `LSD` and `HFS`, that incorporate the density gradient) it is often found that the self consistent electron distribution is rather similar to that obtained from a `LSD` calculation. However, `LSD` calculations need a little bit less time because the derivatives of the basis functions at the grid points are not needed. It may therefore be a sensible approach to calculate the self consistent electron and spin densities at the `LSD` level and then obtain the final total energy at the GGA level. This was advocated by the ADF developers.

This feature is invoked in ORCA by using:

```
%method
   PostSCFGGA true
   end
```

What this does is to use the `PWLDA` local approximation in the SCF iterations and obtain the final energy at whatever functional was chosen in the input file.

More recent test calculations reveal that the performance gain from using this feature is truly minimal while the errors introduced in the final energies are large. Further use of this feature is therefore strongly discouraged.

SpecialGrid Option. Sometimes you will like to increase the integration accuracy for some atoms that need special care while it is not necessary to enlarge the grid generally. This situation for example arises when the basis set contains very steep functions on a few atoms that are required in order to calculate core properties such as isotropic hyperfine couplings. ORCA provides you with a basic mechanism to increase the radial integration accuracy for a few atoms while maintaining the chosen grid for all others.

```
%method
   # a maximum of 64 assignments can be made
   #  in = 0 : no changes are made
   #  in > 0 : the grid will be changed for all atoms with
   #           atomic number=in to IntAcc=an
   #  in < 0 : only the specific n'th atom will have its
   #           IntAcc value changed to an
   SpecialGridAtoms  i1, i2, i3,...,in;
   SpecialGridIntAcc a1,a2,a3,...,an;
   end
```

6.2.2.3 Using the RI-J Approximation to the Coulomb Part

A very useful approximation that greatly speeds up DFT calculations unless the molecule gets very large is the so called 'RI-approximation' [115–121]. RI stands for 'Resolution of the identity'. In short, charge distributions arising from *products of basis functions* are approximated by a linear combination of *auxiliary basis functions*.

$$\phi_i\left(\vec{r}\right)\phi_j\left(\vec{r}\right) \approx \sum_k c_k^{ij}\eta_k(\mathbf{r}) \tag{6.8}$$

There are a variety of different possibilities to determine the expansion coefficients c_k^{ij}. A while ago Almlöf and coworkers [139] have shown that for the approximation of electron repulsion integrals the best choice is to minimize the *residual repulsion*[5].

Define:

$$R_{ij} \equiv \phi_i\left(\vec{r}\right)\phi_j\left(\vec{r}\right) - \sum_k c_k^{ij}\eta_k\left(\vec{r}\right) \tag{6.9}$$

and

$$T_{ij} = \int\int R_{ij}\left(\vec{r}\right)\frac{1}{|\vec{r}-\vec{r'}|}R_{ij}\left(\vec{r}\right)d^3rd^3r' \tag{6.10}$$

and determine $T_{ij} = \int\int R_{ij}\left(\vec{r}\right)\frac{1}{|\vec{r}-\vec{r'}|}R_{ij}\left(\vec{r}\right)d^3rd^3r'$, leading to

$$\mathbf{c}^{ij} = \mathbf{V}^{-1}\mathbf{t}^{ij} \tag{6.11}$$

where:

$$t_k^{ij} = \left\langle\phi_i\phi_j\left|r_{12}^{-1}\right|\eta_k\right\rangle \tag{6.12}$$

$$V_{ij} = \left\langle\eta_i\left|r_{12}^{-1}\right|\eta_j\right\rangle \tag{6.13}$$

Thus an ordinary two electron integral becomes:

$$\left\langle\phi_i\phi_j\left|r_{12}^{-1}\right|\phi_k\phi_l\right\rangle \approx \sum_{p,q}c_p^{ij}c_q^{kl}V_{pq} \tag{6.14}$$

$$= \sum_{p,q}V_{pq}\sum_r\left(\mathbf{V}^{-1}\right)_{pr}t_r^{ij}\sum_s\left(\mathbf{V}^{-1}\right)_{qs}t_s^{kl}$$

$$= \sum_{r,s}\left(\mathbf{V}^{-1}\right)_{rs}t_r^{ij}t_s^{kl} \tag{6.15}$$

and the total Coulomb energy becomes ($\mathbf{P}$ is the total density matrix):

$$E_J = \sum_{i,j}\sum_{k,l}P_{ij}P_{kl}\left\langle\phi_i\phi_j\left|r_{12}^{-1}\right|\phi_k\phi_l\right\rangle \tag{6.16}$$

---

[5]But note that the basic theory behind the method is known for a long time, at least since the late sixties have methods similar to the RI approximation been used, mainly in the context of 'approximate *ab initio* methods' such as LEDO, PDDO and MADO but also in density functional theory in the mid and late seventies by Baerends , Dunlap and others [115–117, 121]

$$\approx \sum_{i,j} \sum_{k,l} P_{ij} P_{kl} \sum_{r,s} \left(\mathbf{V}^{-1} \right)_{rs} t_r^{ij} t_s^{kl}$$

$$= \sum_{r,s} \left(\mathbf{V}^{-1} \right)_{rs} \underbrace{\sum_{i,j} P_{ij} t_r^{ij}}_{\mathbf{X}_r} \underbrace{\sum_{k,l} P_{kl} t_s^{kl}}_{\mathbf{X}_s} \tag{6.17}$$

In a similar way the Coulomb contribution to the Kohn-Sham matrix is calculated. There are substantial advantages from this approximation: the quantities to be stored are the matrix $\mathbf{V}^{-1}$ which depends only on two indices and the three index auxiliary integrals t_r^{ij}. This leads to a tremendous data reduction and storage requirements relative to a four index list of repulsion integrals. Furthermore the Coulomb energy and the Kohn-Sham matrix contributions can be very quickly assembled by simple vector/matrix operations leading to large time savings. This arises because each auxiliary basis function $\eta_k\left(\vec{r}\right)$ appears in the expansion of many charge distributions $\phi_i\left(\vec{r}\right)\phi_j\left(\vec{r}\right)$. Unfortunately a similar strategy is less easily (or with less benefit) applied to the Hartree-Fock exchange term. In addition, the two and three index electron repulsion integrals are easier to compute than the four index integrals leading to further reductions in processing time.

If the auxiliary basis set $\{\eta\}$ is large enough, the approximation is also highly accurate. Since any DFT procedure already has a certain, sometimes sizable, error from the noise in the numerical integration of the XC part it might be argued that a similarly large error in the Coulomb part is perfectly acceptable without affecting the overall accuracy of the calculation too much. Furthermore the errors introduced by the RI method are usually much smaller than the errors in the calculation due to basis set incompleteness in the first place. I therefore recommend the use of the RI procedure for pure DFT's. However, one should probably not directly mix absolute total energies obtained from RI and non-RI calculations because the error in the total energy accumulates and will rise with increasing molecular size while the errors in the relative energies will tend to cancel.

There are several choices for AUX basis sets described in the next section which depend on the choice of the primary GTO basis used to expand the molecular orbitals[6].

In ORCA all that is needed to invoke the RI approximation is to type:

```
%method
  RI  on   # do use the RI-J approximation
      off  # do not use the RI-J approximation
  end
```

Recall:

- If you use RI you *must* specify an auxiliary basis set in the %basis section. For some standard bases the program will automatically choose an AUX basis. However, it is poor practice to rely on an adequate automatic choice.

---

[6] It probably should be noted that a slightly awkward step in the procedure is the inversion of the auxiliary integral matrix $\mathbf{V}$ which can easily become very large. Matrix inversion is an $O(N^3)$ process such that for large molecules this step takes some real time. However, in ORCA this is only done *once* during the calculation whereas other programs that constrain the fit to also exactly reproduce the number of electrons need to perform a similar process *each iteration*. Starting from Version 2.2.09 ORCA implements Cholesky decomposition in favor of matrix inversion which removes any bottleneck concerning the solution of the linear equation system.

6.2.2.4 The Split-RI-J Coulomb Approximation

There is an improved version of the RI-algorithm that has been implemented in version 2.2.09. This algorithm yields the same Coulomb energy as the standard RI-algorithm but is significantly faster if the basis set contains many high angular momentum functions (d-, f-, g-functions). For small basis sets there is virtually no difference between the two algorithms except that Split-RI-J uses more memory than standard RI. However, calculations with ca. 2000 basis functions need about 13 MB extra for Split-RI-J which is a trivial requirement on present day hardware.

The Split-RI-J algorithm is invoked with:

```
! Split-RI-J
```

Split-RI-J is presently only available for SCF calculations. The gradient, coupled-perturbed Kohn-Sham and TD-DFT modules will use the standard RI approximation instead.

NOTE:

- The Split-RI-J algorithm is the default if RI is turned on via ! RI. If you do not want to use Split-RI-J please insert the keyword ! NoSplit-RI-J

6.2.2.5 Using the RI Approximation for Hartree-Fock and Hybrid DFT (RIJONX)

The RI approximation can be used, although with less benefit, for hybrid DFT and Hartree-Fock (RHF and UHF) calculations. In this case a different algorithm[7] is used that allows a fair approximation to the Hartree-Fock exchange matrix. In my experience it is difficult to make this approximation highly accurate. It is, however, usefully fast compared to direct SCF if the molecule is 'dense' enough. There are special auxiliary basis sets that were taken from the TurboMole libraries and were originally developed for RI-MP2 calculations [140].

```
%method
  RI   on        # do use the RI approximation
  end

%basis
  Aux SV_C     # (=VDZ_C) aux basis for the SVP
               # and comparable orbital bases
      TZV_C    # (=VTZ_C) aux basis for the TZVP
               # and comparable orbital bases
      TZVPP_C # (=VTZPP_C) aux basis for the VTZPP
               # and comparable orbital bases (very large)
      QZVP_C  # aux basis for QZVP (even larger)
      QZVPP_C # aux basis for QZVPP

      # with diffuse functions
      aug_SV_C    # for the aug-SVP basis
      aug_TZV_C   # for the aug-TZVPP basis
```

---

[7] This algorithm was described by Kendall and Früchtl [118].

```
        aug_TZVPP_C # for the aug-TZVPP basis
        # Dunning bases correlation fitting
        cc_pVDZ_C
        cc_pVTZ_C
        aug_cc_pVDZ_C
        aug_cc_pVTZ_C
    end
```

NOTE:

- There has been little experimentation with this feature. It is provided on an experimental basis here.

Another feature that was implemented is to use the RI method for the Coulomb term and the standard treatment for the exchange term. This method is called RIJONX because the exchange term should tend towards linear scaling for large molecules. You can use this feature for Hartree-Fock and hybrid DFT calculation by using:

```
    %method
      RI       on      # do use the RI approximation
      RIFlags 1        # ...but treat exchange exactly
    end


    # equivalently use the following keyword AND DON'T FORGET
    # TO ASSIGN AN AUXILIARY BASIS SET!


    ! RIJONX
```

The requirements for the auxiliary basis are the same as for the normal RI-J method.

6.2.2.6 Using the RI Approximation for Hartree-Fock and Hybrid DFT (RIJCOSX)

The aim of this approximation is to efficiently compute the elements of exchange-type matrices:[8]

$$K_{\mu\nu} = \sum_{\kappa\tau} P_{\kappa\tau}(\mu\kappa|\nu\tau) \tag{6.18}$$

where $\mathbf{P}$ is some kind of density-type matrix (not necessarily symmetric) and the two-electron integrals are defined over the basis set $\{\varphi\}$ by:

$$(\mu\kappa|\nu\tau) = \int \mu(\mathbf{r}_1)\kappa(\mathbf{r}_1)\nu(\mathbf{r}_2)\tau(\mathbf{r}_2)r_{12}^{-1}d\mathbf{r}_1 d\mathbf{r}_2 \tag{6.19}$$

The approximation pursued here can be written as follows:

---

[8] The theory of this approach together with all evaluations and implementation details is described in [69]. References to earlier work can also be found there

$$K_{\mu\nu} \approx \sum_g X_{\mu g} \sum_\tau A_{\upsilon\tau}(\mathbf{r}_g) \sum_\kappa P_{\kappa\tau} X_{\kappa g} \tag{6.20}$$

Here the index 'g' refers to grid points $\mathbf{r}_g$ and:

$$X_{\kappa g} = w_g^{1/2} \kappa(\mathbf{r}_g) \tag{6.21}$$

$$A_{\upsilon\tau}(\mathbf{r}_g) = \int \frac{\nu(\mathbf{r})\tau(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|} d\mathbf{r} \tag{6.22}$$

where w_g denotes the grid weights. Thus, the first integration is carried out numerically and the second one analytically. Note that this destroys the Hermitian character of the two-electron integrals.

Equation 6.20 is perhaps best evaluated in three steps:

$$F_{\tau g} = (\mathbf{PX})_{\tau g} \tag{6.23}$$

$$G_{\nu g} = \sum_\tau A_{\nu\tau}(\mathbf{r}_g) F_{\tau g} \tag{6.24}$$

$$K_{\mu\nu} = (\mathbf{XG}^+)_{\mu\nu} \tag{6.25}$$

As such the equations are very similar to the pseudo-spectral method extensively developed and discussed by Friesner and co-workers since the mid 1980s and commercially available in the Jaguar quantum chemistry package. The main difference at this point is that instead of $X_{\kappa g}$ there appears a least-square fitting operator $Q_{\kappa g}$ in Friesner's formulation. Note that an analogue of the fitting procedure has also been implemented in ORCA, which however does not need specially optimized grids as in Friesner's pseudospectral method. The basic idea is to remove the grid errors within the basis set by "fitting" the numerical overlap to the analytical one. Due to its nature, overlap fitting is supposed to work better with larger basis sets.

Given the exchange matrix, the exchange energy is given by (a sum over spin cases is left out here for simplicity):

$$E_X = \frac{1}{2} \sum_{\mu\nu} P_{\mu\nu} K_{\mu\nu}(\mathbf{P}) \tag{6.26}$$

Assuming that E_X refers to the nonrelativistic, variational SCF energy, the derivative with respect to parameter λ can be re-arranged to the following form:

$$\frac{\partial E_X}{\partial \lambda} \approx 2 \sum_g \sum_{\mu\nu} \frac{\partial F_{\mu g}}{\partial \lambda} G_{\nu g} \tag{6.27}$$

with:

$$\frac{\partial F_{\mu g}}{\partial \lambda} = w_g^{1/2} \sum_\kappa P_{\kappa\mu} \frac{\partial X_{\mu g}}{\partial \lambda} \tag{6.28}$$

In this formulation, the gradient arises as a minor modification of the exchange matrix formation code. In particular, the derivatives of the analytic integrals are not needed, merely the derivatives of the basis functions on the grid.

In our implementation, we have defined grids of increasing quality. Higher and more accurate (and more expensive) grids for the COSX approximation are chosen by increasing n from 1 to 9 in

```
! GridXn
```

From version 2.9, overlap fitting is used by default. This means that there might be some differences with respect to previous versions. Firstly, smaller grids are used by default. The old default grid can be retained by typing

```
! GridXOLD
```

If consistency with previous versions of ORCA is desired, overlap fitting can be removed using the keyword `!NoSFitting`. Note that overlap fitting yields better results as the basis and/or the grid increases. It was adapted as the default procedure with the smallest reliable grid, but it is compatible with all possible grids. Grids tested in the overlap fitting paper ([141]) are available as `!GridXS1` and `!GridXS2`. The latter grid is the default.

For expert users, the grid parameters for the exchange grids can be even more finely controlled:

```
%method
   IntAccX  Acc1, Acc2, Acc3
   GridX    Ang1, Ang2, Ang3
   XCorrection Corr1, Corr2, Corr3
   UseFinalGridX true
   end
```

There are three grids involved: the smallest grid (Acc1, Ang1) that is used for the initial SCF iterations, the medium grid (Acc2, Ang2) that is used until the end of the SCF and the largest grid (Acc3, Ang3) that is used for the final energy and the gradient evaluations. UseFinalGridX turns this last grid on or off. 'Accn' refers to the radial integration accuracy, GridX refers to the angular grid resolution and XCorrection (=0 or 1) refers to the method used for correcting the grid error. For '0' no one-center correction is applied, for '1' a one-center correction is computed analytically for the two-electron integrals. Note that in cases where convergence difficulties arise, it is advisable to increase the smallest grid to a larger value, or even equal to the medium grid using the Acc1, Ang1 parameters.

To modify the overlap fitting parameters

```
%method
  UseSFitting false # equals to NoSFitting in the simple
                    # input. Default is true.
  UseQGradFit true  # uses the SCF fitting matrix for
                    # gradient calculations. Default
                    # is false.
end
```

Note that overlap fitting works for HF and MP2 gradients as well without specifying any additional keyword. The UseQGradFit parameter merely uses the same fitting matrix for the gradients as for the energy calculation. However, this does not save significant time, neither is it more accurate, therefore it is turned off by default.

6.2.2.7 Choice of the COSX Grid and Convergence Issues

Other then the !Gridn (where n is an integer number) keywords used to specify the DFT integration grids, the grid structures used for HF exchange evaluation, called by the keywords !GridXn have also been discussed in previous chapters. The purpose of this chapter is to give a more specific introduction into the properties of COSX grids, and discuss some more common problems while using these.

As a start, it will be useful to relate the simple keywords GridXn, to the method block keywords detailed above. Sometimes it is necessary to increase the grids using the method block parameters directly, as they provide a more flexible control over the grid size. For some of the more common grids the simple keywords and the equivalent method block parameters are given below

```
!GRIDX4
%method
GridX 1,1,3
IntAccX 3.34,4.01,4.34
end


!GRIDX5
%method
GridX 1,1,3
IntAccX 3.67,4.01,4.67
end


!GRIDX6
%method
GridX 1,2,3
IntAccX 3.67,4.01,4.34
end


!GRIDX7
%method
GridX 1,2,4
IntAccX 3.67,4.01,4.34
end
```

and for the default grid

```
!GRIDXS2
%method
GridX 1,1,2
IntAccX 3.34,3.67,4.01
end
```

Symptoms of convergence issues: Erratic convergence behavior, often starting from the first SCF step, or possibly setting in at a later stage, producing crazy energy values, with "megahartree" jumps. If overlap fitting is on, the following error message may also be encountered: "Error in Cholesky inversion of numerical overlap".

Convergence issues may arise when the chosen grid has difficulties in representing the basis set. This is the "grid equivalent" of a linear dependence problem, discussed in 6.3.4. It should also be mentioned that the grid related problem discussed here often goes hand in hand with basis set linear dependence, although not necessarily. The most straightforward way of dealing with these is to increase the size of the integration grid. This however, is not always desirable or possible, and in the remainder of this chapter some other methods will be discussed.

One way to avoid the Cholesky inversion issue is to turn overlap fitting off (`!NoSFitting`), however, this means that the numerical problems are still present, only they are ignored. Due to the fact that overlap fitting operates with the numerical overlap, and its inverse, it is more sensitive to linear dependence issues, and turning off the fitting procedure may lead to convergence. This may be a pragmatic, but by no means clean solution, since it relies on the assumption that the numerical errors are small.

On the other hand, overlap fitting also gives a similar tool to deal with linear dependence issues as the one discussed in 6.3.4 for basis sets. The eigenvalues of the numerical overlap can be inspected similarly, and small values can be screened out. There is unfortunately no universal way to determine this screening parameter, but see 6.3.4 for typical values.

The parameters influencing the method used for inversion and obtaining the fitting matrix are

```
%method
  SFitInvertType Cholesky   # Use Cholesky inversion. Default
                 Cholesky_Q # Cholesky + full Q matrix
                 Diag       # Inversion via diagonalization
                 Diag_Q     # Diag + full Q matrix
  SInvThresh     1e-8       # inversion threshold for Diag and Diag_Q, default 1 e-8
end
```

By default, the inversion procedure proceeds through Cholesky decomposition as the fastest alternative. Ideally the overlap matrix is non-singular, as long as the basis set is not linear dependent. For singular matrices the Cholesky procedure will fail. It should be noted at this point that the numerical overlap can become linear dependent even if the overlap of basis functions is not, and so a separate parameter will be needed to take care of grid related issues. To achieve this, a diagonalization procedure (`Diag`) can be used instead of Cholesky with the corresponding parameter to screen out eigenvectors belonging to eigenvalues below a threshold (`SInvThresh`). For both Cholesky and diagonalization procedures a 'full Q' approach is also available (`Cholesky_Q` and `Diag_Q`), which corresponds to the use of a more accurate (untruncated) fitting matrix.

6.2.2.8 Improved Analytical Evaluation of the Coulomb Term: Split-J

ORCA features a method that gives the *exact* Coulomb term at significantly reduced computational cost. It can most profitably be applied to the case where no HF exchange is present. Thus if you use LDA or GGA functionals and you do not want to apply the RI approximation (perhaps because you use a special basis set for which no fit-set is available), the Split-J is an attractive alternative to the traditional evaluation. The advantages of Split-J increase with the quality of the basis set used, i.e. if you have basis sets with high-angular momentum functions split-J can be more effective by a factor of 2-5 compared to the traditional evaluation. For smaller basis sets (i.e. SV(P) and the like) the advantages are smaller but still significant. However, Split-J is also significantly slower than RI-J (but recall that Split-J is exact while RI-J is an approximation).

A small job that uses the Split-J feature is shown below:

```
    ! RKS LSD TZVPP TightSCF Direct
    %scf     jmatrix 1  # turns on the Split-J feature
            end
    *int 0 1
       C  0 0 0 0.00   0.0    0.00
       O  1 0 0 1.20   0.0    0.00
       H  1 2 0 1.10 120.0    0.00
       H  1 2 3 1.10 120.0  180.00
    *
```

6.2.2.9 The Standard Basis Sets and Computational Levels

The definition of the standard basis sets and computational levels is described below. As explained in section 6.2 these features designed in the hope to assist the user in carrying out well reproducible calculations at a consistent level of theory and a reasonable balance between cost and accuracy with a minimum of input.

Standard calculation levels are defined below

Parameter	Calculation-Level						
	Quick DFT	Quick Opt	Medium Opt	Good Opt	Acc Opt	DFT Energy	DFT Energy+
RunTyp	Energy	Opt	Opt	Opt	Opt	Energy	Energy
Functional	BP	BP	BP	BP	BP	B3LYP	B3LYP
Basis[1]	1	1	2	3	4	4	5
RI	On	On	on	on	on	off[2]	off[2]
SCFConv	Loose	Tight	Tight	Tight	Tight	Normal	Normal
GeoConv	-	Normal	Normal	Normal	Tight	-	-
Grid[3]	2/3	2/4	2/4	3/5	4/5	4/5	4/5

1 – these bases refer to the ORCA default bases that are based on the Ahlrichs SV and TZV basis sets to be described below

2 – The RI approximation can be used in these calculations by using ! RIJONX in the input

3 – The first and second number refer to the dual-grid feature of ORCA

You can easily override these defaults by simply typing the appropriate keyword that changes one of the settings given above (for example if you want `MediumOpt` but with the `PBE` functional simply give ! `MediumOpt PBE`).

The default-basis sets are of increasing quality and are defined below:

Default-Basis	H	Main-Group	Transition-Metals
1	SV	SV	TZV
2	SV	SV(d)	TZV(P)
3	TZV	TZV(d)	TZV(2pf)
4	TZV(p)	TZV(2d)	TZV(2pf)
5	TZV(p/s)	TZV(2d/sp)	TZV(2f/sppd)
6	TZV(2p/s)	TZV(2df/sp)	TZV(2fg/sppd)

The notation SV(xxx/yyy) refers to the SV basis set with polarization functions xxx and diffuse functions yyy

A special feature of these basis sets is that the appropriate auxiliary basis for DFT within the RI approximation is automatically selected.

6.2.2.10 Treatment of Dispersion Interactions with DFT-D3

Introduction

DFT-D3 is an atom-pairwise (atom-triplewise) dispersion correction which can be added to the KS-DFT energies and gradient [2]:

$$E_{DFT-D3} = E_{KS-DFT} + E_{disp} \tag{6.29}$$

E_{disp} is then the sum of the two- and three-body contributions to the dispersion energy: $E_{disp} = E^{(2)} + E^{(3)}$. The most important is the two-body term which is given at long range by:

$$E_{disp} = -\sum_{A<B} \sum_{n=6,8} s_n \frac{C_n^{AB}}{r_{AB}^n} \tag{6.30}$$

C_n^{AB} denotes the averaged (isotropic) nth-order dispersion coefficient for atom pair AB, and r_{AB} is their internuclear distance. s_n is a functional-dependent scaling factor (see below). In the general case, an adequate damping function must be employed.

Damping Functions

In order to avoid near singularities for small r_{AB}, the dispersion contribution needs to be damped at short distances. One possible way is to use rational damping as proposed by Becke and Johnson [142–144]:

$$E^{(2)} = -\sum_{A<B} \sum_{n=6,8} s_n \frac{C_n^{AB}}{r_{AB}^n + f(R_0^{AB})^n} \tag{6.31}$$

with [144]

$$R_0^{AB} = \sqrt{\frac{C_8^{AB}}{C_6^{AB}}} \tag{6.32}$$

and

$$f(R_0^{AB}) = a_1 R_0^{AB} + a_2. \tag{6.33}$$

Damping the dispersion contribution to zero for short ranges (as in Ref. [2]) is also possible:

$$E^{(2)} = -\sum_{A<B} \sum_{n=6,8} s_n \frac{C_n^{AB}}{r_{AB}^n} f_{d,n}(r_{AB}) \tag{6.34}$$

with

$$f_{d,n} = \frac{1}{1 + 6(\frac{r_{AB}}{s_{r,n} R_0^{AB}})^{-\alpha_n}} \tag{6.35}$$

Note that the R_0^{AB} used with this damping are from Ref. [2]. For more information on the supported damping functions, see Ref [74]. In the ORCA program the dispersion correction with zero damping is invoked by the keyword

! D3ZERO. The default is the recommended variant with Becke-Johnson damping and is invoked by the keyword ! D3BJ or simply ! D3.

Three-body term

It is possible to calculate the three-body dispersion contributions with DFT-D3, according to

$$E^{(3)} = - \sum_{A<B<C} \frac{C_9^{ABC}(3\cos\theta_a \cos\theta_b \cos\theta_c + 1)}{(r_{AB}r_{BC}r_{CA})^3} f_{d,(3)}(\overline{r}_{ABC}), \tag{6.36}$$

where θ_a, θ_b and θ_c are the internal angles of the triangle formed by r_{AB}, r_{BC} and r_{CA}. The C_9 coefficient is approximated by

$$C_9^{ABC} \approx -\sqrt{C_6^{AB} C_6^{AC} C_6^{BC}}. \tag{6.37}$$

The three-body contribution has a small effect on medium-sized molecules and is damped according to equation 6.36. The damping function $f_{d,(3)}(\overline{r}_{ABC})$ is similar to the one shown in equation 6.35 with $\overline{r}_{ABC}$ being the geometric mean of r_{AB}, r_{BC} and r_{CA}. It can be used with both variants of the $E^{(2)}$ term. However, the three-body term itself will always be calculated using the zero damping scheme. Furthermore, only numerical gradients are available. Adding three-body correction is a rather new feature, but has proven to give quite accurate results for the thermochemistry of supramolecular systems [145]. In the ORCA program, the dispersion correction with zero damping and the three-body contribution is invoked by the keyword ! D3ZERO ABC. Becke-Johnson damping for $E^{(2)}$ and zero damping for $E^{(3)}$ is invoked by ! D3BJ ABC.

Options

Note that correcting Hartree-Fock (HF) is only parameterized with BJ-damping. For a constantly updated list of supported functionals, check the website of the Grimme group [146]. If there is a functional on this website that is parametrized, but the parameters are not implemented into the ORCA program yet, you can specify the parameters manually as shown below (using the respective parameters from Ref. [146]). In the same fashion, one could also use own parameters but this is not recommended.

Important: GGA and hybrid functionals should only be used with $s_6 = 1.0$ to ensure asymptotically correct behaviour. Double-hybrid functionals already account for parts of the dispersion interaction and should therefore be used with $s_6 < 1.0$. Within the "%method" block it is possible to change the parameters s6, a1, s8 and a2 for the variant with Becke-Johnson damping.

```
! d3bj b2plyp
%method
D3S6 0.64
D3A1 0.3065
D3S8 0.9147
D3A2 5.0570
end
```

The variant with zero damping offers the parameters s6, rs6, s8 and α_6.

```
! d3zero blyp
%method
D3S6 1.0
D3RS6 1.094
D3S8 1.682
D3alpha6 14
end
```

If a geometry optimization is performed (! opt) then the program automatically calls the DFT-D3 gradient. There are also special functional parameters, which were optimized for triple-zeta basis sets. This option is is only available with zero damping and can be invoked by the keywords ! D3ZERO D3TZ. Preliminary results in the SI of Ref. [2] indicate that results are slightly worse than with the default parameters and QZVP type basis sets. This option should be carefully tested for future use in very large computations.

Example input files

In the following, we list some example input files. A compuation using the DFT-D3 dispersion correction with BJ-damping using the D3BJ keyword may look like this. As it is the default, the use of the keyword D3 is identical.

```
! pbe svp d3bj
* xyz 0 1
 C     0.000000     0.000000     0.000000
 O     0.000000     0.000000     1.400000
 O     0.000000     0.000000    -1.400000
*
```

The output for the dispersion correction in the ORCA output will look like this:

```
-------------------------------------------------------------------------------
                         DFT DISPERSION CORRECTION

                           DFTD3 V2.1  Rev 6
                         USING Becke-Johnson damping
-------------------------------------------------------------------------------
The default PBE functional is recognized
Active option DFTDOPT                   ...          4

molecular C6(AA) [au] = 156.562480


          DFT-D V3
 parameters
 s6 scaling factor         :      1.0000
 a1 scaling factor         :      0.4289
 s8 scaling factor         :      0.7875
 a2 scaling factor         :      4.4407
 ad hoc parameters k1-k3   :     16.0000     1.3333    -4.0000

 Edisp/kcal,au: -0.563071585638  -0.000897311593
```

```
     E6   /kcal   :  -0.390909076
     E8   /kcal   :  -0.172162510
     % E8         :  30.575598941


     ------------------------   ----------------
     Dispersion correction          -0.000897312
     ------------------------   ----------------



     ------------------------   --------------------
     FINAL SINGLE POINT ENERGY      -188.137098479095
     ------------------------   --------------------
```

E_{disp} is given as the "Dispersion correction". It will be automatically included into the singlepoint energy. As discussed above, the parameters s6, a1, s8 and a2 may be defined manually by:

```
     ! pbe svp d3bj
     %method
     D3S6 1.0
     D3A1 0.4289
     D3S8 0.7875
     D3A2 4.4407
     end
     *xyz 0 1
       C     0.000000     0.000000     0.000000
       O     0.000000     0.000000     1.400000
       O     0.000000     0.000000    -1.400000
     *
```

The calculation of the same system with D3ZERO is invoked by:

```
     ! pbe svp d3zero
     *xyz 0 1
       C     0.000000     0.000000     0.000000
       O     0.000000     0.000000     1.400000
       O     0.000000     0.000000    -1.400000
     *
```

In the same fashion as above, the s6, rs6, s8 and the exponent α_6 can be defined by the user. The next example shows this along with the call for the three-body correction using ABC:

```
     ! pbe svp d3zero abc
     %method
     D3S6 1.0000
     D3RS6 1.2170
```

```
    D3S8 0.7220
    D3ALPHA6 14.0
    end
    *xyz 0 1
     C      0.000000      0.000000      0.000000
     O      0.000000      0.000000      1.400000
     O      0.000000      0.000000     -1.400000
     *
```

6.2.2.11 DFT Calculations with the Non-Local, Density Dependent Dispersion Correction: DFT-NL

Accounting for the missing van der Waals (vdW, dispersion) forces in standard Kohn – Sham Density Functional Theory (DFT) has become essential in many studies of chemical and physical electronic structure problems. Common approaches use atom pair-wise additive schemes as the popular DFT-D3 [2, 74] method, which is also available in ORCA by invoking the keyword "! D3" (for more information see section 6.2.2.10).

A different route is followed by the van der Waals Density Functional (vdW-DF) as pioneered by Langreth and Lundquist [147]. These methods use as input to compute the Non-Local (NL) dispersion contribution only the electron density. The recently developed vdW functional VV10 of Vydrov and Van Voorhis [148] currently seems to be the most promising candidate for a general and accurate electronic structure method.

We use the term DFT-NL for any (hybrid)GGA density functional in combination with the non-local part of the VV10 functional with an optimized parameter b, which will be defined below. The performance of these methods has been evaluated in Ref. [149] using the GMTKN30 [150–152] database and the S66 set [153]. The performance of weak hydrogen bonds were evaluated in Ref. [154].

We recommend the DFT-D3 dispersion correction in general and the DFT-NL method for checking purposes by single-point calculations. DFT-NL and DFT-D3 perform very similar but NL is to be preferred for metallic systems or when the basic electronic structure changes significantly (e.g. oxidations or ionizations). Because the NL correction increases the computational cost in particular of GGA treatments using RI significantly, it is more useful in combination with hybrid functionals (and RIJK or RIJCOSX techniques) where the computational overhead is marginal if it is done non-selfconsistently (see below).

The total exchange-correlation (XC) energy of VV10 type functionals is defined in eq. 6.38. It is composed of standard exchange (X) and correlation (C) parts and the non-local (NL) term, which covers (mainly) long-range dispersive energy:

$$E_{XC}^{DFT-NL} = E_X^{(hybrid)GGA} + E_C^{GGA} + E_{c-NL}^{VV10} \tag{6.38}$$

The NL term is given by the following double-integral:

$$E_{c-NL}^{VV10} = \int dr \rho(r) \left[\beta + \frac{1}{2} \int dr^{'} \rho(r^{'}) \varphi(r, r^{'}) \right] \tag{6.39}$$

where ρ is the total electron density, and the definition of the kernel φ(r,r') and β is as follows (in a.u.):

$$\varphi\left(r, r^{'}\right) = -\frac{3}{2gg^{'}\left(g+g^{'}\right)} \qquad\qquad g\left(r\right) = \omega_0\left(r\right) R^2 + \kappa(r) \qquad\qquad R = |r - r^{'}|$$

$$\omega_0\left(r\right) = \sqrt{C\left|\frac{\nabla\rho(r)}{\rho(r)}\right|^4 + \frac{4\pi}{3}\rho(r)} \qquad\qquad \kappa\left(r\right) = b\frac{3\pi}{2}\left[\frac{\rho(r)}{9\pi}\right]^{1/6} \qquad\qquad \beta = \frac{1}{32}\left[\frac{3}{b^2}\right]^{3/4}$$

In the original definition, the short-range attenuation parameter b appearing in κ and β was fitted to the S22 set [155] of non-covalent interactions (b=5.9 for the rPW86PBE GGA). The other parameter C=0.0093 appearing in $\omega0$, which determines the long-range behavior, was set to its original value. Other DFT-NL functionals are constructed analogously. For a detailed discussion of the derivation of the formulas and their physical meaning and basis see the references given and those given therein.

The defined energy of the non-local DFT-NL exchange-correlation functional can computed non-self-consistently based on a converged SCF density but a self-consistent treatment is also possible. We take B3LYP as an example.

In our implementation of the non-self-consistent B3LYP-NL functional, in the first step a self-consistent B3LYP computation is performed. In the second step the optimized electron density from the B3LYP computation is taken as input for the energy calculation of the non-local part. This procedure is invoked by the combination of the keywords "! B3LYP NL". Use of the keywords " ! B3LYP SCNL" would request a self-consistent treatment in which orbitals and density are optimized in the presence the full B3LYP-NL exchange-correlation potential (see below). According to many test calculations, an SCNL treatment is rarely necessary for normal energy evaluations.

The computation of the double-integral given in eq. 6.39 requires using an integration grid, just like for normal exchange-correlation functionals. The grid size is used analogously to the regular grids available in the ORCA program and can be specified by invoking the keyword "! vdwgridX", where "X" can be 1,2,3,4,5,6 and 7. The default grid size is "! vdwgrid2". In test calculations we found, that going beyond grid size of 3 is not necessary in normal applications.

In the following we compute the energy of an argon dimer at the distance of 3.76 Å with the def2-TZVP basis set using the B3LYP hybrid functional as an example with the non-self-consistent variant of the DFT-NL dispersion correction. The original VV10 method is based on the rPW86PBE GGA.

```
! b3lyp nl
! def2-tzvp def2-tzvp/jk rijk grid4 vdwgrid2 nofinalgrid nososcf nopop
*xyz 0 1
Ar  0.0 0.0 0.0
Ar  0.0 0.0 3.76
*
```

The DFT-NL output for this example is shown below:

```
-------------------------------------------------------------------------------
                        DFT-NL dispersion correction
-------------------------------------------------------------------------------


SC Energy -1054.960659618
NL parameter b = 4.80
NL    Energy 0.209416556
SC+NL Energy -1054.751243062
NL done in   0.3 sec
-------------------------------------------------------------------------------
```

```
                ----------------
                TOTAL SCF ENERGY
                ----------------

                Total Energy       :         -1054.75124306 Eh            -28701.24047 eV

                Components:
                Nuclear Repulsion  :            45.59931263 Eh              1240.82038 eV
                Electronic Energy  :         -1100.35055569 Eh            -29942.06085 eV
                NL         Energy  :             0.20941656 Eh                 5.69851 eV
```

Here we find the B3LYP total energy ("SC Energy") of -1054.9607 Hartree, the parameter b = 4.8, the non-local contribution ("NL Energy") of 0.2094 Hartree and the final total energy ("SC+NL Energy") of -1054.7512 Hartree, which is the sum of the SC and NL energy. In the "Components" section the non-local-contribution is listed separately ("NL Energy") in order to be consistent with the "! SCNL" option (see below).

In the current version of ORCA there are seven GGA and hybrid functionals available, which can be used with the DFT-NL method. The GGA functionals are rPW86PBE (b=5.9), BLYP (b=4.0) and revPBE (b=3.7). The hybrid functionals are B3LYP (b=4.8), B3PW91 (b=4.5), revPBE0 (b=4.3) and revPBE38 (b=4.7). In addition, we also added the non-local term to Hartree-Fock (HF) with a parameter of b=3.9. The parameter C was not changed. All these functionals are ready to use by invoking the keyword "! DF NL", where DF stands for the seven density functionals. Hartree-Fock is invoked with the keyword "! HF NL".

All DFT-NL methods can be used for closed shell and open shell systems.

All density functionals that are available in ORCA (but for which no b parameter has been determined yet) can also be used with the DFT-NL method by providing a value for the parameter b as shown here:

```
    %method
    NLb    5.0
    End
```

For example, for the functional BP86 the parameter b has not been fitted yet and, therefore, using the option "! BP86 NL" will give an error, but by providing a parameter b (5.0 just as an example!) one can use this functional in combination with the DFT-NL dispersion correction. Users who want to use such new combinations might contact the Grimme group.

```
    ! bp86 nl
    ! def2-tzvp def2-tzvp/j ri grid4 vdwgrid2 nofinalgrid nososcf nopop
    %method
    NLb     5.0
    end
    *xyz 0 1
    Ar  0.0 0.0 0.0
    Ar  0.0 0.0 3.76
    *
```

In addition, for any of the already fitted functionals one can define a user specific value for the parameter b, which might be different to the default value, therefore, overriding the default parameter b. For example, for B3LYP the fitted parameter is b=4.8 which is changed below to a more repulsive value of b=5.2.

```
   ! b3lyp nl
   ! def2-tzvp def2-tzvp/jk rijk grid4 vdwgrid2 nofinalgrid nososcf nopop
   %method
   NLb    5.2
   end
   *xyz 0 1
   Ar  0.0 0.0 0.0
   Ar  0.0 0.0 3.76
   *
```

The other parameter C=0.0093 appearing in $\omega 0$ may also be changed with the "`NLC`" keyword as shown in the following example. Of course, both parameters b and C can be changed with the keywords "`NLb`" and "`NLC`" at the same time.

```
   ! b3lyp nl
   ! def2-tzvp def2-tzvp/jk rijk grid4 vdwgrid2 nofinalgrid nososcf nopop
   %method
   NLC    0.0083
   end
   *xyz 0 1
   Ar  0.0 0.0 0.0
   Ar  0.0 0.0 3.76
   *
```

Self-consistent computations with the DFT-NL dispersion correction

Self-consistent calculations with the DFT-NL dispersion correction are possible by invoking the keyword "`!  SCNL`" in combination with one of the available density functionals (rPW86PBE, revPBE, BLYP, B3LYP, B3PW91, revPBE0 and revPBE38). However, as explained above any density functional can be used also in combination with the "`SCNL`" keyword by providing a parameter b with the "`NLb`" keyword. Note, that due to technical reasons self-consistent calculations are not possible with the Hartree-Fock method.

In the following example we use the B3LYP hybrid functional with the self-consistent DFT-NL variant:

```
   ! b3lyp scnl
   ! def2-tzvp def2-tzvp/jk rijk grid4 vdwgrid2 nofinalgrid nososcf nopop
   *xyz 0 1
   Ar  0.0 0.0 0.0
   Ar  0.0 0.0 3.76
   *
```

The DFT-NL output of this example is shown below:

```
        --------------------------------------------------------------------------------
                        Selfconsistent DFT-NL dispersion correction
        --------------------------------------------------------------------------------


        NL parameter b = 4.80


        --------------
        SCF ITERATIONS
        --------------
        ITER      Energy         Delta-E        Max-DP      RMS-DP      [F,P]     Damp
          0  -1054.7512430617    0.000000000000 0.00166322  0.00007167  0.0007919 0.0000
          1  -1054.7512472736   -0.000004211912 0.00077400  0.00002662  0.0005919 0.0000
          2  -1054.7512477044   -0.000000430774 0.00004435  0.00000152  0.0000304 0.0000
                        **** Energy Check signals convergence ****


                ****************************************************
                *                    SUCCESS                      *
                *            SCF CONVERGED AFTER   3 CYCLES        *
                ****************************************************



        ----------------
        TOTAL SCF ENERGY
        ----------------


        Total Energy       :          -1054.75124767 Eh           -28701.24059 eV


        Components:
        Nuclear Repulsion  :             45.59931263 Eh             1240.82038 eV
        Electronic Energy  :          -1100.35056030 Eh           -29942.06097 eV
        NL        Energy   :              0.20940947 Eh                5.69832 eV
```

The procedure for the self-consistent computation is similar to the non-self-consistent one: In the first step a self-consistent B3LYP run is performed and in the second step the converged electron density is used as input for the self-consistent SCNL run, as shown in the output example above.

In the output the "Total Energy" is the energy of an exchange-correlation functional (B3LYP) plus the non-local (NL) contribution. In the "Components" section the non-local contribution ("NL Energy") is listed separately in order to be consistent with the "! NL" option (see above).

As can be seen from this example only two cycles of the self-consistent NL run were performed. The difference in energy (5.0e-06) by comparing the self-consistent (-1054.751248) and non-self-consistent (-1054.751243) computations is very small which seems to be typical also even for larger systems.

In the current version of ORCA analytical gradients for any DFT-NL dispersion correction are not available and, therefore, geometry optimizations are not possible using analytical gradients. As mentioned above we recommend to use DFT-D3 optimized geometries for single point DFT-NL calculations.

6.2.2.12 DFT and HF Calculations with the geometrical Counterpoise Correction: gCP

The central idea of the gCP correction [156] is to add in a semi-empirical fashion a correction ΔE_{gCP} to the energies of molecular systems, in order to remove artificial overbinding effects from BSSE (see section 5.1.6). The correction uses atomic corrections and thus also yields the ability to correct for intramolecular BSSE. The parametrization is

Table 6.5: Overview of parametrized basis sets. The `level` keyword in `!GCP(level)` is a compound of `HF` or `DFT` and the `basis set` keyword. Valid examples are: `!GCP(HF/MINIS)`, `!GCP(DFT/LANL)`, `!GCP(HF/TZ)`, `!GCP(DFT/631GD)` ⋯

parametrized basis set	HF	DFT	basis set keyword
MINIS	yes	yes	MINIS
SV	yes	yes	SV
6-31G(d)	yes	yes	631GD
6-31G(d) + LANL2DZ (Sc-Zn)	no	yes	LANL
def2-SV(P)	no	yes	SV(P)
def2-SV(P/h,c)	no	yes	SVX or SV(P/H,C)
def2-SVP	yes	yes	SVP
def2-TZVP	yes	yes	TZ

constructed such that it approximates the Boys and Bernadi counterpoise (CP) correction ΔE_{CP} in the intermolecular case

$$\Delta E_{CP} \approx \Delta E_{gCP} \ , \tag{6.40}$$

where e.g. for a complexation reaction $A + B \to C$ our correction is given by

$$\Delta E_{gCP} = E_{gCP}(C) - E_{gCP}(A) - E_{gCP}(B) \ . \tag{6.41}$$

In practice, E_{gCP} can simply be added to the HF/DFT energy

$$E_{total} = E_{HF/DFT} + E_{gCP} \ , \tag{6.42}$$

which is also done in ORCA. The `FINAL SINGLE POINT ENERGY` is the sum of the HF/DFT energy and the gCP correction.

The central equation over all atoms N reads:

$$E_{gCP} = \sigma \cdot \sum_a^N \sum_{b \neq a}^N e_a^{miss} \cdot f_{dec}(R_{ab}) \ , \tag{6.43}$$

where the energy e_a^{miss} is a measure for the incompleteness of the chosen target basis set (that is typically small), and $f_{dec}(R_{ab})$ is a decay function that depends on the inter-atomic distance R_{ab}.

The scaling factor σ is one out of 4 parameters needed for every `method/basis set` combination. More details on this can be found in the original gCP paper [156].

Analytical gradients are available for geometry optimization. Due to its semi-empirical nature the correction itself is calculated within seconds even for very large systems.

The correction can be invoked by using `!GCP(level)` keyword, where `level` is a compound of the method (=`HF` or `DFT`) and the `basis set`. See table 6.5 for the available basis sets and the corresponding keyword. For a B3LYP calculation using the def2-SV(P) basis set a typical input would be:

```
! B3LYP def2-SVP GCP(DFT/SV(P))
*xyzfile 0 1 example.xyz
```

The output states the level, the 4 parameters mentioned above and the correction itself:

```
              --------------------------------------------------------------------

                      g C P  - geometrical counterpoise correction


              --------------------------------------------------------------------
              Method:  dft/sv(p)

              Parameters:   sigma        eta        alpha        beta
                           0.2424     1.2371     0.6076      1.4078
              Egcp:       0.0381115568 a.u.
              ------------------   -----------------
              gCP correction              0.038111557
              ------------------   -----------------


              -------------------------   --------------------
              FINAL SINGLE POINT ENERGY     -152.809257799447
              -------------------------   --------------------
```

There may be issued several warnings and notices: Elements of the 4th and higher period are treated as their 3rd period analog, i.e. Sn is treated as if it were Ge in terms of parameters. If this is the case a notice is printed. If an element has no parameters it will give no contributions to the correction. Both cases can occur at once:

```
    ! HF SV GCP(hf/minis)
    *xyzfile 0 1 example.xyz
```

The corresponding output:

```
    Method:  hf/minis

     ** NOTE ** -> element sn will be treated as ge
     ** WARNING ** -> element ge has no parameters       (no contribution)!
    Parameters:   sigma        eta        alpha        beta
                 0.1290     1.1526     1.1549      1.1763
    Egcp:       0.1593457998 a.u.
    WARNING: Basis set mismatch between ORCA and otool_gcp:
    ORCA: 280  gCP: 88
    ------------------   -----------------
    gCP correction              0.159345800
    ------------------   -----------------
```

This is done for the user convenience. It enables the computation of molecules that has un-parametrized elements. If only one or a few atoms in a larger molecule is treated inaccurate or not at all the error can be expected to be small. In case of a mismatch between the ORCA basis set and the selected basis set for gCP another warning is printed. This was done on purpose in above's example (GCP(hf/minis) instead of GCP(hf/sv)). This is meant to be a safety check. However, a small deviation may cause only a minor error (be careful nonetheless!). Use your own judgment if you want to use an unparametrized basis sets: Number of basis functions <u>and</u> exponents should be very similar!

For debug issues, !LARGEPRINT will print additional information about parameters and individual atomic contributions.

The gCP input can also be defined through the method section

```
%method
  DoGCP true/false          # turn gCP on/off
  GCPMETHOD "method"        # define method string for otool_gcp, eg. dft/svp
  GCP.D3MINIS true/false    # use special DFT-D3 refit for HF/MINIS (default=true)
  end
```

Further comments:
CP(HF/MINIS) sets automatically the refitted D3 parameter as proposed in the original gCP paper.

The gCP method is implemented via an external tool called otool_gcp, which is based on the original Fortran program used in the publication. Thus, the otool_gcp binary can also be called directly via the command line (otool_gcp -h gives an overview of the options). It is also possible to read an external parameter file ($HOME/.gcppar) using !GCP(FILE). For further information, please look at the manual for the gcp program as provided by Prof. S. Grimme[9]. During the calculation some temporary output files are written by ORCA: basename.gcp.in.tmp and basename.gcp.out.tmp will contain the input for otool_gcp and its output.

It has been demonstrated that the popular combination of B3LYP with 6-31G(d) can be strongly improved using DFT-D3 and gCP [157]. For convenience, the following compound keyword is defined ! B3LYP-gCP-D3/6-31G*. This equals the keywords: ! B3LYP 6-31G* D3BJ GCP(DF/631GD).

General Advices:

- Small basis sets show not only a large BSSE, but general shortcomings. These effects are not always clearly distinguishable.

- If computationally affordable, large basis sets (triple-ζ and higher) are always preferable for a given system.

- gCP makes only sense for somewhat large molecules

- gCP should always be applied together with a dispersion correction for DFT: gCP-D3 is well tested, but gCP-NL does also work well. (see sections 6.2.2.10 for DFT-D3 and 6.2.2.11 for DFT-NL)

6.2.2.13 HF-3c: Hartree-Fock with three corrections

HF-3c is a fast Hartree-Fock based method developed for computation of structures, vibrational frequencies and non-covalent interaction energies in huge molecular systems [158]. The starting point for evaluating the electronic energy is a standard HF calculation with a small Gaussian AO basis set. The used so-called MINIX basis set consists of different sets of basis functions for different groups of atoms as shown in table 6.6.

Three terms are added to correct the HF energy $E_{tot}^{HF/MINIX}$ in order to include London dispersion interactions, to account for the BSSE and to correct for basis set deficiencies, i.e. overestimated bond lengths. The corrected total energy is therefore calculated as

$$E_{tot}^{HF-3c} = E_{tot}^{HF/MINIX} + E_{disp}^{D3(BJ)} + E_{BSSE}^{gCP} + E_{SRB}. \tag{6.44}$$

The first correction term $E_{disp}^{D3(BJ)}$ is the atom-pair wise London dispersion energy from the D3 dispersion correction scheme [2] applying Becke-Johnson (BJ) damping [142–144] (see section 6.2.2.10). The second term E_{BSSE}^{gCP} denotes the geometrical counterpoise (gCP) correction [156] to treat the BSSE (see section 6.2.2.12). For the HF-3c method, the three usual D3 parameters s_8, a_1 and a_2 were re-fitted using reference interaction energies of the complexes of the

---
[9]http://www.thch.uni-bonn.de/tc/

Table 6.6: Composition of the MINIX basis set.

element	basis
H-He, B-Ne	MINIS
Li-Be	MINIS+1(p)
Na-Mg	MINIS+1(p)
Al-Ar	MINIS+1(d)
K-Zn	SV
Ga-Kr	SVP
Rb-Xe	def2-SV(P) with ECP SD(28,MDF) or OLD-SD(28,MDF)

S66 test set [153]. This results in $s_8 = 0.8777$, $a_1 = 0.4171$ and $a_2 = 2.9149$. The parameter s_6 was set to unity as usual to enforce the correct asymptotic limit and the gCP correction was already applied in this fitting step.

The last term E_{SRB} is a short-ranged correction to deal with basis set deficiencies which occur when using small or minimal basis sets. It corrects for systematically overestimated covalent bond lengths for electronegative elements and is calculated as a sum over all atom pairs:

$$E_{SRB} = -s \sum_A^{Atoms} \sum_{A \neq B}^{Atoms} (Z_A Z_B)^{3/2} \exp(-\gamma (R_{AB}^{0,D3})^{3/4} R_{AB}) \tag{6.45}$$

Here, $R_{AB}^{0,D3}$ are the default cut-off radii as determined *ab initio* for the D3 scheme [2] and Z_A, Z_B are the nuclear charges. This correction is applied for all elements up to argon. The empirical fitting parameters $s = 0.03$ and $\gamma = 0.7$ were determined to produce vanishing HF-3c total atomic forces for B3LYP-D3(BJ)/def2-TZVPP equilibrium structures of 107 small organic molecules. More details can be found in the original publication [158].

The HF-3c method can only be invoked with a simple keyword:

```
! HF-3c
```

`! HF-3c` is a compound keyword and equals `! HF MINIX D3BJ GCP(HF/MINIX) PATOM`, hence no basis set etc has to be specified. The `PATOM` guess is selected since the grid construction for the default guess can take more time than an actual SCF step. The guess can only be overwritten manually in the %method section.

The default mode for the integral handling is set to `Conventional`. The storing of the two-electron integrals on disk or in memory if possible leads to large computational savings. In case one want to use the `Direct` mode, this has to be specified in the %scf input section:

```
%scf
    SCFmode Direct
    end
```

The output gives the used parameters and the correction itself for D3 and gCP separately. As the SRB correction is also calculated with the otool_gcp, the results are given in the gCP output section. The `Total correction to HF/MINIX` is the sum of all three corrections (D3, gCP and SRB) and `FINAL SINGLE POINT ENERGY` is the total HF-3c energy as given in equation 6.44.

```
--------------------------------------------------------------------------------
                          DFT DISPERSION CORRECTION

                             DFTD3 V2.1  Rev 6
                         USING Becke-Johnson damping
--------------------------------------------------------------------------------
The default Hartree-Fock  is recognized
Active option DFTDOPT                    ...           4


molecular C6(AA) [au] = 1689.256597



          DFT-D V3
 parameters
 using HF/MINIX parameters
 s6 scaling factor       :     1.0000
 a1 scaling factor       :     0.4171
 s8 scaling factor       :     0.8777
 a2 scaling factor       :     2.9149
 ad hoc parameters k1-k3  :    16.0000      1.3333     -4.0000


 Edisp/kcal,au: -32.163184627631  -0.051255291794
 E6   /kcal   : -18.007221978
 E8   /kcal   : -14.155962649
 % E8         :  44.012938437


------------------------  ----------------
Dispersion correction         -0.051255292
------------------------  ----------------




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

                g C P  - geometrical counterpoise correction

--------------------------------------------------------------------------------
Method:  hf/minix

Parameters:    sigma        eta        alpha        beta
               0.1290     1.1526     1.1549       1.1763
Egcp:       0.0723150636 a.u.
Ebas:      -0.0636976872 a.u.
-----------------  ----------------
gCP+bas correction        0.008617376
-----------------  ----------------
--------------------------  ----------------
Total correction to HF/MINIX      -0.042637915
--------------------------  ----------------



------------------------  --------------------
 FINAL SINGLE POINT ENERGY    -163.002895262171
------------------------  --------------------
```

For the elements Rb to Xe ECPs are automatically used. In case the default initial guess does not work (due to convergence problems), the guess mode has to be changed to HCORE in the %scf input block. For calculations with

only lighter elements and therefore no ECPs, the ECP printouts in the output file can be ignored.

6.2.3 Second Order Møller-Plesset Perturbation Theory

A second order Møller-Plesset perturbation theory (MP2) [96, 140, 159–170] module was implemented in ORCA. At present it can be used to estimate the correlation energy to second order. Analytic gradients or response densities have been implemented too. The use of the MP2 module is fairly easy.

```
%method  Method MP2
         End

Or simply
! MP2
or
! RI-MP2
```

This command does the following (a) it changes the `Method` to `HFGTO` and (b) it sets the flag `DoMP2` to `true`. The program will then first carry out a Hartree-Fock SCF calculation and then estimate the correlation energy by MP2 theory. RHF, UHF and high-spin ROHF reference wavefunctions are permissible and the type of MP2 calculation to be carried out (for high-spin ROHF the gradients are not available) is automatically chosen based on the value of `HFTyp`. If the SCF is carried out conventionally the MP2 calculation will also be done in a conventional scheme unless the user forces the calculation to be direct. For `SCFMode=Direct` or `SCFMode=SemiDirect` the MP2 energy evaluation will be fully in the integral direct mode.

The following variables can be adjusted in the block [`MP2`]

```
%mp2
  EMin -6.0          # orbital energy cutoff that defines the
                     # frozen core (in Eh; default=-3.0)
  EMax 10.0          # orbital energy cutoff that defines the
                     # neglected virtual orbitals (in Eh;
                     # default=1000.0)
  EWin EMin,EMax     # the same, but accessed as array
                     #(respects settings in %method block!)
  MaxCore 750        # maximum amount of memory (in MB) to be
                     # used for integral buffering
                     # (default=50)
  ForceDirect false  # Force the calculation to be integral
                     # direct (default=false)
  RI      off        # use the RI approximation (default=off)
  Q1Opt              # For non-RI calculations a flag how to perform
                     # the first quarter transformation
                     #  1 - use double precision buffers (default)
                     #  2 - use single precision buffers. This reduces
                     #      the memory usage in the bottleneck step by
```

```
                               #       a factor of two. If several passes are re-
                               #       quired, the number of passes is reduced by
                               #       a factor of two.
                               # -1 - Use a disk based algorithm. This respects
                               #       the flags UCFLOAT,CFLOAT,UCDOUBLE and
                               #       CDOUBLE. (but BE CAREFUL with FLOAT)
       PrintLevel 2           # How much output to produce. PrintLevel 3 produces
                               # also pair correlation energies and other info.
       DoSCS   false          # use spin-component scaling ?
       Ps     1.2             # scaling factor for ab pairs
       Pt     0.333           # scaling factor for aa and bb pairs
       Densities none         # no density construction
                   unrelaxed  # only "unrelaxed densities"
                   relaxed    # full relaxed densities
       NatOrbs    true        # default is false
```

In "memory mode" (Q1Opt>0) basically the program treats batches of occupied orbitals at the same time. Thus, there must be at least enough memory to treat a single occupied MO at each pass. Otherwise the MP2 module will fail. Thus, potentially, MP2 calculations on large molecules take significant memory and may be most efficiently done through the RI approximation.

Alternatively, in the "disk based mode" (Q1Opt=-1) the program performs a half transformation of the exchange integrals and stores the transformed integrals on disk. A bin-sort then leads to the AO operator $K^{ij}(\mu,\nu) = (i\mu|j\nu)$ in (11|22) integral notation. These integrals are then used to make the final $K^{ij}(a,b)$ (a,b=virtual MOs) and the EMP2 pair energy contributions. In many cases, and in particular for larger molecules, this algorithm is much more efficient than the memory based algorithm. It depends, however, much more heavily on the I/O system of the computer that you use. It is important, that the program uses the flags `CFLOAT, UCFLOAT, CDOUBLE` or `UCDOUBLE` in order to store the unsorted and sorted AO exchange integrals. Which flag is used will influence the performance of the program and to some extent the accuracy of the result (float based single precision results are usually very slightly less accurate; microhartree-range deviations from the double precision result[10]). Finally, gradients are presently only available for the memory based algorithm since in this case a much larger set of integrals is required.

The program can be run with very large benefits in the RI mode [140, 168–170]. Here you *must* specify a suitable auxiliary basis set. So far auxiliary basis sets that were developed by Weigend et al. and obtained from the TurboMole library are implemented [140]:

```
   %basis
     Aux SV_C           # (=SVP_C,VDZ_C) suitable for split valence
                        # double-zeta basis sets like SVP
         TZV_C          # (=TZVP_C,VTZ_C) suitable for split valence
                        # triple-zeta basis sets like TZVP
         TZVPP_C        # (=VTZPP_C) suitable for ext. bases
         QZV_C          # (=QZVP_C) aux basis for QZVP (even larger)
         QZVPP_C        # aux basis for QZVPP (in ORCA identical with QZVP
```

---

[10]However, sometimes, and in particular when transition metals and core orbitals are involved we have met unpleasantly large errors. So – be careful and double check when using floats!

```
                        # both were taken from the QZVPP cbasen TurboMole
                        # library)
        DEF2_SV_C       # (=DEF2_SVP_C) suitable for def2-SVP
        DEF2_TZV_C      # (=DEF2_TZVP_C) suitable for def2-TZVP
        DEF2_TZVPP_C    # suitable for def2-TZVPP
        DEF2_QZV_C      # (=DEF2_QZVP_C) aux basis for QZVP (even larger)
        DEF2_QZVPP_C    # aux basis for def2-QZVPP (in ORCA identical with def2-QZVP
                        # both were taken from the QZVPP cbasen TurboMole
                        # library)
        cc_pVDZ_C       # suitable for cc-pVDZ
        cc_pVTZ_C       # suitable for cc-pVTZ
        cc_pVQZ_C       # suitable for cc-pVQZ
        cc_pV5Z_C       # suitable for cc-pV5Z
        cc_pV6Z_C       # suitable for cc-pV6Z
        cc_aug_pVDZ_C  # suitable for aug-cc-pVDZ
        cc_aug_pVTZ_C  # suitable for aug-cc-pVTZ
        cc_aug_pVQZ_C  # suitable for aug-cc-pVQZ
        cc_aug_pV5Z_C  # suitable for aug-cc-pV5Z
        cc_aug_pV6Z_C  # suitable for aug-cc-pV6Z
        cc_aug_SV_C     # (=cc_aug_SVP_C,cc_aug_VDZ_C) suitable for SVP plus diffuse
        cc_aug_TZV_C    # (=cc_aug_TZVP_C,cc_aug_VTZ_C) suitable for TZVP plus diffuse
        cc_aug_TZVPP_C # (=cc_aug_VTZPP) suitable for TZVPP plus diffuse
    end
```

These auxiliary bases cover a fairly extensive range of orbital expansion bases and they will also be appropriate for other orbital bases of similar quality. You can always try by using a very large aux-basis like cc_aug_TZVPP_C which will almost always be appropriate and see if the same result is reproduced with a smaller aux-basis too (of course within a given tolerance).

6.2.4 Semiempirical Methods

The present version of ORCA has inherited the capability of doing semiempirical calculations from the earlier versions. A number of methods based on the 'neglect of differential overlap' [171, 172] are currently implemented for energies and analytic gradients (for geometry optimization).

```
    %method
      Method  CNDO
              INDO
              NDDO
      # for Method=CNDO
      Version CNDO_1
              CNDO_2
              CNDO_S
      # for Method=INDO
```

```
    Version INDO_1
            INDO_2
            ZINDO_1
            ZINDO_2
            ZINDO_S
    # for Method=NDDO
    Version ZNDDO_1
            ZNDDO_2
            MNDO
            AM1
            PM3
 end
```

The methods MNDO [173–175], AM1 [176] and PM3 [177] are available for main group elements only and arise from the work of the Dewar group. They have been optimized to reproduce molecular structure and energetics. The older CNDO/1,2 and INDO/1,2 were developed by the Pople group [178–186] and were designed to roughly mimic minimal basis *ab initio* calculations. The methods of the Zerner group (ZINDO/1,2 and ZINDO/S) are closely related to the older methods but have been well parameterized for transition metals too [187–197]. ZINDO/1 (and less so ZINDO/2) are suitable for geometry optimization. ZINDO/S gives good results for electronically excited states at moderate configuration interaction levels and is also successful for the calculation of electron and spin distributions in large transition metal complexes [192–197]. The ZNDDO/1,2 methods have been implemented into ORCA as straightforward extensions of the corresponding INDO methods without changing any parameter. However, the methods benefit from the somewhat more accurate representation of the Coulomb interaction within the NDDO approximation [198, 199]. The preliminary experience with these methods is that they are better than the corresponding INDO methods for calculation of transition metal complex structures but on the whole have also similar deficiencies.

The analytic gradients are available for all of these methods and can be used to produce reasonable molecular structures at low computational cost or to get preliminary insight in the behavior of the system under investigation[11].

There is also a mechanism for simplified input. Instead of giving values for `Method` and `Version` separately you can also assign the value that would normally belong to `Method` to `Version`. The program will recognize that and assign the correct values to both `Method` and `Version`.

```
 %method
   # shortcut to Method=NDDO; and Version=AM1;
   Method  AM1
 end
```

- If you want you can also combine semiempirical methods with MP2 (energies only). For example use `Method=AM1;` and `DoMP2=true;` It is questionable if this makes the results of semiempirical calculations any better but at least it is possible in ORCA.

You can change the built-in semiempirical parameters in a straightforward fashion. For example:

---

[11]However, do not try to use ZINDO/S (or CNDO/S) for structure optimizations - it doesn't make sense and will lead to disastrous results because there is no accurate representation of nuclear repulsion in these methods.

```
    ! RHF ZINDO/S TightSCF DIIS NoRICO NoMOPrint


%cis    NRoots 20
        MaxDim 60
        end


%ndoparas  P[6,25] 20
           P[6,26] 20
           end
```

The **%ndoparas** block is there in order to let you input your favorite personal parameters. The "molecular" parameters are set using "**INTFA**" ('interaction factors');

```
%ndoparas  INTFA[PP_PI] 0.585
  # The interaction factors exist for
  #  ss_sigma
  #  sp_sigma
  #  sd_sigma
  #  pp_sigma
  #  pd_sigma
  #  dd_sigma
  #  pp_pi
  #  pd_pi
  #  dd_pi
  #  dd_delta
  # the parameter entering the Coulomb integrals
  # in INDO/S
  FGAMMA 1.2
  end
  end
```

All atomic parameters are collected in an array "P". The first index is the atomic number of the element whose parameters you want to change. The second index identifies which parameter. The list of parameters follows below. Most of them will only be interesting for expert users. The most commonly modified parameters are the Beta's (number 25 through 28). Note that most programs require a negative number here. In ORCA the resonance integrals are defined in a way that makes the Beta's positive.

```
# core integrals (in eV)
US      0
UP      1
UD      2
UF      3
# Basis set parameters (double zeta for generality)
NSH     4  # number of shells for the element
```

```
NZS    5   # number of Slater type orbitals for the s-shell
ZS1    6   # first exponent
ZS2    7   # second exponent
CS1    8   # first contraction coefficient
CS2    9   # second contraction coefficient
NZP   10   # number of Slater type orbitals for the p-shell
ZP1   11   # ...
ZP2   12
CP1   13
CP2   14
NZD   15   # number of Slater type orbitals for the d-shell
ZD1   16   # ...
ZD2   17
CD1   18
CD2   19
NZF   20   # number of Slater type orbitals for the f-shell
ZF1   21   # ...
ZF2   22
CF1   23
CF2   24
# Resonance integral parameters (in eV)
BS    25   # s-shell beta
BP    26   # p-shell beta
BD    27   # d-shell beta
BF    28   # f-shell beta
# Number of electrons in the g.s.
NEL   29   # total number of electrons (integer)
NS    30   # fractional occupation number of the s-shell
NP    31   # fractional occupation number of the p-shell
ND    32   # fractional occupation number of the d-shell
NF    33   # fractional occupation number of the f-shell
# The one center repulsion (gamma) integrals (in eV)
GSS   34
GSP   35
GSD   36
GSF   37
GPP   38
GPD   39
GPF   40
GDD   41
GDF   42
GFF   43
# The Slater Condon parameters (in eV)
F2PP  44
F2PD  45
```

```
    F2DD   46
    F4DD   47
    G1SP   48
    G1PD   49
    G2SD   50
    G3PD   51
    R1SPPD 52
    R2SDPP 53
    R2SDDD 54
    # The nuclear repulsion parameters for Dewar type models
    NR1    55
    NR2    56
    NR3    57
    NR4    58
    NR5    59
    NR6    60
    NR7    61
    NR8    62
    NR9    63
    NR10   64
    NR11   65
    NR12   66
    NR13   67
    # The nuclear attraction/repulsion parameter for MNDO/d
    RHO    68
    # Spin orbit coupling parameters
    SOCP   69 # SOC for the p shell
    SOCD   70 # SOC for the d shell
    SOCF   71 # SOC for the f shell
```

6.3 Choice of Basis Set

A fair number of reasonable basis sets is hardwired in the program as will be described in the next section. In addition whole basis sets can be read from a file, basis sets can be assigned for all atoms of a given type or, at the highest resolution, basis sets can be assigned to individual atoms which is convenient if different parts of the molecule are to be treated at different levels of accuracy. All hard wired basis sets were obtained from the EMSL library [200] and the input format in ORCA is consistent with the EMSL library format.

6.3.1 Built-in Basis Sets

The basis set is specified in the block [BASIS]. The format is straightforward.

```
%basis
  #-----------------------------------------------
  # the basic orbital expansion basis
  #-----------------------------------------------
  Basis MINI      # Huzinaga minimal basis [201]
        MINIS     # Huzinaga minimal basis (valence scaled version) [201]
        MIDI      # Huzinaga split valence [201]
        MINIX     # Grimme minimal basis set
        STO_NG    # STO-NG expansion. Default N=3 [202] [203] [204] [205] [206]
                  # [207]
        _3_21G    # Pople style 3-21G basis [208] [209] [210] [211]
        _6_31G    # Pople style 6-31G basis [212] [213] [214] [215] [216] [217]
        _6_311G   # Pople style 6-311G basis [218] [219] [220] [221] [222] [216]
        _3_21GSP  # Buenker's modification of 3-21G [223] [224]
        _4_22GSP  # Buenker 4-22G basis [223] [224]
        VDZ       # Ahlrichs VDZ basis (usually refered to
        SV        # as SV basis) [225]
        VTZ       # Ahlrichs VTZ basis [225]
        TZV       # Ahlrichs TZV basis. Similar but not
                  # identical to VTZ. Coincides with the
                  # TurboMole TZV basis set
        QZV       # TurboMole QZV basis set
        DZ        # a good quality double-zeta set from
                  # the TurboMole library
        DGauss    # DGauss DZVP-DFT basis [226]
        cc_VDZ    # Dunning cc-VDZ set with no pol. [227] [228] [229] [230]
        cc_VTZ    # Dunning cc-VTZ set with no pol. [227] [228] [229] [230]
        cc_VQZ    # Dunning cc-VQZ set with no pol. [227] [228] [229] [230]
        cc_V5Z    # Dunning cc-V5Z set with no pol. [227] [228] [229] [230]
        cc_V6Z    # Dunning cc-V6Z set with no pol. [227] [228] [229] [230]
        Sadlej_pVTZ# Sadlej's triple-zeta plus pol.
        Sadlej    # basis optimized for polariza-
                  # bility calculations.
                  # It is of valence-triple-zeta plus 2d
                  # quality.
        Dunning_DZ # Dunnings double-zeta basis
        CP        # 'core-properties' basis. More extended
        COREPROP  # in the core region. At the moment
                  # only for first row transition metals.
        # These are fairly large "polarization consistent"
        # basis sets from Frank Jensen that are available for
        # H-Ar
        PC_1 (PC1)
        PC_2 (PC2)
        PC_3 (PC3)
```

```
            PC_4 (PC4)
            Aug_PC_1 (Aug_PC1)
            Aug_PC_2 (Aug_PC2)
            Aug_PC_3 (Aug_PC3)
            Aug_PC_4 (Aug_PC4)
            # These basis sets already contain appropriate
            # polarization functions. EPRII+IGLO are only
            # available for some light main group atoms,
            # Wachter's is only available for the first tran-
            # sition row
              EPRII    # Barone's basis for hyperfine calcs [231]
              EPRIII   # Barone's basis for hyperfine calcs [231]
              IGLO_II  # Kutzelnigg's NMR/EPR basis set [232]
              IGLO_III # Kutzelnigg's NMR/EPR basis set [232]
              Wachters # Wachter's basis (+3f->(2,1)) [233] [234]
            # Basis sets for relativistic
            # ZORA(IORA) and DKH calculations. See more in the
            # chapter on relativistic methods.
              SV_ZORA
              TZV_ZORA
              QZV_ZORA
              SV_DKH
              TZV_DKH
              QZV_DKH
              CC_PVTZ_DKH
              CC_PVQZ_DKH
              CC_PV5Z_DKH
            # ANO bases with built-in polarization functions
              Bonn_ANO_DZP     # double-zeta-ANO contraction (3d->1d)
              Bonn_ANO_TZ2P    # triple-zeta ANO contraction (3d->2d)
              Bonn_ANO_TZ3P    # triple-zeta ANO contraction (3d2f->2d1f)
              BNANO_DZP        # double-zeta-ANO contraction (3d->1d)
              BNANO_TZ2P       # triple-zeta ANO contraction (3d->2d)
              BNANO_TZ3P       # triple-zeta ANO contraction (3d2f->2d1f)
              NASA_AMES_ANO    # the NASA/AMES ANO bases
              Bauschlicher_ANO # first row transition metal ANOs
              Roos_ANO_DZP     # augmented Roos double-zeta ANO. It is
                               # of the same size as our TZ2P
              Roos_ANO_TZP     # Very large and expensive ANO set
            # Various ANO basis sets
              ANO_SZ
              ANO_PVDZ
              ANO_PVDZ
              ANO_PVTZ
              ANO_PVQZ
```

```
            ANO_PV5Z
            ANO_PV6Z
            SAUG_ANO_SZ
            SAUG_ANO_PVDZ
            SAUG_ANO_PVTZ
            SAUG_ANO_PVQZ
            SAUG_ANO_PV5Z
            AUG_ANO_SZ
            AUG_ANO_PVDZ
            AUG_ANO_PVTZ
            AUG_ANO_PVQZ
            AUG_ANO_PV5Z
       # These are HF limit quality uncontracted bases
       # that do not contain polarization functions by
       # themselves
         Partridge_1  # large (up to Sr) [235] [236]
         Partridge_2  # larger (up to Kr) [235] [236]
         Partridge_3  # even larger (up to Zn) [235] [236]
       # Bases using STOs
         STO_DZ
         STO_DZP
         STO_VDZ
         STO_VDZP
         STO_SZ
         STO_POPLE
         ADF_II
         ADF_IV
#----------------------------------------------------
# the polarization functions
#----------------------------------------------------
Pol  0            # no polarization functions
     _d           # = Pople(d) (from 6-311+(d) [218] [219] [220] [221] [222]
                  # [216])
     _dp          # = Pople (d,p) (from 6-311+(d) [218] [219] [220] [221] [222]
                  # [216])
     _2d          # = Pople (2d) [218] [219] [220] [221] [222] [216]
     _2d2p        # = Pople (2d,2p) [218] [219] [220] [221] [222] [216]
     _2df         # = Pople (2df) [218] [219] [220] [221] [222] [216]
     _2df2pd      # = Pople (2df,2pd) [218] [219] [220] [221] [222] [216]
     _3df         # = Pople (3df) [218] [219] [220] [221] [222] [216]
     _3df3pd      # = Pople (3df,3pd) [218] [219] [220] [221] [222] [216]
     Ahlrichs_d   # Ahlrichs (d) [237]
     Ahlrichs_dp  # Ahlrichs (d,p) [237]
     Ahlrichs_2d      # Ahlrichs (2d) [237]
     Ahlrichs_2d2p    # Ahlrichs (2d,2p) [237]
```

```
        Ahlrichs_2df    # Ahlrichs (2df) [237]
        Ahlrichs_2df2pd # Ahlrichs (2df,2pd) [237]
        Ahlrichs_ppp    # Ahlrichs extended pol. [237]
                        # for Sc-Zn (2pf) from Wachters and
                        # the TurboMole library
        VDZ_D           # (= VTZ_D) polarization for SV(P)
        VDZ_PD          # (= VTZ_PD) polarization for SVP,TZVP, ...
        GAMESS_d     # Gamess (d) [238]
        GAMESS_dp    # Gamess (d,p) [238]
        cc_d         # pol. of cc-pVDZ but not for H [227] [228] [229] [230]
        cc_dp        # pol. of cc-pVDZ [227] [228] [229] [230]
        cc_2df       # pol. of cc-pVTZ but not for H [227] [228] [229] [230]
        cc_2df2pd    # pol. of cc-pVTZ [227] [228] [229] [230]
   # for STO basis sets
        POL_SZ_CLEMENTI
        POL_ADF
#--------------------------------------------------
# Core polarization functions
#--------------------------------------------------
CorePol    CPVDZ  # core polarization functions for cc-pCVDZ
           CPVTZ  # core polarization functions for cc-pCVTZ
           CPVQZ  # core polarization functions for cc-pCVQZ
           CPV5Z  # core polarization functions for cc-pCV5Z
#--------------------------------------------------
#diffuse functions
#--------------------------------------------------
Diff  0           # no diffuse functions
    _p            # = Pople + [218] [219] [220] [221] [222] [216]
    _pp           # = Pople ++ [218] [219] [220] [221] [222] [216]
    GAMESS_p      # = Gamess + [218] [219] [220] [221] [222] [216]
    GAMESS_pp     # = Gamess ++ [218] [219] [220] [221] [222] [216]
    aug_cc_VDZ    # diff of aug-cc-pVDZ but not for H [227] [228] [229] [230]
    aug_cc_VDZ_H # diff of aug-cc-pVDZ [227] [228] [229] [230]
    aug_cc_VTZ    # diff of aug-cc-pVTZ but not for H [227] [228] [229] [230]
    aug_cc_VTZ_H # diff of aug-cc-pVTZ [227] [228] [229] [230]
    aug_cc_VQZ # diff of aug-cc-pVQZ
    aug_cc_V5Z # diff of aug-cc-pV5Z
    aug_cc_V6Z # diff of aug-cc-pV6Z
    RYD_SZ_BURNS # for STO calculations
#--------------------------------------------------
# STO Core functions
#--------------------------------------------------
Core CORE_SZ_CLEMENTI
     CORE_DZ_CLEMENTI
     CORE_SZ_POPLE
```

```
        CORE_SZ_ADF
        CORE_DZ_ADF
    #------------------------------------------------
    # STO Valence functions
    #------------------------------------------------
    Val  VAL_SZ_CLEMENTI
         VAL_DZ_CLEMENTI
         VAL_SZ_POPLE
         VAL_ADF_II
         VAL_ADF_IV
    #------------------------------------------------
    # A flag to decontracting the basis
    #------------------------------------------------
    Decontract false  # if chosen "true" the program will
                      # decontract the basis set
  end
```

WARNING:

- If you use `Decontract`: if your basis set arises from general contraction it will contain duplicate primitives in several contractions and these will be removed such that only uniqe primitives remain and there is no problem with redundancy.

In addition to these basis sets there are a few assignments for `Basis` that are compounds keys and automatically assign values to `Pol` and `Diff`. Note also that the DGauss basis [226] has polarization functions built in and specifying additional 'Pol' functions will probably lead to disastrous results due to near linear dependence of the basis set.

A note of caution: some of the Pople style polarization exponents were optimized with 6D and 10F functions whereas ORCA can only use 5D and 7F functions. The Pople style _d and _dp polarization exponents were taken from the 6-311++G** basis set [216, 218–222]. They have different d-polarization exponents from the 6-31G** basis. This is to be taken into account when results are compared with other programs. In the case that a more original 6-31G** basis set is wanted you have to use `Basis=_6_31G_d` or `_6_31G_dp`. This will produce the correct exponents but still ORCA will use 5D and 7F functions.

The compound keys refer to the Ahlrichs [225] - (that are highly recommended because of their accuracy and efficiency) and Dunning [227–230] basis set (that are accurate but expensive in ORCA because they are treated as segmented contractions but have been developed for general contraction).

```
  %basis
    # the basic orbital expansion basis
    Basis VDZ_P       # = Ahlrichs SV(P). No polarization on H
          VDZP        # = Ahlrichs SVP. Plus polarization on H
                      # (synonyms are SV_P,SVP and PVDZ)
          VDZ2D       # with Ahlrichs-(2d)
          VDZ2D2P     # with Ahlrichs-(2d,2p)
          VDZ_PP      # with Ahlrichs-pp. No polarization on H
          VDZPP       # with Ahlrichs-pp. Plus polarization on H
```

```
          VDZPPP      # with Ahlrichs-ppp. Plus polarization on H
          VTZ_P       # Ahlrichs VTZ. No polarization on H
          VTZP        # Ahlrichs VTZ. Plus polarization on H
                      # (synonym PVTZ)
          VTZ2D       # with Ahlrichs-(2d)
          VTZ2D2P     # with Ahlrichs-(2d,2p)
          VTZ_PP      # with Ahlrichs-pp. No polarization on H
          VTZPP       # with Ahlrichs-pp. Plus polarization on H
          VTZPPP      # with Ahlrichs-ppp. Plus polarization on H
          TZV_P       # similar for TZV basis. No polarization on H
          TZVP        # similar for TZV basis. Plus polarization on H
          TZV2D       # similar with Ahlrichs-(2d)
          TZV2D2P     # similar with Ahlrichs-(2d,2p)
          TZV_PP      # similar with Ahlrichs-pp. No polarization on H
                      # (synonym TZV2DF)
          TZVPP       # similar with Ahlrichs-pp (synonym TZV2DF2PD)
          TZVPPP      # similar with Ahlrichs-ppp
          QZVP        # quadruple zeta with polarization sets
          QZVPP       # more polarized (identical to def2-QZVPP)
          QZVPPP      # extensively polarized
          Def2SV_P  # SV with "new" (P) sets
          Def2SVP     # SV with "new" P sets
          Def2TZVP    # TZV with "new" P sets
          Def2TZVPP   # TZV with "new" PP sets
          Def2QZVPP   # QZVP with "new" extensive polarization
          Def2SVPD    # Rappoport's property optimized D double-zeta basis set
          Def2TZVPD   # Rappoport's property optimized D triple-zeta basis set
          Def2TZVPPD # Rappoport's property optimized D triple-zeta basis set
          Def2QZVPD   # Rappoport's property optimized D quadruple-zeta basis set
          Def2QZVPPD # Rappoport's property optimized D quadruple-zeta basis set
          ma-def2-SVP   # Minimally augmented diffuse def2-SVP basis set.
          ma-def2-TZVP  # Minimally augmented diffuse def2-TZVP basis set.
          ma-def2-TZVPP # Minimally augmented diffuse def2-TZVPP basis set.
          ma-def2-QZVPP  # Minimally augmented diffuse def2-QZVPP basis set.
          ma-ZORA-def2-SVP   # All-electron ZORA relativistic ma-def2-SVP basis set.
          ma-ZORA-def2-TZVP  # All-electron ZORA relativistic ma-def2-TZVP basis set.
          ma-ZORA-def2-TZVPP # All-electron ZORA relativistic ma-def2-TZVPP basis set.
          ma-ZORA-def2-QZVPP  # All-electron ZORA relativistic ma-def2-QZVPP basis set.
          ma-DKH-def2-SVP   # All-electron DKH relativistic ma-def2-SVP basis set.
          ma-DKH-def2-TZVP  # All-electron DKH relativistic ma-def2-TZVP basis set.
          ma-DKH-def2-TZVPP # All-electron DKH relativistic ma-def2-TZVPP basis set.
          ma-DKH-def2-QZVPP  # All-electron DKH relativistic ma-def2-QZVPP basis set.
          DZVP_DFT    # Godbouts double-zeta basis set designed for DFT
                      # (synonym DZVP_DGAUSS)
        # ... the same is possible with VTZ instead of TZV
```

```
            # it is also possible to add "aug_" in front in order
            # to get the diffuse functions from the Dunning bases
            # together with the Ahlrichs basis sets
             DZ_P          # similar for the DZ basis
             DZP           #
             DZ2D          # with Ahlrichs-(2d)
             DZ2D2P        # with Ahlrichs-(2d,2p)
             DZ_PP         # with Ahlrichs-pp except for H
             DZPP          # with Ahlrichs-pp
             DZPPP         # with Ahlrichs-ppp
             cc_pVDZ       # Dunning cc-pVDZ basis
             cc_pVTZ       # Dunning cc-pVTZ basis
             cc_pVQZ       # Dunning cc-pVQZ basis
             cc_pV5Z       # Dunning cc-pV5Z basis
             cc_pV6Z       # Dunning cc-pV6Z basis
             aug_cc_pVDZ   # Dunning aug-cc-pVDZ basis
             aug_cc_pVTZ   # Dunning aug-cc-pVTZ basis
             aug_cc_pVQZ   # Dunning aug-cc-pVQZ basis
             aug_cc_pV5Z   # Dunning aug-cc-pV5Z basis
             aug_cc_pV6Z   # Dunning aug-cc-pV6Z basis
             cc_pCVDZ      # Dunning core-polarized cc-pVDZ basis
             cc_pCVTZ      # Dunning core-polarized cc-pVTZ basis
             cc_pCVQZ      # Dunning core-polarized cc-pVQZ basis
             cc_pCV5Z      # Dunning core-polarized cc-pV5Z basis
             cc_pCV6Z      # Dunning core-polarized cc-pV6Z basis
             Dunning_DZ_P # the popular Dunning double-zeta plus
                           # polarization basis except for H
             Dunning_DZP  # the popular Dunning double-zeta plus
                           # polarization basis
             6_31G_dp      # the 6-31G** and 6-31G* bases with
             6_31G_d       # the correct pol. exponents BUT
                           # still only 5d and  7f functions!
        end
```

The auxiliary basis is specified by **AUX**:

```
    %basis
      # the auxiliary basis suitable for methods without HF
      # exchange (normally 'pure' DFT)
      Aux  SV_J           # (=VDZ_J,SVP_J) RI-J aux basis for SVP basis and
                          # similar bases. [119] [120]
           SV_J_SMALL     # (=VDZ_J_SMALL,SVP_J_SMALL,VDZ_J_NOF,SVP_J_NOF,SV_J_NOF)
                          # smaller RI-J aux basis for SVP basis and similar bases.
           TZV_J          # (=VTZ_J,TZVP_J) RI-J aux basis for TZVP and higher
```

```
                              # [119] [120]
        TZV_J_SMALL      # (=VTZ_J_SMALL,TZVP_J_SMALL,VTZ_J_NOF,TZVP_J_NOF,TZV_J_NOF)
                         # smaller RI-J aux basis for TZVP basis and similar bases.
        TZVPP_J          # RI-J aux basis for TZVPP
        TZVPP_J_SMALL    # smaller RI-J aux basis for TZVPP
        DEF2_SV_J        # (=DEF2_SVP_J) newer SV aux basis
        DEF2_TZV_J       # (=DEF2_TZVP_J) newer TZV aux basis
        DEF2_TZVPP_J     # newer TZVPP aux basis
        DEF2_QZVPP_J     # newer QZVPP aux basis
                         # in my experience TZV/J also works in conjunction
                         # with EPR and IGLO bases.
        DGauss_A1_J      # (=A1) RI-J basis for DGauss basis [226]
        DGauss_A1_X      # RI-X basis for DGauss basis [226]
        DGauss_A2_J      # (=A2) RI-J basis for DGauss basis [226]
        DGauss_A2_X      # RI-X basis for DGauss basis [226]
        DeMon_J          # (=DeMon) RI-J basis of the DeMon program [226]
  # for correlation calculations (MP2,MR-CI) and also for
  # TD-DFT/CIS calculations with Hartree-Fock exchange
  Aux  SV_C             # RI-C basis for SVP and similar
        TZV_C            # RI-C basis for TZVP and similar
        TZVPP_C          # RI-C basis for TZVPP and similar
        QZVP_C           # RI-C basis for QZVP (and QZVPP)
        DEF2_SV_C        # RI-C basis for SVP and similar
        DEF2_TZV_C       # RI-C basis for TZVP and similar
        DEF2_TZVPP_C     # RI-C basis for TZVPP and similar
        DEF2_QZVP_C      # RI-C basis for QZVP (and QZVPP)
        CC_pVDZ_C        # RI-C basis for cc-pVDZ and similar
        CC_pVTZ_C        # RI-C basis for cc-pVTZ and similar
        CC_pVQZ_C        # RI-C basis for cc-pVQZ and similar
        CC_pV5Z_C        # RI-C basis for cc-pV5Z and similar
        CC_pV6Z_C        # RI-C basis for cc-pV6Z and similar
        AUG_CC_pVDZ_C    # RI-C basis for aug-cc-pVDZ
        AUG_CC_pVTZ_C    # RI-C basis for aug-cc-pVTZ
        AUG_CC_pVQZ_C    # RI-C basis for aug-cc-pVQZ
        AUG_CC_pV5Z_C    # RI-C basis for aug-cc-pV5Z
        AUG_CC_pV6Z_C    # RI-C basis for aug-cc-pV6Z
        AUG_SV_C         # RI-C for SVP+diffuse functions
        AUG_TZV_C        # RI-C for TZVP + diffuse functions
        AUG_TZVPP_C      # RI-C for TZVPP + diffuse
  # Aux bases for Coulomb and exchange fitting
        SV_JK            # RI-JK basis for SVP and similar
        TZV_JK           # RI-JK basis for TZVP and similar
        TZVPP_JK         # RI-JK basis for TZVPP and similar
        QZVP_JK          # RI-JK basis for QZVP (and QZVPP)
        DEF2_SV_JK       # RI-JK basis for SVP and similar
```

```
        DEF2_TZV_JK      # RI-JK basis for TZVP and similar
        DEF2_TZVPP_JK    # RI-JK basis for TZVPP and similar
        DEF2_QZVP_JK     # RI-JK basis for QZVP (and QZVPP)
        CC_pVDZ_JK       # RI-JK basis for cc-pVDZ and similar
        CC_pVTZ_JK       # RI-JK basis for cc-pVTZ and similar
        CC_pVQZ_JK       # RI-JK basis for cc-pVQZ and similar
        CC_pV5Z_JK       # RI-JK basis for cc-pV5Z and similar
        CC_pV6Z_JK       # RI-JK basis for cc-pV6Z and similar
        AUG_CC_pVDZ_JK # RI-JK basis for aug-cc-pVDZ
        AUG_CC_pVTZ_JK # RI-JK basis for aug-cc-pVTZ
        AUG_CC_pVQZ_JK # RI-JK basis for aug-cc-pVQZ
        AUG_CC_pV5Z_JK # RI-JK basis for aug-cc-pV5Z
        AUG_CC_pV6Z_JK # RI-JK basis for aug-cc-pV6Z
    # a general purpose fitting basis can be constructed
    # automatically. This creates ucontracted aux-bases that
    # are larger than the built in ones but give similar
    # accuracy. Useful in case that no built-in set
    # is available or one wants to "play safe"
    Aux   auto  # synonym is autobuild
  end
```

A comment on the DGauss_A1_J basis: the DGauss program uses a more accurate fitting scheme and can therefore get away with a smaller fitting basis than ORCA. Consequently applying the DGauss_A1_J auxiliary basis together with the DGauss DZVP orbital basis will give rise to significant errors. This is *not* the fault of the DGauss expansion basis but is most likely a consequence of the less accurate fit used in ORCA. It is recommended to use the SV_J aux basis together with the DGauss basis (and also 6-31G and other split valence double zeta basis sets) because it is larger and will usually give rise to more accurate results. For example for the molecule $[Cu(NH_3)_4]^{2+}$ I obtained an error in the total energy of 4.3 mEh for the SV_J aux basis and DGauss orbital basis and 20.3 mEh with the DGauss A1 expansion basis compared to the exact result (with the same integration grid). Remarkably in my test calculation the DeMon_J aux basis was as accurate as the SV_J aux basis despite the fact that it is smaller. This however is an isolated example and an extended set of test calculations would be necessary to find out whether this is a general conclusion. Note that beyond Ne the TZV_J expansion basis is the same as SV_J but for the first row TZV_J is more accurate than SV_J.

One thing that is certainly feasible and reasonable if you do not want to depend on the RI approximation is to converge a RI-J calculation and then take the resulting orbitals as initial guess for a calculation with exact Coulomb term. This should converge within a few cycles and the total execution time should still be lower than just converging the calculation directly with exact Coulomb treatment.

6.3.2 Assigning or Adding Basis Functions to an Element

In order to assign a new basis set to a given element, use:

```
%basis
  NewGTO 8    # new basis for oxygen
   S   3
```

```
      1   910.10034975            0.03280967
      2   137.19711335            0.23422391
      3    30.85279077            0.81490980
   L   2
      1      1.72885887           0.27389659          0.48155753
      2      0.39954770           0.79112437          0.65447861
   L   1
      1      8.35065975           1.00000000          1.00000000
    end
  end
```

Note that for simplicity and consistency the input format is the same as that used in the basis set files. In this format the first line carries first the angular momentum of the shell to be added (s, p, d, f, g or L) and the number of primitives. Then for each primitive one line follows which has (a) the index of the primitive (1, 2, 3, ...) (b) the exponent of the primitive and (c) the contraction coefficient (unnormalized). Note that ORCA always uses spherical harmonic gaussian functions and that L-shells are internally *not* handled as such but are treated as separate s- and p-shells. There also is the possibility to include a **SCALE X** statement after the number of primitives in the first line to indicate that the basis function exponents should be scaled.

In order to add basis functions to the basis of a given element (for example because you don't like the standard polarization functions) use **AddGTOs** instead or **NewGTOs**. In **NewGTOs** or **AddGTOs** you can also use the nicknames of internally stored basis sets. An example is:

```
  %basis
    NewGTO 8      # new basis for oxygen
     "6-31G"
      D  1
      1  0.4  1.0
    end
  end
```

In this example the 6-31G basis is assigned to oxygen and in addition a polarization function with exponent 0.4 is added to the oxygen basis. The nickname mechanism also understands Pople style assignments like "6-311++G(d,p)".

A similar mechanism was established for the auxiliary basis sets in RI calculations:

```
  %basis
    NewAuxGTO 8     # new auxiliary basis for oxygen
      s  1
      1  350  1.0
      ... etc
    end
    AddAuxGTO 8    # add a shell to the auxiliary basis for
                   # oxygen
      D  1
```

```
      1  0.8  1.0
    end
  end
```

6.3.3 Assigning or Adding Basis Functions to Individual Atoms

Sometimes you may want to not treat all carbon atoms with the same basis set but to assign a specific basis set to a specific atom in the molecules. This is also possible in ORCA and takes place in the section [COORDS]. The format is the same as described above. An example may help to make things clear:

```
%coords
  CTyp   =  Internal;   # choose internal coordinates
  Units  =  Angs;       # choose Angström units
  Charge =    0;        # total charge
  Mult   =    1;        # spin multiplicity
  Coords                # start coordinate assignments
    C(1)  0 0 0 0.00   0.0   0.00
      AddGTO
        D  1
          1  1.0  1.0
      end;
    O(2)  1 0 0 1.13   0.0   0.00
      NewGTO
        "6-311G"
        D  1
          1  1.2  1.0
      end;
  end;
end
```

In this example an extra d-shell with exponent 1.0 is added to the first carbon atom and the basis for the oxygen atom is changed to 6-311G with an extra d-function of exponent 1.2 added.

Note:

- AUX basis sets can not yet individually specified. This will change in the future.

6.3.4 Linear Dependence

The previous sections describe the assessment of a desired molecular basis set from appropriately parametrized functions at various locations within the molecule (normally centered on atoms). The parametrization of these functions is such that the chance for redundancy is minimal. Since however, one is limited to work with finite numerical precision, and furthermore these parameters also depend on the molecular geometry, redundancies cannot be completely eliminated in advance. Redundancy means that the subspace spanned by the given basis functions at given values of parameters (including geometry), can be identically spanned by a smaller number of *linear independent* basis

functions. Linear dependent (redundant) function sets however may cause numerical instabilities. Linear dependence is normally identified by searching for zero eigenvalues of the overlap matrix. Note that the inverse of the overlap (or related matrices) are used for orthogonalization purposes, and it follows that if near zero eigenvalues are not treated properly, the inverse becomes ill-defined, and the SCF procedure numerically unstable.

From the previous discussion, it is evident that the crucial parameter for curing linear dependence is the threshold below which an overlap eigenvalue is considered zero. This parameter may be changed using the following keyword

```
%scf
   sthresh 1e-7 # default 1e-8
   end
```

Although there is no strict limit for the value of the above parameter, it should reasonably be somewhere between 1e-5 and 1e-8 (the latter is the default). One may get away with 1e-9 or perhaps even lower without convergence problem, but there is a risk that the result is contaminated with noise caused by the near zero vectors. In difficult cases, an 1e-7 threshold was often found to work smoothly, and above that one risks throwing away more and more functions, which also influence comparability of results with other calculations. To monitor the behaviour of the small eigenvalues, one should look for the following block in the output

```
   Diagonalization of the overlap matrix:
   Smallest eigenvalue                   ... -1.340e-17
   Time for diagonalization              ...    0.313 sec
   Threshold for overlap eigenvalues     ... 1.000e-08
   Number of eigenvalues below threshold ... 1
   Time for construction of square roots ...    0.073 sec
   Total time needed                     ...    0.387 sec
```

Here, the smallest eigenvalue is printed, along with the currently used overlap threshold, and the number of functions below this (which will be dropped). It is a recommended consistency check to look for an equal number of zero entries among orbital energies once the SCF procedure converged. Note that for functions belonging to zero eigenvalues *no level shifts* are applied!

6.3.5 Reading Orbital and Auxiliary Basis Sets from a File

By using the variables `GTOName` and `GTOAuxName` a basis set can be read from an ASCII file[12]. The format is that used for 'GAMESS' in the EMSL library [200].

```
   %basis
     # read an externally specified orbital basis
     GTOName   = "MyBasis.bas";
     # read an externally specified basis for RI
     # calculations
     GTOAuxName= "MyAuxBasis.bas";
     # for STO basis
     STOName= "MySTOBasis.bas"
     end
```

---

[12] In order to not confuse the program it is recommended to assign some value to `Basis` and `Aux` and then 'overrule' this default with the external basis set. The externally specified basis set then takes priority.

To give an example of how this format looks like here is a part of the `3-21GSP` basis of Buenker and coworkers [223,224]:

```
3-21GSP                   the name of the basis
!                         all these lines in the beginning with '!' are comments
! BASIS="3-21GSP"
!Elements                           References
!--------                           ----------
! H - Ne: A.V. Mitin, G. Hirsch, R. J. Buenker, Chem. Phys. Lett. 259, 151 (1996)
! Na - Ar: A.V. Mitin, G. Hirsch, R. J. Buenker, J. Comp. Chem. 18, 1200 (1997).
!
HYDROGEN      ! (3s) -> [2s]
  S   2
   1           4.50036231          0.15631167
   2           0.68128924          0.90466909
  S   1
   1           0.15137639          1.00000000
CARBON        ! (6s,3p) -> [3s,2p]
  S   3
   1         499.24042249          0.03330322
   2          75.25419194          0.23617745
   3          16.86538669          0.81336259
  L   2        L shells are a s and a p shell with identical exponents
   1           0.89739483          0.24008573          0.46214684
   2           0.21746772          0.81603757          0.66529098
  L   1
   1           4.52660451          1.00000000          1.00000000
STOP
```

In this way you can construct your favorite standard basis set and load it easily into the program. A word of caution: in C/C++ the backslashes in directory assignments must be given twice to be correctly understood! The file format for the auxiliary basis sets is exactly the same.

6.3.6 Advanced Specification of Effective Core Potentials

Library ECPs and Basis Sets

Besides the simple input line (section 3.2.5), assignment of ECPs and associated basis sets can be done within the `%basis` block using the `NewECP` keyword. This provides low-level access to the internal ORCA ECP library, so that specific combinations of pseudopotentials and valence basis sets that already exist in the program library can be assigned to specific elements. Moreover, it allows manual specification of new ECPs directly in the input file.

For library ECPs, one of the keywords listed in the second column of table 3.6 (page 26) must be used. If required, a valence basis set is requested in the same line, using the keywords listed in the fourth column of table 3.6. The utility program `orca_ecplib` can be used to get a list of all library ECPs and valence basis sets for any element.

In its most simple format, the `NewECP` keyword is used as in the following example, which is equivalent (for Pt) to the simple `ECP{def2-TZVP}` or the more verbose `ECP{def2-SD,def2-TZVP}` keyword line statements:

```
%basis
    NewECP Pt "SD(60,MWB)" "def2-TZVP" end
end
```

Similarly for a Los Alamos ECP and valence basis set:

```
        NewECP Fe "LANL(10)" "LANLTZ" end
```

Note that the second argument in the above examples (the valence basis set) is actually optional in terms of program functionality; only the ECP name is compulsory. However, no corresponding valence basis set is assigned automatically by the program if none is specified. A library valence basis set can be explicitly requested here, or alternatively any custom basis set can be provided manually with the `NewGTO` command.

A variant of the `NewECP` keyword can be used for individual atoms inside the geometry definition:

```
    * xyz ...
    ...
    S   0.0  0.0  0.0  NewECP "LANL(10)" "LANLDZ" end
    ...
    *
```

Manual Input of ECP Parameters

To manually specify ECP parameters, the `NewECP` keyword is followed by the element for which an ECP is to be entered, the number of core electrons to be replaced (N_core) and the maximum angular momentum (lmax). The ECP specification is finished by giving the definitions of the individual shells that constitute the angular dependent potentials U_l.

```
    %basis
        NewECP element
          N_core (number of core electrons)
          lmax (max. angular momentum)
          [shells]
        end
    end
```

For each ECP shell, first the angular momentum l has to be given, followed by the number of primitives. The primitives themselves are then specified by giving a running index and the respective tuple of exponent a_{kl}, expansion coefficient d_{kl} and radial power n_{kl}.

```
    # ECP shell
        l (number of primitives)
        1    a1l    d1l    n1l
        2    a2l    d2l    n3l
        ...
```

As an example, consider the SD(10,MDF) for Vanadium. The name indicates a Stuttgart–Dresden type ECP that replaces 10 core electrons and is derived from a relativistic calculation for the neutral atom. It consists of 4 shells with angular momentum s, p, d and f. Note that the f shell has an expansion coefficient of 0.0 and thus will not

contribute at all to this effective core potential. This is typical for all SD potentials (but may be different for program packages like TURBOMOLE that do not support arbitrary angular momentum with respect to the ECP and therefore use recontractions of the original parameter sets).

```
# ECP SD(10,MDF) for V
# M. Dolg, U. Wedig, H. Stoll, H. Preuss,
# J. Chem. Phys. 86, 866 (1987).
NewECP V
  N_core 10
  lmax f
  s 2
    1       14.4900000000     178.4479710000  2
    2        6.5240000000      19.8313750000  2
  p 2
    1       14.3000000000     109.5297630000  2
    2        6.0210000000      12.5703100000  2
  d 2
    1       17.4800000000     -19.2196570000  2
    2        5.7090000000      -0.6427750000  2
  f 1
    1        1.0000000000       0.0000000000  2
end
```

6.3.7 Embedding Potentials

Computations on cluster models sometimes require the presence of embedding potentials in order to account for otherwise neglected repulsive terms at the border [239]. In order to simplify these kind of calculations with ORCA the ECP embedding can be accomplished quite easy:

```
* ...
# atom>  charge x     y     z     optional ECP declaration
Zr>       4.0    0.0   0.0   0.0   NewECP "SD(28,MWB)" end
...
*
```

The declaration of such a coreless ECP centre takes place in the coordinates section by appending a bracket '>' to the element symbol. Note that embedding ECPs are treated as point charges in ORCA, so the charge has to be given next. The coordinates of the coreless ECP centre have to be specified as usual and may be followed by an optional ECP assignment. In general, calculations that employ an ECP embedding procedure should be single point calculations. However if the need arises to perform a geometry optimization, make sure to set up explicit Cartesian constraints for the coreless ECP centres.

6.4 Choice of Initial Guess and Restart of SCF Calculations

The initial guess is an important issue in each SCF calculation. If this guess is reasonable the convergence of the procedure will be much better. ORCA makes some effort to provide a good initial guess and give the user enough flexibility to tailor the initial guess to his or her needs.

The initial guess is also controlled via the `%scf` block and the variables `Guess`, `MOInp` and `GuessMode`.

```
    %scf Guess HCore          # One electron matrix
               Hueckel        # Extended Hückel guess
               PAtom          # Polarized atomic densities
               PModel         # Model potential
               MORead         # Restart from an earlier calc.
         MOInp "Name.gbw"     # orbitals used for MORead
         GuessMode FMatrix    # FMatrix projection
                   CMatrix    # Corresponding orbital projection
         AutoStart true       # try to use the orbitals from the existing
                              # GBW file of the same name (if possible)
                              # (default)
                   false      # don't use orbitals from existing GBW file
         end
```

6.4.1 AutoStart feature

Previous versions of ORCA always created a new GBW file at the beginning of the run no matter whether a file of the same name existed or perhaps contained orbitals. This philosophy was changed in version 2.5-03. The program now automatically checks if a `.gbw` file of the same name exists. If yes, the program checks if it contains orbitals and all other necessary information for a restart. If yes, the variable `Guess` is set to `MORead`. The existing `.gbw` file is renamed to `BaseName.ges` and `MOInp` is set to this filename. If the `AutoStart` feature is not desired set `AutoStart false` in the `%scf` block or give the keyword `!NoAutoStart` in the simple input line.

6.4.2 One Electron Matrix Guess

The simplest guess is to diagonalize the one electron matrix to obtain starting orbitals. This guess is very simple but usually also a disaster because it produces orbitals that are far too compact.

6.4.3 Basis Set Projection

The remaining guesses (may) need the projection of initial guess orbitals onto the actual basis set. In ORCA there are two ways this can be done. `GuessMode FMatrix` and `GuessMode CMatrix` The results from the two methods are usually rather similar. In certain cases `GuessMode Cmatrix` may be preferable. `GuessMode FMatrix` is simpler and faster. In short the `FMatrix` projection defines an effective one electron operator:

$$f = \sum_i \varepsilon_i a_i^+ a_i \tag{6.46}$$

where the sum is over all orbitals of the initial guess orbital set, a_i^+ is the creation operator for an electron in guess MO i, a_i is the corresponding destruction operator and ε_i is the orbital energy. This effective one electron operator is diagonalized in the actual basis and the eigenvectors are the initial guess orbitals in the target basis. For most wavefunctions this produces a fairly reasonable guess.

`CMatrix` is more involved. It uses the theory of corresponding orbitals to fit each MO subspace (occupied, partially occupied or spin up and spin down occupied) separately [240, 241]. After fitting the occupied orbitals the virtual

starting orbitals are chosen in the orthogonal complement of the occupied orbitals. In some cases, especially when restarting ROHF calculations this may be an advantage. Otherwise I haven't met cases where `CMatrix` was grossly superior to `FMatrix` but that doesn't mean that such cases may exist and they may not even be exotic.

6.4.4 PModel Guess

The PModel guess (chosen by `Guess PModel` in the `%scf` block or simply a keywordline with `!PModel`) is one that is usually considerably successful. It consists of building and diagonalizing a Kohn-Sham matrix with an electron density which consists of the superposition of spherical neutral atoms densities which are predetermined for both relativistic and nonrelativistic methods. This guess is valid for both Hartree-Fock and DFT methods but not for semiempirical models. However, due to the complexity of the guess it will also take a little computer time (usually less than one SCF iteration). The model densities are available for most atoms of the periodic table and consequently the PModel guess is usually the method of choice (particularly for molecules containing heavy elements) unless you have more accurate starting orbitals available.

6.4.5 Hueckel and PAtom Guesses

The extended Hückel guess proceeds by performing a minimal basis extended Hückel calculation and projecting the MOs from this calculation onto the actual basis set using one of the two methods described above. The minimal basis is the STO-3G basis set. The Hückel guess may not be very good because the STO-3G basis set is so poor. There is also accumulating evidence that the superposition of atomic densities produces a fairly good initial guess. The critique of the atomic density method is that the actual shape of the molecule is not taken into account and it is more difficult to reliably define singly occupied orbitals for ROHF calculations or a reasonable spin density for UHF calculations. Therefore ORCA chooses a different way in the `PAtom` guess (which is the default guess) - the Hückel calculation is simply carried out *for all electrons* in a minimal basis of atomic SCF orbitals. These were determined once and for all and are stored inside the program. This means that the densities around the atoms are very close to the atomic ones, all orbitals on one center are exactly orthogonal, the initial electron distribution already reflects the molecular shape and there are well defined singly occupied orbitals for ROHF calculations. This guess has to the best of my knowledge not been described before.

6.4.6 Restarting SCF Calculations

To restart SCF calculations, it can be very helpful and time-saving to read in the orbital information of a previous calculation. To do this, specify:

```
! moread
% moinp "name.gbw"
```

The program has an inconvenience here — the input `.gbw` file from the earlier calculation must have a different name than the new calculation, because in the very beginning of a calculation, a new `.gbw` file is written. If the names are the same, the `.gbw` file from the earlier calculation is overwritten and all information is lost. Therefore, if you want to restart a calculation with an input file of the same name as the previous calculation, you have to rename the `.gbw` file first.

The program stores the current orbitals in every SCF cycle. Should a job crash, it can be restarted from the orbitals that were present at this time by renaming the `.gbw` file and using it as input. In addition, an effort has been made to

make `.gbw` files from different releases compatible with each other. If your input `.gbw` file is from an older release, use `! rescue moread noiter` with `% moinp "name.gbw"` to produce an up-to-date `.gbw`. When the `rescue` keyword is invoked, only the orbital coefficients are read from the `.gbw` file, and everything else from the input file. Thus, make sure that the geometry and the basis set of the old `.gbw` file and the new input match.

Within the same ORCA version, neither the geometry nor the basis set stored in `name.gbw` need to match the present geometry or basis set. The program merely checks if the molecules found in the current calculation and `name.gbw` are consistent with each other and then performs one of the possible orbital projections. If the two basis sets are identical the program by default only reorthogonalizes and renormalizes the input orbitals. However, this can be overruled by explicitly specifying `GuessMode` in the `% scf` block as `CMatrix` or `FMatrix`.

There is an additional aspect of restarting SCF calculations — if you have chosen `SCFMode = Conventional` or `SCFMode = SemiDirect` the program stores a large number of integrals that might have been time consuming to calculate on disk. Normally the program deletes these integrals at the end of the calculation. However, if you want to do a closely related calculation that requires the same integrals (i.e. the *geometry*, the *basis set* and the *threshold* `Thresh` are the same) it is faster to use the integrals generated previously. This is done by using `KeepInts = true` in the `% scf` block of the first calculation and then use `ReadInts = true` in the the `% scf` block of the second calculation. If the second calculation has a different name than the first calculation you have to use `IntName = "FirstName"` to tell the program the name of the integral files. Note that the file containing the integrals does not have an extension — it is simply the name of the previous input file with `.inp` stripped off.

```
%scf   KeepInts true          # Keep integrals on disk
       ReadInts true          # Read integrals from disk
       IntName  "MyInts"    # Name of the integral files without extension
   end
```

Note that if you want to reuse the integrals for `SemiDirect` calculations it is forbidden to change the values of `TCost` and `TSize` in the `% scf` block. The program will not check for that but your results will be totally meaningless. In general, restarting calculations with old integral files requires the awareness and responsibility of the user. If properly used, this feature can save considerable amounts of time.

6.4.7 Changing the Order of Initial Guess MOs and Breaking the Initial Guess Symmetry

Occasionally you will want to change the order of initial guess MOs — be it because the initial guess yielded an erroneous occupation pattern or because you want to converge to a different electronic state using the orbitals of a previous calculation. Reordering of MOs and other tasks (like breaking the symmetry of the electronic wavefunction) are conveniently handled with the `Rotate` feature in ORCA. `Rotate` is a subblock of the SCF block that allows you to linearly transform pairs of MOs.

```
%scf
  Rotate
   { MO1, MO2, Angle }
   { MO1, MO2, Angle, Operator1, Operator2 }
   end
end
```

Here, `MO1` and `MO2` are the indices of the two MOs of interest. Recall that ORCA starts counting MOs with index 0, i.e. the MO with index 1 is the *second* MO. `Angle` is the rotation angle in degrees. A rotation angle of 90° corresponds to flipping two MOs, an angle of 45° leads to a 50:50 mixture of two MOs, and a 180° rotation leads to a change of phase. `Operator1` and `Operator2` are the orbitals sets for the rotation. For UHF calculations spin-up orbitals belong to operator 0 and spin-down orbitals to operator 1. RHF and ROHF calculations only have a single orbital set.

The `Rotate` feature can be used to produce broken symmetry solutions say in transition metal dimers. In order to do that first perform a high spin calculation, then find the pairs of MOs that are symmetric and antisymmetric combinations of each other. Take these MOs as the initial guess and use rotations of 45° for each pair to localize the startup MOs. If you are lucky and the broken symmetry solution exists, you have a good chance of finding it this way. The physical meaning of such a solution is a different question that will not be touched here.

6.5 SCF Convergence

SCF convergence is a pressing problem in any electronic structure package because the total execution times increases linearly with the number of iterations. Thus, it remains true that the best way to enhance the performance of an SCF program is to make it converge better. In some cases, especially for open shell transition metal complexes, convergence may be very difficult. ORCA makes a dedicated effort to achieve reasonable SCF convergence for these cases without compromising efficiency. However, there are cases that do not converge, but find a list of things to try first before giving up in this chapter.

6.5.1 Convergence Tolerances

Before discussing how to converge a SCF calculation it should be defined what is meant by 'converged'. ORCA has a variety of options to control the target precision of the energy and the wavefunction that can be selected in the `% scf` block, or with a simple input line keyword that merges the criterion label with "SCF", e.g. ! `StrongSCF` or ! `VeryTightSCF`:

```
%scf
  Convergence              # The default convergence is between medium and strong
          Sloppy     # very weak convergence
          Loose      # still weak convergence
          Medium     # intermediate accuracy
          Strong     # stronger
          Tight      # still stronger
          VeryTight  # even stronger
          Extreme    # close to numerical zero of the computer
                     # in double precision arithmetic
  end
```

Like other keys, `Convergence` is a compound key that assigns default values to a variety of other variables given in the box below. The default convergence criteria are reasonable and should be sufficient for most purposes. For a cursory look at populations weaker convergence may be sufficient, whereas other cases may require stronger than default convergence. Note that `Convergence` does not only affect the target convergence tolerances but also the integral accuracy as discussed in the section about direct SCF and alike. This is very important: if the error in the integrals is larger than the convergence criterion, a direct SCF calculation cannot possibly converge.

The convergence criteria are always printed in the output. Given below is a list of the convergence criteria for ! `TightSCF`, which is often used for calculations on transition metal complexes.

```
%scf
    TolE        1e-8  # energy change between two cycles
    TolRMSP     5e-9  # RMS density change
    TolMaxP     1e-7  # maximum density change
    TolErr      5e-7  # DIIS error convergence
    TolG        1e-5  # orbital gradient convergence
    TolX        1e-5  # orbital rotation angle convergence
    ConvCheckMode  2  # = 0: check all convergence criteria
                      # = 1: stop if one of criterion is met, this is sloppy!
                      # = 2: check change in total energy and in one-electron energy
                      # Converged if delta(Etot)<TolE and delta(E1)<1e3*TolE
    end
```

If `ConvCheckMode=0`, *all* convergence criteria have to be satisfied for the program to accept the calculation as converged, which is a quite rigorous criterion. In this mode, the program also has mechanisms to decide that a calculation is converged even if one convergence criterion is not fulfilled but the others are overachieved. `ConvCheckMode=1` means that one criterion is enough. This is quite dangerous, so ensure that none of the criteria are too weak, otherwise the result will be unreliable. The default `ConvCheckMode=2` is a check of medium rigour — the program checks for the change in total energy and for the change in the one-electron energy. If the ratio of total energy and one-electron energy is constant, the self-consistent field does not fluctuate anymore and the calculation can be considered converged. If you have small eigenvalues of the overlap matrix, the density may not be converged to the number of significant figures requested by `TolMaxP` and `TolRMSP`.

6.5.2 Dynamic and Static Damping

Damping is the oldest and simplest convergence aid. It was already invented by Douglas Hartree when he did his famous atomic calculations. Damping consists of mixing the old density with the new density as:

$$P_{new,damped} = (1 - \alpha) P_{new} + \alpha P_{old} \tag{6.47}$$

where α is the damping factor, which must have a value of less than 1. Thus the permissible range (not checked by the program) is $0 \ldots 0.999999$. For α values larger than 1, the calculation cannot proceed since no new density is admixedfall. Damping is important in the early stages of a calculation where P_{old} and P_{new} are very different from each other and the energy is strongly fluctuating. Many schemes have been suggested that vary the damping factor dynamically to give strong damping in the beginning and no damping in the end of an SCF. The scheme implemented in ORCA is that by Hehenberger and Zerner [242] and is invoked with `CNVZerner=true`. Static damping is invoked with `CNVDamp=true`. These convergers are mutually exclusive. They can be used in the beginning of a calculation when it is not within the convergence radius of DIIS or SOSCF. Damping works reasonably well, but most other convergers in ORCA are more powerful.

If damping used in conjunction with DIIS or SOSCF, the value of `DampErr` is important: once the DIIS error falls below `DampErr`, the damping is turned off. In case SOSCF is used, `DampErr` refers to the orbital gradient value at which the damping is turned off. The default value is 0.1 Eh. In difficult cases, however, it is a good idea to choose

`DampErr` much smaller, e.g. 0.001. This is — to some extent — chosen automatically together with the keyword `! SlowConv`.

```
%scf
  # control of the Damping procedure
  CNVDamp   true    # default: true
  CNVZerner false   # default: false
  DampFac   0.98    # default: 0.7
  DampErr   0.05    # default: 0.1
  DampMin   0.1     # default: 0.0
  DampMax   0.99    # default: 0.98
  # more convenient:
  Damp fac 0.98 ErrOff 0.05 Min 0.1 Max 0.99 end
end
```

6.5.3 Level Shifting

Level shifting is a frequently used technique. The basic idea is to shift the energies of the virtual orbitals such that after diagonalization the occupied and virtual orbitals mix less strongly and the calculation converges more smoothly towards the desired state. Also, level shifting should prevent flipping of electronic states in near-degenerate cases. In a special context it has been shown by Saunders and Hillier [243, 244] to be equivalent to damping.

Similar to `DampErr` described in the previous section, `ShiftErr` refers to the DIIS error at which the level shifting is turned off.

```
%scf
  # control of the level shift procedure
  CNVShift  true  # default: true
  LShift    0.1   # default: 0.25, energy unit is Eh.
  ShiftErr  0.1   # default: 0.0
  # more convenient:
  Shift Shift 0.1 ErrOff 0.1 end
end
```

6.5.4 Direct Inversion in Iterative Subspace (DIIS)

The Direct Inversion in Iterative Subspace (DIIS) is a technique that was invented by Pulay [245, 246]. It has become the *de facto* standard in most modern electronic structure programs, because DIIS is robust, efficient and easy to implement. Basically DIIS uses a criterion to judge how far a given trial density is from self consistency. The commutator of the Fock and density matrices $[\mathbf{F},\mathbf{P}]$ is a convenient measure for this error. With this information, an extrapolated Fock matrix from the present and previous Fock matrices is constructed, which should be much closer to self consistency. In practice this is usually true, and better than linear convergence has been observed with DIIS. In some rare (open shell) cases however, DIIS convergence is slow or absent after some initial progress. As self consistency is approached, the set of linear equations to be solved for DIIS approaches linear dependency and it is useful to bias DIIS in favor of the SCF cycle that had the lowest energy using the factor `DIISBfac`. This is achieved

by multiplying all diagonal elements of the DIIS matrix with this factor unless it is the Fock matrix/density which leads to the lowest energy. The default value for `DIISBfac` is 1.05.

The value of `DIISMaxEq` is the maximum number of old Fock matrices to remember. Values of 5-7 have been recommended, while other users store 10-15 Fock matrices. Should the standard DIIS not achieve convergence, some experimentation with this parameter can be worthwhile. In cases where DIIS causes problems in the beginning of the SCF, it may have to be invoked at a later stage. The start of the DIIS procedure is controlled by `DIISStart`. It has a default value of 0.2 Eh, which usually starts DIIS after 0-3 cycles. A different way of controlling the DIIS start is adjusting the value `DIISMaxIt`, which sets the maximum number of cycles after which DIIS will be started irrespective of the error value.

```
%scf
  # control of the DIIS procedure
  CNVDIIS     true  # default: true
  DIISStart   0.1   # default: 0.2
  DIISMaxIt   5     # default: 12
  DIISMaxEq   7     # default: 5
  DIISBFac    1.2   # default: 1.05
  DIISMaxC    15.0  # default: 10.0
  # more convenient:
  DIIS Start 0.1  MaxIt 5  MaxEq 7  BFac 1.2  MaxC 15.0  end
  end
```

6.5.5 An alternative DIIS algorithm: KDIIS

An alternative algorithm that makes use of the DIIS concept is called KDIIS in ORCA. The KDIIS algorithm is designed to bring the orbital gradient of any energy expression to zero using a combination of DIIS extrapolation and first order perturbation theory. Thus, the method is diagonalization-free. In our hands it is superior to the standard DIIS algorithm in many cases, but not always. The algorithm is invoked with the keyword ! `KDIIS` and is available for RHF, UHF and CASSCF.

6.5.6 Approximate Second Order SCF (SOSCF)

SOSCF is an approximately quadratically convergent variant of the SCF procedure [247, 248]. The theory is relatively involved and will not be described here. In short – SOSCF computes an initial guess to the inverse orbital hessian and then uses the BFGS formula in a recursive way to update orbital rotation angles. As information from a few iterations accumulates, the guess to the inverse orbital hessian becomes better and better and the calculation reaches a regime where it converges superlinearly. As implemented, the procedure converges as well or slightly better than DIIS and takes a somewhat less time. However, it is also *a lot* less robust, so that DIIS is the method of choice for many problems (see also the description of the full Newton-Raphson procedure in the next section). On the other hand, SOSCF is useful when DIIS gets stuck at some error around ~ 0.001 or 0.0001. Such cases were the primary motive for the implementation of SOSCF into ORCA.

The drawback of SOSCF is the following: in the beginning of the SCF, the orbital gradient (the derivative of the total energy with respect to rotations that describe the mixing of occupied and virtual MOs) is large, so that one is far from the quadratic regime. In such cases, the procedure is not successful and may even wildly diverge. Therefore it is recommended to only invoke the SOSCF procedure in the very end of the SCF where DIIS may lead to "trailing"

convergence. SOSCF is controlled by the variables `SOSCFStart` and `SOSCFMaxIt`. `SOSCFStart` is a threshold for the orbital gradient. When the orbital gradient, or equivalently the DIIS Error, fall below `SOSCFStart`, the SOSCF procedure is initiated. `SOSCFMaxIt` is the latest iteration to start the SOSCF even if the orbital gradient is still above `SOSCFStart`.

```
%scf
      # control of the SOSCF procedure
      CNVSOSCF    true  # default: false
      SOSCFStart  0.1   # default: 0.01
      SOSCFMaxIt  5     # default: 1000
      # more convenient:
      SOSCFStart 0.1  MaxIt 5  end
      end
```

For many calculations on transition metal complexes, it is a good idea to be conservative in the startup criterion for SOSCF, it may diverge otherwise. A choice of 0.01 or lower can be recommended.

6.5.7 Full Second Order SCF (Newton-Raphson SCF)

The Newton–Raphson (NR) procedure is quite powerful once the calculations is within its radius of convergence: it then usually converges extremely fast and to high accuracy. It should be considered that the time requirement for a NR iteration is much higher than that for a regular SCF calculation. However, this may be overcompensated by the reduced number of cycles (see section 5.1.5 for a numerical example). For a "normal" well-behaved molecule which converges in 10–15 iterations with the DIIS procedure, it is unlikely that invoking the NR converger pays off. For more problematic cases it is more likely that it will. Another area where NR-SCF may be useful is when you desire accurate SCF solutions, for example for numerical frequency calculations. In these cases, a different option is to decrease the convergence tolerance of the CP-SCF solver (`Z_Tol`).

In version 2.4.45 the full Newton–Raphson method (and its augmented Hessian analogue) was implemented for converging closed-shell or spin-unrestricted Hartree–Fock and density functional calculations.

The NR solver itself has two options, which are very similar to the analogous SOSCF options:

```
%scf
      # no Newton-Raphson by default
      NR MaxIt 10   # iteration at which NR is switched on, default 1000
      Start 0.05    # DIIS error at which NR is switched on, default 0.02
      end
    end
```

As soon as the Newton–Raphson procedure starts within one SCF procedure, all other convergers are switched off. After convergence is achieved, the orbitals are canonicalized again through a single diagonalization of the Fock operator. In order to help the convergence of the CP-SCF procedure, the orbitals are quasi-canonical in between iterations. This means that after the orbital update, the occ–occ and virt–virt blocks of the Fock operator are diagonalized separately and the occupied and virtual orbitals are chosen to be eigenfunctions of these parts of the Fock operator.

There is a shortcut to the NR-converger. If ! `NRSCF` is chosen, this will invoke the NRSCF once the DIIS error falls below 0.02 (the default value). If `SlowConv` is also specified, the NR converger will only be switched on at a DIIS error of 0.002. This is more conservative since the NR-solver may also have convergence problems otherwise.

In particularly difficult cases, the Newton–Raphson step may run into trouble if the orbital Hessian is singular or nearly singular. It is then preferable to take a more cautious step, by invoking the augmented-Hessian method which was implemented for such cases. This method is called by ! `AHSCF`. It sets the same values as NRSCF but puts an augmented Hessian solver into operation. It will not always help, but in some difficult situations this feature has already been successful.

Technically speaking, the program solves a large set of linear equations in each NR iteration. It turns out that the equations to be solved are precisely equivalent to the CP-SCF equations for an electric-field-like equation but with the right hand side being replaced by the occupied/virtual blocks of the Fock operator in the MO basis. Thus, the implementation calls the CP-SCF program in each SCF iteration. The converged CP-SCF first order wavefunction coefficients are then used to update the current orbitals. Thus, the cost of each NR iteration can be precisely specified as the cost of one electric field type CP-SCF calculation for a single perturbation on the same system.

One important aspect of the NR implementation is that the NR solver checks itself for convergence. The orbital gradient is calculated *before* the CP-SCF program is called. If the orbital gradient is below its threshold (`TolG`), convergence is signaled irrespective of what the other convergence indicators are. The orbital gradient is a rigorous criterion and therefore the results are considered reliable. If tighter convergence is desired, the thresholds `TolG` and `TolX` should be lowered.

The following example for a "normal" case of a Hartree–Fock geometry optimization converges in 5 cycles:

```
! RHF TZVPP Opt TightSCF
! NRSCF

* xyz 0 1
N 0 0 0
N 0 0 1
*
```

With the alternative first order SCF it also takes 5 cycles, but in each geometry optimization step there are more SCF iterations. In these examples, the total time for the calculations is comparable (less than 30 seconds), but that may be different for other cases.

```
! RHF TZVPP Opt VeryTightSCF

* xyz 0 1
N 0 0 0
N 0 0 1
*
```

The CP-SCF program is called with the convergence and method flags that are also used in Z-vector calculations (e.g. in MP2 or CIS/TD-DFT gradient calculations). They are accessible through the `%method` block:

```
    %method Z_solver   Pople      # Pople's solver. Usually great!
                       CG         # conjugate gradient
                       DIIS       # Pulays DIIS
                       AugHess    # Augmented Hessian (for SCF only)
              Z_MaxIter 64        # maximum number of CP-SCF iterations
              Z_MaxDIIS 10        # for DIIS: maximum number of expansion vectors
              Z_Shift   0.3       # for DIIS: level shift
              Z_Tol     1e-6      # convergence tolerance on residual
        end
```

The range of applicability of the NR method is precisely the same as that of the CP-SCF program itself and the SCF program will immediately benefit from any improvement made in the CP-SCF module.

6.5.8 Fractional Occupation Numbers

Only a very basic implementation of fractional occupation numbers is presently provided. It is meant to deal with orbitally degenerate states in the UHF/UKS method. Mainly it was implemented to avoid symmetry breaking in DFT calculations on orbitally degenerate molecules and atoms. The program checks the orbital energies of the initial guess orbitals, finds degenerate sets and averages the occupation numbers among them. Currently the criterion for degenerate orbitals is 10^{-3} Eh. The fractional occupation number option is invoked by:

```
    %scf FracOcc true
      end
```

Clearly, the power of fractional occupation numbers goes far beyond what is presently implemented in the program and future releases will likely make more use of them. The program prints a warning whenever it uses fractional occupation numbers. The fractionally occupied orbitals should be checked to ensure they are actually the intended ones.

NOTE:

- Fractional occupation numbers *cannot* be used together with the `CNVRico` or `SOSCF` methods.

- Using `GuessMode = CMatrix` will cause problems because there are no orbital energies for the initial guess orbitals. The program will then average over *all* orbitals — which makes no sense at all.

A finite temperature can be used to apply a Fermi-like occupation number smearing over the orbitals of the system, which may sometimes help to get convergence of the SCF equations in near hopeless cases. Through the smearing, the electrons are distributed according to Fermi statistics among the available orbitals. The "chemical potential" is found through the condition that the total number of electrons remains correct. Gradients can be computed in the presence of occupation number smearing.

```
    %scf SmearTemp 5000  # 'temperature' in Kelvin
      end
```

6.6 Choice of Wavefunction and Integral Handling

6.6.1 Choice of Wavefunction Type

The basic variable that controls the type of wavefunction to be computed is the variable `HFTyp` in the `%scf` block. If nothing is specified for `HFTyp`, the program will check the multiplicity given in the input: for closed shell molecules with multiplicity 1, RHF/RKS is assumed; for open shell molecules with multiplicity larger than 1, UHF/UKS is invoked. `RHF` will lead to a spin restricted closed shell type computation [96]. For DFT calculations, `RKS`, `UKS` and `ROKS` can be used as synonyms for `RHF`, `UHF` and `ROHF`. The restricted open shell DFT method (`ROKS`) is only operative for high-spin states that have n unpaired electrons and $S = n/2$. `UKS` wavefunctions will not be spin-purified.

```
%scf HFTyp   RHF      # closed shell (RKS for DFT)
             UHF      # unrestricted open shell (UKS for DFT)
             ROHF     # restricted open shell (ROKS for DFT)
             CASSCF   # complete active space SCF
    end
```

In certain cases you may want to run open shell molecules with RHF/RKS to get a 'half-electron' type wavefunction [249]. The total energy is not corrected! Sometimes these half-electron computations lead to acceptable convergence, and the resulting orbitals may be used as input for ROHF, UHF or MRCI calculations. Especially for transition metal complexes the orbitals are quite different from ROHF or UHF orbitals, so that it is not recommended to over-interpret the wavefunctions from such calculations. The calculation is set up in the following way:

```
%method AllowRHF true end
# or simply: ! AllowRHF
```

For ROHF calculations [250–258] the program will try to figure out what type of open shell situation is present on the basis of the initial guess orbitals and their energies. Most 'simple' cases are well recognized but sometimes a little help from the user is needed. The ROHF code has a very powerful feature that goes back to insights of Mike Zerner [259, 260]. It can average over either *all* states of a given configuration (`CAHF`) or *all states of a given spin* for a given configuration (`SAHF`). Especially the `SAHF` feature gives you easy access to most degenerate high symmetry situations and the orbitals resulting from such calculations will be very convenient as input for CI calculations. In this way one can approach results from MCSCF calculations.

The input variables described above are accessed in the `%scf` block. `ROHFOPT Case User` is a convenient way to input details. For example for the high spin case with three electrons in three orbitals gives two operators with vector coupling coefficients $a = 1$ and $b = 2$ (Zerner convention).

```
%scf HFTyp       ROHF
     ROHF_case   CAHF        # configuration averaged HF
                 SAHF        # spin averaged HF
                 HighSpin    # high spin case, n unpaired e- in n orbitals with S=n/2
                 ROHF_NumOp   3       # number of operators (3, 2 or 1)
                 ROHF_NOrb[1] 2,1     # number of orbitals in each open shell
                 ROHF_NEl[1]  1,1     # number of electrons in each open shell
```

```
        end
        ROHFOP  Case    User  # manual input of ROHF variables
                Nop     2     # number of operators
                Norb[1] 3     # number of open shell orbitals
                Nel[1]  3     # number of open shell electrons
                A[1,1]  1     # Coulomb vector in the open shell
                B[1,1]  2     # Exchange vector in the open shell
        end
    end
```

The hypothetical example below could represent an excited state of an octahedral d^3 transition metal complex. In this case there are five open shell orbitals. The first three open shell orbitals contain two electrons and the last two one electron. The input for a SAHF calculation is identical, just replace CAHF with SAHF.

```
    %scf HFTyp        ROHF
         ROHF_case    CAHF    # configuration averaged HF
         ROHF_NumOp   3       # 3 operators in this case: closed, open1, open2
         ROHF_NOrb[1] 3,2     # 3 orbitals in first open shell, 2 in the second
         ROHF_NEl[1]  2,1     # 2 electrons in first open shell, 1 in the second
    end
```

For a ROHF_case HighSpin calculation an analogous input can be used, but the number of operators is always two (one closed and one open shell), therefore:

```
    %scf HFTyp ROHF
         ROHF_case HighSpin   # high spin case
         ROHF_NEl[1] 4        # n=4 unpaired electrons, S=n/2
         end
```

One awkward feature of the ROHF theory is that the Fock operator is somewhat arbitrarily defined. Different choices lead to the same wavefunction, but have different convergence properties that may vary from system to system. ORCA thus lets the user choose the desired variant. Playing around with these choices may turn a divergent or slowly converging ROHF calculation into a successful calculation!

The ROHF_Restrict feature is another feature that may be useful. If you suspect that the ROHF calculation does not converge because an open shell and a closed shell orbital are flipping back and forth, you can try to avoid this behavior by choosing ROHF_Restrict= true. Of course there is no guarantee that it will work, and no guarantee that the system stays in the desired state. However, it decreases the chances of large, uncontrolled steps.

```
    %scf ROHF_Mode  0   # construct F according to Pulay (default)
                    1   # construct F as in the Gamess program
                    2   # construct F according to Kollmar
         ROHF_Restrict false   # restrict orbital interchanges and off-diagonal elements
                               # (default=false)
```

```
        # a complete list of ROHF variables
ROHFOP  Case      User   # manual input of ROHF variables
        Nop       2      # number of operators
        Norb[1]   3      # number of open shell orbitals
        Nel[1]    3      # number of open shell electrons
        A[1,1]    1      # Coulomb vector in the open shell
        B[1,1]    2      # Exchange vector in the open shell
        Mode      2      # use the Kollmar operator
        Restrict  false  # don't restrict
        end
end
```

6.6.2 UHF Natural Orbitals

The program can produce the UHF natural orbitals (UNOs). With these, the open shell wavefunction can be pictured conveniently. The syntax is simple:

```
%scf UHFNO true
  end
# or simply: ! UNO
```

There are various printing options for UNOs described in the output section 6.22. The UNOs can also be plotted as described in the plots section 6.23. In general the program stores a file `BaseName.uno`, where `BaseName` is by default the name of you input file with `.inp` stripped off. Accordingly, the gbw file is named `BaseName.gbw`. The `.uno` file is a normal gbw file that contains the geometry, basis set and the UNO orbitals. It could be used, for example, to start a ROHF calculation.

6.6.3 Integral Handling (Conventional, Direct, SemiDirect)

As the number of nonzero integrals grows very rapidly and reaches easily hundreds of millions even with medium sized basis sets in medium sized molecules, storage of all integrals is not generally feasible. This desperate situation prevented SCF calculations on larger molecules for quite some time, so that Almlöf [261–263] made the insightful suggestion to repeat the integral calculation, which was already the dominant step, in *every SCF cycle* to solve the storage problem. Naively, one would think that this raises the effort for the calculation to P_{new} (where n_{iter} is the number of iterations and $t_{integrals}$ is the time needed to generate the nonzero integrals). However, this is not the case because only the change in the Fock matrix is required from one iteration to the next, but not the Fock matrix itself. As the calculations starts to converge, more and more integrals can be skipped. The integral calculation time will still dominate the calculation quite strongly, so that ways to reduce this burden are clearly called for. An important contribution to this subject was made by Häser and Ahlrichs [264] who suggested to store some of the integrals and to only recalculate the remaining ones in the direct SCF calculation. Obvious candidates for storage are those integrals that are expensive to calculate. As integrals are calculated in *batches*[13] the cost of evaluating the given batch of shells p, q, r, s may be estimated as:

---

[13] A *batch* is a set of integrals that arises from all components of the shells involved in the integral. For example a $\langle pp|pp \rangle$ batch gives rise to $3 \times 3 \times 3 \times 3 = 81$ integrals due to all possible combinations of p_x, p_y and p_z functions in the four shells. Computations based on batches lead to great computational advantages because the 81 integrals involved in the $\langle pp|pp \rangle$ batch share many common intermediate quantities.

$$cost \approx n_p n_q n_r n_s \left(2l_p + 1\right) \left(2l_q + 1\right) \left(2l_r + 1\right) \left(2l_s + 1\right) \tag{6.48}$$

Here, n_p is the number of primitives involved in shell p, and l_p is the angular momentum for this shell. *Large* integrals are also good candidates for storage, because small changes in the density that multiply large integrals are likely to give a nonzero contribution to the changes in the Fock matrix.

ORCA thus features three possibilities for integral handling, which are controlled by the variable `SCFMode`. In the mode `Conventional`, all integrals above a given threshold are stored on disk (in a compressed format that saves much disk space). In the mode `Direct`, all two-electron integrals are recomputed in each iteration. In the mode `SemiDirect`, the Häser/Ahlrichs-type hybrid method as described above is implemented.

Two further variables are of importance: In the `Conventional` and `SemiDirect` modes the program may write enormous amounts of data to disk. To ensure this stays within bounds, the program first performs a so-called 'statistics run' that gives a pessimistic estimate of how large the integral files will be. Oftentimes, the program will overestimate the amount of disk space required by a factor of two or more. The maximum amount of disk space that is allowed for the integral files is given by `MaxDisk` (in Megabytes).

On the other hand, if the integral files in `Conventional` and `SemiDirect` runs are small enough to fit into the central memory, it is faster to do this since it avoids I/O bottlenecks. The maximum amount of memory allocated for integrals in this way is specified by `MaxIntMem` (in Megabytes). If the integral files are larger than `MaxIntMem`, no integrals will be read into memory.

```
%scf
  MaxIter   100      # Max. no. of SCF iterations
  SCFMode Direct     # default, other choices: Conventional, SemiDirect
  Thresh  1e-8       # Threshold for neglecting integrals / Fock matrix contributions
                     # Depends on the chosen convergence tolerance (in Eh).
  TCut    1e-10      # Threshold for neglecting primitive batches. If the prefactor
                     # in the integral is smaller than TCut, the contribution of the
                     # primitive batch to the total batch is neglected.
  TCost   100        # Threshold for considering integrals as costly in
                     # SemiDirect SCF. Batches with Cost > TCost are stored.
  TSize   0.01       # In SemiDirect SCF: stores integrals with estim. size = TSize
  UseCheapInts false   # default: false
  DirectResetFreq 20   # default: 15
  MaxDisk  2500        # Max. amount of disk for 2 el. ints. (MB)
  MaxIntMem 400        # Max. amount of RAM for 2 el. ints. (MB)
end
```

The flag `UseCheapInts` has the following meaning: In a `Direct` or `SemiDirect` SCF calculation, the oscillations in the total energy and density are initially quite large. High accuracy in the integrals is therefore not crucial. If `UseCheapInts` is switched on, the program loosens the threshold for the integrals and thus saves a lot of computational time. After having obtained a reasonable initial convergence, the thresholds are tightened to the target accuracy. One pitfall with this method is that the number of cycles required to reach convergence may be larger relative to a calculation with full integral accuracy throughout.[14] When restarting calculations that are close to convergence, it is recommended to switch `UseCheapInts` off. `UseCheapInts` has no meaning in a conventional SCF.

---

[14] This might be an undesirable feature of the current implementation.

The value of `DirectResetFreq` sets the number of incremental Fock matrix builds after which the program should perform a full Fock matrix build in a `Direct` or `SemiDirect` SCF calculation. To prevent numerical instabilities that arise from accumulated errors in the recursively build Fock matrix, the value should not be too large, since this will adversely affect the SCF convergence. If the value is too small, the program will update more frequently, but the calculation will take considerably longer, since a full Fock matrix build is more expensive than a recursive one.

The thresholds `TCut` and `Thresh` also deserve a closer explanation. `Thresh` is a threshold that determines when to neglect two-electron integrals. If a given integral is smaller than `Thresh` Eh, it will not be stored or used in Fock matrix construction. Additionally, contributions to the Fock matrix that are smaller than `Thresh` Eh will not be calculated in a `Direct` or `SemiDirect` SCF.

Clearly, it would be wasteful to calculate an integral, then find out it is good for nothing and thus discard it. A useful feature would be an efficient way to estimate the size of the integral *before it is even calculated*, or even have an estimate that is a *rigorous upper bound* on the value of the integral. Häser and Ahlrichs [264] were the first to recognize that such an upper bound is actually rather easy to calculate. They showed that:

$$|\langle ij\,|kl\rangle| \leqslant \sqrt{\langle ij\,|ij\rangle}\sqrt{\langle kl\,|kl\rangle} \tag{6.49}$$

where:

$$\langle ij\,|kl\rangle = \int\int \phi_i\left(\vec{r}_1\right)\phi_j\left(\vec{r}_1\right)r_{12}^{-1}\phi_k\left(\vec{r}_2\right)\phi_l\left(\vec{r}_2\right)d\vec{r}_1 d\vec{r}_2 \tag{6.50}$$

Thus, in order to compute an upper bound for the integral only the right hand side of this equation must be known. This involves only two index quantities, namely the matrix of two center exchange integrals $\langle ij\,|ij\rangle$. These integrals are easy and quick to calculate and they are all $\geqslant 0$ so that there is no trouble with the square root. Thus, one has a powerful device to avoid computation of small integrals. In an actual calculation, the Schwartz prescreening is not used on the level of individual basis functions but on the level of shell batches because integrals are always calculated in batches. To realize this, the largest exchange integral of a given exchange integral block is looked for and its square root is stored in the so called *pre-screening* matrix **K** (that is stored on disk in ORCA). In a direct or semidirect SCF this matrix is not recalculated in every cycle, but simply read from disk whenever it is needed. The matrix of exchange integrals on the level of individual basis function is used in conventional or semidirect calculations to estimate the disk requirements (the 'statistics' run).

Once it has been determined that a given integral batch survives it may be calculated as:

$$\langle ij\,|kl\rangle = \sum_p d_{pi}\sum_q d_{qj}\sum_r d_{kr}\sum_s d_{sl}\,\langle i_p j_q\,|k_r l_s\rangle \tag{6.51}$$

where the sums p, q, r, s run over the primitive gaussians in each basis function i, j, k, l and the d's are the contraction coefficients. There are more powerful algorithms than this one and they are also used in ORCA. However, if many terms in the sum can be skipped and the total angular momentum is low, it is still worthwhile to compute contracted integrals in this straightforward way. In equation 6.51, each primitive integral batch $\langle i_p j_q\,|k_r l_s\rangle$ contains a prefactor I_{pqrs} that depends on the position of the four gaussians and their orbital exponents. Since a contracted gaussian usually has orbital exponents over a rather wide range, it is clear that many of these primitive integral batches will contribute negligibly to the final integral values. In order to reduce the overhead, the parameter `TCut` is introduced. If the common prefactor I_{pqrs} is smaller than `TCut`, the primitive integral batch is skipped. However, I_{pqrs} is *not* a rigorous upper bound to the true value of the primitive integral. Thus, one has to be more conservative with `TCut` than with `Thresh`. In practice it appears that choosing `TCut=0.01*Thresh` provides sufficient accuracy, but the user is encouraged to determine the influence of `TCut` if it is suspected that the accuracy reached in the integrals is not sufficient.

HINT:

- If the direct SCF calculation is close to convergence but fails to finally converge, this maybe related to a numerical problem with the Fock matrix update procedure – the accumulated numerical noise from the update procedure prevents sharp convergence. In this case, set `Thresh` and `TCut` lower and/or let the calculation more frequently reset the Fock matrix (`DirectResetFreq`).

NOTE:

- For a `Direct` or `SemiDirect` calculation, there is no way to have `Thresh` larger than `TolE`. If the errors in the Fock matrix are larger than the requested convergence of the energy, the change in energy can never reach `TolE`. The program checks for that.

- In a `SemiDirect` SCF, the number of stored integrals increases with molecular size. It may be necessary to *increase* `TSize` and `TCost` to make storage possible. It is usually not a good idea to produce integral files many gigabytes in size, because this will lead to quite some I/O penalty, especially in the late stages of a calculation. Within reasonable bounds however, `SemiDirect` calculations can be quite a bit faster than `Direct` calculations.

- The actual disk space used for *all* temporary files may easily be larger than `MaxDisk`. `MaxDisk` only pertains to the two-electron integral files. Other disk requirements are not currently checked by the program and appear to be uncritical.

6.7 The Second Order Many Body Pertubation Theory Module (MP2)

Throughout this section, indices $i, j, k, \ldots$ refer to occupied orbitals in the reference determinant, $a, b, c, \ldots$ to virtual orbitals and $p, q, r, \ldots$ to general orbitals from either set while $\mu, \nu, \kappa, \tau, \ldots$ refer to basis functions.

6.7.1 Standard MP2

The standard (or full accuracy) MP2 module has two different branches. One branch is used for energy calculations, the other for gradient calculations.

For standard MP2 energies, the program performs two half-transformations and the half-transformed integrals are stored on disk in compressed form. This appears to be the most efficient approach that can also be used for medium sized molecules. There is little the user can control in this module since there are no special requirements on core memory. The module should parallelize acceptably well as long as I/O is not limiting.

For standard MP2 gradients, the program performs four quarter transformations that are ordered by occupied orbitals. Here, the program massively benefits from large core memory (`%maxcore`) since this minimizes the number of batches that are to be done. I/O demands are minimal in this approach.

6.7.2 RI-MP2

The RI-MP2 module is of a straightforward nature. The program first transforms the three-index integrals (ia|P), where 'i' is occupied, 'a' is virtual and 'P' is an auxiliary basis function that is orthogonalized against the Coulomb metric. These integrals are stored on disk, which is not critical, even if the basis has several thousand functions. The integral transformation is parallelized and has no specifically large core memory requirements.

In the next step, the integrals are read ordered with respect to the occupied labels and the exchange operators $K^{ij}(a,b)= (ia|jb) = \text{sum}((ia|P)(P|jb), P=1..NAux)$ are formed in the rate limiting $O(N^5)$ step. This step is done with high efficiency by a large matrix multiplication and parallelizes well. From the exchange operators, the MP2 amplitudes and the MP2 energy is formed. The program mildly benefits from large core memory (%maxcore) as this minimizes the number of batches and hence reads through the integral list.

The RI-MP2 gradient is also available. Here, all necessary intermediates are made on the fly.

In the RI approximation, one introduces an auxiliary fitting basis $\eta_P(\mathbf{r})$ and then approximates the two-electron integrals in the Coulomb metric as:

$$(pq|rs) \approx \sum_{PQ} (pq|P) V_{PQ}^{-1} (Q|rs) \tag{6.52}$$

where $V_{PQ} = (P|Q)$ is a two-index electron-electron repulsion integral. As first discussed by Weigend and Häser, the closed-shell case RI-MP2 gradient takes the form:

$$E_{RI-MP2}^x = 2 \sum_{\mu\nu P} (\mu\nu|P)^{(x)} \sum_i c_{\mu i} \Gamma_{i\nu}^P + \sum_{RS} V_{RS}^x \left(\mathbf{V}^{-1/2} \gamma \mathbf{V}^{-1/2}\right)_{RS} + \langle \mathbf{DF}^x \rangle \tag{6.53}$$

The $\mathbf{F}$-matrix derivative terms are precisely handled as in the non-RI case and need not be discussed any further. Γ_{ia}^P is a three-index two-particle 'density':

$$\Gamma_{ia}^P = \sum_{jbQ} (1 + \delta_{ij}) \tilde{t}_{ab}^{ij} V_{PQ}^{-1/2} (Q|jb) \tag{6.54}$$

Which is partially transformed to the AO basis by:

$$\Gamma_{i\nu}^P = \sum_a c_{\nu a} \Gamma_{ia}^P \tag{6.55}$$

The two-index analogue is given by:

$$\gamma_{PQ} = \sum_{iaR} \Gamma_{ia}^Q (ia|R) V_{RP}^{-1/2} \tag{6.56}$$

The RI contribution to the Lagrangian is particularly convenient to calculate:

$$L_{ai}^{RI} = \sum_\mu c_{\mu a} \left[2 \sum_{PQ\nu} \Gamma_{i\nu}^P (\mu\nu|Q) V_{PQ}^{-1/2} \right] \tag{6.57}$$

In a similar way, the remaining contributions to the energy weighted density matrix can be obtained efficiently. Note, however, that the response operator and solution of the CP-SCF equations still proceed via traditional four-index

integrals since the SCF operator was built in this way. Thus, while the derivatives of the three-index integrals are readily and efficiently calculated, one still has the separable contribution to the gradient, which requires the derivatives of the four-index integrals.

The RI-MP2 energy and gradient calculations can be drastically accelerated by employing the RIJCOSX or the RIJDX approximation.

6.7.3 'Double-Hybrid' Density Functional Theory

A slightly more general form is met in the double hybrid DFT gradient. The theory is briefly described below.

The energy expression for perturbatively and gradient corrected hybrid functionals as proposed by Grimme is:

$$
\begin{aligned}
E &= V_{NN} + \langle \mathbf{Ph^+} \rangle + \frac{1}{2} \int \int \frac{\rho\left(\mathbf{r}_1\right)\rho\left(\mathbf{r}_2\right)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 - \frac{1}{2} a_x \sum_{\mu\nu\kappa\tau\sigma} P^\sigma_{\mu\kappa} P^\sigma_{\nu\tau} \left(\mu\nu|\kappa\tau\right) + c_{DF} E_{XC}\left[\rho_\alpha, \rho_\beta\right] + c_{PT} E_{PT2} \\
&= E_{SCF} + c_{PT} E_{PT2}
\end{aligned}
\tag{6.58}
$$

Here V_{NN} is the nuclear repulsion energy and $h_{\mu\nu}$ is a matrix element of the usual one-electron operator which contains the kinetic energy and electron-nuclear attraction terms ($\langle \mathbf{ab} \rangle$ denotes the trace of the matrix product $\mathbf{ab}$). As usual, the molecular spin-orbitals are expanded in atom centered basis functions ($\sigma = \alpha, \beta$):

$$
\psi^\sigma_p\left(\mathbf{r}\right) = \sum_\mu c^\sigma_{\mu p} \varphi_\mu\left(\mathbf{r}\right)
\tag{6.59}
$$

with MO coefficients $c^\sigma_{\mu p}$. The total density is given by (real orbitals are assumed throughout):

$$
\rho\left(\mathbf{r}\right) = \sum_{i\sigma} \left|\psi^\sigma_i\left(\mathbf{r}\right)\right|^2 = \sum_{\mu\nu\sigma} P^\sigma_{\mu\nu} \varphi_\mu\left(\mathbf{r}\right)\varphi_\nu\left(\mathbf{r}\right) = \rho^\alpha\left(\mathbf{r}\right) + \rho^\beta\left(\mathbf{r}\right)
\tag{6.60}
$$

Where $\mathbf{P} = \mathbf{P}^\alpha + \mathbf{P}^\beta$ and $P^\sigma_{\mu\nu} = \sum_{i_\sigma} c^\sigma_{\mu i} c^\sigma_{\nu i}$.

The second term of eq. 6.58 represents the Coulombic self-repulsion. The third term represents the contribution of the Hartree-Fock exchange with the two-electron integrals being defined as:

$$
\left(\mu\nu|\kappa\tau\right) = \int \int \phi_\mu\left(\mathbf{r}_1\right)\phi_\nu\left(\mathbf{r}_1\right) r^{-1}_{12} \phi_\kappa\left(\mathbf{r}_2\right)\phi_\tau\left(\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2
\tag{6.61}
$$

The mixing parameter a_x controls the fraction of Hartree-Fock exchange and is of a semi-empirical nature. The exchange correlation contribution may be written as:

$$
E_{XC}\left[\rho_\alpha, \rho_\beta\right] = \left(1 - a_x\right) E^{GGA}_x\left[\rho_\alpha, \rho_\beta\right] + b E^{GGA}_c\left[\rho_\alpha, \rho_\beta\right]
\tag{6.62}
$$

Here, $E^{GGA}_x\left[\rho_\alpha, \rho_\beta\right]$ is the exchange part of the XC-functional in question and $E^{GGA}_c\left[\rho_\alpha, \rho_\beta\right]$ is the correlation part. The parameter b controls the mixing of DFT correlation into the total energy and the parameter c_{DF} is a global scaling factor that allows one to proceed from Hartree-Fock theory ($a_x = 1, c_{DF} = 0, c_{PT} = 0$) to MP2 theory ($a_x = 1, c_{DF} = 0, c_{PT} = 1$) to pure DFT ($a_x = 0, c_{DF} = 1, c_{PT} = 0$) to hybrid DFT ($0 < a_x < 1, c_{DF} = 1, c_{PT} = 0$) and finally to the general perturbatively corrected methods discussed in this work ($0 < a_x < 1, c_{DF} = 1, 0 < c_{PT} < 1$). As discussed in detail by Grimme, the B2-PLYP functional uses the Lee-Yang-Parr (LYP) functional as correlation

part, the Becke 1988 (B88) functional as GGA exchange part and the optimum choice of the semi-empirical parameters was determined to be $a_x = 0.53$, $c_{PT} = 0.27$, $c_{DF} = 1$, $b = 1 - c_{PT}$. For convenience, we will suppress the explicit reference to the parameters a_x and b in the XC part and rewrite the gradient corrected XC energy as:

$$E_{XC}\left[\rho^\alpha, \rho^\beta\right] = \int f\left(\rho^\alpha, \rho^\beta, \gamma^{\alpha\alpha}, \gamma^{\beta\beta}, \gamma^{\alpha\beta}\right) d\mathbf{r} \tag{6.63}$$

with the gradient invariants $\gamma^{\sigma\sigma'} = \vec{\nabla}\rho^\sigma \vec{\nabla}\rho^{\sigma'}$. The final term in eq (48) represents the scaled second order perturbation energy:

$$E^{PT2} = \frac{1}{2}\sum_{i_\alpha < j_\alpha}\left\langle \mathbf{t}^{i_\alpha j_\alpha}\bar{\mathbf{K}}^{i_\alpha j_\alpha +}\right\rangle + \frac{1}{2}\sum_{i_\beta < j_\beta}\left\langle \mathbf{t}^{i_\beta j_\beta}\bar{\mathbf{K}}^{i_\beta j_\beta +}\right\rangle + \sum_{i_\alpha, j_\beta}\left\langle \mathbf{t}^{i_\alpha j_\beta}\bar{\mathbf{K}}^{i_\alpha j_\beta +}\right\rangle \tag{6.64}$$

The PT2 amplitudes have been collected in matrices $\mathbf{t}^{i_\sigma j_{\sigma'}}$ with elements:

$$t^{i_\sigma j_{\sigma'}}_{a_\sigma b_{\sigma'}} = \bar{K}^{i_\sigma j_{\sigma'}}_{a_\sigma b_{\sigma'}}\left(\varepsilon^\sigma_i + \varepsilon^{\sigma'}_j - \varepsilon^\sigma_a - \varepsilon^{\sigma'}_b\right)^{-1} \tag{6.65}$$

Where the orbitals were assumed to be canonical with orbital energies ε^σ_p. The exchange operator matrices are $K^{i_\sigma j_{\sigma'}}_{a_\sigma b_{\sigma'}} = (i_\sigma a_\sigma | j_{\sigma'} b_{\sigma'})$ and the anti-symmetrized exchange integrals are defined as $\bar{K}^{i_\sigma j_{\sigma'}}_{a_\sigma b_{\sigma'}} = (i_\sigma a_\sigma | j_{\sigma'} b_{\sigma'}) - \delta_{\sigma\sigma'}(i_\sigma b_\sigma | \sigma a_\sigma)$.

The orbitals satisfy the SCF equations with the matrix element of the SCF operator given by:

$$F^\sigma_{\mu\nu} = h_{\mu\nu} + \sum_{\kappa\tau} P_{\kappa\tau}(\mu\nu|\kappa\tau) - a_x P^\sigma_{\kappa\tau}(\mu\kappa|\nu\tau) + c_{DF}(\mu|V^\sigma_{XC}|\nu) \tag{6.66}$$

The matrix elements of the XC–potential for a gradient corrected functional are: [96]

$$(\mu|V^\alpha_{XC}|\nu) = \int \left\{ \frac{\delta f}{\delta\rho_\alpha(\mathbf{r})}(\varphi_\mu\varphi_\nu) + 2\frac{\delta f}{\delta\gamma_{\alpha\alpha}}\vec{\nabla}\rho_\alpha\vec{\nabla}(\varphi_\mu\varphi_\nu) + \frac{\delta f}{\delta\gamma_{\alpha\beta}}\vec{\nabla}\rho_\beta\vec{\nabla}(\varphi_\mu\varphi_\nu) \right\} d\mathbf{r} \tag{6.67}$$

The energy in equation 6.58 depends on the MO-coefficients, the PT2-amplitudes and through V_{NN}, V_{eN} (in h) and the basis functions also explicitly on the molecular geometry. Unfortunately, the energy is only stationary with respect to the PT2 amplitudes since they can be considered as having been optimized through the minimization of the Hylleraas functional:

$$E_{PT2} = \min_{\mathbf{t}}\left\{ \frac{1}{2}\sum_{i_\alpha < j_\alpha}\left\langle \mathbf{t}^{i_\alpha j_\alpha}\bar{\mathbf{K}}^{i_\alpha j_\alpha +}\right\rangle + \frac{1}{2}\sum_{i_\beta < j_\beta}\left\langle \mathbf{t}^{i_\beta j_\beta}\bar{\mathbf{K}}^{i_\beta j_\beta +}\right\rangle + \sum_{i_\alpha j_\beta}\left\langle \mathbf{t}^{i_\alpha j_\beta}\bar{\mathbf{K}}^{i_\alpha j_\beta +}\right\rangle + \left\langle \mathbf{D}'^\alpha\mathbf{F}^{\alpha+}\right\rangle + \left\langle \mathbf{D}'^\beta\mathbf{F}^{\beta+}\right\rangle \right\} \tag{6.68}$$

The unrelaxed PT2 difference density is defined as:

$$D'^\alpha_{ij} = -\sum_{k_\alpha}\left\langle \mathbf{t}^{i_\alpha k_\alpha}\mathbf{t}^{k_\alpha j_\alpha}\right\rangle - \sum_{k_\beta}\left\langle \mathbf{t}^{i_\alpha k_\beta}\mathbf{t}^{k_\beta j_\alpha}\right\rangle \tag{6.69}$$

$$D'^\alpha_{ab} = \sum_{i_\alpha < j_\alpha}\mathbf{t}^{i_\alpha j_\alpha}\mathbf{t}^{i_\alpha j_\alpha +} + \sum_{i_\beta j_\alpha}\mathbf{t}^{i_\beta j_\alpha +}\mathbf{t}^{i_\beta j_\alpha} \tag{6.70}$$

With analogous expressions for the spin-down unrelaxed difference densities. Minimization of this functional with respect to the amplitudes yields the second order perturbation energy. The derivative of the SCF part of equation 6.58 with respect to a parameter 'x' is straightforward and well known. It yields:

$$E_{SCF}^x = V_{NN}^x + \langle \mathbf{P}\mathbf{h}^x \rangle + \left\langle \mathbf{W}^{SCF}\mathbf{S}^{(x)} \right\rangle + \sum_{\mu\nu\kappa\tau} \Gamma_{\mu\nu\kappa\tau} (\mu\nu|\kappa\tau)^{(x)}$$
$$+ \underbrace{\sum_{\sigma}}_{(\sigma' \neq \sigma)} \int \left\{ \tfrac{\delta f}{\delta \rho_\sigma(\mathbf{r})} \rho_\sigma^{(x)} + 2 \tfrac{\delta f}{\delta \gamma_{\sigma\sigma}} \vec{\nabla}\rho_\sigma \vec{\nabla}\rho_\sigma^{(x)} + \tfrac{\delta f}{\delta \gamma_{\sigma\sigma'}} \vec{\nabla}\rho_{\sigma'} \vec{\nabla}\rho_\sigma^{(x)} \right\} d\mathbf{r} \qquad (6.71)$$

Superscript 'x' refers to the derivative with respect to some perturbation 'x' while a superscript in parenthesis indicates that only the derivative of the basis functions with respect to 'x' is to be taken. For example:

$$\rho_\sigma^{(x)} = \sum_{\mu\nu} P_{\mu\nu}^\sigma \left\{ \tfrac{\partial \varphi_\mu}{\partial x} \varphi_\nu + \varphi_\mu \tfrac{\partial \varphi_\nu}{\partial x} \right\}$$
$$h_{\mu\nu}^x = \left(\tfrac{\partial \varphi_\mu}{\partial x} |\hat{h}| \varphi_\nu \right) + \left(\varphi_\mu |\hat{h}| \tfrac{\partial \varphi_\nu}{\partial x} \right) + \left(\varphi_\mu | \tfrac{\partial \hat{h}}{\partial x} | \varphi_\nu \right) \qquad (6.72)$$

In equation 6.71, $\mathbf{S}$ is the overlap matrix and $\mathbf{W}^{SCF}$ the energy weighted density:

$$W_{\mu\nu}^{SCF} = W_{\mu\nu}^{\alpha;SCF} + W_{\mu\nu}^{\beta;SCF} = - \sum_{i\sigma} c_{\mu i}^\sigma c_{\nu i}^\sigma \varepsilon_i^\sigma \qquad (6.73)$$

At this point, the effective two-particle density matrix is fully separable and reads:

$$\Gamma_{\mu\nu\kappa\tau} = \frac{1}{2} P_{\mu\nu} P_{\kappa\tau} - \frac{1}{2} a_x P_{\mu\kappa}^\alpha P_{\nu\tau}^\alpha - \frac{1}{2} a_x P_{\mu\kappa}^\beta P_{\nu\tau}^\beta \qquad (6.74)$$

The derivative of the PT2 part is considerably more complex, since E_{PT2} is not stationary with respect to changes in the molecular orbitals. This necessitates the solution of the coupled-perturbed SCF (CP-SCF) equations. We follow the standard practice and expand the perturbed orbitals in terms of the unperturbed ones as:

$$\psi_p^{\sigma;x}(\mathbf{r}) = \sum_q U_{qp}^{\sigma;x} \psi_q^\sigma(\mathbf{r}) \qquad (6.75)$$

The occupied-occupied and virtual-virtual blocks of $\mathbf{U}$ are fixed, as usual, through the derivative of the orthonormality constraints:

$$U_{ij}^{\sigma;x} = -\frac{1}{2} S_{ij}^{\sigma(x)} \qquad (6.76)$$

$$U_{ab}^{\sigma;x} = -\frac{1}{2} S_{ab}^{\sigma(x)} \qquad (6.77)$$

$$U_{ia}^{\sigma;x} = -S_{ia}^{\sigma(x)} - U_{ai}^{\sigma;x} \qquad (6.78)$$

Where $S_{pq}^{\sigma(x)} = \sum_{\mu\nu} c_{\mu p}^\sigma c_{\nu q}^\sigma S_{\mu\nu}^{(x)}$. The remaining virtual-occupied block of $\mathbf{U}^x$ must be determined through the solution of the CP-SCF equations. However, as shown by Handy and Schaefer, this step is unnecessary and only a single set of CP-SCF equations (Z-vector equations) needs to be solved. To this end, one defines the Lagrangian:

$$L_{ai}^{\alpha} = R^{\sigma} \left(\mathbf{D}'\right)_{ai} + 2 \sum_{j_{\alpha} b_{\alpha} c_{\alpha}} \left(a_{\alpha} c_{\alpha} | j_{\alpha} b_{\alpha}\right) t_{c_{\alpha} b_{\alpha}}^{i_{\alpha} j_{\alpha}} - 2 \sum_{j_{\alpha} k_{\alpha} b_{\alpha}} \left(k_{\alpha} i_{\alpha} | j_{\alpha} b_{\alpha}\right) t_{a_{\alpha} b_{\alpha}}^{k_{\alpha} j_{\alpha}}$$
$$+ 2 \sum_{j_{\beta} b_{\beta} c_{\alpha}} \left(a_{\alpha} c_{\alpha} | j_{\beta} b_{\beta}\right) t_{b_{\beta} c_{\alpha}}^{j_{\beta} i_{\alpha}} - 2 \sum_{j_{\beta} k_{\alpha} b_{\beta}} \left(k_{\alpha} i_{\alpha} | j_{\beta} b_{\beta}\right) t_{b_{\beta} a_{\alpha}}^{j_{\beta} k_{\alpha}} \tag{6.79}$$

An analogous equation holds for L_{ai}^{β}. The matrix elements of the response operator $R^{\alpha}\left(\mathbf{D}'\right)$ are best evaluated in the AO basis and then transformed into the MO basis. The AO basis matrix elements are given by:

$$R^{\alpha}\left(\mathbf{D}'\right)_{\mu\nu} = \sum_{\kappa\tau} 2 D'_{\kappa\tau} \left(\mu\nu | \kappa\tau\right) - D'^{\alpha}_{\kappa\tau} \left[\left(\mu\kappa | \nu\tau\right) + \left(\nu k | \mu\tau\right)\right]$$
$$+ \sum_{\zeta} \int \left[\frac{\delta^2 f}{\delta\rho_{\alpha}\delta\zeta} \zeta\left(\mathbf{D}'\right)\left(\phi_{\mu}\phi_{\nu}\right) + \left(2\frac{\delta^2 f}{\delta\gamma_{\alpha\alpha}\delta\zeta}\vec{\nabla}\rho_{\mathbf{P}}^{\alpha} + \frac{\delta^2 f}{\delta\gamma_{\alpha\beta}\delta\zeta}\vec{\nabla}\rho_{\mathbf{P}}^{\beta}\right) \zeta\left(\mathbf{D}'\right) \vec{\nabla}\left(\phi_{\mu}\phi_{\nu}\right)\right.$$
$$\left. + \left(2\frac{\delta f}{\delta\gamma_{\alpha\alpha}}\vec{\nabla}\rho_{\mathbf{D}'}^{\alpha} + \frac{\delta f}{\delta\gamma_{\alpha\beta}}\vec{\nabla}\rho_{\mathbf{D}'}^{\beta}\right) \vec{\nabla}\left(\phi_{\mu}\phi_{\nu}\right)\right] d\mathbf{r}$$

where

$$\zeta\left(\mathbf{D}'\right) = \rho_{\mathbf{D}'}^{\alpha}, \rho_{\mathbf{D}'}^{\beta}, \gamma_{\alpha\alpha}\left(\mathbf{D}'\right), \gamma_{\beta\beta}\left(\mathbf{D}'\right), \gamma_{\alpha\beta}\left(\mathbf{D}'\right) \tag{6.80}$$

The ζ-gradient-parameters are evaluated as a mixture of PT2 difference densities and SCF densities. For example:

$$\gamma_{\alpha\alpha}(\mathbf{D}') = 2\vec{\nabla}\rho_{\mathbf{D}'}^{\alpha}\vec{\nabla}\rho_{\mathbf{P}'}^{\alpha} \tag{6.81}$$

With

$$\rho_{\mathbf{D}'}^{\alpha}\left(\mathbf{r}\right) = \sum_{\mu\nu} D'^{\alpha}_{\mu\nu}\phi_{\mu}\left(\mathbf{r}\right)\phi_{\nu}\left(\mathbf{r}\right) \tag{6.82}$$

$$\rho_{\mathbf{P}}^{\alpha}\left(\mathbf{r}\right) = \sum_{\mu\nu} P^{\alpha}_{\mu\nu}\phi_{\mu}\left(\mathbf{r}\right)\phi_{\nu}\left(\mathbf{r}\right) \tag{6.83}$$

Having defined the Lagrangian, the following CP-SCF equations need to be solved for the elements of the 'Z-vector':

$$\left(\varepsilon_a^{\sigma} - \varepsilon_i^{\sigma}\right) Z_{ai}^{\sigma} + R^{\sigma}\left(\mathbf{Z}\right)_{ai} = -L_{ai}^{\sigma} \tag{6.84}$$

The solution defines the occupied-virtual block of the relaxed difference density, which is given by:

$$\mathbf{D}^{\sigma} = \mathbf{D}'^{\sigma} + \mathbf{Z}^{\sigma} \tag{6.85}$$

For convenience, $\mathbf{D}^{\sigma}$ is symmetrized since it will only be contracted with symmetric matrices afterwards. After having solved the Z-vector equations, all parts of the energy weighted difference density matrix can be readily calculated:

$$W_{ij}^{\alpha;PT2} = -\frac{1}{2}D_{ij}^{\alpha}\left(\varepsilon_i^{\alpha} + \varepsilon_j^{\alpha}\right) - \frac{1}{2}R\left(\mathbf{D}\right)_{ij} - \sum_{k_{\alpha} a_{\alpha} b_{\alpha}} \left(i_{\alpha} a_{\alpha} | k_{\alpha} b_{\alpha}\right) t_{a_{\alpha} b_{\alpha}}^{j_{\alpha} k_{\alpha}} - \sum_{k_{\beta} a_{\alpha} b_{\beta}} \left(i_{\alpha} a_{\alpha} | k_{\beta} b_{\beta}\right) t_{b_{\beta} a_{\alpha}}^{k_{\beta} j_{\alpha}} \tag{6.86}$$

$$W_{ab}^{\alpha;PT2} = -\frac{1}{2}D_{ab}^{\alpha}\left(\varepsilon_a^{\alpha} + \varepsilon_b^{\alpha}\right) - \sum_{i_{\alpha} j_{\alpha} c_{\alpha}} \left(i_{\alpha} a_{\alpha} | j_{\alpha} c_{\alpha}\right) t_{b_{\alpha} c_{\alpha}}^{i_{\alpha} j_{\alpha}} - \sum_{i_{\alpha} j_{\beta} c_{\beta}} \left(i_{\alpha} a_{\alpha} | j_{\beta} c_{\beta}\right) t_{c_{\beta} b_{\alpha}}^{j_{\beta} i_{\alpha}} \tag{6.87}$$

$$W_{ai}^{\alpha;PT2} = -2 \sum_{j_\alpha k_\alpha b_\alpha} (k_\alpha i_\alpha | j_\alpha b_\alpha)\, t_{a_\alpha b_\alpha}^{k_\alpha j_\alpha} - 2 \sum_{j_\beta k_\alpha b_\beta} (k_\alpha i_\alpha | j_\beta b_\beta)\, t_{b_\beta a_\alpha}^{j_\beta k_\alpha} \tag{6.88}$$

$$W_{ia}^{\alpha;PT2} = -\varepsilon_i^\alpha Z_{ai}^\alpha \tag{6.89}$$

Once more, analogous equations hold for the spin-down case. With the relaxed difference density and energy weighted density matrices in hand, one can finally proceed to evaluate the gradient of the PT2 part as ($\mathbf{W}^{PT2} = \mathbf{W}^{\alpha;PT2} + \mathbf{W}^{\beta;PT2}$):

$$\begin{aligned} E_{PT2}^x &= \langle \mathbf{D}\mathbf{h}^x \rangle + \left\langle \mathbf{W}^{PT2}\mathbf{S}^{(x)} \right\rangle + \sum_{\mu\nu\kappa\tau} \Gamma_{\mu\nu\kappa\tau}^{PT2} \left(\mu\nu|\kappa\tau\right)^{(x)} \\ &+ \sum_{\substack{\sigma \\ (\sigma\neq\sigma')}} \int \left\{ \frac{\delta f}{\delta \rho_\sigma(\mathbf{r})} \rho_\sigma^{(x)} + 2\frac{\delta f}{\delta\gamma_{\sigma\sigma}} \overset{r}{\nabla}\rho_\sigma \overset{r}{\nabla}\rho_\sigma^{(x)} + \frac{\delta f}{\delta\gamma_{\sigma\sigma}} \overset{r}{\nabla}\rho_{\sigma'} \overset{r}{\nabla}\rho_\sigma^{(x)} \right\} d\mathbf{r} \end{aligned} \tag{6.90}$$

The final derivative of eq. 6.58 is of course the sum $E_{SCF}^x + c_{PT} E_{PT2}^x$. Both derivatives should be evaluated simultaneously in the interest of computational efficiency.

Note that the exchange-correlation contributions to the gradient take a somewhat more involved form than might have been anticipated. In fact, from looking at the SCF XC-gradient (eq. 6.71) it could have been speculated that the PT2 part of the gradient is of the same form but with $\rho_{\mathbf{P}}^{\sigma(x)}$ being replaced by $\hat{H}$, the relaxed PT2 difference density. This is, however, not the case. The underlying reason for the added complexity apparent in equation 6.90 is that the XC contributions to the PT2 gradient arise from the contraction of the relaxed PT2 difference density with the derivative of the SCF operator. Since the SCF operator already contains the first derivative of the XC potential and the PT2 energy is not stationary with respect to changes in the SCF density, a response type term arises which requires the evaluation of the second functional derivative of the XC-functional. Finally, as is well known from MP2 gradient theory, the effective two-particle density matrix contains a separable and a non-separable part:

$$\Gamma_{\mu\nu\kappa\tau}^{PT2} = D_{\mu\nu}P_{\kappa\tau} - D_{\mu\kappa}^\alpha P_{\nu\tau}^\alpha - D_{\mu\kappa}^\beta P_{\nu\tau}^\beta + \Gamma_{\mu\nu\kappa\tau}^{NS} \tag{6.91}$$

$$\Gamma_{\mu\nu\kappa\tau}^{NS} = \sum_{i_\alpha j_\alpha a_\alpha b_\alpha} c_{\mu i}^\alpha c_{\nu a}^\alpha c_{\kappa j}^\alpha c_{\tau b}^\alpha t_{a_\alpha b_\alpha}^{i_\alpha j_\alpha} + \sum_{i_\beta j_\beta a_\beta b_\beta} c_{\mu i}^\beta c_{\nu a}^\beta c_{\kappa j}^\beta c_{\tau b}^\beta t_{a_\beta b_\beta}^{i_\beta j_\beta} + 2\sum_{i_\alpha j_\beta a_\alpha b_\beta} c_{\mu i}^\alpha c_{\nu a}^\alpha c_{\kappa j}^\beta c_{\tau b}^\beta t_{a_\alpha b_\beta}^{i_\alpha j_\beta} \tag{6.92}$$

Thus, the non-separable part is merely the back-transformation of the amplitudes from the MO to the AO basis. It is, however, important to symmetrize the two-particle density matrix in order to be able to exploit the full permutational symmetry of the AO derivative integrals.

6.7.4 Orbital Optimized MP2

The MP2 energy can be regarded as being stationary with respect to the MP2 amplitudes, since they can be considered as having been optimized through the minimization of the Hylleraas functional:

$$E_{MP2} = \min_{\mathbf{t}} \left\{ 2 \left\langle \Psi_1 | \hat{H} | \Psi_0 \right\rangle + \left\langle \Psi_1 | \hat{H}_0 - E_0 | \Psi_1 \right\rangle \right\} \tag{6.93}$$

$\hat{H}$ is the 0th order Hamiltonian as proposed by Møller and Plesset, Ψ_0 is the reference determinant, Ψ_1 is the first-order wave function and $E_0 = E_{HF} = \langle \Psi_{HF} | \hat{H} | \Psi_{HF} \rangle$ is the reference energy. The quantities $\mathbf{t}$ collectively denote the MP2 amplitudes.

The fundamental idea of the OO-MP2 method is to not only minimize the MP2 energy with respect to the MP2 amplitudes, but to minimize the total energy additionally with respect to changes in the orbitals. Since the MP2 energy is not variational with respect to the MO coefficients, no orbital relaxation due to the correlation field is taken into account. If the reference determinant is poor, the low-order perturbative correction then becomes unreliable. This may be alleviated to a large extent by choosing better orbitals in the reference determinant. Numerical evidence for the correctness of this assumption will be presented below.

In order to allow for orbital relaxation, the Hylleraas functional can be regarded as a functional of the wavefunction amplitudes $\mathbf{t}$ and the orbital rotation parameters $\mathbf{R}$ that will be defined below. Through a suitable parameterization it becomes unnecessary to ensure orbital orthonormality through Lagrange multipliers. The functional that we minimize reads:

$$L\{\mathbf{t}, \mathbf{R}\} = E_0[\mathbf{R}] + 2\langle\Psi_1|\hat{H}|\Psi_0\rangle + \langle\Psi_1|\hat{H}_0 - E_0|\Psi_1\rangle \tag{6.94}$$

Ψ_0 is the reference determinant. However, it does no longer correspond to the Hartree-Fock (HF) determinant. Hence, the reference energy $E_0[\mathbf{R}] = \langle\Psi_0[\mathbf{R}]|\hat{H}|\Psi_0[\mathbf{R}]\rangle$ also changes during the variational process and is no longer stationary with respect to the HF MO coefficients. Obviously, $E_0[\mathbf{R}] \geqslant E_{HF}$ since the HF determinant is, by construction, the single determinant with the lowest expectation value of the full Hamiltonian.

The reference energy is given as:

$$E_0[\mathbf{R}] = \sum_i \langle i|h|i\rangle + \frac{1}{2}\sum_{ij}\langle ij||ij\rangle \tag{6.95}$$

The first-order wave function excluding single excitations is:

$$|\Psi_1\rangle = \frac{1}{4}\sum_{ijab} t_{ab}^{ij}|\Psi_{ij}^{ab}\rangle \tag{6.96}$$

A conceptually important point is that Brillouin's theorem [265] is no longer obeyed since the Fock matrix will contain off-diagonal blocks. Under these circumstances the first-order wavefunction would contain contributions from single excitations. Since the orbital optimization brings in all important effects of the singles we prefer to leave them out of the treatment. Any attempt to the contrary will destroy the convergence properties. We have nevertheless contemplated to include the single excitations perturbatively:

$$E_{Singles}^{(2)} = -\sum_{ia}\frac{|F_{ia}|^2}{\varepsilon_a - \varepsilon_i} \tag{6.97}$$

The perturbative nature of this correction would destroy the stationary nature of the total energy and is hence not desirable. Furthermore, results with inclusion of single excitation contributions represent no improvement to the results reported below. They will therefore not be documented below and henceforth be omitted from the OO-MP2 method by default.

The explicit form of the orbital-optimized MP2 Hylleraas functional employing the RI approximation (OO-RI-MP2) becomes:

$$L_\infty[\mathbf{t}, \mathbf{R}] = \sum_i \langle i|\hat{h}|i\rangle + \frac{1}{2}\sum_{ij}\langle ij||ij\rangle + \sum_{iaP}(ia|P)\Gamma_{ia}^{'P} + \sum_{ij}D_{ij}F_{ij} + \sum_{ab}D_{ab}F_{ab} \tag{6.98}$$

with:

$$\Gamma_{ia}'^{P} = \sum_Q V_{PQ}^{-1} \sum_{jb} (Q|jb) t_{ab}^{ij} \tag{6.99}$$

$$(ia|P) = \int \int \psi_i(\mathbf{r_1}) \psi_a(\mathbf{r_1}) \frac{1}{|\mathbf{r_1} - \mathbf{r_2}|} \eta_{\mathbf{P}}(\mathbf{r_2}) \mathbf{dr_1 dr_2} \tag{6.100}$$

$$(P|Q) = \int \int \eta_p(\mathbf{r_1}) \frac{1}{|\mathbf{r_1} - \mathbf{r_2}|} \eta_{\mathbf{Q}}(\mathbf{r_2}) \mathbf{dr_1 dr_2} \tag{6.101}$$

Here, $\{\psi\}$ is the set of orthonormal molecular orbitals and $\{\eta\}$ denotes the auxiliary basis set. F_{pq} denotes a Fock matrix element:

$$F_{pq} = \langle p| \hat{h} |q\rangle + \sum_k \langle pk||qk\rangle \tag{6.102}$$

and it is insisted that the orbitals diagonalize the occupied and virtual subspaces, respectively:

$$\begin{aligned} F_{ij} &= \delta_{ij} F_{ii} = \delta_{ij} \varepsilon_i \\ F_{ab} &= \delta_{ab} F_{aa} = \delta_{ab} \varepsilon_a \end{aligned} \tag{6.103}$$

The MP2 like density blocks are,

$$\begin{aligned} D_{ij} &= -\tfrac{1}{2} \sum_{kab} t_{ab}^{ik} t_{ab}^{jk} \\ D_{ab} &= \tfrac{1}{2} \sum_{ijc} t_{ac}^{ij} t_{bc}^{ij} \end{aligned} \tag{6.104}$$

where the MP2 amplitudes in the case of a block diagonal Fock matrix are obtained through the condition $\frac{\partial L_\infty}{\partial t_{ab}^{ij}} = 0$:

$$t_{ab}^{ij} = -\frac{\langle ij||ab\rangle}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \tag{6.105}$$

The orbital changes are parameterized by an anti-Hermitian matrix $\mathbf{R}$ and an exponential Ansatz,

$$\begin{aligned} \mathbf{c^{new}} &= \mathbf{c^{old}} \exp(\mathbf{R}) \\ \mathbf{R} &= \begin{pmatrix} 0 & \mathbf{R_{ia}} \\ -\mathbf{R_{ia}} & 0 \end{pmatrix} \end{aligned} \tag{6.106}$$

The orbitals changes to second order are,

$$\begin{aligned} \exp(\mathbf{R}) |\mathbf{i}\rangle &= |\mathbf{i}\rangle + \sum_{\mathbf{a}} \mathbf{R_{ai}} |\mathbf{a}\rangle - \tfrac{1}{2} \sum_{\mathbf{jb}} \mathbf{R_{bi}} \mathbf{R_{bj}} |\mathbf{j}\rangle + ... \\ \exp(\mathbf{R}) |\mathbf{a}\rangle &= |\mathbf{a}\rangle - \sum_{\mathbf{i}} \mathbf{R_{ai}} |\mathbf{i}\rangle - \tfrac{1}{2} \sum_{\mathbf{jb}} \mathbf{R_{aj}} \mathbf{R_{bj}} |\mathbf{b}\rangle + ... \end{aligned} \tag{6.107}$$

Through this Ansatz it is ensured that the orbitals remain orthonormal and no Lagrangian multipliers need to be introduced. The first-order expansion of the Fock operator due to the orbital rotations are:

$$F_{pq}[R] = F_{pq}[0] + R_{pq}^{(1)} + \sum_r R_{rp} F_{rq}[0] + R_{rq} F_{pr}[0] \tag{6.108}$$

$$R_{pq}^{(1)} = \sum_{kc} R_{ck} \left\{ \langle pc||qk \rangle + \langle pk||qc \rangle \right\} \tag{6.109}$$

The first-order energy change becomes $\left(h_{pq} \equiv \langle p | \hat{h} | q \rangle , g_{pqrs} \equiv \langle pq||rs \rangle \right)$:

$$
\begin{aligned}
L_\infty [\mathbf{t}, \mathbf{R}] = {} & \sum_{ic} R_{ci} \left(h_{ci} + h_{ic} \right) + \tfrac{1}{2} \sum_{ijc} R_{ci} \left(g_{cjij} + g_{ijcj} \right) + R_{cj} \left(g_{icij} + g_{ijic} \right) \\
& + 2 \sum_{iacP} R_{ci}(ac|P)\Gamma_{ia}'^{P} - 2 \sum_{ikaP} R_{ak}(ik|P)\Gamma_{ia}'^{P} \\
& - \sum_{ij} D_{ij} \left(R_{ij}^{(1)} + \sum_{c} \left(R_{ci} F_{cj} + R_{cj} F_{ic} \right) \right) \\
& + \sum_{ab} D_{ab} \left(R_{ab}^{(1)} - \sum_{k} \left(R_{ak} F_{kb} + R_{bk} F_{ak} \right) \right)
\end{aligned} \tag{6.110}
$$

The condition for the energy functional to be stationary with respect to the orbital rotations $\left(\frac{\partial L_\infty [\mathbf{t}, \mathbf{R}]}{\partial R_{ai}} = 0 \right)$, yields the expression for the orbital gradient and hence the expression for the OO-RI-MP2 Lagrangian.

$$
\begin{aligned}
\frac{\partial L_\infty [\mathbf{t}, \mathbf{R}]}{\partial R_{ai}} \equiv g_{ai} = {} & 2F_{ai} + 2\sum_{j} D_{ij} F_{aj} - 2\sum_{b} D_{ab} F_{ib} + R^{(1)}(\mathbf{D})_{ai} \\
& + 2\sum_{cP} (ac|P)\Gamma_{ia}'^{P} - 2\sum_{kP} (ik|P)\Gamma_{ia}'^{P}
\end{aligned} \tag{6.111}
$$

The goal of the orbital optimization process is to bring this gradient to zero. There are obviously many ways to achieve this. In our experience, the following simple procedure is essentially satisfactory. We first build a matrix $\mathbf{B}$ in the current MO basis with the following structure:

$$
\begin{aligned}
\mathbf{B_{ij}} &= \delta_{\mathbf{ij}} \mathbf{F_{ii}} \\
\mathbf{B_{ab}} &= \delta_{\mathbf{ab}} (\mathbf{F_{aa}} + \mathbf{\Delta}) \\
\mathbf{B_{ai}} &= \mathbf{B_{ia}} = \mathbf{g_{ai}}
\end{aligned} \tag{6.112}
$$

where Δ is a level shift parameter. The occupied/occupied and virtual/virtual blocks of this matrix are arbitrary but their definition has a bearing on the convergence properties of the method. The orbital energies of the block diagonalized Fock matrix appear to be a logical choice. If the gradient is zero, the $\mathbf{B}$-matrix is diagonal. Hence one obtains an improved set of orbitals by diagonalizing $\mathbf{B}$.

In order to accelerate convergence a standard DIIS scheme is used. [97, 159] However, in order to carry out the DIIS extrapolation of the $\mathbf{B}$-matrix it is essential that a common basis is used that does not change from iteration to iteration. Since the B-matrix itself is defined in the molecular orbitals of the current iteration we choose as a common set of orthonormal orbitals the MOs of the HF calculation. The extrapolation is carried out in this basis and the extrapolated $\mathbf{B}$-matrix is transformed back to the current set of MOs prior to diagonalization. Obviously, the same strategy can be used for orbital optimization in any method for which an orbital gradient is available.

For well behaved cases this simple scheme converges in 5-10 iterations. Transition metals and more complicated molecules may require up to 20 iterations and level shifting in order to achieve convergence.

Upon convergence the sum of the matrix $\mathbf{D}$ and the density of the reference determinant $P_{\mu\nu} = \sum_{i} c_{\mu i} c_{\nu i}$ form the true one-particle density matrix of the OO-MP2 approach that can be used for property or gradient calculations.

6.8 The Single Reference Correlation Module

ORCA features a variety of single-reference correlation methods for single point energies (restricted to a RHF or RKS determinant in the closed-shell case and a UHF or UKS determinant in the open-shell case; quasi-restricted orbitals (QROs) [266] are also supported in the open-shell case). They are all fairly expensive but maybe be used in order to obtain accurate results in the case that the reference determinant is a good starting point for the expansion of the many-body wavefunction. The module is called `orca_mdci` for "matrix driven configuration interaction". This is a rather technical term to emphasize that if one wants to implement these methods (CCSD, QCISD etc) efficiently, one needs to write them in terms of matrix operations which can drive pretty much every computer at peak performance. Let us first briefly describe the theoretical background of the methods that we have implemented in ORCA.

6.8.1 Theory

We start from the full CI hierarchy in which the wavefunction is expanded as:

$$|\Psi\rangle = |0\rangle + |S\rangle + |D\rangle + |T\rangle + |Q\rangle + ... \tag{6.113}$$

where $|0\rangle$ is a single-determinant reference and S,D,T,Q,... denote the single, double, triple quadruple and higher excitations relative to this determinant at the spin-orbital level. As usual, labels i, j, k, l refer to occupied orbitals in $|0\rangle$, a, b, c, d to unoccupied MOs and p, q, r, s to general MOs. The action of the second quantized excitation operators $a_i^a = a_a^+ a_i$ on $|0\rangle$ lead to excited determinants $|\Phi_i^a\rangle$ that enter $|\Psi\rangle$ with coefficients C_a^i. The variational equations are:

$$\langle \Phi_i^a | H - E_0 | 0 + S + D \rangle = E_C C_a^i - \langle \Phi_i^a | H - E_0 | T \rangle \tag{6.114}$$

$$\left\langle \Phi_{ij}^{ab} | H - E_0 | 0 + S + D \right\rangle = E_C C_{ab}^{ij} - \left\langle \Phi_{ij}^{ab} | H - E_0 | T + Q \right\rangle \tag{6.115}$$

Further equations coupling triples with singles through pentuples etc.

The total energy is the sum of the reference energy $E_0 = \langle 0 | H | 0 \rangle$ and the correlation energy

$$E_C = \langle 0 | H | S + D \rangle \tag{6.116}$$

which requires the exact singles- and doubles amplitudes to be known. In order to truncate the series to singles- and doubles one may either neglect the terms containing the higher excitations on the right hand side (leading to CISD) or approximate their effect thereby loosing the variational character of the CI method (CCSD, QCISD and CEPA methods). Defining the one- and two-body excitation operators as $\hat{C}_1 = \sum_{ia} C_a^i a_i^a$, $\hat{C}_2 = \frac{1}{4} \sum_{ijab} C_{ab}^{ij} a_{ij}^{ab}$ one can proceed to approximate the triples and quadruples by the disconnected terms:

$$|T\rangle = \hat{C}_1 \hat{C}_2 |0\rangle \tag{6.117}$$

$$|Q\rangle = \frac{1}{2} \hat{C}_2^2 |0\rangle \tag{6.118}$$

As is well known, the CCSD equations contain many more disconnected contributions arising from the various powers of the $\hat{C}_1$ operator (if one would stick to CC logics one would usually label the cluster amplitudes with $t_a^i, t_{ab}^{ij}, \ldots$ and the n-body cluster operators with $\hat{T}_n$; we take a CI point of view here). In order to obtain the CEPA type equations from (6.114-6.118), it is most transparent to relabel the singles and doubles excitations with a compound label P for the internal indices (i) or (ij) and x for (a) or (ab). Then, the approximations are as follows:

$$\frac{1}{2} \left\langle \Phi_P^x \left| (H - E_0) \hat{C}_2^2 \right| 0 \right\rangle = \frac{1}{2} \sum_{QRyz} C_y^Q C_z^R \left\langle \Phi_P^x \left| H - E_0 \right| \Phi_{QR}^{yz} \right\rangle \tag{6.119}$$

$$\approx C_x^P \sum_{Qy} C_y^Q \left\langle \Phi_P^x \left| H \right| \Phi_{PQ}^{xy} \right\rangle \tag{6.120}$$

$$= C_x^P \sum_{Qy} C_y^Q \left\langle 0 \left| H \right| \Phi_Q^y \right\rangle - C_x^P \sum_{Qy \cup Px} C_y^Q \left\langle 0 \left| H \right| \Phi_Q^y \right\rangle \tag{6.121}$$

$$\approx C_x^P \left(E_C - \sum_{Q \cup P} \varepsilon_Q \right) \tag{6.122}$$

Here the second line contains the approximation that only the terms in which either Qy or Rz are equal to Px are kept (this destroys the unitary invariance) and the fourth line contains the approximation that only 'exclusion principle violating' (EPV) terms of internal labels are considered. The notation $Qy \cup Px$ means "Qy joint with Px" (containing common orbital indices) and ε_Q is the pair correlation energy. The EPV terms must be subtracted from the correlation energy since they arise from double excitations that are impossible due to the fact that an excitation out of an occupied or into an empty orbital of the reference determinant has already been performed. Inserting eq 6.122 into eq 6.115 $C_x^P E_C$ cancels and effectively is replaced by the 'partial correlation energy' $\sum_{Q \cup P} \varepsilon_Q$.

The resulting equations thus have the appearance of a diagonally shifted ('dressed') CISD equation $\left\langle \Phi_P^x \left| H - E_0 + \Delta \right| 0 + S + D \right\rangle = 0$. If the second approximation mentioned above is avoided Malrieu's (SC)^2-CISD arises. [98, 187–190] Otherwise, one obtains CEPA/3 with the shift:

$$-\Delta_{ab}^{ij} = \sum_k (\varepsilon_{ik} + \varepsilon_{jk}) - \varepsilon_{ij} \tag{6.123}$$

CEPA/2 is obtained by $-\Delta_{ab}^{ij} = \varepsilon_{ij}$ and CEPA/1 is the average of the CEPA/2 and CEPA/3. As mentioned by Ahlrichs, [267] no consensus appears to exist in the literature for the appropriate shift on the single excitations. If one proceeds straightforwardly in the same way as above, one obtains:

$$\left\langle \Phi_i^a \left| (H - E_0) \hat{C}_1 \hat{C}_2 \right| 0 \right\rangle \approx C_a^i \left(E_C - 2 \sum_k \varepsilon_{ik} \right) \tag{6.124}$$

as the appropriate effect of the disconnected triples on the singles. In has been assumed here that only the singles $|\Phi_i^a\rangle$ in $\hat{C}_1$ contribute to the shift. If $|0\rangle$ is a HF determinant, the effect of the disconnected triples in the doubles projection vanishes under the same CEPA approximations owing to Brillouin's theorem. Averaged CEPA models are derived by assuming that all pair correlation energies are equal (except $\varepsilon_{ii} = 0$). As previously discussed by Gdanitz [268], the averaging of CEPA/1 yields $\frac{2}{n} E_C$ and CEPA/3 $E_C \frac{4n-6}{n(n-1)}$ where n is the number of correlated electrons. These happen to be the shifts used for the averaged-coupled pair functional (ACPF [269]) and averaged quadratic coupled cluster (AQCC [270]) methods respectively. However, averaging the singles shift of eq 6.124 gives $\frac{4}{n} E_C$. The latter is also the leading term in the expansion of the AQCC shift for large n. In view of the instability of ACPF in certain situations, Gdanitz has proposed to use the AQCC shift for the singles and the original ACPF

shift for the doubles and called his new method ACPF/2 [269]. Based on what has been argued above, we feel that it would be most consistent with the ACPF approach to simply use $\frac{4}{n}E_C$ as the appropriate singles shift. We refer to this as NACPF.

It is readily demonstrated that the averaged models may be obtained by a variation of the modified correlation energy functional:

$$E_C = \frac{\langle 0 + S + D \,|H - E_0|\, 0 + S + D \rangle}{1 + g_s \langle S|S \rangle + g_D \langle D|D \rangle} \tag{6.125}$$

with g_S and g_D being the statistical factors $\frac{4}{n}$, $\frac{2}{n}$, $\frac{4n-6}{n(n-1)}$, as appropriate for the given method. Thus, unlike the CEPA models, the averaged models fulfil a stationarity principle and are unitarily invariant. However, if one thinks about localized internal MOs, it appears evident that the approximation of equal pair energies must be one of rather limited validity and that a more detailed treatment of the electron pairs is warranted. Maintaining a stationarity principle while providing a treatment of the pairs that closely resembles that of the CEPA methods was achieved by Ahlrichs and co-workers in an ingenious way with the development of the CPF method [63]. In this method, the correlation energy functional is written as:

$$E_C = 2 \sum_{Px} \frac{\langle \Phi_p^x \,|H|\, 0 \rangle}{N_P} + \sum_{PQxy} \frac{\langle \Phi_p^x \,|H - E_0|\, \Phi_Q^y \rangle}{\sqrt{N_P N_Q}} \tag{6.126}$$

with

$$N_P = 1 + \sum_Q T_{PQ} \sum_y (C_y^Q)^2 \tag{6.127}$$

The topological matrix for pairs $P = (ij)$ and $Q = (kl)$ is chosen as: [118]

$$T_{PQ} = \frac{\delta_{ik} + \delta_{il}}{2n_i} + \frac{\delta_{jk} + \delta_{jl}}{2n_j} \tag{6.128}$$

with n_i being the number of electrons in orbital i in the reference determinant. The singles out of orbital i are formally equated with $P = (ii)$. At the spin-orbital level, $n_i = 1$, for closed shells $n_i = 2$. Using the same toplogical matrix in $\Delta_P = \sum_Q T_{PQ}\varepsilon_Q$ one recovers the CEPA/1 shifts for the doubles in eq 6.124. It is straightforward to obtain the CPF equivalents of the other CEPA models by adjusting the T_{PQ} matrix appropriately. In our program, we have done so and we refer below to these methods as CPF/1, CPF/2 and CPF/3 in analogy to the CEPA models (CPF/1 ≡CPF). In fact, as discussed by Ahlrichs and co-workers, variation of the CPF-functional leads to equations that very closely resemble the CEPA equation and can be readily implemented along the same lines as a simple modification of a CISD program. Ahlrichs et al. argued that the energies of CEPA/1 and CPF/1 should be very close. We have independently confirmed that in the majority of cases, the total energies predicted by the two methods differ by less than 0.1 mEh.

An alternative to the CPF approach which is also based on variational optimization of an energy functional is the VCEPA method [271]. The equations resulting from application of the variational principle to the VCEPA functional are even closer to the CEPA equations than for CPF so that the resulting energies are practically indistinguishable from the corresponding CEPA values. The VCEPA variants are referred to as VCEPA/1, VCEPA/2, and VCEPA/3 in analogy to CEPA and CPF. A strictly size extensive energy functional (SEOI) which is invariant with respect to unitary transformations within the occupied and virtual orbital subspaces is also available [272] (an open-shell version is not implemented yet).

Again, a somewhat critical point concerns the single excitations. They do not account for a large fraction of the correlation energy. However, large coefficients of the single excitations lead to instability and deterioration of the results. Secondly, linear response properties are highly dependent on the effective energies of the singles and their balanced treatment is therefore important. Since the CEPA and CPF methods amount to shifting down the diagonal energies of the singles and doubles, instabilities are expected if the effective energy of an excitation approaches the reference energy of even falls below it. In the CPF method this would show up as denominators N_P that are too small. The argument that the CPF denominators are too small has led Chong and Langhoff to the proposal of the MCPF method which uses a slightly more elaborate averaging than $(N_P N_Q)^{1/2}$ [62].[15] However, their modification was solely based on numerical arguments rather than physical or mathematical reasoning. In the light of eq 6.124 and the performance of the NACPF, it appears to us that for the singles one should use twice the T_{PQ} proposed by Ahlrichs and co-workers. The topological matrix T_{PQ} is modified in the following way for the (very slightly) modified method to which we refer to as NCPF/1:

$$T_{ij,kl} = \frac{\delta_{ik} + \delta_{il}}{2n_i} + \frac{\delta_{jk} + \delta_{jl}}{2n_j} \tag{6.129}$$

$$T_{ij,k} = 0 \tag{6.130}$$

$$T_{i,kl} = 2\frac{\delta_{ik} + \delta_{il}}{n_i} \tag{6.131}$$

$$T_{i,k} = 0 \tag{6.132}$$

(note that $T_{PQ} \neq T_{QP}$ for this choice). Thus, the effect of the singles on the doubles is set to zero based on the analysis of the CEPA approximations and the effect of the singles on the singles is also set to zero. This is a sensible choice since the product of two single excitations is a double excitation which is already included in the SD space and thus none of them can belong to the outer space. It is straightforward to adapt this reasoning about the single excitations to the CEPA versions as well as to NCPF/2 and NCPF/3.

The aforementioned ambiguities arising from the use of single excitations in coupled pair methods can be avoided by using correlation-adapted orbitals instead of Hartree-Fock orbitals thus eliminating the single excitations. There are two alternatives: (a) Brueckner orbitals and (b) optimized orbitals obtained from the variational optimization of the electronic energy with respect to the orbitals. Both approaches have already been used for the coupled cluster doubles (CCD) method [273, 274] and later been extended to coupled pair methods [275]. In the case of CCD, orbital optimization requires the solution of so-called Λ (or Z vector) equations [276]. There is, however, a cheaper alternative approximating the Z vector by a simple analytical formula [277].

Futhermore, the parametrized coupled cluster (pCCSD) method of Huntington and Nooijen [278], which combines the accuracy of coupled pair type methods for (usually superior to CCSD, at least for energies and energy differences) with the higher stability of the coupled cluster methods, is an attractive alternative. Comprehensive numerical tests [279] indicate that particularly pCCSD(-1,1,1) (or pCCSD/1a) and pCCSD (-1.5,1,1) (or pCCSD/1b) have a great potential for accurate computational thermochemistry.

---

[15]This method – although it has been rather extensively used in the past – is not implemented in ORCA. We recommend to use our NCPF/1 instead.

6.8.2 Closed Shell Equations

Proceeding from spin-orbitals to the spatial orbitals of a closed-shell determinant leads to the actual working equations of this work. Saebo, Meyer and Pulay have exploited the generator state formalism to arrive at a set of highly efficient equations for the CISD problem [66]. A similar set of matrix formulated equations for the CCSD and QCISD cases has been discussed by Werner and co-workers [280] and the MOLPRO implementation is widely recognized to be particular efficient. Equivalent explicit equations for the CISD and CCSD methods were published by Scuseria et al. [281][16] The doubles equations for the residual 'vector' σ are ($i \leqslant j$, all a, b):

$$
\begin{aligned}
\sigma_{ab}^{ij} = {} & K_{ab}^{ij} + K\left(\mathbf{C}^{ij}\right)_{ab} + \left\{\mathbf{F}^V \mathbf{C}^{ij} + \mathbf{C}^{ij}\mathbf{F}^V\right\}_{ab} - \sum_k \left\{F_{jk} C_{ab}^{ik} + F_{ik} C_{ab}^{kj}\right\} + \sum_{kl} K_{kl}^{ij} C_{ab}^{kl} \\
& + \sum_k \left\{\left(2\mathbf{C}^{ik} - \mathbf{C}^{ik+}\right)\left(\mathbf{K}^{kj} - \tfrac{1}{2}\mathbf{J}^{kj}\right) + \left(\mathbf{K}^{ik} - \tfrac{1}{2}\mathbf{J}^{ik}\right)\left(2\mathbf{C}^{kj} - \mathbf{C}^{kj+}\right)\right\}_{ab} \\
& - \sum_k \left\{\tfrac{1}{2}\mathbf{C}^{ik+}\mathbf{J}^{jk+} + \tfrac{1}{2}\mathbf{J}^{ik}\mathbf{C}^{kj+} + \mathbf{J}^{jk}\mathbf{C}^{ik} + \mathbf{C}^{kj}\mathbf{J}^{ik+}\right\}_{ab} \\
& + C_a^i F_b^j + C_b^j F_a^i - \sum_k \left\{K_{ka}^{ji} C_b^k + K_{kb}^{ij} C_a^k\right\} + \left\{\mathbf{K}^{ia}\mathbf{C}^j + \mathbf{K}^{ja}\mathbf{C}^i\right\}_b \\
& - \Delta^{ij} C_{ab}^{ij}
\end{aligned}
\tag{6.133}
$$

The singles equations are:

$$
\begin{aligned}
\sigma_a^i = {} & F_a^i + \left\{\mathbf{F}^V \mathbf{C}^i\right\}_a - \sum_j F_{ij} C_a^j - \sum_{jkb}\left(2K_{jb}^{ik} - J_{jb}^{ik}\right) C_{ba}^{kj} \\
& + \sum_j \left\{\left(2\mathbf{K}^{ij} - \mathbf{J}^{ij}\right)\mathbf{C}^j + \mathbf{F}^j\left(2\mathbf{C}^{ij+} - \mathbf{C}^{ij}\right) + \left\langle\left(2\mathbf{K}^{ia} - \mathbf{K}^{ia+}\right)\mathbf{C}^{ij+}\right\rangle\right\}_a \\
& - \Delta^i C_a^i
\end{aligned}
\tag{6.134}
$$

The following definitions apply:

$$
K\left(\mathbf{C}^{ij}\right)_{ab} = \sum_{cd}(ac|bd)\, C_{cd}^{ij}
\tag{6.135}
$$

$$
K_{rs}^{pq} = (pr|qs)
\tag{6.136}
$$

$$
J_{rs}^{pq} = (pq|rs)
\tag{6.137}
$$

$$
\langle\mathbf{AB}\rangle = \sum_{pq} A_{pq} B_{qp}
\tag{6.138}
$$

The two-electron integrals are written in (11|22) notation and $\mathbf{F}$ is the closed-shell Fock operator with $\mathbf{F}^V$ being its virtual sub-block. We do not assume the validity of Brillouin's theorem. The amplitudes C_a^i, C_{ab}^{ij} have been collected in vectors $\mathbf{C}^i$ and matrices $\mathbf{C}^{ij}$ wherever appropriate. The shifts Δ^i and Δ^{ij} are dependent on the method used and are defined in Table 6.7 for each method implemented in ORCA.

---

[16]Our coupled cluster implementation is largely based on this nice paper. The equations there have been extensively verified to be correct.

Table 6.7: Summary of the diagonal shifts used in various singles- and doubles methods discussed in this chapter. The quantities ε_i and ε_{ij} are the correlation energy increments brought about by the single- and the double excitations respectively. The partial denominators for the CPF type methods N_i and N_{ij} are specified in eq 6.127.

Method	Doubles Shift	Singles Shift
CISD	E_C	$\mathbf{E_C}$
CEPA/0	0	**0**
CEPA/1	$\frac{1}{2}(\varepsilon_i + \varepsilon_j) + \frac{1}{4}\sum_k (\varepsilon_{ik} + \varepsilon_{jk})$	$\frac{1}{2}\varepsilon_{ii} + \frac{1}{2}\sum_k \varepsilon_{ik}$
CEPA/2	$\delta_{ij}\varepsilon_i + \varepsilon_{ij}$	$\varepsilon_i + \varepsilon_{ii}$
CEPA/3	$(\varepsilon_i + \varepsilon_j) - \delta_{ij}\varepsilon_i - \varepsilon_{ij} + \frac{1}{2}\sum_k (\varepsilon_{ik} + \varepsilon_{jk})$	$\varepsilon_i + \sum_k \varepsilon_{ik}$
NCEPA/1	$\frac{1}{4}\sum_k (\varepsilon_{ik} + \varepsilon_{jk})$	$\varepsilon_{ii} + \sum_k \varepsilon_{ik}$
NCEPA/2	ε_{ij}	$2\varepsilon_{ii}$
NCEPA/3	$-\varepsilon_{ij} + \frac{1}{2}\sum_k (\varepsilon_{ik} + \varepsilon_{jk})$	$2\sum_k \varepsilon_{ik}$
CPF/1	$N_{ij}\left\{\frac{1}{2}\left(\frac{\varepsilon_i}{N_i} + \frac{\varepsilon_j}{N_j}\right) + \frac{1}{4}\sum_k \left(\frac{\varepsilon_{ik}}{N_{ik}} + \frac{\varepsilon_{jk}}{N_{jk}}\right)\right\}$	$N_i\left\{\frac{1}{2}\frac{\varepsilon_{ii}}{N_{ii}} + \frac{1}{2}\sum_k \frac{\varepsilon_{ik}}{N_{ik}}\right\}$
CPF/2	$N_{ij}\left\{\delta_{ij}\frac{\varepsilon_i}{N_i} + \frac{\varepsilon_{ij}}{N_{ij}}\right\}$	$N_i\left\{\frac{\varepsilon_i}{N_i} + \frac{\varepsilon_{ii}}{N_{ii}}\right\}$
CPF/3	$N_{ij}\left\{\frac{\varepsilon_i}{N_i}(1 - \delta_{ij}) + \frac{\varepsilon_j}{N_j} - \frac{\varepsilon_{ij}}{N_{ij}} + \frac{1}{2}\sum_k \left(\frac{\varepsilon_{ik}}{N_{ik}} + \frac{\varepsilon_{jk}}{N_{jk}}\right)\right\}$	$N_i\left\{\frac{\varepsilon_i}{N_i} + \sum_k \frac{\varepsilon_{ik}}{N_{ik}}\right\}$
NCPF/1	$\frac{1}{4}N_{ij}\sum_k \left(\frac{\varepsilon_{ik}}{N_{ik}} + \frac{\varepsilon_{jk}}{N_{jk}}\right)$	$N_i\left\{\frac{\varepsilon_{ii}}{N_{ii}} + \sum_k \frac{\varepsilon_{ik}}{N_{ik}}\right\}$
NCPF/2	$N_{ij}\frac{\varepsilon_{ij}}{N_{ij}}$	$2N_i\frac{\varepsilon_{ii}}{N_{ii}}$
NCPF/3	$N_{ij}\left\{-\frac{\varepsilon_{ij}}{N_{ij}} + \frac{1}{2}\sum_k \left(\frac{\varepsilon_{ik}}{N_{ik}} + \frac{\varepsilon_{jk}}{N_{jk}}\right)\right\}$	$2N_i\sum_k \frac{\varepsilon_{ik}}{N_{ik}}$
ACPF	$\frac{2}{n}E_C$	$\frac{2}{n}E_C$
ACPF/2	$\frac{2}{n}E_C$	$\left[1 - \frac{(n-3)(n-2)}{n(n-1)}\right]E_C$
NACPF	$\frac{2}{n}E_C$	$\frac{4}{n}E_C$
AQCC	$\left[1 - \frac{(n-3)(n-2)}{n(n-1)}\right]E_C$	$\left[1 - \frac{(n-3)(n-2)}{n(n-1)}\right]E_C$

The QCISD method requires some slight modifications. We found it most convenient to think about the effect of the nonlinear terms as a "dressing" of the integrals occurring in equations 6.133 and 6.134. This attitude is close to the recent arguments of Heully and Malrieu and may even open interesting new routes towards the calculation of excited states and the incorporation of connected triple excitations. [282] The dressed integrals are given by:

$$\bar{F}_{ik} = F_{ik} + \sum_l \left\langle \mathbf{C}^{il}\left(2\mathbf{K}^{kl} - \mathbf{K}^{kl+}\right)\right\rangle \tag{6.139}$$

$$\bar{F}_{ab} = F_{ab} - \sum_{kl} \left\{\mathbf{C}^{kl}\left(2\mathbf{K}^{kl} - \mathbf{K}^{kl+}\right)\right\}_{ab} \tag{6.140}$$

$$\bar{F}_{kc} = F_{kc} + \sum_l \left(2\mathbf{K}^{kl} - \mathbf{K}^{kl+}\right)\mathbf{C}^l \tag{6.141}$$

$$\bar{K}_{kl}^{ij} = K_{kl}^{ij} + \left\langle \mathbf{K}^{kl}\mathbf{T}^{kl+}\right\rangle \tag{6.142}$$

$$\bar{K}_{ab}^{ij} = K_{ab}^{ij} + \sum_k \left\{\mathbf{C}^{ik}\left(\mathbf{K}^{kj} - \frac{1}{2}\mathbf{K}^{jk}\right) + \mathbf{C}^{ki}\mathbf{K}^{kj}\right\}_{ab} \tag{6.143}$$

$$\bar{J}_{ab}^{ij} = J_{ab}^{ij} + \sum_k \left\{ \mathbf{C}^{ki} \mathbf{K}^{jk} \right\}_{ab} \tag{6.144}$$

The CCSD method can be written in a similar way but requires 15 additional terms that we do not document here. They may be taken conveniently from our paper about the LPNO-CCSD method [283].

A somewhat subtle point concerns the definition of the shifts in making the transition from spin-orbitals to spatial orbitals. For example, the CEPA/2 shift becomes in the generator state formalism:

$$-\left\langle \tilde{\Phi}_{ij}^{ab} | \Delta^{ij} | \Psi \right\rangle = C_{ab}^{ij} \left(\frac{1}{3} \varepsilon_{ij}^{\alpha\alpha} + \frac{2}{3} \varepsilon_{ij}^{\alpha\beta} \right) + C_{ba}^{ij} \left(-\frac{1}{3} \varepsilon_{ij}^{\alpha\alpha} + \frac{1}{3} \varepsilon_{ij}^{\alpha\beta} \right) \tag{6.145}$$

($\tilde{\Phi}_{ij}^{ab}$ is a contravariant configuration state function, see Pulay et al. [276]. The parallel and antiparallel spin pair energies are given by:

$$\varepsilon_{ij}^{\alpha\alpha} = \frac{1}{2} \sum_{ab} \left[K_{ab}^{ij} - K_{ba}^{ij} \right] \left(C_{ab}^{ij} - C_{ba}^{ij} \right) \tag{6.146}$$

$$\varepsilon_{ij}^{\alpha\beta} = \frac{1}{2} \sum_{ab} K_{ab}^{ij} C_{ab}^{ij} \tag{6.147}$$

This formulation would maintain the exact equivalence of an orbital and a spin-orbital based code. Only in the (unrealistic) case that the parallel and antiparallel pair correlation energies are equal the CEPA/2 shift of 6.7 arise. However, we have not found it possible to maintain the same equivalence for the CPF method since the electron pairs defined by the generator state formalism are a combination of parallel and antiparallel spin pairs. In order to maintain the maximum degree of internal consistency we have therefore decided to follow the proposal of Ahlrichs and co-workers and use the topological matrix T_{PQ} in equation 6.128 and the equivalents thereof in the CEPA and CPF methods that we have programmed.

6.8.3 Open Shell Equations

We have used a non-redundant set of three spin cases ($\alpha\alpha$, $\beta\beta$, $\alpha\beta$) for which the doubles amplitudes are optimized separately. The equations in the spin-unrestricted formalism are straightforwardly obtained from the corresponding spin orbital equations by integrating out the spin. For implementing the unrestricted QCISD and CCSD method, we applied the same strategy (dressed integrals) as in the spin-restricted case. The resulting equations are quite cumbersome and will not be shown here explicitly [284].

Note that the definitions of the spin-unrestricted CEPA shifts differ from those of the spin-restricted formalism described above (see Kollmar et al. [284]). Therefore, except for CEPA/1 and VCEPA/1 (and of course CEPA/0), for which the spin-adaptation of the shift can be done in a consistent way, CEPA calculations of closed-shell molecules yield slightly different energies for the spin-restricted and spin-unrestricted versions. Since variant 1 is also the most accurate among the various CEPA variants [285], we recommend to use variant 1 for coupled pair type calculations. For the variants 2 and 3, reaction energies of reactions involving closed-shell and open-shell molecules simultaneously should be calculated using the spin-unrestricted versions only.

A subtle point for open-shell correlation methods is the choice of the reference determinant [286]. Single reference correlation methods only yield reliable results if the reference determinant already provides a good description of the systems electronic structure. However, an UHF reference wavefunction suffers from spin-contamination which can spoil the results and lead to convergence problems. This can be avoided if quasi-restricted orbitals (QROs) are used [266,282] since the corresponding wavefunction is an eigenfunction of the $\hat{S}^2$ operator and thus, no spin-contamination will

appear. The coupled pair and coupled cluster equations will be still solved in a spin-unrestricted formalism but the energy will be slightly higher compared to the results obtained with a spin-polarized UHF reference determinant. Furthermore, especially for more difficult systems like e.g. transition metal complexes, it is often advantageously to use Kohn-Sham (KS) orbitals instead of HF orbitals.

6.8.4 Use of the MDCI Module

The MDCI module is fairly easy to use. The flags for the "simple" input lines have been described in section 3.2. The detailed listing of options is found below:

```
    %mdci citype CISD      # CI singles+doubles
                 QCISD     # quadratic CI (singles+doubles)
                 CCSD      # coupled cluster singles+doubles
                 CEPA_1    # couplet-electron pair approximation '1'
                 CEPA_2    #
                 CEPA_3    #
                 CPF_1     # Coupled pair functional approximation '1'
                 CPF_2     # (note that CPF/1 is identical with
                 CPF_3     #  the original CPF of Ahlrichs et al.)
                 VCEPA_1   # Variational CEPA approximation '1'
                 VCEPA_2   #
                 VCEPA_3   #
                 NCEPA_1   # our slightly modified versions of CEPA
                 NCEPA_2   # and CPF
                 NCEPA_3   #
                 NCPF_1    #
                 NCPF_2    #
                 NCPF_3    #
                 VNCEPA_1  #
                 VNCEPA_2  #
                 VNCEPA_3  #
                 ACPF      # averaged coupled pair functional
                 ACPF_2    # Gdanitz modification of it
                 NACPF     # our modification of it
                 AQCC      # Szalay + Bartlett
                 SEOI      # a strictly size extensive energy functional
                           # maintaining unitary invariance (not yet
                           # available for UHF)
                 MP3       # MP3 calculation. With UseSCS=true it is
                           # SCS-MP3
            ewin -3,1e3    # orbital energy window to determine which
                           # MOs are included in the treatment
          # (respects settings in %method block)
              Singles true  # include single excitations in the
                            # treatment (default true)
```

```
                 Triples 0     # (T) correction in CCSD(T)/QCISD(T)
                               # default is no triples
                      1        # Algorithm 1 (lots of memory, fast)
                      2        # Algorithm 2 (less memory, ~2x slower)
                 Brueckner true   # use Brueckner orbitals
                                  # (default false)
                 Denmat none       # no evaluation of density matrices
                       linearized # density matrix obtained by retaining
                                  # only CEPA_0-like terms, i.e., those
                                  # linear in the excitation amplitudes
                       unrelaxed  # unrelaxed density matrices, i.e.,
                                  # density matrices without orbital
                                  # relaxation
                       orbopt     # perform orbital optimization yielding
                                  # fully relaxed density matrices (if
                                  # citype chosen as CCSD or QCISD this option
                                  # implies evaluation of the Z vector).
                                  # (default: linearized)
                 ZSimple true     # simplified evaluation of the Z vector
                                  # in case of orbital optimized CCD
                                  # (citype chosen as CCSD or QCISD and
                                  # Denmat as orbopt) by using an
                                  # analytical formula
                       false      # explicit solution of Z vector
                                  # equations
                                  # in case of orbital optimized CCD
                                  # (default: false)
                 UseQROs          # use of quasi-restricted orbitals
                                  # (default false)
                 Localize 0    # use localized MOs. Presently very little
                               # use is made of locality. It may help
                               # for interpretations. Localization is
                               # incompatible with the (T) correction
                         PM    # Use Pipek-Mezey localized MOs
                         FB    # use Foster-Boys localized MOs
                 NatOrbIters 0 # Perform natural orbital iterations.
                               # default is none. Not possible for CCSD
                               # and QCISD
                 pCCSDAB       # the three parameters for parametrized
                 pCCSDCD       # coupled cluster (default is 1.0 which
                 pCCSDEF       # corresponds to normal CCSD
                 # this defines how the rate limiting step is handled
                 # MO and AOX need lots of disk and I/O but if they
                 # can be done they are fast
                 KCOpt KC_MO    # Perform full 4-index transformation
```

```
            KC_AOBLAS# AO direct with BLAS (preferred)
            KC_AO    # AO direct handling of 3,4 externals
            KC_RI    # RI approximation of 3,4 externals
            KC_RI2   # Alternative RI (not recommended)
            KC_AOX   # Do it from stored AO exchange integrals
PrintLevel 2    # Control the amount of output. For 3 and
                # higher things like pair correlation
                # energies are printed.
MaxIter 35      # Max. number of iterations
# How the integral transformation is done.
# Note that it is fine to do AOX or AO or AOBLAS
# together with trafo_ri
TrafoType trafo_jk # Partial trafo to J+K operators
          trafo_ri   # RI transformation of all
                     # integrals up to 2-externals
                     # (3-ext for (T))and rest on the
                     # fly
          trafo_full # Full four index transformation.
                     # Automatically chosen for
                     # KCOpt=KC_MO
MaxCoreWork 350 # Memory in MB used for integral
                # trafos and batching. The program
                # may, however, occasionally use more
                # than that - so don't be too generous
MaxCoreIntAmp   # Max. memory in MB for integrals
                # and amplitudes
STol  1e-5      # Max. element of the residual vector
                # for convergence check
LShift 0.3      # Level shift to be used in update of
                # coefficients
MaxDIIS  7      # Max number of DIIS vectors to be stored
# this let's you control how much and what is residing
# in central memory. May speed up things. Note that
# MaxCore is not respected here 9
InCore   0   # nothing in core
         1   #  + sigma-vector and amplitudes (default)
         2   #  + Jij(a,b) Kij(a,b) operators
         3   #  + DIIS vectors
         4   #  + 3-exernal integral Kia(b,c)
         5   #  + 4-external integrals Kab(c,d)
             # this is identical to ALL
         # the default is AUTO which means that incore
         # is chosen based on MaxCoreIntAmp
end
```

6.9 The Complete Active Space Self-Consistent Field (CASSCF) Module

6.9.1 General Description

The complete active space self-consistent field (CASSCF) method is a special form a multiconfigurational SCF method and can be thought of as an extension of the Hartree-Fock method. It is a very powerful method to study static correlation effects and a solid basis for MR-CI and MR-PT treatments. It can be applied to the ground state and excited states or averages thereof. The implementation in ORCA is fairly general and reasonably efficient. However, CASSCF calculations are fairly complex and ultimately require a lot of insight from the user in order to be successful.

The wavefunction. The wavefunction of a given CASSCF state is written as:

$$\left|\Psi_I^S\right\rangle = \sum k C_{kl} \left|\Phi_k^S\right\rangle \tag{6.148}$$

Here, $\left|\Psi_I^S\right\rangle$ is the CASSCF N-electron wavefunction for state I with total spin S. The sect of $\left|\Phi_k^S\right\rangle$ is a set of configuration state functions (for example linear combination of Slater determinants) each adapted to a total spin S. The expansion coefficients C_{kI} represent the first set of variational parameters. Each CSF is constructed from a common set of orthonormal molecular orbitals $\psi_i(\mathbf{r})$ which are in turn expanded in basis functions $\psi_i(\mathbf{r}) = \sum_\mu c_{\mu i}\phi_\mu(\mathbf{r})$. The MO coefficients $c_{\mu i}$ form the second set of variational parameters.

The energy. The energy of the CASSCF wavefunction is given by the Rayleigh quotient:

$$E(\mathbf{c}, \mathbf{C}) = \frac{\left\langle \Psi_I^S \left| \hat{H}_{BO} \right| \Psi_I^S \right\rangle}{\left\langle \Psi_I^S | \Psi_I^S \right\rangle} \tag{6.149}$$

And represent and upper bound to the true total energy. However, CASSCF calculations are *not* designed to provide values for total energy which are close to the exact energy. The value of a CASSCF calculation is rather to provide a qualitatively correct wavefunction of reasonable quality and which forms (if desired) a good starting point for a treatment of dynamic correlation.

The CASSCF method is fully variational in the sense that the energy is made stationary with respect to variations in both sets of MO and CI coefficients. At convergence, the gradient of the energy with respect to the MO and CI coefficients vanishes:

$$\frac{\partial E(\mathbf{c}, \mathbf{C})}{\partial c_{\mu i}} = 0 \tag{6.150}$$

$$\frac{\partial E(\mathbf{c}, \mathbf{C})}{\partial C_{kI}} = 0 \tag{6.151}$$

Orbital spaces. In CASSCF calculations, the MO space is divided into three user defined subspaces:

- The 'internal orbitals' are the orbitals which are doubly occupied in all configuration state functions (labels i, j, k, l).

- The 'active orbitals' are the orbitals with variable occupation numbers in the various CSFs (labels t, u, v, w).

- The 'external orbitals' (labels a, b, c, d)

The wavefunction and energy is invariant with respect to unitary transformations within the three subspaces. The special feature of a CASSCF wavefunction is that a fixed number of electrons is assigned to each subspace. The internal subspace is of course completely filled but the CSFs in the active space constitute a full-CI of n-electrons in m-orbitals. The CSF list is constructed such, however, that a wavefunction of well defined total spin (and potential space) symmetry results. Such a wavefunction is referred to as a **CASSCF(n,m)** wavefunction. The CSF list grows extremely quickly with the number of active orbitals and the number of active electrons (basically factorially). Depending on the system, the limit of feasibility is roughly around ~14 active orbitals or about one million CSFs in the active space (future ORCA releases may be able to treat even longer expansions efficiently).

Since the orbitals within the subspaces are only defined up to a unitary transformation, the program needs to make some canonicalization choice.

In ORCA, the final orbitals by default are:

1. natural orbitals in the active space,

2. orbitals which diagonalize the CASSCF Fock matrix in the internal space and

3. orbitals which diagonalize the CASSCF Fock matrix in the external space.

State averaging. In many circumstances, it is desirable to optimize the orbitals not for a single state but for the average of several states. In order to see what is done, the energy for state I is re-written as:

$$E_I\left(\mathbf{c},\mathbf{C}\right) = \sum_{pq}\Gamma_q^{p(I)}h_{pq} + \sum_{pqrs}\Gamma_{qs}^{pr(I)}\left(pq\,|rs\right) \tag{6.152}$$

Here, $\Gamma_q^{p(I)}$ and $\Gamma_{qs}^{pr(I)}$ are the one-and two-particle reduced electron density matrices for this state (labels p, q, r, s span the internal and active subspaces):

$$\Gamma_q^{p(I)} = \left\langle \Psi_I^S \left| E_q^p \right| \Psi_I^S \right\rangle \tag{6.153}$$

$$\Gamma_{qs}^{pr(I)} = \frac{1}{2}\left\langle \Psi_I^S \left| E_q^p E_s^r - \delta_{qr}E_s^p \right| \Psi_I^S \right\rangle \tag{6.154}$$

The average energy is simply obtained from averaging the density matrices using arbitrary weights w_I that are user defined but are constrained to sum to unity.

$$\Gamma_q^{p(av)} = \sum_I w_I \Gamma_q^{p(I)} \tag{6.155}$$

$$\Gamma_{qs}^{pr(av)} = \sum_I w_I \Gamma_{qs}^{pr(I)} \tag{6.156}$$

$$\sum_I w_I = 1 \tag{6.157}$$

Optimization of CASSCF wavefunctions. In general, except for trivial cases, CASSCF wavefunctions are considerably more difficult to optimize than RHF (or UHF) wavefunctions. The underlying reason is that variations in $\mathbf{c}$ and $\mathbf{C}$ maybe strongly coupled and the energy functional may have many local minima in $(\mathbf{c},\mathbf{C})$ space. Consequently, the choice of starting orbitals is of really high-importance and the choice which orbitals and electrons are included in the active space has decisive influence on the success of a CASSCF study. In general, after transformation to natural

orbitals, one can classify the active space orbitals by their occupation numbers which vary between 0.0 and 2.0. In general, convergence problems are almost guaranteed if orbitals with occupation numbers close to zero or close to 2.0 are included in the active space. Occupation numbers between 0.02 and 1.98 are typically very reasonable and should not lead to large convergence problems. The reason for the occurrence of convergence problems is that the energy is only very weakly dependent on rotations between internal and active orbitals if the active orbital is almost doubly occupied and similarly for the rotations between external and weakly occupied active orbitals. However, in some cases (for example in the study of potential energy surfaces) it may not be avoidable to include weakly or almost inactive orbitals in the active space and in these cases the use of the most powerful convergence aids is necessary (*vide infra*). As in the case of single-determinant wavefunctions (RHF, UHF, RKS, UKS) there are first and second order converging methods available. The first order CASSCF methods require the integrals transformed integrals (tu|vx) with x belonging to any subspace. This is a very small subspace of the total transformed integral list and is readily held in central storage even for larger calculations. On the other hand, second order CASSCF methods require the integrals $(pq|xy)$ and $(px|qy)$ (p, q = internal, active; x, y = any orbital). This is a fairly large set of integrals and their generation is laborious in terms of CPU time and disk storage. Second order CASSCF calculations are therefore more limited in the size of the molecules which can be well treated. It would be possible to basically avoid the integral transformation also in the case of second-order CASSCF calculations and proceed to fully direct calculations. Such calculations may become quite time consuming since there may be a large number of Fock matrix builds necessary.

The augmented Hessian method solves the eigenvalue problem:

$$\begin{pmatrix} 0 & \mathbf{g} \\ \mathbf{g} & \mathbf{H} \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{t} \end{pmatrix} = \varepsilon \begin{pmatrix} 1 \\ \mathbf{t} \end{pmatrix} \tag{6.158}$$

Here, $\mathbf{g}$ is the orbital gradient (derivative of the total energy with respect to a non-redundant rotation between two orbitals) and $\mathbf{H}$ is the orbital Hessian (second derivative of the energy with respect to two non-redundant orbital rotations). The vector $\mathbf{t}$ (in intermediate normalization obtained from the CI like vector) summarizes the rotation angles. The angles are used to define the antisymmetric matrix ($X_{pq} = -X_{qp}$ is thus the rotation angle between orbitals p and q):

$$\mathbf{X} = \begin{pmatrix} \mathbf{0} & \mathbf{t} \\ -\mathbf{t} & \mathbf{0} \end{pmatrix} \tag{6.159}$$

Which is used to parametrize the unitary matrix $\mathbf{U} = \exp(\mathbf{X})$ which is used to update the orbitals according to:

$$\mathbf{c}^{new} = \mathbf{c}^{old}\mathbf{U} \tag{6.160}$$

(where $\mathbf{c}$ is an MO coefficient matrix).

Starting orbitals. You cannot be careful enough with your starting orbitals - Canonical orbitals or HF orbitals from a RHF calculation are seldom a good choice! Usually DFT orbitals (quasi-restricted or RKS) often do better. Alternatively, if CASSCF orbitals from a previous run or a close by geometry are available this is a good choice. In my hands, the PAtom guess produces more reliable start orbitals than PModel. However, the best natural orbitals from a simple correlation calculation like MP2 or a calculation with the MRCI module are usually the best choice. For example:

```
#
# First job provides reasonable natural orbitals
#
```

```
    ! RHF RI-MP2 SVP SV/C SmallPrint

%mp2 natorbs true
     density relaxed
     end

* int 0 1
   C  0 0 0 0.00   0.0    0.00
   O  1 0 0 1.20   0.0    0.00
   H  1 2 0 1.10 120.0    0.00
   H  1 2 3 1.10 120.0  180.00
*
```

Now examine the occupation numbers of the natural orbitals (you will find that in the output of the MP2 part of the calculation):

```
Natural Orbital Occupation Numbers:
N[  0] =   2.00000622
N[  1] =   2.00000233
N[  2] =   1.98685690
N[  3] =   1.97743095
N[  4] =   1.97533829
N[  5] =   1.96915811
N[  6] =   1.96443726
N[  7] =   1.93992919
N[  8] =   0.05371120
N[  9] =   0.02543285
N[ 10] =   0.02444375
N[ 11] =   0.01355852
N[ 12] =   0.01093080
N[ 13] =   0.00984154
N[ 14] =   0.00648238
N[ 15] =   0.00626053
```

A rule of thumb is that orbitals with occupation numbers between 1.98 and 0.02 should be in the active space. Thus, in the present case we speculate that a 10 electrons in 8 orbitals active space would be appropriate for the CASSCF of the ground state. Let's try:

```
#
# Run a CASSCF calculation for the ground state of H2CO
#
! RHF  SVP SV/C SmallPrint
! moread
%moinp "Test-CASSCF-MP2-H2CO.mp2nat"

%casscf nel        10
        norb        8
```

```
         switchstep nr
         switchconv 0.08
         end


 * int 0 1
    C  0 0 0 0.00    0.0     0.00
    O  1 0 0 1.20    0.0     0.00
    H  1 2 0 1.10 120.0      0.00
    H  1 2 3 1.10 120.0   180.00
 *
```

If we run that calculation it converges (somewhat a little slowly in 8 iterations) and produces the following:

```
MACRO-ITERATION   8:
   --- Inactive Energy E0 = -83.06255267 Eh
   E(CAS)=  -113.889431430 Eh DE=     -0.000017713
   N(occ)=  1.99745 1.99621 1.98486 1.97930 1.94281 0.05939 0.02155 0.01843
   ||g|| =       0.000603382 Max(G)=    -0.000306830 Rot=9,0
                    ---- THE CAS-SCF GRADIENT HAS CONVERGED ----
                         --- FINALIZING ORBITALS ---
                    ---- DOING ONE FINAL ITERATION FOR PRINTING ----
   --- Forming Natural Orbitals
   --- Canonicalize Internal Space
   --- Canonicalize External Space
```

From which we see that we had *two orbitals too much in the active space* with occupation numbers very close to two. This caused the convergence problems even in the second order CASSCF method. You can actually monitor this during the iterations and if orbitals approach occupation 0 or 2 the calculation is unlikely to converge and might as well be stopped. You have to look for another active space or better starting orbitals then.

In the present case, we re-run the CASSCF with 6 active electrons in six orbitals. The result is:

```
    MACRO-ITERATION   2: --- DAMPED 0.618 ---
       --- Inactive Energy E0 = -101.29208454 Eh
       E(CAS)=  -113.881218708 Eh DE=     -0.009625847
       N(occ)=  1.98108 1.97898 1.94000 0.06082 0.02110 0.01802
       ||g|| =       0.070797765 Max(G)=    -0.020381394 Rot=19,7
       ===>>> Convergence to 8.0e-02 achieved - switching to Step=NR
       --- Increasing StepSize to 0.742 %
       --- Scaling the gradient (0.742%)
       --- Orbital Update [       NR]
       --- Density not converged (0.004). Exact active Fock build

    MACRO-ITERATION   3:
       --- Inactive Energy E0 = -100.73608230 Eh
       E(CAS)=  -113.885226615 Eh DE=     -0.004007907
       N(occ)=  1.98130 1.97901 1.94104 0.05956 0.02126 0.01783
       ||g|| =       0.020517383 Max(G)=    -0.007941891 Rot=19,7
       --- Increasing StepSize to 0.890 %
       --- Scaling the gradient (0.890%)
```

```
        --- Orbital Update [        NR]
        --- Density not converged (0.001). Exact active Fock build


  MACRO-ITERATION   4:
        --- Inactive Energy E0 = -100.55893099 Eh
        E(CAS)=  -113.885572046 Eh DE=    -0.000345431
        N(occ)=  1.98161 1.97924 1.94164 0.05889 0.02107 0.01754
        ||g|| =      0.004400745 Max(G)=    -0.002411060 Rot=19,7
        --- Reached Maximum Stepsize (1.0)
        --- Orbital Update [        NR]
        --- Density not converged (0.001). Exact active Fock build


  MACRO-ITERATION   5:
        --- Inactive Energy E0 = -100.52680306 Eh
        E(CAS)=  -113.885584053 Eh DE=    -0.000012007
        N(occ)=  1.98168 1.97931 1.94188 0.05865 0.02101 0.01747
        ||g|| =      0.001229592 Max(G)=    -0.000702657 Rot=19,7
        --- Orbital Update [        NR]
        --- Density not converged (0.000). Exact active Fock build


  MACRO-ITERATION   6:
        --- Inactive Energy E0 = -100.52392934 Eh
        E(CAS)=  -113.885584414 Eh DE=    -0.000000361
        N(occ)=  1.98169 1.97932 1.94195 0.05857 0.02100 0.01746
        ||g|| =      0.000375433 Max(G)=    -0.000236680 Rot=19,7
                    ---- THE CAS-SCF GRADIENT HAS CONVERGED ----
                        --- FINALIZING ORBITALS ---
                    ---- DOING ONE FINAL ITERATION FOR PRINTING ----
        --- Forming Natural Orbitals
        --- Canonicalize Internal Space
        --- Canonicalize External Space


  MACRO-ITERATION   7:
        --- Inactive Energy E0 = -100.52392934 Eh
        E(CAS)=  -113.885584415 Eh DE=    -0.000000001
        N(occ)=  1.98172 1.97933 1.94202 0.05851 0.02100 0.01743
        ||g|| =      0.000375432 Max(G)=     0.000236661 Rot=20,7
        --- Number of divergent steps = 1
  --------------
  CASSCF RESULTS
  --------------


  Final CASSCF energy       : -113.885584415 Eh   -3098.9843 eV
```

You see that after switching to the second-order procedure, the calculation converges very quickly and the occupation numbers show you that all of these orbitals are actually needed in the active space. The omission of the two orbitals from the active space came at an increase of the energy by ∼4 mEh which seems to be tolerable. Let's look what we have in the active space in figure 6.2.

Thus, we can see that we got a fairly nice result: our calculation has correlated the in-plane oxygen lone pair, the C-O σ and the C-O π bond. For each strongly occupied bonding orbital, there is an accompanying weakly occupied antibonding orbital in the active space that is characterized by one more node. In particular, the correlating lone pair and the C-O σ^* orbital would have been hard to find with any other procedure than the one chosen based on natural orbitals. We have now done it blindly and looked at the orbitals only after the CASSCF — a better approach is normally to look at the starting orbitals *before* you enter a potentially expensive CASSCF calculation. If you have

(a) MO5   (b) MO6   (c) MO7

(d) MO10   (e) MO9   (f) MO8

Figure 6.2: Orbitals of the active space for the CASSCF(6,6) calculation of H_2CO.

bonding/antibonding pairs in the active space plus perhaps the singly-occupied MOs of the system you probably have chosen a reasonable active space.

We can play the game now somewhat more seriously and optimize the geometry of the molecule using a reasonable basis set:

```
! TZV(2d,2p) TZV/C SmallPrint Opt
! moread
%moinp "Test-CASSCF-MP2-H2CO.mp2nat"

%casscf nel          6
        norb         6
        switchstep nr
        end

* int 0 1
    C  0 0 0 0.00    0.0     0.00
    O  1 0 0 1.20    0.0     0.00
    H  1 2 0 1.10  120.0     0.00
    H  1 2 3 1.10  120.0   180.00
```

```
        *
```

and get:

```
    ----------------------------------------------------------------------
                      Redundant Internal Coordinates

                       --- Optimized Parameters ---
                         (Angstroem and degrees)

         Definition                  OldVal   dE/dq    Step    FinalVal
    ----------------------------------------------------------------------
      1. B(O   1,C   0)              1.2114  0.000026 -0.0000   1.2114
      2. B(H   2,C   0)              1.0926 -0.000005  0.0000   1.0926
      3. B(H   3,C   0)              1.0926 -0.000005  0.0000   1.0926
      4. A(O   1,C   0,H   3)        121.99  0.000001  -0.00    121.99
      5. A(H   2,C   0,H   3)        116.02 -0.000002   0.00    116.02
      6. A(O   1,C   0,H   2)        121.99  0.000001  -0.00    121.99
      7. I(O   1,H   3,H   2,C   0)   -0.00 -0.000000   0.00     -0.00
    ----------------------------------------------------------------------
```

Let us compare to MP2 geometries (this job was actually run first):

```
    ! RHF RI-MP2 TZV(2d,2p) TZV/C SmallPrint TightSCF Opt


    %mp2 natorbs true
          end


    * int 0 1
        C  0 0 0 0.00   0.0    0.00
        O  1 0 0 1.20   0.0    0.00
        H  1 2 0 1.10 120.0    0.00
        H  1 2 3 1.10 120.0  180.00
    *
```

```
    ----------------------------------------------------------------------
                      Redundant Internal Coordinates

                       --- Optimized Parameters ---
                         (Angstroem and degrees)

         Definition                  OldVal   dE/dq    Step    FinalVal
    ----------------------------------------------------------------------
      1. B(O   1,C   0)              1.2127  0.000374 -0.0002   1.2125
      2. B(H   2,C   0)              1.0991 -0.000031  0.0001   1.0992
      3. B(H   3,C   0)              1.0991 -0.000031  0.0001   1.0992
      4. A(O   1,C   0,H   3)        121.77  0.000023  -0.00    121.77
      5. A(H   2,C   0,H   3)        116.45 -0.000046   0.01    116.46
      6. A(O   1,C   0,H   2)        121.77  0.000023  -0.00    121.77
      7. I(O   1,H   3,H   2,C   0)   -0.00 -0.000000   0.00     -0.00
    ----------------------------------------------------------------------
```

The results are actually extremely similar (better than 1 pm agreement). Compare to RHF:

```
            ---------------------------------------------------------------------
                          Redundant Internal Coordinates

                            --- Optimized Parameters ---
                             (Angstroem and degrees)

             Definition                  OldVal   dE/dq     Step     FinalVal
            ---------------------------------------------------------------------
             1. B(O   1,C   0)             1.1784 -0.000164  0.0001    1.1785
             2. B(H   2,C   0)             1.0921  0.000010 -0.0000    1.0921
             3. B(H   3,C   0)             1.0921  0.000010 -0.0000    1.0921
             4. A(O   1,C   0,H   3)      121.93  -0.000003   -0.00    121.93
             5. A(H   2,C   0,H   3)      116.13   0.000005    0.00    116.13
             6. A(O   1,C   0,H   2)      121.93  -0.000003   -0.00    121.93
             7. I(O   1,H   3,H   2,C   0)  0.00   0.000000   -0.00     -0.00
            ---------------------------------------------------------------------
```

Thus, one can observe that the correlation brought in by CASSCF or MP2 has an important effect on the C=O distance ($\sim$4 pm), while the rest of the geometry is not much affected.

More on the technical use of the CASSCF program.

The most elementary input information which is always required for CASSCF calculations is the specification of the number of active electrons and orbitals.

```
   %casscf nel  4  # number of active space electrons
           norb 6  # number of active orbitals
           end
```

The CASSCF program in ORCA can average of states of several multiplicities. The multiplicities are given as a list. For each multiplicity the number of roots should be specified:

```
   %casscf mult   1,3 # here: multiplicities singlet and triplet

           nroots 4,2 # four singlets, two triplets
           end
```

If the symmetry handling in ORCA is enabled (! UseSym) each multiplicity block must have an irreducible representation assigned. Numbers corresponding to the "irrep" within a given symmetry are printed in the output of ORCA.

```
   %casscf mult   1,3 # here: multiplicities singlet and triplet
           irrep  0,1 # here: irrep for each mult. block

           nroots 4,2 # four singlets, two triplets
           end
```

Several roots and multiplicities usually imply a state average CASSCF (SA-CASSCF) calculation. In this case, the user should specify the weight for each block and for each root:

```
%casscf mult 1,3   # here: multiplicities singlet and triplet
        nroots 4,2  # four singlets, two triplets
        bweight 2,1 # singlets and triplets weighted 2:1
        weights[0] = 0.5,0.2,0.2,0.2 # singlet weights
        weights[1] = 0.7,0.3         # triplet weights
        end
```

The program will automatically normalize these weights such that the sum of the weights over all roots is unity. If the weights are not given, the program simply assumes that all weights should be weighted equally. If convergence on an excited state is desired then the `weights[0]` array may look like `0.0,0.0,1.0` (this would give orbital optimization for the third excited state. If several states cross during the orbital optimization this will ultimately cause convergence problems.

Orbital optimization methods. Since orbital optimization is so difficult, there are several methods available for optimization.

```
%casscf OrbStep SuperCI     # SuperCI (first order)
                DIIS         # DIIS (first order)
                KDIIS        # KDIIS (first order)
                SOSCF        # approx. Newton-Raphson (first order)
                NR           # augmented Hessian Newton-Raphson
                             # unfolded two-step procedure
                             # - still not true second order
                SuperCI_PT  # perturbative SuperCI (first order)
        end
```

The different convergers have different strengths. The SuperCI converger is normally quite effective at the beginning of the optimization process when the process is far from convergence but is slow towards the end. In the default method the optimization process is started with SuperCI and then switched to DIIS. Even DIIS may sometimes converge slowly or "trail" towards the end such that real convergence is never reached. The DIIS stores orbital update vectors from previous macro-iterations and uses them to extrapolate the new step. The KDIIS [256] — based on perturbation theory — is an approximation to the regular DIIS procedure avoiding redundant rotations. Both DIIS schemes avoid linear dependencies in the expansion space.

In difficult cases the use of the Newton-Raphson method is highly recommended even if each individual iteration is considerably more expensive. It is strong towards the end but it would be a waste to start orbital optimization with the expensive NR method since its radius of quadratic convergence is quite small. The computationally cheaper alternative is the SOSCF procedure belonging to the family of quasi-Newton updates. The switching is controlled as follows:

```
%casscf

  SwitchStep         DIIS       # switch to DIIS
```

```
                        KDIIS       # switch to the diagonalization free DIIS.
                        NR          # switch to Newton-Raphson
                        SOSCF       # switch to SOSCF
                        SuperCI_PT # switch to SuperCI_PT


    SwitchConv 0.03 # gradient at which to switch
    SwitchIter 15   # iteration at which the switch takes place
                    # irrespective of the gradient


    MaxIter     75  # Maximum number of macro-iterations
  end
```

There are more refined options available for the augmented Hessian Newton-Raphson step. Below are list of keywords including their default settings.

```
    %casscf aughess
            Solver   0       # Davidson (default)
                     1       # Pople (pure NR steps)
                     2       # DIIS
            MaxIter 35       # max. no. of CI iters.
            MaxDim  35       # max. dim. of Davidson iterations.
            MaxDIIS  12      # max. number of DIIS vectors
            UseSubMatrixGuess true # diag a submatrix of the Hessian
                             # as an initial guess
            NGuessMat 2064   # size of initial guess matrix (part of
                             # the Hessian exactly diagonalized)
            ExactDiagSwitch 2064 # up to this dimension the Hessian
                             # is exactly diagonalized (small problems)
            PrintLevel 1     # amount of output during AH iterations
            Tol    1e-6      # convergence tolerance
            UseFloat false   # use float values in storage of inte-
                             # grals and Hessian files
            Compress true    # use compressed storage
            DiagShift 0.0    # shift of the diagonal elements of the
                             # Hessian
            UseDiagPrec true # use the diagonal in updating
            SecShift 0.0     # shift the higher roots in the Davidson
                             # secular equations
            UpdateShift 0.5  # shift of the denominator in the
                             # update of the AH coefficients
            end
```

Convergence is strongly influenced by numerical noise, especially in the final iterations. One source of numerical noise is the incremental Fock build. Thus, it can help to enforce more frequent full Fock matrix formation.

```
    ResetFreq 1 # reset frequency for direct SCF
```

If the orbital change in the active space is small, the active Fock matrix in ORCA is approximated using the density matrix from the previous cycle saving a second Fock matrix build. However, this approximation might also be a source of numerical instability. The threshold "SwitchDens" can be set to zero to enable the exact build. The program default starts with a rather large value of value (1e-2) and will reduce this parameter automatically when necessary.

```
    switchdens  0.0001 # ˜gtol * 0.1
```

In all of the implemented orbital optimization schemes the step-size correlates with the gradient-norm. A constant damping factor can be set with the keyword `GradScaling`. A better choice is to let program reduce the step-size when the calculation starts to diverge. The respective keyword is `DampBadSteps`. Once the calculation is going into a good direction, restrictions to the step-size will be discarded slowly. The amount of damping and relaxation is controlled with `DampingRate` and `RelaxationRate`. Alternatively, setting the keyword `DropBadSteps` the program will retry the diverging steps with a smaller step-size. Adding the keyword `NROnBadSteps` forces the program to do or redo a few steps using the more expensive Newton-Raphson. **If none of the previous options are selected manually, ORCA will pick the best choice for the given orbital optimization step.**

```
        GradScaling  0.5   # constant damping in all steps
        # Handling divergent steps (DE > 0.0)
        DampBadSteps true  # continue steps with smaller step-size
        DropBadSteps false # retry steps with smaller step-size
        NRonBadSteps false # retry/continue with NR converger

        # Damping and relaxation factors of the step-size control
        # in "DampBadSteps" and "DropBadSteps"
        DampingRate    0.618  # DampingRate of the step-size
        RelaxationRate 1.2    # RelaxtionRate of the step-size

        MaxDampIter  4     # maximum number of retries for the same step
                           # with DropBadSteps
```

If orbital occupations are close to 0.0 or 2.0, convergence is usually hampered. For the SuperCI, DIIS and KDIIS approaches introducing a level-shift will damp rotations with the active orbitals. Convergence is also deterioted, when the orbital energies of the active space overlap with the orbital energies of the internal or external space. If no shift is specified in the input, ORCA will choose a level-shift to guarantee an energy separation between the subspaces.

```
        ShiftUp     1.0   # up-shift the virtual orbitals
        ShiftDn     1.0   # down-shift the internal orbitals
```

There are situations when the active space has been chosen carefully, but the initial gradient is still far off. To keep the "good" active space, we can suppress all rotation but the inactive-external ones until the gradient-norm is small enough to continue safely. The threshold can be set with `FreezeIE` keyword. Once the components of the gradient in the inactive-external direction have a weight of less than `FreezeIE`, all constrains are lifted. ORCA by default freezes

active rotations if the total gradient norm is larger than 1.0 and the active rotations have a weight of less than 5%. The feature can be turned off setting the threshold to zero.

Similarly, rotations of the almost doubly occupied orbitals with the inactive orbitals can be damped using the thresh `FreezeActive`. Rotations of this type are damped as long as all their weight is smaller than `FreezeActive`. In contrast to the `ShiftDn`, it damps just the "troublesome" parts of internal-active rotations. This option applies to all of the orbital optimization schemes.

If the CASSCF starts from a converged Hartree-Fock calculation, the lowest core orbitals should not change dramatically by the CASSCF optimization. Setting a the thresh `FreezedGrad` rotations with orbitals that would have been affected by the frozencore approximation are damped. Once again, this constrain will be lifted if these orbital rotations have a weight that is larger than the given `FreezeGrad`.

```
    FreezeIE 0.4       # keep active space until int-ext rotation have
                       # a contribution of less than 40%
    FreezeActive 0.03  # keep almost doubly occupied orbitals as long as
                       # their contribution is less than 3%
    FreezeGrad   0.02  # keep the frozencore orbitals as long as
                       # their contribution is less than 2%
```

In the case of DIIS, the size of the extrapolation factor does matter. Quit often the DIIS extrapolation contains some near linear dependent vectors leading to numerical instability. Linear dependent vectors are skipped by default using the parameter `DIISThresh`.

```
        MaxDIIS    15    # max. no of DIIS vectors to keep
        DIISThresh 1e-7  # overlap criteria for linear dependency
```

A good way to speed up the calculations at the expense of "only" obtaining approximate results is to introduce the RI approximation (best done together with the "jk" auxiliary bases). The RI approximation is invoked by:

```
        TrafoStep  RI    # RI used in transformation
                         # NOTE: NEEDS and Aux-Basis
                   Exact # exact transformation (default)
```

Once the calculation has converged, ORCA will do a final macro-iteration, where the orbital are "finalized". The default final orbitals are canonical in the internal and external space with respect to state-averaged Fock operator. The active orbitals are chosen as natural orbitals. Other orbital types can be chosen for the individual subspaces.

```
    #internal space
    intorbs canonorbs # canonical
            locorbs   # localized
            unchanged # no changes


    #external space
```

```
    extorbs canonorbs # canonical
            locorbs   # localized
            unchanged # no changes


    #active space
    actorbs natorbs   # natural
            canonorbs # canonical
            locorbs   # localized
            unchanged # no changes
            dorbs     # purify metal d-orbital
            forbs     # purify metal f-orbital
```

If the active space is formed by some metal d-orbitals, the default setting might lead to orbitals that are generally mixed. Then, the d-orbitals can be purified if the `actorbs` is set to `dorbs`. The same holds for f-orbitals and the option `forbs`. For analysis purposes, a compact printing of the complete wavefunction can be requested.

```
    PrintWF false     # extended wavefunction printing
```

During the iterations, the active orbitals can be chosen to actively maximize the overlap with active orbitals from the previous iterations or to form approximate natural orbitals. Maximizing the overlap does not make any restrictions on the nature of the orbitals. Thus initially localized orbitals will stay localized and ordered, which is sometimes a desired feature e.g. in the density matrix renormalization group (DMRG) approach, where it is turned on automatically. However, natural orbitals will improve the stability of the SuperCI procedure. Thus the program default figures out whats best for the current situation. The flag can be set manually with `ActConstrains`.

```
    ActConstrains  0 # nothing at all
        1 # maintain shape and ordering of active orbitals (DMRG default)
        2 # make natural orbitals in every iteration
        3 # make canonical orbitals in every iteration
        4 # localize and reorder orbitals
```

As mentioned earlier, the CASSCF procedure requires a CI-step in every macro-iteration. There are three different options available to perform this full CI step.

```
    %casscf
    CIStep CSFCI  # CSF based CI (default)
    DMRGCI # density matrix renormalization group instead of the CI
```

The CI-step default setting is CSF based and is done in the present program by generating a partial "formula tape" which is read in each CI iteration. The tape may become quite large beyond several hundred thousand CSFs which limits the applicability of the CASSCF module. Details of CI part can be controlled in a sub-block. **Not all of the options and properties are available for CISteps apart from the default!**

```
%casscf ci
        MaxIter 35   # max. no. of CI iters.
        MaxDim  -1   # max. dim. of Davidson iterations.
                     # By default 3, which is 3*NumberRoots
        NGuessMat -1 # size of initial guess matrix. Default
                     # is 2064/NRoots
        PrintLevel 1 # amount of output during CI iterations
        ETol    -1   # default 0.1*ETol in CASSCF
        RTol    -1   # default 0.1*ETol in CASSCF
        end
```

The `CI-step` DMRGCI interfaces **BLOCK** program developed in the group of G. K.-L. Chan [287–290]. A detailed description of the **BLOCK** program, its input parameters, general information and examples on the density matrix renormalization group (DMRG) approach, are available in the section 6.11 of the manual.

The implementation of DMRG in **BLOCK** is fully spin-adapted. However, spin-densities and related properties are not available in the current version of the **BLOCK** code. To start a DMRG calculation add the keyword "CIStep DMRGCI" into a regular CASSCF input. ORCA will set default parameters and generate and input for the **BLOCK** program. In general, DMRG is not invariant to rotation in the active space. The program by **default** will run an automatic ordering procedure (`Fiedler`). More and refined options can be set in the `dmrg` sub-block of CASSCF - see section 6.11 for a complete list of keywords.

```
%casscf
 nel  8
 norb 6
 mult 3
 CIStep DMRGCI

# Detailed settings
 dmrg
 # more/refined options
 ...
 end
 end
```

It is highly recommended to start the calculation with split-localized orbitals. Any set of starting orbitals (gbw file) can be localized using the `orca_loc` program. Typing `orca_loc` in the shell will return a small help-file with details on how to setup an input for the localization. Examples for DMRG including the localization are in the corresponding section of the manual 6.11. The utility program `orca_loc` is documented in section 6.24.5. Split-localization refers to an independent localization of the internal and virtual part of the desired active orbitals.

NOTE:

- Let us stress again: it is strongly recommended to first LOOK at your orbitals and make sure that the ones that will enter the active space are really the ones that you want to be in the active space! Many problems can be solved by thinking about the desired physical contents of the reference space before starting a CASSCF. A poor choice of orbitals results in poor convergence or poor accuracy of the results! Choosing the active orbitals always requires chemical and physical insight into the molecules that you are studying!

- Please try the program with default settings before playing with the more advanced options. If you encounter convergence problems, have a look into your output, read the warning and see how the gradient and energy evolves. Increasing MaxIter will not help in many cases.

6.9.2 CASSCF Properties

The CASSCF program is able to calculate UV transition, CD spectra, SOC, Zeeman splittings, g-tensors, and MCD spectra. The techniques used to calculate SOC, and Zeeman splittings are identical to those implemented into the MRCI program. Input and keywords mimic the ones in the MRCI module. As an example, the input file to calculated g-values of CO^+ is listed below:

```
!TZVPP  TightSCF Bohrs Conv
%casscf nel  9
        norb 8
        nroots 9
        mult  2
        switchstep NR
        etol 1e-7
        gtol 1e-05
        rel
        dosoc true
        gtensor true
        printlevel 2
        end
        shiftup 2 shiftdn 2
        end
* xyz 1 2
 C 0 0  0.0
 O 0 0  2.3504
*
```

Note, that Spin-Spin couplings are momentarily not accessible via the CASSCF module. However, it is possible to run CAS-CI type calculations in MRCI. Converged CASSCF orbitals can be read setting the following flags

```
!MOREAD NOITER ALLOWRHF TZVPP  TightSCF Bohrs
%moinp "convergedCASSCF.gbw"

%mrci
 ...
 TPre 0.0
 citype mrci

 newblock 2 *
 excitations none
```

```
    refs CAS(9,8) end
    end

    soc
    DoSC true
    ...
    end
end
* xyz 1 2
  C 0 0  0.0
  O 0 0  2.3504
*
```

A common way to introduce dynamical correlation for the property computation, is to replace the energies entering the quasi-degenerate perturbation theory. If the NEVPT2 energy correction is computed in CASSCF, there will be additional printings where CASSCF energies are replaced by the more accurate NEVPT2 values. Alternatively, these diagonal energies can be taken from the input file similarly how it is described for the MRCI module. A more detailed documentation is presented in the MRCI property section.

6.9.3 Fully Variational Spin-Orbit Coupled CASSCF

The fully variational spin-orbit coupled CASSCF approach allows the user to account for spin-orbit coupling effects variationally. In this method the CASSCF wavefunction is presented as a linear combination of several parts of pre-selected multiplicities S and all possible $M_s = S, S-1, \ldots, -S$.

$$\Psi^{REL} = \Psi^{SM_S} + \Psi^{SM_S-1} + \ldots + \Psi^{S-1M_S-1} + \Psi^{S-1M_S-2} + \ldots \tag{6.161}$$

Each of the building block wavefucntions Ψ^{SM_S} is itself represented by a sum of CSFs

$$\Psi^{SM_S} = \sum_l^{NCSF} C_l^{SM_S} \Phi_l^{SM_S} \tag{6.162}$$

The presented approach optimizes the CI coefficients and MO coefficients of a sum of the Born-Oppenheimer (BO) Hamiltonian and the spin-orbit mean-field Hamiltonian (SOMF) that are given by

$$\hat{H}_{REL} = \hat{H}_{BO} + \hat{H}_{SOMF} \tag{6.163}$$

The procedure employs the intermediate coupling scheme to account for SOC. This allows to employ complex-valued CI coefficients, but real-valued molecular orbital coefficients.

As an example, the input file to calculated g-values of HgH is listed below:

```
    ! TZVP  TightSCF Conv DKH  Pmodel

%casscf
   nel  3               # number of electrons
   norb 5               # number of orbitals
   nroots 3             # number of roots for the guess calculation
   mult   2             # multiplicities
   etol 1e-9
   gtol 1e-05
   maxiter 100


   rel
      dosoc true            # accounts for the SOC interaction
      soctype          1 # runs a variational calculation
                       0 # corresponds to the QDPT case
      orbstep       superci # SuperCI algorithm for convergence
                    diis    # DIIS algorithm for convergence
                    qn      # quasi-Newton algorithm for
                            # convergence
      switchstep    diis
      gradscaling      0.1 # gradient scaling factor
      maxiter          100 # number of iteration
                         0 # corresponds to the diagonalization
                           # of the BO and SOC operators in the
                           # basis of CSFs
      printlevel 2
      nroots 6              # number of relativistic roots
      etol 1e-9
      gtol 1e-3
      socints            0 # SOC integrals are calculated once
                           # with guess MOs and used during
                           # every CASSCF iteration
                         1 # SOC integrals are re-calculated
                           # on every CASSCF calculation
                        -1 # SOC integrals are taken from files
                           # input_name.mrci.sox.tmp etc
      gtensor true          # calculate g-tensor
      usekramerssym    true # use Kramers symmetry for the CI
                            # step
      Weights = 0.166667, 0.166667, 0.166667, 0.166667, 0.166667, 0.166667
              # weights for the state-averaging
      ci
         rtol 5e-14            # residual tolerance for the Davidson
                              # procedure
         maxdim   18          # maximal dimension of space
```

```
                                # expansion in the Davidson procedure
           guesstype           0 # guess CI coefficient are obtained
                                # from diagonalization of
                                # a small CI matrix
                              1 # CI guess vectors a taken as final
                                # vectors from the previous CI
                                # iteration
        end
     end
 end


 %rel
 soctype                      2 # reads atomic densities from files
                                # 0.gbw for the atom 0, 1.gbw for the
                                # atom 1 etc.
                              4 # constructs atomic densities
                                # automatically
 picturechange true            # include picture change effects for
                                # for the SOC and the Zeeman operator

 end

 * xyz 0 2
 Hg 0 0 0
 H 0 0 1.766
 *
```

Here, the non-relativistic (or scalar relativistic, real-valued) CI vectors are calculated first. They are used as guess vectors for the following variational calculation. The relativistic procedure starts with the message

```
 STARTING RELATIVISTIC CASSCF

 INIT CI
 SOLVE CI
 ORBITAL-IMPROVEMENT-STEP:
 Algorithm                    ... DIIS

   Upward level shift           ...  1.00e+00
   Downward level shift         ...  1.00e+00
   AO integral handling         ... CONVENTIONAL
   Energy convergence tolerance ...  1.00e-09
   Orbital gradient convergence ...  1.00e-03
   Max. number of iterations    ... 35

    WEIGHTS:  0.166667  0.166667  0.166667  0.166667  0.166667 0.166667
```

First, the variational procedure diagonalizes the relativistic Hamiltonian in the basis of CSFs. This corresponds, in fact, to the optimization of the CI coefficients using the Davidson procedure and guess MOs, and denoted in the output as `ZERO MACRO-ITERATION`

```
    ZERO MACRO-ITERATION


    --------------------------------
    COMPLEX DAVIDSON-DIAGONALIZATION
    --------------------------------


    Dimension of the eigenvalue problem        ...      80
    Number of roots to be determined           ...       6
    Maximum number of CI iterations            ...      64
    Maximum size of the expansion space        ...      18
    Convergence tolerance for the residual     ... 1.00e-14
    Orthogonality tolerance                    ... 1.00e-15
    Data storage type                          ... COMPLEX DOUBLE
    Data compression type                      ... UNCOMPRESSED
```

At the end of the Davidson procedure the resulting energies and roots contributions are printed:

```
    Eh              cm-1      residual norm


    -19596.632795883       0.00000   0.000000000000 # energies and
    -19596.632795883       0.00001   0.000000000000 # residual norms
    -19596.522641797   24176.02753   0.000000000000
    -19596.522641797   24176.02754   0.000000000000
    -19596.506502189   27718.26205   0.000000000000
    -19596.506502189   27718.26205   0.000000000000
          *** CONVERGENCE OF RESIDUAL NORM REACHED ***



    -------------------------------------------# relativistic
      RELATIVISTIC CASSCF STATES               # states are
    -------------------------------------------# printed
    ROOT    0:  E= -19596.6327958833 Eh

    S = 0.5  MS = 0.5                          # contribution to
                                               # the root 0 of CSFs
                                               # with S = 1/2 and
                                               # MS = 1/2


    (    -0.88880371,        0.17857507)  0    # real and image parts
                                               # of the CI
                                               # coefficient of the
                                               # CFS "0"
    (    -0.02017328,        0.00746783)  1
    (    -0.00746783,       -0.02017328)  2
    (    -0.08345160,        0.01676678)  3
    (    -0.02929834,        0.00588651)  4
    (    -0.08258516,        0.01659270)  9
    (     0.07876099,       -0.01582436) 10
    (     0.04482426,       -0.00900592) 11
    (     0.04482426,       -0.00900592) 16
    (    -0.02273506,        0.00456784) 19
    (     0.05706131,       -0.01146454) 22
    (     0.03892008,       -0.00781967) 23
    (     0.03892008,       -0.00781967) 28
```

```
(      0.05898014,      -0.01185007)   31


S = 0.5  MS = -0.5

(      0.09374074,       0.34621370)    0
(     -0.00586884,      -0.05405190)    1
(     -0.05405190,       0.00586884)    2
(      0.00880151,       0.03250671)    3
(      0.00309005,       0.01141252)    4
(      0.00871013,       0.03216921)    9
(     -0.00830680,      -0.03067959)   10
(     -0.00472754,      -0.01746029)   11
(     -0.00472754,      -0.01746029)   16
(     -0.00601817,      -0.02222696)   22
(     -0.00410484,      -0.01516045)   23
(     -0.00410484,      -0.01516045)   28
(     -0.00622054,      -0.02297440)   31
```

Next, the properties specified in the input are calculated. Next, the obtained complex CI coefficients are used to develop the orbital gradient as it is described in detail in the literature [291]. The orbital gradient is used for optimizing real-valued MOs.

```
MACRO-ITERATION   1:

Scaling of the SOC contribution    1.000


--------------------------------
COMPLEX DAVIDSON-DIAGONALIZATION
--------------------------------


Dimension of the eigenvalue problem         ...      80
Number of roots to be determined            ...       6
Maximum number of CI iterations             ...      64
Maximum size of the expansion space         ...      18
Convergence tolerance for the residual      ... 1.00e-14
Orthogonality tolerance                     ... 1.00e-15
Data storage type                           ... COMPLEX DOUBLE
Data compression type                       ... UNCOMPRESSED
```

After the CASSCF calculation converged

```
  -19596.632736469        0.00000   0.000000000000 (     0.00)
  -19596.632736469        0.00000   0.000000000000
  -19596.522536003    24186.20658   0.000000000000
  -19596.522536003    24186.20659   0.000000000000
  -19596.506572822    27689.71986   0.000000000000
  -19596.506572822    27689.71986   0.000000000000
         *** CONVERGENCE OF RESIDUAL NORM REACHED ***

    E(CAS)=-19596.553948432 Eh DE=      0.000000510

    CI-PROBLEM SOLVED
```

```
    Making density matrices

    SOC Part:
    ||g||=        0.024945604 Max(G)=      -0.012476958 Rot=111,43
    ||g||=        0.000827087 Max(G)=      -0.000742821 Rot=111,43
                       ---- THE CAS-SCF GRADIENT HAS CONVERGED ---
```

the complex CI vectors are printed again using the same format as indicated above. The program is able to calculate UV transitions, SOC splittings, and g-tensors for doublet states. The options are similar to those used for the CASSCF used to calculate SOC splittings are identical to those implemented into the MRCI program. The output for the calculated g-tensor of HgH is printed in the same format as produced the by MRCI program.

```
    -------------------
    KRAMERS PAIR 1 :
    -------------------
    Matrix elements Re<1|S|1>    -0.441689  -0.391275   0.793281
    Matrix elements Re<1|S|2>     0.272168  -0.908540  -0.292499
    Matrix elements Im<1|S|2>    -0.846822  -0.087922  -0.507771
    Matrix elements Re<1|L|1>     0.039367   0.034874   0.005531
    Matrix elements Re<1|L|2>    -0.024258   0.080977  -0.002039
    Matrix elements Im<1|L|2>     0.075476   0.007836  -0.003540


    -------------------
    ELECTRONIC G-MATRIX
    -------------------


    raw-matrix g-matrix:
        0.496395    -1.657046    -0.589760
        1.544481     0.160356     1.023809
       -0.805576    -0.713629     1.599477


    -------------------
    ELECTRONIC gTg-MATRIX
    -------------------


        3.280781    -0.000000     0.000000
       -0.000000     3.280781     0.000000
        0.000000     0.000000     3.954328
    diagonalized g**2 matrix (sqrt of eigenvalues taken):
        1.811293     1.811293     1.988549


        0.993455     0.114223     0.000000
        0.114223    -0.993455     0.000000
       -0.000000     0.000000     1.000000


    g-Shifts in ppm (for completeness; 1e6*(g[i]-gel))
        g1 = -191026.7
        g2 = -191026.7
        g3 =  -13770.1
```

6.10 N-Electron Valence State Pertubation Theory

ORCA has implemented the *strongly contracted* NEVPT2 [292–294] and its quasi-degenerate formulation [295]. It belongs to the family of internally contracted multireference perturbation theories. NEVPT2 applies to CASSCF type MCSCF wavefunctions and thus is similar to the CASPT2 [296] and MSCASPT2 [297] approach. It has many desirable properties - among them:

- It is **intruder state free** due to the choice of the Dyall Hamiltonian [298] as the 0'th order Hamiltonian.

- It is "**strongly contracted**". Perturber functions are orthogonal and hence no time is wasted on orthogonalization issues.

- The **0'th order Hamiltonian is diagonal** in the perturber space. Therefore no linear equation system needs to be solved.

- It is **strictly size consistent**. The total energy of two non-interacting systems is equal to the sum of two isolated systems.

- It is **invariant under unitary transformations** within the active subspaces.

The two computationally most demanding steps of the NEVPT2 calculation are the initial integral transformation involving the two-external labels and the formation and contraction of the fourth order density matrix (D4). Efficient approximations to both issues are available in ORCA.

RI Approximation

Setting the RI approximation on CASSCF level, will set the RI options for NEVPT2 respectively. Alternatively setting ! `RI-NEVPT2` as simple input does the job.

```
! def2-svp def2-svp/c RI-NEVPT2
```

With the default settings for RI, the three index integrals are computed are computed and stored. Three index integral with two internal labels are kept in main memory. The two-electron integrals are assembled on the fly. Other options are available with `nev_ristep`, they differ in the way the integrals are stored or assembled. If only limited memory is available, the recommend option is `1` storing all the three integrals on disk.

```
%casscf
  ...
  nevpt2      true
  nev_ristep  -1 # no RI for the integral transformation
 0 # assembles and stores the two-electron integrals (ia|jb) using RI
 1 # stores  the three index integrals (ij|v) and (ia|v) as containers on disk
   # (ia|jb) integrals are assembled on the fly
 2 # as option 1, but (ij|v) are kept in memory (_MC_INCORE)
 3 # default: as option 2, but (ij|v) are kept as matrix array
end
```

Additional speedups can be obtained if the Fock operator formation is approximated using the ! RIJCOSX or ! RIJK techniques.

```
! def2-svp def2-svp/c RI-NEVPT2 RIJCOSX
```

Approximations for large active CASSCF space

For CASSCF spaces of [8, 8] and larger the formation of the fourth order density matrix becomes more and more the time dominating step of the NEVPT2 calculation. To improve this situation, ORCA truncates the CASSCF wavefunction during the formation of the fourth and third order density matrices. Only configurations with a weight larger than a given parameter nev_tpre are taken into account. The same approximation is available for the third order density matrix controlling nev_d3tpre. Both of the parameters can be adjusted within the %casscf block.

```
%casscf
  ...
  nevpt2    true
  nev_tpre  1e-10  #default
  nev_d3tpre 1e-14  #default
end
```

These approximations naturally can affect the "configuration RI" as well. In this context, it should be noted that a configuration corresponds to a set of configuration state functions (CSF) with identical orbital occupation. For each state the dimension of the CI and and RI space is printed.

```
D3 Build        ... CI space truncated: 141 -> 82   CFGs
                ... RI space truncated: 141 -> 141 CFGs
D4 Build        ... CI space truncated: 141 -> 82   CFGs
                ... RI space truncated: 141 -> 141 CFGs
```

The default values usually produce errors of less than 1 mEh. However, the error introduced by the nev_tpre is system dependent and should be double checked. The exact NEVPT2 energy is recovered with the parameters set to zero.

Storage of the fourth order density matrix can easily reach several gigabytes and thus cannot be kept in core memory for a large active space. We have implemented a few strategies for the computation of fourth order density matrix:

1. "fly" , compute elements and immediately contract them

2. "lfly", less memory demanding version of "fly"

3. "on disk", the full density matrix is stored on disk

4. "in core", the full density matrix is kept in core memory

```
    %casscf
      ...
      nev_d4step disk  # dumped on disk
                 fly   # on the fly (default)
                 core  # in core memory
                 lfly  # less memory demanding
                       # compared to fly, but substantially slower!
    end
```

State-averaged NEVPT2

In the definition of the Dyall Hamiltonian [298] the CASSCF orbitals are chosen to diagonalize the Fock operator (pseudo-canonicalized). Therefore in a state-averaged CASSCF the NEVPT2 procedure involves the construction and diagonalization of the "state-specific" Fock operators and is thus resulting in a unique set of orbitals for each state. This becomes quickly inefficient for large number of states or large molecular systems since each orbital set implies an integral-transformation. This is the **default** setting for NEVPT2 and is printed in the output

```
    NEVPT2-SETTINGS:
    Orbitals      ... canonical for each state
```

Other orbital options can be set using the keyword nev_canonstep.

```
    %casscf
      ...
      nev_canonstep  0 # state-averaged orbitals and specific orbital energies
                     1 # canonical for each state
                     2 # state-averaged orbitals and orbital energies
                     3 # 1-step orbital relaxation
                       # and canonical for each state (partially relaxed)
    end
```

The final orbitals of the state-averaged CASSCF diagonalize the state-averaged Fock operator. Large computational savings can be made if these orbitals are employed for all of the states. nev_canonstep 0 chooses Orbital energies as diagonal elements of the state-specific Fock operators. In previous version of ORCA, this has been the default-setting. These options work best if the averaged states are similar in nature.

nev_canonstep 2 is even more approximate and employs orbital energies from the state-averaged calculation. Thus there is no contribution to excitation energies from the perturber class V_{ij}^{ab} at this level of approximation.

If the states under consideration are substantially different, these approximations will be of poor quality and should be turned off. Better results can be achieved, if the state-averaged orbitals are partially relaxed for each state before the actual NEVPT2 calculation. [299] Often it is not possible to optimize the excited states separately. Thus nev_canonstep 3 will try a single steepest descent step for each state before running the actual NEVPT2 calculation with canonicalized orbitals. Optionally, instead of a steepest descent using an approximate diagonal Hessian, a single Newton-Raphson step can be made.

```
nev_gstep SOSCF true  # steepest descent step
          NR    false # Newton-Raphson   step
```

Despite a converged state-averaged calculation, the gradient for the individual states can be surprisingly large. As a consequence, the orbital relaxation might fail as both methods might be outside their convergence radius. ORCA will retry the relaxation with an increased damping. If the orbital update still fails, the program will stick with the initial orbitals. Setting an overall damping manually, might help the relaxation procedure.

```
nev_gscaling 0.5 # damp gradient with a pre-factor
```

Quasi-Degenerate NEVPT2
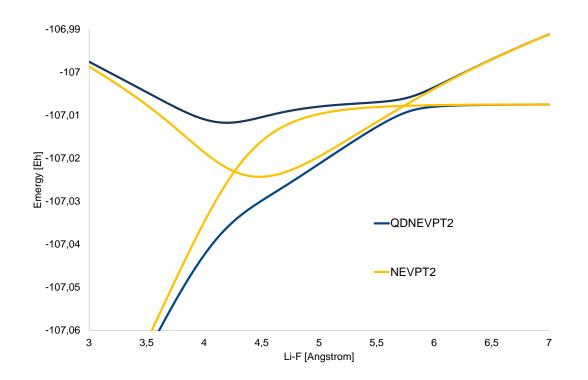
NEVPT2 as it is presented in the previous subsections follows the recipe of "diagonalize and perturb". The zero'th order wavefunction is determined by the diagonalization of the CAS-CI matrix. The space spanned by the CAS-CI vectors is often referred to as model-space. The subsequent perturbation theory is constructed based on the assumption that the states under consideration are well described within the model space. Consequently, the first order correction to the wavefunction $\Psi_I^{(1)}$ does not affect the composition of the reference state $|I>$. Corrections to the wavefunction and energy arise from the interaction of the reference state with the functions $|k>$ of the contributing first order interacting space

$$\Psi_I^{(1)} = \sum_k C_k |k> \tag{6.164}$$

$$E_I^{(2)} = \sum_k \frac{<I|H|k><k|H|I>}{E_I^{(0)} - E_k}. \tag{6.165}$$

This is problematic, when the interaction/mixing of states are falsely described at the CASSCF level. A typical example is the dissociation of Lithium-Fluoride.

```
!def2-tzvp nevpt2
%casscf
nel 2
norb 2 #Li(2s), F(2pz)
mult 1
nroots 2
end

%paras
r = 3,7,200
end
*xyz 0 1
Li 0 0 0
F 0 0 r
*
```

Here, the ground and first excited state of Σ^+ should not cross. However, at the NEVPT2 level, an erratic double crossing is observed instead.

Figure 6.3: NEVPT2 and QDNEVPT2 Li-F dissociation curves of the ground and first excited states for a CAS(2,2) reference

A re-organizing of the reference states can be introduced in the framework of quasi-degenerate perturbation theory. In practice, an eff. Hamiltonian is constructed allowing "off-diagonal" corrections to the second order energy

$$H_{IJ} = \delta_{IJ} E_I^{(0)} + \sum_k \frac{<I|H|k><k|H|J>}{E_I^{(0)} - E_k} \tag{6.166}$$

Diagonalization of this eff. Hamiltonian yields improved energies and rotation matrix (right eigenvectors) that introduces the desired re-mixing of the reference states. The quasi-degenerate extension to NEVPT2 [295] can be switched on with the keyword `nev_qdtype` .

```
%casscf
...
nevpt2 true
nev_qdtype 0 # no QDPT2
   1 # non-Hermetian eff. Hamiltonian
   2 # Hermitian eff. Hamiltonian (Cloizeaux)
end
```

ORCA will print the eff. Hamiltonian matrix and its eigenvectors at the end of the calculation.

```
      ============================================================
                         QD-NEVPT2 Results
      ============================================================
        *********************
         MULT 1,
        *********************


         Total Hamiltonian to be diagonalized
                         0            1
            0     -107.074594  -0.012574
            1        -0.011748-107.003810


         Right Eigenvectors
                         0            1
            0      -0.987232    0.170171
            1      -0.159292   -0.985414


      ---------------------------
             ROOT = 0
      ---------------------------

             Total Energy Correction : dE = -0.25309172934720
             Zero Order Energy        : E0 = -106.82353108218946
             Total Energy (E0+dE)     : E  = -107.07662281153667


      ---------------------------
             ROOT = 1
      ---------------------------

             Total Energy Correction : dE = -0.23103459727281
             Zero Order Energy        : E0 = -106.77074682157986
             Total Energy (E0+dE)     : E  = -107.00178141885267
```

By construction the Hamiltonian is non-Hermitian. If the matrix is ill-conditioned, the eigenvalues can be imaginary. ORCA will print a warning in this case. The energy denominator in the quasi-degenerate NEVPT2 is very sensitive to approximations. The canoncalization options with averaged orbitals and orbitals energies (nev_canonstep 0/2) have the tendency to lessen the energy-denominator. To avoid artifacts, the calculation is restricted to nev_canonstep 1 - each state has its own orbitals. For the computation of properties e.g. expectation values, the non-Hermitian formulation requires left- and right eigenvectors. A Hermitian variant [300] with a single set of eigenvectors ("right") can be constructed. The transformation will not change the energies but affects the mixing of states. Dispite the simplification, the recommended option should be the non-Hermitian variant!

If properties are requested within the casscf module i.e. zero-field splitting, there will be an additional printing with the "improved" CI vectors and energies. For technical reasons, properties that are not computed in CASSCF such as the Moessbauer parameters do not benefit from the QD-NEVPT2 correction.

6.11 Density Matrix Renormalization Group

The `BLOCK` *code in ORCA is only available on the Linux platform!*

`BLOCK` is an implementation of the density matrix renormalization group (DMRG) algorithm from the Chan group [287–290, 301].

The DMRG is a variational wavefunction method. It can be viewed as (i) an efficient method for strong correlation in large complete active spaces, (ii) a brute force method to systematically approach FCI for a large number of electrons and orbitals, (iii) a polynomial cost route to exact correlation in pseudo-one-dimensional molecules, such as chains and rings.

Although the algorithm is somewhat complicated compared to many quantum chemistry methods, significant effort has been devoted in `BLOCK` to ensure that it can be run in a simple black-box fashion. In most cases, only a single keyword needs to be specified.

To provide an idea of how the DMRG can be used, here are some examples. The timings will vary depending on your computational setup, but the following are calculations that run in a few hours to a day, on a single 12-core Xeon Westmere cluster node:

- Complete active space (CAS) CI calculations for active spaces with up to 30 electrons in 30 active orbitals, targetting up to 1-10 states, e.g. Jacobsen's catalyst in a 32 electron, 25 orbital active space,

- One-dimensional chain molecules, with "widths" of up to 4 orbitals, and about 100 orbitals in total, e.g. the π-active space of a 4×25 graphene nanoribbon,

- FCI benchmark solutions in molecules with fewer than 20 electrons, and up to 100 orbitals, e.g. C_2 in a cc-pVTZ basis, D_{2h} symmetry (12 electrons in 60 orbitals),

- Accuracies in energy differences or total energies of about 1 kcal/mol.

The following are calculations which are possible with the `BLOCK` code, but which are challenging, and require large memory (e.g. up to 8 Gb per core) and computational time (e.g. from a day to more than a week on up to 6 12-core Xeon Westmere nodes),

- Complete active space (CAS) CI calculations in active spaces with around 40 electrons in 40 active orbitals, targetting a few states, for example, an Fe(II)-porphine (40 electrons in 38 orbitals) with an active space of Fe 3d, 4d and all porphine π and σ donor orbitals, or an Fe 3d, S 3p active space calculation for $[Fe_4S_4(SCH_3)_4]^{2-}$,

- One-dimensional chain molecules, with "widths" of up to 6 orbitals, and about 100 total orbitals,

- Champion FCI benchmark solutions in small molecules, such as butadiene in a cc-pVDZ basis (22 electrons in 82 orbitals),

- Accuracies in energy differences or total energies of about 1 kcal/mol.

If any these calculations interest you, then you might want to try a DMRG calculation with `BLOCK`!

6.11.1 Technical capabilities

Currently, `BLOCK` implements the following

- An efficient DMRG algorithm for quantum chemistry Hamiltonians

- Full spin-adaptation (SU(2) symmetry) and Abelian point-group symmetries

- State-averaged excited states

Note that the standalone version of `BLOCK` may provide more capabilities than are available through the external interface. See the `BLOCK` website for details [302].

6.11.2 How to cite

We would appreciate if you cite the following papers in publications resulting from the use of `BLOCK` :

- G. K.-L. Chan and M. Head-Gordon, *J. Chem. Phys.* **116**, 4462 (2002),

- G. K.-L. Chan, *J. Chem. Phys.* **120**, 3172 (2004),

- D. Ghosh, J. Hachmann, T. Yanai, and G. K.-L. Chan, *J. Chem. Phys.*, **128**, 144117 (2008),

- G. K-.L. Chan and S. Sharma, *Ann. Rev. Phys. Chem.* **62**, 465 (2011),

- S. Sharma and G. K-.L. Chan, *J. Chem. Phys.* **136**, 124121 (2012).

In addition, useful DMRG references relevant to quantum chemistry can be found in the review above by Chan and Sharma.

6.11.3 Overview of BLOCK input and calculations

Within `ORCA`, the `BLOCK` program is accessed as part of the CASSCF module. `BLOCK` can be run in two modes: CASCI mode (no orbital optimization) or CASSCF mode. To enable CASCI mode, set `maxiter 1`.

```
%casscf
maxiter 1 # remove if doing CASSCF
CIStep DMRG
...
end
```

For small molecule CASCI it may be possible to correlate all orbitals. In general, similar to a standard CASSCF calculation, it is necessary to select a sensible active space to correlate. (See Section 6.11.4.1 on CASSCF). This is the responsibility of the user.

6.11.4 Standard commands

Once the orbitals to correlate have been chosen, and the wavefunction symmetries and quantum numbers are specified, the accuracy of the DMRG calculation is governed by two parameters: the maximum number of renormalized states M; and, the order and localization of the orbitals.

The most important parameter in the DMRG calculation is M, the number of renormalized states. This defines the maximum size of the wave-function expansion, which is $O(M^2)$ in length in the renormalized basis. As M is increased, the DMRG energy converges to the exact (FCI or CASCI) limit.

The DMRG maps orbitals onto a 1D lattice, thus the best results are achieved if strongly interacting orbitals are placed next to each other. For this reason, the DMRG energy is not generally invariant to orbital rotations within the active space, and orbital rotation and ordering can improve the DMRG energy for a given M. As M is increased, the DMRG energy becomes less and less sensitive to the orbital ordering and localization.

To minimize the number of optimization steps, it is often advantageous to perform DMRG calculations at small M, then increase M to the final maximum value. This sequence of optimizations is governed by the *sweep schedule*, which specifies how many optimization steps (sweeps) to perform at each intermediate value of M.

The above may seem to make running a DMRG calculation more complicated than a usual quantum chemistry calculation, however, BLOCK provides a set of default settings which eliminate the need to specify the above parameters by hand. We highly recommend that you first learn to use the BLOCK program *with these default settings*. In the default mode, the orbitals are ordered automatically (Fiedler vector method [303–306]) and a default sweep schedule is set.

An example of a default CASCI calculation on the C_2 molecule correlating all electrons in a VTZ basis, is given here:

```
   !cc-pvtz pal4
   %MaxCore 1600000
   %casscf nel 8
    norb 58
    nroots 1
    mult 1
    maxiter 1
    CIStep DMRGCI
    DMRG
     maxM 5000
   end
   end

   * xyz 0 1
   C 0 0 -0.621265
   C 0 0  0.621265
   *
```

Once you are familiar with the default mode, we recommend exploring the localization of orbitals. In general, DMRG benefits from the use of localized orbitals, and these should be used unless the high-symmetry of the molecule (e.g.,

D$_{2h}$ symmetry) provides compensating computational benefits. We recommend using "split-localized" orbitals, which correspond to localizing the occupied and virtual orbitals separately. An example of a split-localized default DMRG calculation on the porphine molecule, correlating the full π-space (26 electrons in 24 orbitals), in a cc-pVDZ basis is given in Sec. 6.11.5.

For a given `maxM`, it can take a long time to tightly converge DMRG calculations (e.g. to the default 1e-9 tolerance). To decrease computation time, you may wish to loosen the default tight sweep tolerance or control the maximum number of sweep iterations with the commands `sweep_tol` and `maxIter`.

6.11.4.1 Orbital optimization

Orbital optimization (mixing the external/internal space with the active space, not to be confused with orbital rotation and ordering in the active space) in DMRG calculation can be performed by using the `BLOCK` program as the "CIStep" within a CASSCF calculation, as described above. For the moment, spin-densities and related properties are not available for this `CIStep`.

During the optimization iterations it is important that the active orbitals maintain their overlap and ordering with previous iterations. This is done using `actConstrains`. This flag is set by default.

```
%casscf
ActConstrains  1 # maintain shape and ordering of active orbitals
...
end
```

In general, performing a DMRG calculation with orbital optimization is quite expensive. Therefore, it is often best to carry out the orbital optimization using a small value of `maxM` (enabled by the default parameters), and to carry out a final single-point calculation using a larger value of `maxM`.

6.11.4.2 Advanced options

There may be times when one wants finer control of the DMRG calculation. All keywords are shown in the *complete set of BLOCK options* 6.11.4.4 below. The `startM` command allows to change the starting number of states in DMRG calculations. It is also possible to specify the entire sweep schedule manually. A sweep schedule example follows:

```
%casscf
   ...
  dmrg

  MaxIter         14
  TwoDot_to_oneDot 12
  NSchedule       3
  sche_iteration  0,    4,    8
  sche_M         50,   100,  500
  sche_sweeptol 1e-4,  1e-6, 1e-9
```

```
      sche_noise     1e-8, 1e-11,   0.0


      end
    end
```

The commands above are:

- **MaxIter**, corresponds to the maximum number of sweeps done by DMRG;

- **NSchedule**, specifies the total number of schedule parameters we will specify;

- **Sche_iteration**, details the sweep number at which to change the parameters of the calculation. Notice count begins at 0;

- **Sche_M**, is the number of renormalized states at each sweep;

- **Sche_sweeptol**, is the tolerance of the Davidson algorithm;

- **Sche_noise**, is the amount of perturbative noise we add each sweep;

- **Twodot_to_onedot**, specifies the sweep at which the switch is made from a twodot to a onedot algorithm. The recommended choice is to start with twodot algorithm and then switch to onedot algorithm a few sweeps after the maximum M has been reached. To do a calculation entirely with the twodot or the onedot algorithm, replace the **twodot_to_onedot** line with **twodot 1** or **onedot 1**.

The default schedule corresponds to the following: starting from a given **startM** (where the default is 500 and 8 sweeps), increase to a value of 1000 (8 sweeps) and increment by 1000 every 4 iterations until **maxM** is reached. The algorithm switches from twodot to onedot two sweeps after the maximum M has been reached.

For better control of the orbital ordering, we also provide a genetic algorithm minimization method of a weighted exchange matrix. The genetic algorithm usually provides a superior orbital ordering to the default ordering, but can itself take some time to run for large numbers of orbitals. The genetic algorithm can be enabled by

```
    %casscf
     ...
     DMRG
     auto_ordering GAOPT
     end
    end
```

within the **%casscf** input.

6.11.4.3 Troubleshooting

The two most common problems with DMRG calculations are that (i) convergence with `maxM` is slower than desired, or (ii) the DMRG sweeps get stuck in a local minimum. (i) is governed by the orbital ordering / choice of orbitals. To improve convergence, turn on the genetic algorithm orbital ordering.

If you suspect (ii) is occurring, the simplest thing to do is to increase the starting number of states with the `startM` (e.g. from 500 to 1000 states). Local minima can also sometimes be avoided by increasing the noise in the DMRG schedule, e.g. by a factor of 10. To check that you are stuck in a local minimum, you can carry out a DMRG extrapolation (see extended Manual in the `BLOCK` website).

Note that the warmup (initial) sweep in DMRG currently assumes that the input orbitals supplied from ORCA are ordered by their energies. While the canonical orbitals are supplied in this order by default, this is not the case if the user has performed an orbital localization, unless the user manually rearranges the orbitals. If the input orbitals are not in their energy order, then the initial DMRG sweep may produce an energy higher than the HF energy (due to a wrong initial guess) even if the input orbitals are ultimately canonical or split-localized canonical in nature. While the DMRG calculation often recovers from this guess in later sweeps, it can occasionally stay stuck. **This will be fixed in the next release**. The current fix is to make sure the input orbitals are ordered by energy or to increase the starting number of states through the `startM` command.

6.11.4.4 Complete set of BLOCK options

```
%casscf
 ...
 dmrg


 startM 500 # number of re-normalized states for a single root
 maxM 500 # number of re-normalized states for a single root
 DryRun false  # just create an input for Block
 SweepTol 1e-9 # tolerance for the sweeps
 auto_ordering NOREORDER # auto_ordering is an int. If set to 0
                         # or the alias NOREORDER, the reordering
                         # is skipped.
 FIEDLER # (default) let Block optimize the active orbital ordering
 GAOPT  # let Block optimize the active orbital ordering using genetic algorithm



 TwoDot_to_OneDot 1 # Switch from two-dot expressions to one-dot
 OneDot 0 # Only 1-dot expressions. In CASCI mode only.
 TwoDot 0 # Only 2-dot expressions In CASCI mode only.


 # Define a schedule for DMRG
 MaxIter 14 # Specify maximum number of iterations
 NSchedule 3 # number of schedule parameters.
 # All schedule parameters must be set if
 # this flag is set manually!
```

```
    sche_iteration 0, 4, 8 # vector with sweep-number to execute changes
    # (schedule parameter)


    sche_M       50,100,500 # vector with corresponding M values (schedule parameter)
    sche_sweeptol 1e-4,1e-6,1e-9 # vector with sweep tolerances (schedule parameter)
    sche_noise   1e-8, 1e-11,0.0 # vector with the noise level (schedule parameter)


    # For developers only
    Scratch dirname  # directory name used as scratch for Block
    FCIDump dumpname # name of the integral files needed for Block
    ConfigFile "MyInput" # MyInput contains a customized Block input to be used instead.
     # If specified, anything but energy computation is turned off



    end
end
```

6.11.5 Appendix: Porphine π-active space calculation

We provide a step-by-step basis on localizing the π-orbitals of the porphine molecules and running a CASSCF-DMRG calculation on this system. It will be important to obtain an initial set of orbitals, rotate the orbitals which are going to be localized, localize them, and finally run the CASSCF calculation. We will abbreviate the coordinates as [. . .] after showing the coordinates in the first input file, but please note they always need to be included.

1. First obtain rhf orbitals:

```
    # To obtain RHF orbitals
    !cc-pvdz
    * xyz 0 1
    N      2.10524     -0.00000      0.00000
    N     -0.00114      1.95475     -0.00000
    N     -2.14882      0.00000     -0.00000
    N     -0.00114     -1.95475      0.00000
    C      2.85587     -1.13749     -0.00000
    C      2.85587      1.13749      0.00000
    C      1.02499      2.75869     -0.00000
    C     -1.10180      2.78036      0.00000
    C     -2.93934      1.13019     -0.00000
    C     -2.93934     -1.13019      0.00000
    C     -1.10180     -2.78036     -0.00000
    C      1.02499     -2.75869      0.00000
    C      4.23561     -0.67410     -0.00000
    C      4.23561      0.67410      0.00000
    C      0.69482      4.18829     -0.00000
    C     -0.63686      4.14584     -0.00000
```

```
C          -4.25427        0.70589       -0.00000
C          -4.25427       -0.70589        0.00000
C          -0.63686       -4.14584        0.00000
C           0.69482       -4.18829        0.00000
H           5.10469       -1.31153        0.00000
H           5.10469        1.31153       -0.00000
H           1.36066        5.02946        0.00000
H          -1.28917        5.00543        0.00000
H          -5.12454        1.34852        0.00000
H          -5.12454       -1.34852       -0.00000
H          -1.28917       -5.00543       -0.00000
H           1.36066       -5.02946       -0.00000
C           2.46219        2.41307        0.00000
C          -2.39783        2.44193        0.00000
C          -2.39783       -2.44193       -0.00000
C           2.46219       -2.41307       -0.00000
H           3.18114        3.22163       -0.00000
H          -3.13041        3.24594       -0.00000
H          -3.13041       -3.24594        0.00000
H           3.18114       -3.22163        0.00000
H           1.08819        0.00000       -0.00000
H          -1.13385       -0.00000        0.00000
*
```

2. We then swap orbitals with π-character so they are adjacent to each other in the active space. (π orbitals are identified by looking at the MO coefficients). When they are adjacent in the active space, they can be easily localized in the next step.

```
#To rotate the orbitals (so that we can localize them in the next step)
!cc-pvdz moread noiter
%moinp "porphine.gbw"
%scf
 rotate
70, 72, 90
65, 71, 90
61, 70, 90
59, 69, 90
56, 68, 90
88, 84, 90
92, 85, 90
93, 86, 90
96, 87, 90
99, 88, 90
102, 89, 90
103, 90, 90
```

```
    104, 91, 90
     end
    end
    * xyz 0 1
    [...]
    *
```

3. After rotating the orbitals, we localize the 13 occupied π-orbitals. This is performed using the orca_loc code. The input file follows.

```
    porphine_rot.gbw
    porphine_loc.gbw
    0
    68
    80
    120
    1e-3
    0.9
    0.9
    1
```

4. After localizing the occuppied orbitals, we localize the 11 virtual π-orbitals using the orca_loc code once again. The input file follows.

```
    porphine_loc.gbw
    porphine_loc2.gbw
    0
    81
    91
    120
    1e-3
    0.9
    0.9
    1
```

5. After these steps are complete, we run a CASSCF-DMRG calculation. The standard input file is shown below

```
    !cc-pvdz moread pal4
    %moinp "porphine_loc2.gbw"

    %casscf nel 26
    norb 24
    nroots 1
    CIStep DMRGCI
    end
    * xyz 0 1
```

```
      [...]
      *
```


6.12 Relativistic Options

The relativistic methods in ORCA are implemented in a fairly straightforward way but do require some caution from the user. The options are controlled through a block called REL which features the following variables:


```
%rel
      #-------------------------------------------------------
      # Basic scalar relativistic method
      #-------------------------------------------------------
      method   DKH        # Douglas-Kroll-Hess
               ZORA       # ZORA (numerical integration)
               IORA       # IORA (numerical integration)
               IORAmm     # IORA with van Wuellens
                          #  modified metric
               ZORA_RI    # ZORA (RI approximation)
               IORA_RI    # IORA (RI approximation)
               IORAmm_RI  # IORA (RI approximation)
                          # and modified metric
               NESC       # NESC (not functional)
      # -------------------------------------------------------
      # Choice of the model potential for ALL methods
      # -------------------------------------------------------
      ModelPot  VeN, VC, VXa, VLDA
               # Flags for terms in the model potential
               # =0 not included =1 included
               # default is 1,1,1,1 for ZORA and IORA and
               # 1,0,0,0 for DKH
               # VeN = nuclear attraction term
               # VC  = model Coulomb potential
               # VXa = model Xalpha potential
               # VLDA= VWN-5 local correlation model pot.
      Xalpha 0.7 # default value for the X-Alpha potential
      # -------------------------------------------------------
      # This variable determines the type of fitted atomic
      # density that enters the model potential
      # -------------------------------------------------------
      ModelDens      rhoDKH  # DKH4 model densities (default)
                     rhoZORA # ZORA model densities
                     rhoHF   # Hartree-Fock model densities
```

```
# -------------------------------------------------
# This flag controls whether only one center terms
# retained. If this is true an approximate treat-
# ment of relativistic effects result but geom-
# try optimizations CAN BE PERFORMED WITH ALL
# METHODS AND MODEL POTENTIAL
# In addition one gets NO gauge noninvariance
# errors in ZORA or IORA
# -------------------------------------------------
OneCenter false     # default value
# -------------------------------------------------
# Specify the speed of light used in relativistic
# calculations
# -------------------------------------------------
C       137.0359895 # speed of light used
                    # synonyms are VELIT, VELOCITY
# -------------------------------------------------
# Picture change for properties
# -------------------------------------------------
PictureChange true # default is false
# -------------------------------------------------
# Order of DKH treatment
# -------------------------------------------------
order 1 # first-order DKH Hamiltonian
      2 # second-order DKH Hamiltonian
# -------------------------------------------------
# SPIN ORBIT COUPLING OPERATORS
# -------------------------------------------------
SOCType  0  # none
         1  # effective nuclear charge
         3  # mean-field/effective potential (default)
# -------------------------------------------------
# Flags for construction of potential.
# Operative only for SOCType 3.
# -------------------------------------------------
SOCFlags 1,2,3,0 # (default)
# Flag 1 = 0 - do not include 1-electron terms
#        = 1 - do include 1-electron terms
# Flag 2 = 0 - do not include Coulomb terms
#        = 1 - compute Coulomb terms fully numeric
#        = 2 - compute Coulomb term seminumeric
#        = 3 - compute Coulomb term with RI approx
#        = 4 - compute Coulomb term exactly
# Flag 3 = 0 - do not include exchange terms
#        = 1 - do include local X-alpha exchange
```

```
#                the X-Alpha parameter can be chosen via
#                % rel Xalpha 0.7 (default)
#        = 2 - same as 1 but with sign reversed
#        = 3 - exchange via one-center exact
#                integrals including the spin-other
#                orbit interaction
#        = 4 - all exchange terms full analytic
#                (this is expensive)
# Flag 4 = 0 - do not include DFT local correlation
#                terms
#        = 1 - do include local DFT correlation (here
#                this is done with VWN5)
#
# Preferred choices:
#  SOCFlags 1,3,3,1 = "complete mean-field" with
#     efficient Coulomb terms via RI. This is the
#     most accurate one available
#  SOCFlags 1,2,3,1 = "complete mean-field" but
#     to be preferred if no fitting basis is
#     available for RI
#  SOCFlags 1,3,1,1 = "DFT-Veff"
#  SOCFlags 1,3,2,1 = "DFT-Veff" with an estimate of
#     spin-other orbit interaction; almost as
#     accurate as the first one
#
SOCMaxCenter 4 # max. number of centers to include in
               # the integrals (not fully consistently
               # implemented yet; better leave equal to 4)
#  Simple input equivalents:
#  SOMF(1X)      = SOCType 3, SOCFlags 1,2,3,0 and SOCMaxCenter 4
#  RI-SOMF(1X)   = SOCType 3, SOCFlags 1,3,3,0 and SOCMaxCenter 4
#  VEFF-SOC      = SOCType 3, SOCFlags 1,3,1,1 and SOCMaxCenter 4
#  VEFF(-2X)-SOC = SOCType 3, SOCFlags 1,3,2,1 and SOCMaxCenter 4
#  AMFI          = SOCType 3, SOCFlags 1,4,3,0 and SOCMaxCenter 1
#  AMFI-A        = SOCType 4, SOCFlags 1,4,3,0 and SOCMaxCenter 1
#                  (AMFI-like approach with generated atomic densities)
# -----------------------------------------------
# For the effective nuclear charge SOC operator
# the nuclear charges can be adjusted.
# -----------------------------------------------
Zeff[26] 0.0   # set the effective nuclear charge
               # of iron (Z = 26) to zero
SOCOff 0,5     # turn off the SOC for atoms 0 and 5
               # this makes sense if the SOC operator
               # has only one center contributions
```

```
                          # (e.g. effective nuclear charge)
        end
```

6.12.1 Approximate Relativistic Hamiltonians

In the relativistic domain, calculations are based on the one-electron, stationary Dirac equation in atomic units (rest mass subtracted)

$$h_D \Psi = \left((\beta - 1) c^2 + c \vec{\alpha} \vec{p} + V \right) \Psi = E \Psi. \tag{6.167}$$

The spinor Ψ can be decomposed in its so-called large and small components

$$\Psi = \begin{pmatrix} \Psi_L \\ \Psi_S \end{pmatrix} \tag{6.168}$$

These are obviously coupled through the Dirac equation. More precisely, upon solving for Ψ_S, the following relation is obtained:

$$\Psi_S = \frac{1}{2c} \left(1 + \frac{E - V}{2c^2} \right)^{-1} \vec{\sigma} \vec{p} \Psi_L = R \Psi_L \tag{6.169}$$

Through the unitary transformation

$$U = \begin{pmatrix} \Omega_+ & -R^+ \Omega \\ R\Omega_+ & \Omega_- \end{pmatrix} \text{ with } \Omega_+ = \frac{1}{\sqrt{1+R^+R}}, \Omega_- = \frac{1}{\sqrt{1+RR^+}},$$

the Hamiltonian can be brought into block-diagonal form

$$U^+ h_D U = \begin{pmatrix} \tilde{h}_{++} & 0 \\ 0 & \tilde{h}_{--} \end{pmatrix} \tag{6.170}$$

The (electronic) large component thus has to satisfy the following relation

$$h_{++} \Psi_L = \Omega_+ \left(h_{++} + h_{\pm} R + R^+ \left(h_{\mp} + h_{--} R \right) \right) \Omega_+ \Psi_L = E_+ \Psi_L. \tag{6.171}$$

The approximate relativistic schemes implemented in ORCA use different methods to substitute the exact relation 6.152 with approximate ones.

Two approximation schemes are available in ORCA: the regular approximation and the Douglas-Kroll-Hess (DKH) approach.

6.12.2 The Regular Approximation

In the regular approximation, 6.152 is approximated by

$$R = \frac{c}{2c^2 - V}\vec{\sigma}\vec{p}. \tag{6.172}$$

At the zeroth-order level (ZORA), $\Omega_\pm = 1$, so that the ZORA transformation is simply

$$U_{\text{ZORA}} = \begin{pmatrix} 1 & -R^+ \\ R & 1 \end{pmatrix} \tag{6.173}$$

and the corresponding Hamiltonian given by

$$\tilde{h}_{++}^{\text{ZORA}} = V + c\vec{\sigma}\vec{p}\frac{1}{2c^2 - V}c\vec{\sigma}\vec{p}. \tag{6.174}$$

At the infinite-order level (IORA), $\Omega_\pm$ is taken into account, so that

$$U_{\text{IORA}} = U_{\text{ZORA}}\begin{pmatrix} \Omega_+ & 0 \\ 0 & \Omega_- \end{pmatrix} \tag{6.175}$$

and

$$\tilde{h}_{++}^{\text{IORA}} = \Omega_+\left(V + c\vec{\sigma}\vec{p}\frac{1}{2c^2 - V}c\vec{\sigma}\vec{p}\right)\Omega_+ \tag{6.176}$$

is the corresponding Hamiltonian. Note that despite the name – *infinite-order* regular approximation – this is still *not* exact.

In ORCA, the spin-free variant of ZORA and IORA are implemented. These are obtained from those above through the replacement

$$\vec{\sigma}\vec{p}\frac{1}{2c^2 - V}\vec{\sigma}\vec{p} \rightarrow \vec{p}\frac{1}{2c^2 - V}\vec{p}. \tag{6.177}$$

The regular Hamiltonians contain only part of the Darwin term and no mass-velocity term. A problem with relation 6.157 is that due to the non-linear dependence of the resulting regular Hamiltonians on V, a change in V that in the Dirac and Schrödinger equations will result in a change of energy

$$E \rightarrow E + const$$

does not so in the regular approximation. Several attempts have been made to circumvent this problem. The scaled ZORA variant is one such procedure. Another one is given through the introduction of model potentials replacing V. Both approaches are available in ORCA.

The scaled ZORA variant

This variant goes back to van Lenthe et al. [307]. The central observation is that the Hamiltonian

$$h_{\text{scaledZORA}} = \frac{h_{\text{ZORA}}}{1 + \left\langle \Psi_L \left| c\vec{\sigma}\vec{p} \frac{1}{(2c^2 - V)^2} c\vec{\sigma}\vec{p} \right| \Psi_L \right\rangle} \tag{6.178}$$

produces constant energy-shifts $E \rightarrow E + const$ when the potential V is changed by a constant – *for hydrogenic ions*. For many-electron systems, the scaled-ZORA Hamiltonian still does not yield simple, constant energy shift for $V \rightarrow V + const$. But it produces the exact Dirac energy for hydrogen-like atoms and performs better than the first-order regular approximation for atomic ionization energies.

The regular approximation with model potential

The scalar relativistic ZORA self-consistent field equation is in our implementation (in atomic units):

$$\left[\mathbf{p} \frac{c^2}{2c^2 - \tilde{V}_{\text{model}}} \mathbf{p} + V_{\text{eff}} \right] \psi_i = \varepsilon_i \psi_i \tag{6.179}$$

where c is the speed of light. It looks like the normal nonrelativistic Kohn–Sham equation with the KS potential V_{eff}:

$$V_{\text{eff}} (\mathbf{r}) = -\sum_A \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} + \int \frac{\rho (\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + V_{xc} [\rho] \tag{6.180}$$

(Z_A is the charge of nucleus A and R_A is its position; $\rho(r)$ is the total electron density and $V_{xc} [\rho]$ the exchange-correlation potential – the functional derivative of the exchange-correlation energy with respect to the density). The kinetic energy operator $T = -\frac{1}{2}\nabla^2$ of the nonrelativistic treatment is simply replaced by the ZORA kinetic energy operator:

$$T^{\text{ZORA}} = \mathbf{p} \frac{c^2}{2c^2 - \tilde{V}_{\text{model}}} \mathbf{p} \tag{6.181}$$

Clearly, in the regions where the potential is small compared to c^2, this operator reduces to the nonrelativistic kinetic energy. $\tilde{V}_{\text{model}}$ could be the actual KS potential. However, this would require to solve the ZORA equations in a special way which demands recalculation of the kinetic energy in every SCF cycle. This becomes expensive and is also undesirable since the ZORA method is not gauge invariant and one obtains fairly large errors from such a procedure unless special precaution is taken. Van Wüllen [82] has therefore argued that it is a reasonable approximation to replace the actual KS potential with a model potential $\tilde{V}_{\text{model}}$ which is constructed as follows:

$$\tilde{V}_{\text{model}} = -\sum_A \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} + \int \frac{\rho_{\text{model}} (\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + V_{xc}^{LDA} \left[\rho^{\text{model}} \right] \tag{6.182}$$

The model density is constructed as a sum over spherically symmetric (neutral) atomic densities:

$$\rho_{\text{model}} (\mathbf{r}) = \sum_A \rho^A (\mathbf{r}) \tag{6.183}$$

Thus, this density neither has the correct number of electrons nor any spin polarization. Yet, in the regions close to the nucleus, where the relativistic effects matter, it is a reasonable approximation. The atomic density is expanded in a sum of s-type gaussian functions like:

$$\rho^A(\mathbf{r}) = \sum_i d_i \exp\left(-\alpha_i |\mathbf{r} - \mathbf{R}_A|^2\right) \tag{6.184}$$

The fit coefficients were determined in three different ways by near basis set limit scalar relativistic atomic HF calculations and are stored as a library in the program. Through the variable `ModelDens` (*vide supra*) the user can choose between these fits and study the dependence of the results in this choice (it should be fairly small except, perhaps, with the heavier elements and the HF densities which are not recommended). The individual components of the model potential can be turned on or off through the use of the variable `ModelPot` (*vide supra*).

Van Wüllen has also shown that the calculation of analytical gradients with this approximation becomes close to trivial and therefore scalar relativistic all electron geometry optimizations become easily feasible within the ZORA approach. However, since T^{ZORA} is constructed by numerical integration it is very important that the user takes appropriate precaution in the use of a suitable integration grid and also the use of appropriate basis sets! In the case of `OneCenter true` the numerical integration is done accurately along the radial coordinate and analytically along the angular variables such that too large grids are not necessary unless your basis set is highly decontracted and contains very steep functions.

6.12.3 The Douglas-Kroll-Hess Method

The Douglas-Kroll-Hess (DKH) method expands the exact relation 6.157 in the external potential V. In ORCA the first- and second-order DKH methods are implemtented. The first-order DKH Hamiltonian is given by

$$\tilde{h}^{(1)}_{++} = E_p + A_p V A_p + B_p V^{(p)} B_p, \tag{6.185}$$

with

$$E_P = \sqrt{c^4 + c^2 p^2}, \ A_p = \sqrt{\frac{E_p + c^2}{2E_p}}, \ B_p = \frac{c}{\sqrt{2E_p(E_p + c^2)}} \tag{6.186}$$

At second order, it reads

$$\tilde{h}^{(2)}_{++} = \tilde{h}^{(1)}_{++} + \frac{1}{2}[W_p, O] \tag{6.187}$$

where

$$\{W_p, E_p\} = \beta O, \ O = A_p[R_p, V]A_p, \ R_p = \frac{c\sigma p}{E_p + c^2} \tag{6.188}$$

define the second-order contribution. In ORCA, the spin-free part of $\tilde{h}^{(2)}_{++}$ is implemented.

The occurrence of the relativistic kinetic energy, E_P, which is not well-defined in position space, makes a transformation to the p^2-eigenspace necessary. Thus any DKH calculation will start with a decontraction of the basis set, to insure a good resolution of the identity. Then the non-relativistic kinetic energy is diagonalized and the E_P-dependent operators calculated in that space. The potential V and $V^{(p)}$ are transformed to p^2-eigenspace. After all contributions

are multiplied to yield the (first- or second-order) Hamiltonian, the transformation back to AO space is carried out and the basis is recontracted.

The (spin-free) DKH-Hamiltonians contain all spin-free, relativistic correction terms, eg the mass-velocity and Darwin terms. As the potential enters linearly, no scaling or model potential is necessary to introduce the correct behaviour of the energy under a change

$$V \to V + const.$$

In all these respects the DKH Hamiltonians are much cleaner than the regular Hamiltonians.

6.12.4 Picture-Change Effects

Irrespective of which Hamiltonian has been used in the determination of the wave function, the calculation of properties requires some special care. This can be understood in two ways: First of all, we changed from the ordinary Schrödinger Hamiltonian $\tilde{h}^{(2)}_{++}$ to the more complicated Hamiltonian. As properties are defined as derivatives of the energy, it is clear that a new Hamiltonian will yield a new expression for the energy and thus a new and different expression for the property in question. Another way of seeing this is that through the transformation U, we changed not only the Hamiltonian but also the wave function. To obtain the property at hand as the expectation value of the property operator with the wave function, we have to make sure that property operator and wave function are actually given in the same space. This is done through a transformation of either the property operator or the wave function.

In any case, the difference between the non-relativistic and (quasi) relativistic property operator evaluated between the (quasi) relativistic wave function is called the picture-change effect. From what was said above, this is clearly not a physical effect. It describes how consistent the quasi relativistic calculation is carried out. A fully consistent calculation requires the determination of the wave function on the (quasi) relativistic level as well as the use of the (quasi) relativistic property operator. This is obtained through the choice

```
%rel picturechange true end
```

in the %rel block. It may be that the (quasi) relativistic and non-relativistic property operator do produce similar results. In this case, a calculation with picture changes turned off (PictureChange false) may be a good approximation. This is, however, not the rule and cannot be predicted before carrying out the calculation. It is therefore highly recommended to set PictureChange true in all (quasi) relativistic property calculations! Consistent picture-change effects on the DKH2 level have been implemented for the g-tensor and the hyperfine coupling tensor. Using

```
%rel picturechange 1 end
```

only first-order changes on the property operators are taken into account. This reduces the computational cost, of course. But as this is in no way a significant reduction, this choice is not recommended.

For magnetic properties, the DKH transformation and consequently the DKH Hamiltonian and the corresponding property operators are not unique. Depending on whether the magnetic field is included in the free-particle Foldy–Wouthuysen (fpFW) transformation carried out in the first step of the DKH protocol or not, two different Hamiltonians result. If the magnetic field is included in the fpFW transformation, the resulting Hamiltonian is a function of the gauge invariant momentum

$$\boldsymbol{\pi} = \mathbf{p} + \mathbf{A}. \tag{6.189}$$

It is therefore gauge invariant under gauge transformations of the magnetic potential and thus are the property operators derived from it. This is referred to as fπFW DKH Hamiltonian. If the magnetic field is not included in the fpFW transformation, the resulting Hamiltonian is a function of the kinetic momentum $\mathbf{p}$ only and thus is not gauge invariant. The latter Hamiltonian is referred to as fpFW DKH Hamiltonian. A comparison of both Hamiltonians is given in Table 6.8.

Table 6.8: Comparison of the properties of the fpFW and fπFW DKH Hamiltonians. For details see Ref. [308].

Criterion	fpFW Hamiltonian	fπFW Hamiltonian
Convergence of Eigenvalues to Dirac Eigenvalues	?	yes
1st order is bounded	no	yes
Reproduces Pauli Hamiltonian	no	yes
Gauge invariance	no	yes
Lorentz invariance	no	no

From this Table, it becomes clear that the fπFW DKH Hamiltonian is clearly preferred over the fpFW Hamiltonian. To obtain the property operators, it is however necessary to take the derivatives of these Hamiltonians. It turns out that in the case of the hyperfine-coupling tensor, the necessary derivatives produce divergent property operators in the case of the fπFW DKH Hamiltonian. This may be due to the unphysical assumption of a point-dipole as a source of the magnetic field of the nucleus. As a physical description of the magnetization distribution of the nucleus is not available due to a lack of experimental data, the magnetization distribution is assumed to be the same as the charge distribution of the nucleus, see Section 6.12.5. This is unphysical as the magnetization is caused by the one unpaired nucleon in the nucleus whereas the charge distribution is generated by the protons in the nucleus. So, physically, the magnetization should occupy a larger volume in space than the charge. This might also be the reason why the resulting finite-nucleus model is insufficient to remedy the divergencies in the fπFW hyperfine-coupling tensor. Consequently, for the hyperfine-coupling tensor only the one resulting from the fpFW DKH Hamiltonian is implemented. In the case of the g-tensor both versions are implemented and accessible via the keyword

```
%rel fpFWtrafo true/false end
```

By default, this keyword is set to `true`. A detailed form of the property operators used for the g-tensor and hyperfine-tensors can be found in Ref. [308].

6.12.5 Finite Nucleus Model

A finite-nucleus model is available for all calculations in the ORCA program package. It is accessible from the `%rel` block via

```
%rel FiniteNuc true/false end
```

By default, this keyword is set to `false`. If the keyword is set to `true`, finite-nucleus effects are considered in the following integrals:

- nucleus potential V

- DKH-integral $V^{(p)}$

- one-electron spin-orbit integrals SOC (also in one-electron part of SOMF)

- electric-field gradient EFG (and thus, as a consequence in the Fermi-contact and spin-dipole terms of the hfc tensor)

- nucleus-orbit integral NUC

- angular-momentum integral l

The finite-nucleus model implemented in ORCA is the Gaussian nucleus model of Ref. [309].

6.12.6 Basis Sets in Relativistic Calculations

For relativistic calculations, special basis sets have been designed, both as DKH and ZORA recontractions of the non-relativistic def2 basis sets for elements up to Xe, and as purpose-built segmented all-electron relativistically contracted (SARC) basis sets for elements beyond Xe [5–9]. For many of the heavy elements, these are the only all-electron basis sets available. They are controlled through the `%basis` block via

```
  ! TZVPP DKH                # this specifies the polarization
                               and diffuse functions added to
                               the basis specified below
                               (TZVPP) and the relativistic
                               method employed (DKH or ZORA)


  %basis
        #----------------------------------------------------
        # Specification of basis
        #----------------------------------------------------
      basis   DKH_TZV  # triple-zeta basis fit for DKH
                            calculations
              DKH_QZV  # quadruple-zeta basis fit for
                            DKH calculations
              ZORA_TZV # triple-zeta basis fit
                            for ZORA/IORA calculations
              ZORA_QZV # quadruple-zeta basis fit for
                            ZORA/IORA calculations
      end
```

NOTE:

It is crucial to include both, the full basis in the one-keyword line and the relativistic basis in the basis block.

This may be a bit counterintuitive but is due to the following reason: Basis sets for heavy elements are simply stripped down to their polarization and diffuse functions. This allows them to be combined either with ECPs replacing the

core electrons or the DKH/ZORA basis sets described above explicitly including the core electrons – and doing so suitably for a relativistic caluculation.

So if only the basis is specified in the one-keyword line, e.g. TZVPP, only polarization functions are used in the basis set. This will, of course, give very wrong results as the basis set will then be reduced to four or five functions at best. If on the other hand, only the DKH/ZORA basis is specified in the basis block, no polarization and diffuse functions are added and again, the calculation is error-prone. It is always advisable to carefully check your output (perhaps also with the `PrintBasis` keyword) in order to verify that the desired basis sets have indeed been assigned, especially in cases where different basis sets are specified for different atoms.

NOTES:

- It is important to recognize that in the one-center approximation (`OneCenter true`) ALL methods can be used for geometry optimization. Several papers in the literature show that this approximation is fairly accurate for the calculation of structural parameters and vibrational frequencies. Since this approximation is associated with negligible computational effort relative to the nonrelativistic calculation it is a recommended procedure.

- The same model potential as in ZORA can also be used in the DKH method. There is no experience with this and one could also probably argue about the theoretical status of such an approximation. It is nevertheless implemented and may prove to be useful.

- The ZORA/RI, IORA/RI and IORAmm/RI methods are also done with the model potential. Here we do the integrals analytically except for the XC terms which has clear advantages. However, the RI approximation is performed in the actual orbital basis sets which means that this set has to be large and flexible. Otherwise significant errors may arise. If the basis sets are large (ZORA/RI) and the numerical integration is accurate (ZORA), the ZORA and ZORA/RI (or IORA and IORA/RI) methods must give identical within to microhartree accuracy.

6.13 Geometry Optimization

ORCA is able to calculate equilibrium structures (minima and transition states) using the quasi Newton update procedure with the well known BFGS update [83–88], the Powell or the Bofill update. The optimization can be carried out in either redundant internal (recommended in most cases) or Cartesian displacement coordinates. As initial Hessian the user can choose between a diagonal initial Hessian, several model Hessians (Lindh, Almloef, Schlegel), an exact hessian and a partially exact Hessian (both recommended for transition state optimization) for both coordinate types. In redundant internal coordinates several options for the type of step to be taken exist. The user can define constraints via two different paths. He can either define them directly (as bond length, angle, dihedral or Cartesian constraints) or he can define several fragments and constrain the fragments internally and with respect to other fragments. The ORCA optimizer can be used as an external optimizer, i.e. without the energy and gradient calculations done by ORCA.

6.13.1 Input Options and General Considerations

The use of the geometry optimization module is relatively straightforward.[17]

---

[17]But that doesn't mean that geometry optimization itself is straightforward! Sometimes, even when it is not expected the convergence can be pretty bad and it may take a better starting structure to come to a stationary point. In particular floppy structures with many possible rotations around single bonds and soft dihedral angle modes are tricky. It may sometimes be advantageous to compute a Hessian matrix at a "cheap" level of theory and then do the optimization in Cartesian coordinates starting from the calculated Hessian.

```
%method RunTyp Opt # use geometry optimization.
                   #(equivalent is RunTyp=Geom)
          end


# or simply "! Opt" in the keyword line


# details of the optimization are controlled here
%geom MaxIter 50  # max. number of geometry iterations
                  #  (default is 3N (N = number of atoms), and at
                  #  least 50 iterations)
      # coordinate type control
      coordsys redundant     # New redundant internal coords
               redundant_old # old set of redundant internal
                             #  coords (molecules might explode)
               deloc         # Delocalized internals
               cartesian     # Cartesian coordinates
      # transition state (TS) optimization
      TS_search EF      # Switch on TS search, EF means
                        #  "eigenvector following"
                        #  alternatively use "! OptTS"
      TS_Mode {M 0} end # Choose the mode to follow uphill in the
                        #  TS optimization. {M X}: eigenvector of
                        #  the Hessian with X. lowest eigenvalue
                        #  (start counting at zero) (default: X=0)
      # Instead of a mode choose an internal coordinate strongly
      #  involved in the eigenmode followed uphill
      TS_Mode {B 0 1} end      # bond between atoms 0 and 1  or
      TS_Mode {A 2 1 0} end    # angle between atoms 2, 1 and 0  or
      TS_Mode {D 3 2 1 0} end  # dihedral of atoms 3, 2, 1 and 0
      # add or remove internal coordinates from the automatically
      #  generated set of redundant internal coords
      modify_internal
          { B 10 0 A }     # add a bond between atoms 0 and 10
          { A 8 9 10 R }   # remove the angle defined
                           #  by atoms 8, 9 and 10
          { D 7 8 9 10 R } # remove the dihedral angle defined
          end              #  by atoms 7, 8, 9 and 10
      # constrain internal coordinates:
      Constraints
          { B N1 N2 value C }       # the bond between N1 and N2
          { A N1 N2 N1 value C }    # the angle defined by N1, N2
                                    #  and N3
          { D N1 N2 N3 N4 value C } # the dihedral defined by N1,
                                    #  N2, N3 and N4
          { C N1 C }                # the cartesian position of N1
```

```
        { B N1 * C}          # all bonds involving N1
        { B * * C}           # all bonds
        { A * N2 * C }       # all angles with N2 as central atom
        { A * * * C }        # all angles
        { D * N2 N3 * C } # all dihedrals with N2 and N3 as
                             #  central atoms
        { D * * * * C }      # all dihedrals
  end
  # scan an internal coordinate:
  Scan B N1 N2 = value1, value2, N end
    # perform constrained optimizations with varying N1-N2-
    #  distance from value1 up to value2 in N steps;
    #  works as well for angles (use A N1 N2 N3) and for
    #  dihedrals (use D N1 N2 N3 N4)
  Scan B N1 N2 [value1 value2 value3 ... valueN] end
    # perform constrained optimizations with N1-N2-distances
    #  as given in the list;
    #  works as well for angles (use A N1 N2 N3) and for
    #  dihedrals (use D N1 N2 N3 N4)
  fullScan true   # if !ScanTS is requested, fullScan assures
                    #  that the relaxed surface scan is fully
                    #  carried out before the TS optimization is
                    #  started (Default is false)
  # fragment optimization:
  #  1. all atoms have to belong to a fragment
  #  2. you have to connect the fragments
  ConnectFragments
      {1 2 C}        # constrain the internal coordinates
                     #  connecting fragments 1 and 2
      {1 2 C N1 N2}# constrain the internal coordinates
                     #  connecting fragments 1 and 2, the
                     #  fragments are connected via atoms
                     #  N1 and N2
      {1 3 O}        # optimize the internal coordinates
                     #  connecting fragments 1 and 3
      {1 3 O N1 N2}# optimize the internal coordinates
                     #  connecting fragments 1 and 3, the
                     #  fragments are connected via atoms
                     #  N1 and N2
  end
  #  3. you can constrain the fragment internally
  ConstrainFragments  # constrain all internal coordinates
      { 1 }             #  containing only atoms of fragment 1
  end
```

```
# optimize hydrogens
optimizeHydrogens true
  # in the context of a normal optimization all internal
  #  coordinates not involving any hydrogens are constrained
  # in the context of a fragment optimization all internal
  #  coordinates involving hydrogens are optimized (also in a
  #  constrained fragment)
# freeze the hydrogen positions with respect to the
#  heteroatoms
freezeHydrogens true
# invert the defined constraints, i.e. optimize the
#  constraints and constrain the remaining coordinates
# this only works for the redundant internal coordinates
# Cartesian coordinates are not affected by invertConstraints
invertConstraints true      # step type control
Step     qn        # quasi-Newton step
         rfo       # Rational function step (Default for !Opt)
         gdiis     # gdiis step
         prfo      # partitioned RFO step (Default for !OptTS)
UseGDIIS false  # use GDIIS step (in Cartesian optimization)
                # Default is false.
GDIISStart 1.0  # Gradient at which to start GDIIS algorithm
                #  (in Cartesian optimization)
GDIISMaxE  10   # number of last steps to use in GDIIS
                #  algorithm
# Step size control
MaxStep 0.3 # maximum step length in internal coordi-
             #  nates. Default is 0.3 au
Trust -0.3 # Initial trust radius. Default is -0.3 au
            # Trust <0 - use fixed trust radius
            #  of size -trust. I.e. -0.3 means fix
            #  the trust radius at 0.3
            # Trust >0 - use trust radius update. I.e. 0.3
            #  means start with trust radius 0.3 and update
            #  the trust radius after each optimization step
# Convergence tolerances. Note that the calculation is
# only converged if all criteria are fullfilled. All
# values given are default values.
TolE 5e-6       # Energy change (a.u.)
TolRMSG  1e-4   # RMS gradient (a.u.)
TolMaxG  3e-3   # Max. element of gradient (a.u.)
TolRMSD  2e-3   # RMS displacement (a.u.)
TolMaxD  4e-3   # Max. displacement (a.u.)
# keyword for frequently used sets of convergence thresholds
Convergence normal     # Default
```

```
                        loose
                        tight
             ProjectTR false # project translation and rotation
                              # default is false. MUST be false for
                              # redundant internals
         end
```

Keywords for the control of the Hessian (especially important for the TS optimization):

```
    # initial Hessian control
         inhess   unit        # unit matrix
                  Read        # Hessian in a .hess file (e.g. from
                              #  a previous NumFreq run), this command
                              #  comes with the following:
                  InHessName "filename.hess" # filename of
                                              #  Hessian input file
                  # these only for redundant's
                  Lindh       # Lindh's model Hessian
                  Almloef     # Almloef's model Hessian
                  Schlegel  # Schlegel's model Hessian
         # additional Hessian control for TS optimization
         Calc_Hess true  # calculate the Hessian numerically at the beginning
         Recalc_Hess  5  # calculate the Hessian at the beginning
                         #  and recalculate it after 5,10,.. cycles
         Hybrid_Hess [0 1 5 6] end # calculates a Hybrid Hessian
                                    # exact calculation for
                                    # atoms 0, 1, 5 and 6; works also
                                    # with Calc_Hess and Recalc_Hess
         NumHess true      # requests use of numerical Hessian
         # modification of the internal Hessian
         Hess_Internal
           {A 3 2 1 D 2.0} # define a diagonal Hessian value of
                           #  2 Eh/Bohr2 for the angle between
                           #  atoms 3 2 1. This can also be done for
                           #  bonds, dihedrals and Cartesian
                           #  coordinates.) The Hessian values of
                           #  multiple coordinates can be modified
           reset 5    # reset the modified internal Hessian values
                      #  after 5 cycles
         # The following is only recommended
         #  after a relaxed surface scan
         #  in this example of the scan coordinate B 1 0;
         #  "basename.004.xyz" contains the optimized structure
```

```
            #  of the scan step with highest energy
            {B 1 0 C}
            XYZ1 "scanName.003.xyz" # the xyz-files of the structures
            XYZ2 "ScanName.005.xyz" #  next to the highest energy point
            GBW1 "ScanName.003.gbw" # the gbw-files of the structures
            GBW2 "ScanName.005.xyz" #  next to the highest energy
                                    # the gbw-files are optional
         end
       # Hessian update procedure
       Update   Powell
                Bofill  # default for TS optimization
                BFGS    # default for geometry optimization
       # Hessian modification (only for P-RFO step)
       HESS_Modification Shift_Diag # shift the diagonal elements
                                    #  (default)
                         EV_Reverse # reverse the
                                    #  diagonal elements
       # Minimal value of Hessian eigenvalues (only P-RFO step)
       HESS_MinEV 0.0001  # if an absolute Hessian eigenvalue
                          #  is smaller than this value, it is
                          #  set to HESS_MinEV
       # Rebuilding the model Hessian after a number of cycles can
       #  accelerate the convergene of the optimization
       NResetHess 20 # Set the number of geometry steps after which
                     #  a new model Hessian is built (only with BFGS
                     #  update)
       NStepsInResetHess 5  # since previous steps and gradients are
                            #  available, it is possible to include
                            #  information about the PES in the
                            #  newly built Hessian (via a BFGS
                            #  update). This number should be
                            #  ? NResetHess
   end
```

As for parameter scan runs ORCA has some special options that may help to speed up the optimization:

```
  %geom UseSOSCF false # switches the converger to SOSCF
                       # after the first point. SOSCF may
                       # converge better than DIIS if the
                       # starting orbitals are good.
                       # default = false
        ReducePrint true # reduce printout after the first
                         # point default=true
   # the initial guess can be changed after the first
```

```
    # point. The default is MORead. The MO's of the pre-
    # vious point will in many cases be a very good guess
    # for the next point. In some cases however, you may
    # want to be more conservative and use a general guess.
    OptGuess   = OneElec; # the one electron matrix
               = Hueckel; # the extended Hueckel guess
               = PAtom;   # the PAtom guess
               = Pmodel   # the PModel guess
               = MORead;  # MO's of the prev. point (default)
    end
```

Redundant Internal Coordinates

There are three types of internal coordinates: redundant internals, old redundant internals (redundant_old) and (nonredundant) delocalized internals. All three sets work with the same 'primitive' space of internal coordinates (stretches, bends, dihedral angles and improper torsions). Only the redundant internals works with one more type of bends in cases where a normal bend would have been approximately $180°$. In redundant internal coordinates the full primitive set is kept and the Hessian and gradient are transformed into this – potentially large – space. A geometry optimization step requires, depending on the method used for the geometry update, perhaps a diagonalization or inversion of the Hessian of dimension equal to the number of variables in the optimization. In redundant internal coordinates this space may be 2-4 times larger than the nonredundant subspace which is of dimension $3N_{\text{atoms}} - 6(5)$. Since the diagonalization or inversion scales cubically the computational overhead over nonredundant spaces may easily reach a factor of 8–64. Thus, in redundant internal coordinates there are many unnecessary steps which may take some real time if the number of primitive internals is greater than 1000 or so (which is not so unusual). The timing problem may become acute in semiempirical calculations where the energy and gradient evaluations are cheap.

In order to deal with this situation, there is a second method implemented in ORCA which is based on (nonredundant) *delocalized internal coordinates*. If everything is well, the two methods show identical convergence since they start from the same Hessian and finally they span the same space. However, the delocalized internals take significantly less time for the geometry update since the dimension of the matrices to be manipulated is only the required $3N_{\text{atoms}} - 6(5)$ and there are no expensive transformation. The (only) drawback of the delocalized internals is that the iterative backtransformation from internal to Cartesian coordinates is — in our experience — less robust than in redundant internal coordinates and may fail. In this case you have to revert to redundant internals. If everything goes well, delocalized internals are a good choice.

In order to appreciate these differences we briefly outline the theoretical background which is not difficult to understand:

Suppose, we have a set of n_I (redundant) primitive internal coordinates $\mathbf{q}$ constructed by some recipe and a set of $n_C = 3N_{\text{atoms}}$ Cartesian coordinates $\mathbf{x}$. The B-matrix is defined as:

$$B_{ij} = \frac{\partial q_i}{\partial x_j} \tag{6.190}$$

This matrix is rectangular. In order to compute the internal gradient one needs to compute the 'generalized inverse of $\mathbf{B}$. However, since the set of primitive internals is redundant the matrix is rank-deficient and one has to be careful. In pratice one first computes the $n_I \times n_I$ matrix $\mathbf{G}$:

$$\mathbf{G} = \mathbf{B}\mathbf{B}^T \tag{6.191}$$

The generalized inverse of $\mathbf{G}$ is denoted $\mathbf{G}^-$ and is defined in terms of the eigenvalues and eigenvectors of $\mathbf{G}$:

$$\mathbf{G}^- = \begin{pmatrix} \mathbf{U} \\ \mathbf{R} \end{pmatrix}^T \begin{pmatrix} \mathbf{\Lambda}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{R} \end{pmatrix} \tag{6.192}$$

Here $\mathbf{U}$ are the eigenvectors belonging to the nonzero eigenvalues $\mathbf{\Lambda}$ which span the nonredundant space and $\mathbf{R}$ are the eigenvectors of the redundant subspace of the primitive internal space. If the set of primitive internals is carefully chosen, then there are exactly $3N_{\mathrm{atoms}} - 6(5)$ nonzero eigenvalues of $\mathbf{G}$. Using this matrix, the gradient in internal coordinates can be readily computed from the (known) Cartesian gradient:

$$\mathbf{g}_q = \mathbf{G}^- \mathbf{B} \mathbf{g}_x \tag{6.193}$$

The initial Hessian is formed directly in the redundant internal space and then itself or its inverse is updated during the geometry optimization.

Before generating the Newton step we have to ensure that the displacements take place only in the nonredundant part of the internal coordinate space. For this purpose a projector P':

$$\mathbf{P}' = \mathbf{G}\mathbf{G}^- = \mathbf{G}^-\mathbf{G} \tag{6.194}$$

is applied on both the gradient and the Hessian:

$$\tilde{\mathbf{g}}_{\mathbf{q}} = \mathbf{P}'\mathbf{g}_{\mathbf{q}} \tag{6.195}$$

$$\tilde{\mathbf{H}}_{\mathbf{q}} = \mathbf{P}'\mathbf{H}_{\mathbf{q}}\mathbf{P}' + \alpha\left(\mathbf{1} - \mathbf{P}'\right) \tag{6.196}$$

The second term for $\tilde{H}$ sets the matrix elements of the redundant part of the internal coordinate space to very large values ($\alpha = 1000$).

Coordinate steps

A Quasi-Newton (QN) step is the simplest choice to update the coordinates and is given by:

$$\Delta\mathbf{q} = -\tilde{\mathbf{H}}_{\mathbf{q}}^{-1}\tilde{\mathbf{g}}_{\mathbf{q}} \tag{6.197}$$

A more sophisticated step is the rational function optimization step which proceeds by diagonalizing the augmented Hessian:

$$\begin{pmatrix} \mathbf{H_q} & \mathbf{g_q} \\ \mathbf{g_q} & 0 \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ 1 \end{pmatrix} = v \begin{pmatrix} \Delta\mathbf{q} \\ 1 \end{pmatrix} \tag{6.198}$$

The lowest eigenvalue ν_0 approaches zero as the equilibrium geometry is approached and the nice side effect of the optimization is a step size control. Towards convergence, the RFO step is approaching the quasi-Newton step and before it leads to a damped step is taken. In any case,each individual element of $\Delta\mathbf{q}$ is restricted to magnitude `MaxStep` and the total length of the step is restricted to `Trust`. In the RFO case, this is achieved by minimizing the predicted energy on the hypersphere of radius `Trust` which also modifies the direction of the step while in the quasi-Newton step, the step vector is simply scaled down.

Thus, the new geometry is given by:

$$\mathbf{q}_{new} = \mathbf{q}_{old} + \Delta\mathbf{q} \tag{6.199}$$

However, which Cartesian coordinates belong to the new redundant internal set? This is a somewhat complicated problem since the relation between internals and Cartesians is very nonlinear and the step in internal coordinates is not infinitesimal. Thus, an iterative procedure is taken to update the Cartesian coordinates. First of all consider the first (linear) step:

$$\Delta\mathbf{x} = \mathbf{A}\Delta\mathbf{q} \tag{6.200}$$

with $\mathbf{A} = \mathbf{B}^T\mathbf{G}^-$. With the new Cartesian coordinates $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$ a trial set of internals $\mathbf{q}_{k+1}$ can be computed. This new set should ideally coincide with $\mathbf{q}_{new}$ but in fact it usually will not. Thus, one can refine the Cartesian step by forming

$$\Delta\Delta\mathbf{q} = \mathbf{q}_{new} - \mathbf{q}_{k+1} \tag{6.201}$$

which should approach zero. This leads to a new set of Cartesians $\Delta\mathbf{x}' = \mathbf{A}\Delta\Delta\mathbf{q}$ which in turn leads to a new set of internals and the procedure is iterated until the Cartesians do not change and the output internals equal $\mathbf{q}_{new}$ within a given tolerance (10^{-7} RMS deviation in both quantities is imposed in ORCA).

Now, consider working with the nonredundant subspace $\mathbf{q}_D = \mathbf{U}\mathbf{q}$. In this case, the whole procedure becomes rather simple. The 'delocalized' B-matrix is:

$$\mathbf{B}_D = \mathbf{U}^T\mathbf{B} \tag{6.202}$$

The corresponding $\mathbf{G}$ matrix is *not* rank deficient and can be readily inverted. Then, the gradient in delocalized internals is calculated as:

$$\mathbf{g}_D = \mathbf{G}_D^-\mathbf{B}_D\mathbf{g}_c \tag{6.203}$$

The initial (diagonal) Hessian in $\mathbf{q}$-space is converted to $\mathbf{q}_D$ simply by $\mathbf{U}^T\mathbf{H}_0\mathbf{U}$. Then the update procedure and perhaps the diagonalization in the RFO step only works in the nonredundant $3N - 6(5)$ subspace instead of the full primitive internal space which is an advantage. The back-transformation of the step in delocalized internals to Cartesians is completely analogous to the redundant internal case. However, in my experience it is less robust than in the latter case. In principle, the nonredundant subspace changes in every iteration since the B-matrix changes in every iteration. However, if one would take this into account the whole computational advantage of the delocalized internals is lost. Consequently, the delocalized internals are kept fixed and unless the geometry undergoes a qualitative change during the optimization this is not expected to have a large influence on the convergence characteristics. The final geometries predicted with any type of coordinates should of course always be the same.

Constrained Optimization

Constraints on the redundant internal coordinates can be imposed by modifying the above projector P' with a projector for the constraints C:

$$\mathbf{P} = \mathbf{P}' - \mathbf{P}'\mathbf{C}\left(\mathbf{C}\mathbf{P}\mathbf{C}\right)^{-1}\mathbf{C}\mathbf{P}' \tag{6.204}$$

C is a diagonal matrix with 1's for the constraints and 0's elsewhere. The gradient and the Hessian are projected with the modified projector:

$$\tilde{g}_q = P g_q \tag{6.205}$$

$$\tilde{H}_q = P H_q P + \alpha \left(1 - P\right) \tag{6.206}$$

Constrained Fragments Optimization

The constrain fragments option was implemented in order to provide a convenient way to handle constraints for systems consisting of several molecules. The difference to a common optimization lies in the coordinate setup. In a common coordinate setup the internal coordinates are built up as described in the following:

In a first step, bonds are constructed between atom pairs which fulfill certain (atom type specific) distance criteria. If there are fragments in the system, which are not connected to each other (this is the case when there are two or more separate molecules), an additional bond is assigned to the nearest atom pair between the nonbonded fragments. All other internal coordinates are constructed on the basis of this set of bonds. Here, in a second step, bond angles are constructed between the atoms of directly neighboured bonds. If such an angle reaches more than $175°$, a special type of linear angles is constructed. In a third step, dihedral angles (and improper torsions) are constructed between the atoms of directly neighboured angles.

If the constrain fragments option is switched on, the set of bonds is constructed in a different way. The user defines a number of fragments. For each fragment a full set of bonds (not seeing the atoms of the other fragments) is constructed as described above. If using this option, the user also has to define which fragments are to be connected. The connection between these fragments can either be user-defined or automatically chosen. If the user defines the connecting atoms N1 and N2, then the interfragmental bond is the one between N1 and N2. If the user does not define the interfragmental bond, it is constructed between the atom pair with nearest distance between the two fragments. Then the angles and dihedrals are constructed upon this (different) set of bonds in the already described fashion.

Now let us regard the definition of the fragment constraints: A fragment is constrained internally by constraining all internal coordinates that contain only atoms of the respective fragment. The connection between two fragments A and B is constrained by constraining specific internal coordinates that contain atoms of both fragments. For bonds, one atom has to belong to fragment A and the other atom has to belong to fragment B. Regarding angles, two atoms have to belong to fragment A and one to fragment B and vice versa. With respect to dihedrals, only those are constrained where two atoms belong to fragment A and the other two belong to fragment B.

6.13.2 Transition State Optimization

As transition state finder we implemented the well-established eigenvector following algorithm using a P-RFO step as implemented by Baker [88]. This algorithm is a quasi-Newton like algorithm.

The Taylor series of the energy, truncated after the quadratic term, is:

$$E = E_0 + g_q{}^+ \Delta q_q + \frac{1}{2} \Delta q{}^+ \mathrm{H}_q \Delta q \tag{6.207}$$

The Newton-Raphson step to get from the actual point to a stationary point is:

$$\Delta q = -\mathrm{H}_q^{-1} g_q = \sum -\frac{V_i^+ g_q V_i}{b_i} \tag{6.208}$$

with V_i and b_i as eigenvectors and eigenvalues of the Hessian H_q. This step leads to the nearest stationary point on the PES. This stationary point can be a minimum or a saddle point, according to the curvature of the PES at the actual point.

With a simple shift of the Hessian eigenvalues b_i in this equation one can guide the step to a stationary point with the required characteristics (Hessian with exactly one negative eigenvalue). The transition state search is separated into two different optimization problems. The energy is maximized along one Hessian eigenmode and minimized along the remaining $3N - 7(6)$ eigenmodes. We introduce two different shift parameters λ_p and λ_n, where λ_p is the shift parameter for the eigenmode being maximized and λ_n shifts the Hessian eigenvalues of the modes being minimized. This method allows us to maximize along any mode, not only the one with smallest eigenvalue. Starting from two different RFO-matrices for the different optimization problems (see description above) we get for λ_p and λ_n:

$$\lambda_p = \frac{1}{2} b_k + \frac{1}{2} \sqrt{b_k^2 + 4F_k^2} \; and \; \sum_{i \neq k} \frac{F_i^2}{\lambda_n - b_i} = \lambda_n \tag{6.209}$$

whereas $F_i = V_i^+ g$ is the component of g along the Hessian eigenmode V_i and λ_n has to get solved iteratively. The solution for λ_n has to be negative and lower than b_2 (or lower than b_1, if not the lowest mode is being followed). If the Hessian has more than one negative eigenvalue, these properties might not be fulfilled, and the Hessian would have to be modified. In our implementation the Hessian diagonal elements are either shifted or reversed in such a case.

Once the shift parameters are known the P-RFO step h is calculated as follows:

$$\Delta q_k = -\frac{\bar{F}_k V_k}{b_k - \lambda_p} \; and \; \Delta q_i = -\frac{\bar{F}_i V_i}{b_i - \lambda_n} \; with \, i = 1 \ldots n, \; i \neq k \tag{6.210}$$

$$\Delta q = \sum_{j=1}^{n} \Delta q_j \tag{6.211}$$

ScanTS option

For TS modes of rather local nature (involving only one bond or an angle; no concerted movements over multiple atoms) we implemented the ScanTS feature. Here the user can carry out a relaxed surface scan and a TS optimization in one calculation. After the relaxed surface scan the algorithm chooses the optimized structure of the scan with highest energy as initial guess structure and the two neighbouring structures for the calculation of the second derivative of the scanned coordinate (e.g., if scan step number 4 gives the structure with highest energy, then structure `basename.004.xyz` is the initial guess for the TS optimization; the structures `basename.003.xyz` and `basename.005.xyz` are used for the calculation of the second derivative). Before the first step of the subsequent TS optimization the energies and gradients for all three structures are calculated. The gradients are then transformed to internal coordinates. The diagonal Hessian value of the scanned coordinate is then calculated via finite difference of the internal gradients of the two given structures (003 and 005 in our example).

For the construction of the initial Hessian a model force field Hessian is built up (this Hessian has got only diagonal entries and zeros as off-diagonal elements). The exactly calculated diagonal Hessian value replaces the model force field Hessian entry for the respective internal coordinate.

If the user already performed a regular relaxed surface scan without the subsequent TS optimization, then he can nevertheless use these structures for the same procedure. A relaxed surface scan always gives you the xyz-files and gbw-files of the optimized structures of each scan step. A separate TS optimization can be carried out where the structure with highest energy is the starting structure. Additionally the two files with the two adjacent structures (as explained above) have to be provided (via the Hess_Internal keyword, see below). Furthermore, the internal coordinate, for which the diagonal Hessian value has to be calculated, has to be given (the previously scanned coordinate). This exact Hessian calculation is only possible for one internal coordinate:

```
%geom
     Hess_Internal
        {B 1 0 C}               # previously scanned coordinate
        XYZ1 "scanName.003.xyz" # the xyz-files of the structures
        XYZ2 "ScanName.005.xyz" #  next to the highest energy point
        GBW1 "ScanName.003.gbw" # the gbw-files of the structures
        GBW2 "ScanName.005.xyz" #  next to the highest energy
                                # the gbw-files are optional
     end
  end
```

Additionally the manipulation of the diagonal Hessian values of the internal Hessian is possible for further internal coordinates, but without an extra calculation. Here the user can just define a value (in $Eh/Bohr^2$).

```
  Hess_Internal
        {A 3 2 1 D 2.0} # define a diagonal Hessian value of
                        #  2 Eh/Bohr2 for the angle between
                        #  atoms 3 2 1
        {B 1 0 D -0.5}  # define a diagonal Hessian value of
                        #  -0.5 Eh/Bohr2 for the bond between
                        #  atoms 1 and 0
     end
```

The definition of such Hessian (diagonal) elements is possible for multiple internal coordinates. These just replace the values of the force field model Hessian.

Hybrid Hessian

We implemented the calculation of a "Hybrid Hessian" as an alternative to the full Hessian calculation for TS optimization. Here only those parts of the Hessian, that are important for the TS optimization, are calculated exactly. For this calculation we define two kinds of atoms: atoms whose couplings with the other atoms are treated exactly (E) and atoms whose couplings are treated approximately (A).

In a first step an Almloef model Hessian is built up in redundant internal coordinates and transformed to Cartesian coordinates. This Hessian gives the second derivative elements for atom pairs A/A. In a second step the second derivative elements between pairs E/E and E/A are calculated numerically as in a numerical frequency calculation:

$$\frac{\Delta \mathbf{E}}{\Delta i_B \Delta j_C} = \frac{\Delta \mathbf{E}}{\Delta j_C \Delta i_B} = \frac{g_{j,C}^{i,B} - g_{j,C}^{eq.}}{displ.} \tag{6.212}$$

with:

i,j : x-, y- or z-direction

B,C : pairs of E/E, E/A, A/E

$displ.$: magnitude of displacement

$g_{j,C}^{eq.}$: force on atom C in direction j in actual geometry

$g_{j,C}^{i,B}$: force on atom C in direction j after displacement of atom B in direction i

Partial Hessian Vibrational Analysis

We implemented the Partial Hessian Vibrational Analysis (PHVA), as published by Li, Jensen in [91], for the analysis of the nature of stationary points of structures obtained with QM/MM optimizations.

```
# PHVA after a QM/MM optimization in the (dispersion-/PC-) field
# caused by the MM-atoms
! NumFreq
%LJCoefficients "temp.LJ" # file with the Lennard Jones
                          #  coefficients for dispersion interaction
                          # obtained from last QM/MM run
%pointcharges "temp.pc"   # file with the point charges for
                          #  electrostatic interaction
                          # obtained from last QM/MM run
 #
%freq
    PARTIAL_Hess {0 1 2}  # atoms which are "frozen" and which make
                          #  the boundary to the MM-system
      end
    end
```

NOTE

- This procedure should be used for QM/MM optimized structures only to verify the nature of the stationary point and have an estimate of the ZPE.

Here we shortly describe the procedure: In PHVA we divide the system into two parts B and C. Let the atom set B belong to the region where the chemical change is localized. The Partial Hessian matrix is built up as follows:

$$\begin{pmatrix} K_{BB} & 0 \\ 0 & K_{CC}^{\varepsilon} \end{pmatrix} \tag{6.213}$$

With:

$$K_{BB} \quad : x-,\ y-\ or\ z-direction$$

$$K_{CC}^{\varepsilon} = \begin{pmatrix} \varepsilon & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \varepsilon \end{pmatrix},\ \varepsilon = 10^{-8} au, \tag{6.214}$$

this corresponds to using near-infinite masses for the atoms in C.

With this procedure we get the following eigenvalue structure:

- Six zero eigenvalues with modes corresponding to translational and rotational motion of the entire molecule.

- 3*(N - n) - 6 small (less than 1 cm^{-1}) eigenvalues with modes corresponding mainly to internal motion within region C.

- Three eigenvalues (typically less than 10 cm^{-1}) with modes corresponding mainly to motion of region C relative to region B.

- 3n - 3 eigenvalues with modes corresponding mainly to relative motion of B and C as well as internal motion within region B.

6.13.3 Minimum Energy Crossing Points

The MECP optimization allows the user to optimize to structures where two different potential energy surfaces (PES1 and PES2) cross each other. In this optimization two conditions apply: the energy E$_1$ of PES1 is minimized while at the same time the energy difference $(E_1 - E_2)^2$ of both surfaces is minimized. For the implementation we follow in principle the suggestions of Harvey et al. in [90].

For the minimization two different gradients are constructed:

The first gradient chosen for the minimization is

$$\mathbf{f} = \frac{\partial}{\partial q} (E_1 - E_2)^2 = 2 (E_1 - E_2) \cdot x_1 \tag{6.215}$$

where x_1 is the gradient difference vector

$$x_1 = \left[\left(\frac{\partial E_1}{\partial q} \right) - \left(\frac{\partial E_2}{\partial q} \right) \right] \tag{6.216}$$

Which is orthogonal to the crossing hyperline near the MECP.

The gradient

$$\mathbf{g} = \left(\frac{\partial E_1}{\partial q} \right) - \frac{x_1}{|x_1|} \left[\left(\frac{\partial E_1}{\partial q} \right) \cdot \frac{x_1}{|x_1|} \right] \tag{6.217}$$

is orthogonal to x_1.

Both gradients are combined to yield the effective surface crossing gradient

$$\mathbf{g_{SC}} = \mathbf{g} + \mathbf{f} \tag{6.218}$$

The crossing hyperline is defined as the $3N - 7$ dimensional subspace of PES1, which is orthogonal to x_1. In the MECP optimization we want to find the point of lowest energy within this subspace.

Our calculation of normal modes and force constants for movements along the crossing hyperline differ from the one proposed by Harvey et al. A standard frequency analysis can not be performed, but a similar procedure is applied:

Let us regard the second-order Taylor expansion for the energy of both surfaces near the MECP for a displacement along the crossing hyperline (orthogonal to x_1):

$$\mathbf{E}_A = E_{MECP} + \frac{1}{2} \Delta q^T H_{eff,A} \Delta q \tag{6.219}$$

with:

    $\mathbf{E}_A$: Energy E_1 on PES1 or E_2 on PES2

    $H_{eff,A}$: effective Hessian for PES1 or PES2

    Δq : displacement along the crossing hyperline

Diagonalization of this effective Hessian gives us the normal modes of the crossing hyperline and thus allows us to decide whether the MECP optimization converged to a minimum in the $3N - 7$ dimensional subspace of the crossing hyperline.

The procedure for the calculation of the effective Hessian is now as follows: For each of both surfaces the second derivative matrix is calculated. Then the 6 rotations and translations and additionally the direction of the gradient difference vector x_1 (this ensures that movement orthogonal to the crossing hyperline, for which we do NOT satisfy the conditions of a stationary point, is excluded) are projected out from the Hessian matrix.

For MECP optimizations the following options exist:

```
%mecp
   SurfCrossOpt true  # switches on the MECP optimization
                      #  alternatively use: ! SurfCrossOpt
   SurfCrossNumFreq true # switches on the MECP effective Hessian
                         #  calculation
                         #  alternatively use: ! SurfCrossNumFreq
 # separate MO input for the second spin state (PES2)
   moinp "Myfile.gbw"# MO input for PES2
 # information on the electronic structure of PES 2
   Mult 3            # multiplicity of PES2
   brokenSym 1,1     # broken symmetry for PES2
 # CASSCF options for PES2 (also see the CASSCF chapter)
   casscf_nel 6       # number of active space electrons
   casscf_norb 6      # number of active orbitals
   casscf_mult 1,3    # multiplicities singlet and triplet
   casscf_nroots 4,2  # four singlets, two triplets
   casscf_bweight 2,1 # singlets and triplets weighted 2:1
   casscf_weights[0] = 0.5,0.2,0.2,0.2 # singlet weights
   casscf_weights[1] = 0.7,0.3         # triplet weights
 end
```

6.13.4 Numerical Gradients

If you want to use numerical instead of analytic gradients you have to use

```
! NumGrad
```

in your input file. Additionally the settings for the numerical differentiation can be changed:

```
%numgrad
  CentralDiff true # (Default) You should use two-sided numerical
                   #  differentiation, but it is possible to switch
                   #  to one-sided numerical differentiation.
  DX 0.005         # Increment in Bohr for the differentiation.
  TransInvar  true # (Default) Take advantage of translation invariance
 end
```

6.13.5 ORCA as External Optimizer

If you want to use only the optimizer of ORCA you have to use

```
! ExtOpt
```

in your input file. All information that you give on the electronic structure is discarded. In each optimization step ORCA writes a file called "`extopt.extcomp.inp`" with the following format:

```
"N (nr. of atoms)" "calculation requested" "charge" "multiplicity"
atomic nr.(atom1)  x-coord.(atom1) y-coord.(atom1) z-coord.(atom1)
...
atomic nr.(atomN)  x-coord.(atomN) y-coord.(atomN) z-coord.(atomN)

example:
        9             1          0           1
        6     -7.183647732744      1.832728827744      -0.116462028781
        9     -6.344271117689     -0.569555640677      -0.053332293594
        9     -6.385938620292      3.010680341363       1.994410067976
        9     -9.725973988287      1.760800299423      -0.041860336809
        6     -6.281650723853      3.228508119832      -2.541654671259
        1     -7.049215894384      2.192752895053      -4.164371259742
        1     -7.120030511673      5.123374809616      -2.502306840221
        8     -3.579612411580      3.262825146858      -2.485537715188
        1     -3.134995715660      4.196025355887      -4.047828323839
```

NOTE: the coordinates are given in Bohr.

There are two types of calculation:

1. calculate energy and gradient

2. calculate energy, gradient and Hessian

If a Hessian is requested, it should be stored in the ORCA Hessian file format and be named "`yourInputFilename.hess`".

ORCA then calls a script (that is not distributed with the ORCA binaries):

   "orca_External   extopt.extcomp.inp   extopt.extcomp.out"

Your script starts the energy, gradient (and Hessian) calculation and finally provides the results in a file called "`extopt.extcomp.out`", which has to be written in the following format:

```
Total Energy
gradient(x) on atom1    gradient(y) on atom1    gradient(z) on atom1
...
gradient(x) on atomN    gradient(y) on atomN    gradient(z) on atomN

example:
-1.135276876846e+02
-1.144030900000e-05   3.458417100000e-05   7.686904800000e-06
-4.450221700000e-05  -2.016015000000e-05  -2.617359400000e-05
 4.460242300000e-05  -3.290908700000e-05   5.698279500000e-06
```

```
     1.026743300000e-06   4.889225700000e-05   3.474765100000e-05
     6.178236500000e-05  -1.506876000000e-04  -1.288283000000e-04
     1.601578300000e-05   1.670611600000e-05   2.747591400000e-05
    -4.360515500000e-06   5.840020000000e-06   8.823311100000e-06
    -1.135615900000e-04   1.384714300000e-04   6.197283200000e-05
     5.043762200000e-05  -4.073673300000e-05   8.597172100000e-06
```

NOTE:

- the energy should be given in Eh, gradients should be given in Eh/Bohr, Hessian values should be given in Eh/Bohr^2

ORCA then performs the next optimization step, checks for convergence, and if the optimization has not yet converged, it goes on to the next optimization step.

Gaussian as External Optimizer. To use the external optimizer from Gaussian in ORCA, the following keywords were provided in the past:

```
%geom
   UseGaussian true # Use the external Gaussian optimizer instead
                    #  of the ORCA optimizer.
   GaussianName "GAU" # String defining the name of the Gaussian
                      #  optimizer
   GauOptFlags   # String indicating the optimization flags
   Gaussian Constraints # List defining the constraints for
                        #  the Gaussian optimizer.
end
```

Since the ORCA team got banned by Gaussian in January 2007 we can no longer support these option flags. They have not been removed from the code and may or may not work. If there is trouble with it we can – unfortunately – not offer any help since we do not have access to the Gaussian code any longer.

6.14 Excited States via CIS and TD-DFT

ORCA features a relatively efficient single-excitation CI (CIS) and time-dependent DFT module that can be used to calculate excitation energies, absorption intensities and CD intensities. Especially TD-DFT became very popular for excited state calculations as it offers significantly better results than HF-CIS at about the same cost. However, there are also many pitfalls of TD-DFT, some of which are discussed in reviews [310] [311]. TD-DFT methods are available for closed shell and spin-unrestricted reference states. Analytic gradients are available. There also is a doubles correction implemented that improves the results (but also the computational cost). It is often used together with double hybrid functionals as explained below. The TD-DFT module of ORCA is also extensively used for the calculation of X-ray absorption spectra at the K-edge of a given element.

6.14.1 General Features

The module is invoked with the block:

```
%cis  end

# or equivalently

%tddft end
```

There are a variety of options. The most important one is the number of excited states that you want to have calculated:

```
%cis NRoots    10
     MaxDim    100
     end
```

The variable `NRoots` gives the number of excited states to be calculated. `MaxDim` is the maximum dimension of the expansion space in the Davidson procedure. If `MaxDim` $\approx$ 5-10 times `NRoots` the calculations will show favorable convergence but also increased disk space demands. In general the larger `NRoots` the more core memory is needed while the disk space requirements are proportional to `MaxDim`.

The convergence tolerances are:

```
%cis ETol  1e-6
     RTol  1e-6
     end
```

The variable `ETol` gives the required convergence of the energies of the excited states (in Eh) and `RTol` is the required convergence on the norm of the residual vectors. Under normal ciorcumstances the calculations need about 5-10 iterations to converge to the default convergence tolerances.

If closed shell references are used the program can calculate the singlet and triplet excited states at the same time by using:

```
%cis triplets   true
     end
```

This is available for all combinations of methods except double hybrid DFT or analytic gradients.

In order to control the orbitals that should be taken into account in the calculation two mechanisms are available. The first mechanism is the default mechanism and consists of specifying and orbital energy window within which all single excitations will be considered:

```
    %cis EWin  -3,3  # (orbital energy window in Eh)
         end
```

Thus, the default is to keep core orbitals frozen and to neglect very high lying virtual orbitals which is a sensible approximation. However, you may want to consider to include all virtual orbitals by choosing for example `EWin -3,10000`. The second mechanism is to explicitly give an orbital energy window for each operator, i.e.

```
    %cis OrbWin[0] 2,-1,-1,14 # orbital window for spin-up MOs
         OrbWin[1] 2,-1,-1,16 # orbital window for spin-down MOs
         end
```

The "-1"'s in the above example mean that the HOMO and LUMO for the spin-.up and spin-down orbitals will be automatically determined by the program.

In using the CIS/TD-DFT module five different types of calculations should be distinguished:

- Semiempirical methods

- Hartree-Fock calculations

- DFT calculations without HF exchange (non-hybrid functionals)

- DFT calculations with HF exchange (hybrid functionals)

- DFT calculations with HF exchange and MP2 correlation (double hybrid functionals)

6.14.2 Semiempirical Methods

The semiempirical INDO/S method is very suitable to calculate absorption spectra of medium sized to large organic and inorganic molecules. It has been parameterized by the late M.C. Zerner for optical spectroscopy and in my experience at least, it tends to work nicely for many systems. With the semiempirical approach it is easy to calculate many states of large molecules. For example, consider the following calculation on a bis-histdine ligated iron-porphyrin model (in the Fe(II) state) that includes 92 atoms and $\approx$ 16,500 CSFs in the single excitation space. Yet the calculation requires only a few minutes on an ordinary computer for the prediction of the first 40 excited states.

The calculated spectrum is in essentially reasonable agreement with experiment in showing a huge band around 400 nm (the famous Soret band) and a smaller but still intense band between 500 and 550 nm (the Q-band). There are no predicted absorptions below $\approx$ 10,000 cm^{-1}.

The input for the job is shown below:

```
    #                          ----------------------
    #                                 ORCA Test
    #                          ----------------------
    #
    # Test CIS in conjunction with INDO/S
    #
    ! RHF ZINDO/S TightSCF DIIS NoRICO NoMOPrint
    %cis  NRoots  40
```

```
      MaxDim 160
      end
* xyz 0 1
Fe    -0.01736    0.71832    -0.30714
C      2.65779    4.03195    -0.13175
C      3.51572    3.02488    -0.24101
C      2.66971    1.82027    -0.30891
C      3.30062    0.51609    -0.42755
C      2.61022   -0.60434    -0.47131
C      3.32146   -1.89491    -0.57434
C      2.35504   -2.79836    -0.57179
C      1.11740   -1.99868    -0.46878
C     -0.04908   -2.61205    -0.44672
C     -1.30967   -1.89127    -0.38984
C     -2.58423   -2.63345    -0.40868
C     -3.50492   -1.68283    -0.37930
C     -2.72946   -0.42418    -0.33711
C     -3.35747    0.73319    -0.28970
C     -2.66935    2.01561    -0.22869
C     -3.31167    3.19745    -0.16277
C     -4.72835    3.62642    -0.14517
C     -5.84825    2.89828    -0.20597
C     -2.21443    4.15731    -0.09763
C     -1.11572    3.39398    -0.14235
C      0.19578    4.02696    -0.10122
C      1.33370    3.36290    -0.15370
C      3.09165    5.44413    -0.02579
C      2.35656    6.55323     0.10940
N      1.43216    2.09428    -0.24815
N      1.34670   -0.74673    -0.42368
N     -1.39885    2.15649    -0.21891
N     -1.47620   -0.63353    -0.34705
C      5.03025    3.02708    -0.28544
C      4.81527   -2.12157    -0.66646
C     -5.01065   -1.83771    -0.38886
C     -2.28137    5.66820    -0.00321
C     -2.73691   -4.14249    -0.43699
C     -2.42579   -4.72805    -1.83259
C      2.45978   -4.31073    -0.64869
C      2.19678   -4.82182    -2.08201
C      1.60835   -6.22722    -2.10748
C     -1.90102   -6.15737    -1.82447
O     -1.96736   -6.92519    -2.75599
O      1.60982   -7.01844    -1.19330
O     -1.15355   -6.41323    -0.74427
O      0.89871   -6.41433    -3.22828
H      4.17823    5.62170    -0.05623
H      2.86221    7.53117     0.17503
H      1.26303    6.57673     0.17212
H      0.21799    5.11603    -0.03468
H     -1.78003    6.14426    -0.87498
H     -3.32281    6.05139     0.01906
H     -1.78374    6.03115     0.92347
H     -4.89690    4.71221    -0.07358
H     -6.82566    3.40843    -0.18007
H     -5.88239    1.80643    -0.28628
```

```
H      -4.44893      0.70720     -0.28575
H      -5.32107     -2.89387     -0.54251
H      -5.45075     -1.49552      0.57400
H      -5.46788     -1.24144     -1.20929
H      -2.05997     -4.55939      0.34045
H      -3.76430     -4.43895     -0.12880
H      -3.33638     -4.66246     -2.47119
H      -1.65517     -4.10119     -2.33605
H      -0.56422     -7.14866     -1.00437
H       0.26056     -7.12181     -3.00953
H       1.48118     -4.13253     -2.58671
H       3.13949     -4.79028     -2.67491
H       3.46153     -4.65168     -0.30336
H       1.73023     -4.75206      0.06633
H       5.26172     -1.51540     -1.48550
H       5.31767     -1.84036      0.28550
H       5.06416     -3.18438     -0.87628
H      -0.07991     -3.70928     -0.48866
H       4.39835      0.46775     -0.47078
H       5.39550      2.59422     -1.24309
H       5.47197      4.04179     -0.19892
H       5.44914      2.41988      0.54738
N       0.01831      0.60829      1.68951
C       0.02054      1.64472      2.54371
C       0.04593     -0.50152      2.45186
N       0.04934      1.20474      3.84418
C       0.06582     -0.16578      3.80848
H       0.00322      2.72212      2.31829
N      -0.05051      0.81937     -2.30431
H       0.05251     -1.53704      2.08183
C       0.11803      1.92670     -3.04495
H       0.05712      1.81091      4.70485
H       0.08982     -0.83278      4.68627
C      -0.24302     -0.18840     -3.17641
C      -0.19749      0.28568     -4.49059
N       0.03407      1.63309     -4.38373
H       0.30109      2.95786     -2.70479
H      -0.41432     -1.24242     -2.91290
H      -0.31761     -0.27403     -5.43315
H       0.12975      2.31943     -5.17616
*
```
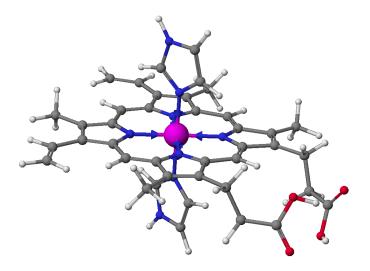
Figure 6.4: Structure of the iron-porphyrin used for the prediction of its absorption spectrum (the structure was obtained from a molecular mechanics calculation and the iron-imidazole bondlength was set to 2.0 Å).
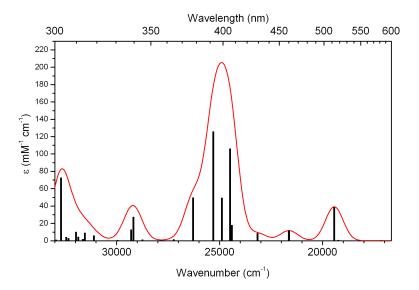
Figure 6.5: The ZINDO/S predicted absorption spectrum of the model iron porphyrin shown above. The spectrum has been plotted using the orca_mapspc tool.

Note that ORCA slightly departs from standard ZINDO/S in using dipole integrals in the intensity calculations that include all one- and two-center terms which are calculated via a STO-3G expansion of the Slater basis orbitals. The calculated intensities are not highly accurate anyways. In the present case they are overestimated by a factor of $\approx$ 2.

6.14.3 Hartree-Fock Wavefunctions

For HF calculations the module will always calculate the CI-singles wavefunctions. The "random-phase approximation" (RPA) is not implemented as it will generally not improve the results and is algorithmically somewhat more involved. In general, CIS calculations do not lead to good agreement with experimental excitation energies and errors of 1-5 eV are common. Therefore HF/CIS is mostly a qualitative tool or can be used with caution for larger molecules if more extensive and more well balanced CI calculations are not computationally tractable.

6.14.4 Non-Hybrid DFT

For non-hybrid functionals there is the choice between the full TD-DFT treatment and the so-called Tamm-Dancoff approximation (TDA). The TDA is the same approximation that leads from RPA to CIS (i.e. neglect of the so-called "B" matrix). The results are usually very similar between the two approaches. The TDA is turned on by:

```
%tddft TDA          true # (default)
       TammDancoff  true # (equivalent)
       end
```

TIP:

- For `TDA=true` it is recommended to put `Etol=-1` and `RTol=1e-5 ... 1e-4` since the convergence is normally significantly slower as for `TDA=false`.

6.14.5 Hybrid DFT

If you insist on using hybrid functionals `TDA=true` is required and will be enforced by the program. There are situations where hybrid functionals give significantly better results than GGA functionals since they suffer less from the self-interaction error. The RIJCOSX procedure [69] [141] leads to very large speedups in such calculations at virtually no loss in accuracy [312].

6.14.6 Double hybrid functionals and Doubles Correction

The program can compute a doubles correction to the CIS excitation energies. The theory is due to Head-Gordon an co-workers. [313] The basic idea is to compute a perturbative estimate (inspired by EOM-CCSD theory) to the CIS excited states that is compatible with the MP2 ground state energy. In many cases this is a significant improvement over CIS itself and comes at a reasonable cost since the correction is computed *a posteriori*. Of course, if the CIS prediction of the excited state is poor, the (D) correction – being perturbative in nature – cannot compensate for qualitatively wrong excited state wavefunctions.

In addition – and perhaps more importantly – the (D) correction is compatible with the philosophy of the double hybrid functionals and should be used if excited states are to be computed with these functionals. The results are usually much better than those from TD-DFT since due to the large fraction HF exchange, the self-interaction error is much smaller than for other functionals and after the (D) correction the results do not suffer from the overestimation of transition energies that usually comes with increased amounts of HF exchange in TD-DFT calculations.

Since the calculations would require a fairly substantial integral transformation that would limit it to fairly small molecules if no approximation are introduced we have decided to only implement a RI version of it. With this approximation systems with more than 1000 basis functions are readily within the reach of the implementation.

Since one always has a triad of computational steps: MP2-CIS solution-(D) correction, we have implemented several algorithms that may each become the method of choice under certain circumstances. The choice depends on the size of the system, the number of roots, the available main memory and the available disk space together with the I/O rate of the system. The formal cost of the (D) correction is $O(N^5)$ and its prefactor is higher than that of RI-MP2. In the best case scenario, the rate limiting step would be the calculation of the pair-contribution in the 'U-term' which requires (for a closed-shell system) twice the effort of a RI-MP2 calculation *per state*.

The use of the (D)-correction is simple. Simply write:

```
! RKS B3LYP/G SV(P) SV/C TightSCF
%cis dcorr n # n=1-4. The meaning of the four algorithms is
             # explained below.
   # algorithm 1 Is perhaps the best for small systems. May use a
   #             lot of disk space
   # algorithm 2 Stores less integrals
   # algorithm 3 Is good if the system is large and only a few
   #             states are calculated. Safes disk and main
   #             memory.
   # algorithm 4 Uses only transformed RI integrals. May be the
   #             fastest for large systems and a larger number
   #             of states
       end
```

Table 6.9: Integral handling in various implementations of the (D) correction (i,j=occupied MOs, a,b=virtual MOs, Q=aux function; NumInt=numerical integration).

DCORR=	1	2	3	4
(ia|jb) integrals	Stored	Stored	Not stored	Not stored
(ij|ab) integrals	Stored	Not made	Not made	Not made
(ab|Q) integrals	Stored	Not made	Not made	Stored
(ij|Q) integrals	Stored	Stored	Stored	Stored
(ia|Q) integrals	Stored	Stored	Stored	Stored
Coulomb CIS	From (ia|jb)	From (ia|jb)	From (ia|Q)	From (ia|Q)
Exchange CIS	From (ij|ab)	RI-AO-direct	RI-AO-direct	From (ab|Q)
XC-CIS	Num. Int.	Num. Int.	Num. Int.	Num. Int.
V-term in (D)	From (ia|jb)	From (ia|jb)	From (ia|Q)	From (ia|Q)
U-term in (D)	From (ab|Q)	RI-AO-direct	RI-AO-direct	From (ab|Q)

NOTE:

- In all three involved code sections (MP2, CIS, (D)) the storage format FLOAT is respected. It cuts down use of disk and main memory by a factor of two compared the default double precision version. The loss of accuracy should be negligible; however it is – as always in science – better to double check.

- The (ab|Q) list of integrals may be the largest for many systems and easily occupies several GB of disk space (hence algorithms 2 and 3). However, that disk-space is often well invested unless you run into I/O bottlenecks.

- The (ia|jb) and (ij|ab) lists of integrals is also quite large but is relatively efficiently handled. Nevertheless, I/O may be a problem.

- Making the exchange contribution to the CIS residual vector in an RI-AO direct fashion becomes quite expensive for a larger number of states. It may be a good choice if only one or two excicted states are to be calculated for a larger system.

6.14.7 Natural Transition Orbitals

Results of TD-DFT or CIS calculations can be tedious to interprete as many individual MO pairs may contribute to a given excited state. In order to facilitate the analysis while keeping the familiar picture of an excited state originating from essentially an electron being promoted from a donor orbital to an acceptor orbital, the device of "natural transition orbitals" can be used.

The procedure is quite straightforward. For example, consider the following job on the pyridine molecule:

```
    ! pbe d3zero def2-svpd def2-svp/j tightscf


    %tddft  nroots 5
            DoNTO   true      # flag to turn on generation of natural transition orbitals
            NTOStates 1,2,3  # States to consider for NTO analysis;
                                    #if empty all will be done
            NTOThresh 1e-4   # threshold for printing occupation numbers
            MaxDim 100
            end

    * xyz 0 1
      N      0.000000    0.000000    1.401146
      C      0.000000    1.146916    0.702130
      C      0.000000   -1.146916    0.702130
      C     -0.000000    1.205574   -0.702848
      C     -0.000000   -1.205574   -0.702848
      C      0.000000   -0.000000   -1.421344
      H     -0.000000    2.079900    1.297897
      H     -0.000000   -2.079900    1.297897
      H     -0.000000    2.179600   -1.219940
      H     -0.000000   -2.179600   -1.219940
      H      0.000000    0.000000   -2.525017
    *
```

which results in:

```
-------------------------------------------
NATURAL TRANSITION ORBITALS FOR STATE    1
-------------------------------------------

Making the (pseudo)densities                        ... done
Solving eigenvalue problem for the occupied space  ... done
Solving eigenvalue problem for the virtual space   ... done
Natural Transition Orbitals were saved in TD-DFT-Example-6.s1.nto
Threshold for printing occupation numbers 0.000100

  E=   0.158492 au      4.313 eV    34785.0 cm**-1
     20a ->  21a  : n=  0.99825296
     19a ->  22a  : n=  0.00067172
     18a ->  23a  : n=  0.00051394
     17a ->  24a  : n=  0.00030846


-------------------------------------------
NATURAL TRANSITION ORBITALS FOR STATE    3
-------------------------------------------

Making the (pseudo)densities                        ... done
Solving eigenvalue problem for the occupied space  ... done
Solving eigenvalue problem for the virtual space   ... done
Natural Transition Orbitals were saved in TD-DFT-Example-6.s3.nto
Threshold for printing occupation numbers 0.000100

  E=   0.197103 au      5.363 eV    43259.2 cm**-1
     20a ->  21a  : n=  0.64493520
     19a ->  22a  : n=  0.34962356
     18a ->  23a  : n=  0.00166855
     17a ->  24a  : n=  0.00112178
     16a ->  25a  : n=  0.00073279
     15a ->  26a  : n=  0.00062556
     14a ->  27a  : n=  0.00045127
     13a ->  28a  : n=  0.00023135
     12a ->  29a  : n=  0.00019911
     11a ->  30a  : n=  0.00017459
     10a ->  31a  : n=  0.00011544


-----------------------------------------------------------------------------
       ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS
-----------------------------------------------------------------------------

State   Energy  Wavelength   fosc         T2        TX        TY        TZ
        (cm-1)    (nm)                   (au**2)    (au)      (au)      (au)
-----------------------------------------------------------------------------
    1   34785.0   287.5   0.004079502   0.03861  -0.19649   0.00000  -0.00000
    2   35118.0   284.8   0.000000000   0.00000   0.00000   0.00000   0.00000
    3   43259.2   231.2   0.024852699   0.18913  -0.00000   0.43490  -0.00000
    4   49592.3   201.6   0.000013122   0.00009   0.00000   0.00000  -0.00933
    5   54405.8   183.8   0.027126662   0.16414  -0.00000  -0.40515  -0.00000
```

We see that there is a weakly allowed transition (S1) that is essentially totally composed of a single NTO pair (20a -¿ 21a : n= 0.99825296), while the third excited state (S3) is strongly allowed and requires two NTO pairs for its description (20a -¿ 21a : n= 0.64493520 and 19a -¿ 22a : n= 0.34962356).

These orbitals are shown below. It is evident that the S1 state donor orbital (NTO10) is a nitrogen lone pair and the

acceptor orbital is a pi* orbital of the ring. For the S3 state the two NTO donor orbitals are comprised of a nearly degenerate set of pi orbitals (they would be degenerate in the parent benzene) and the acceptor orbitals are a pair of nearly degenerate pi* orbitals. It is evident from this example that by looking at the NTOs one can obtain a nicely pictorial view of the transition process, even if many orbital pairs contribute to a given excited state in the canonical basis.
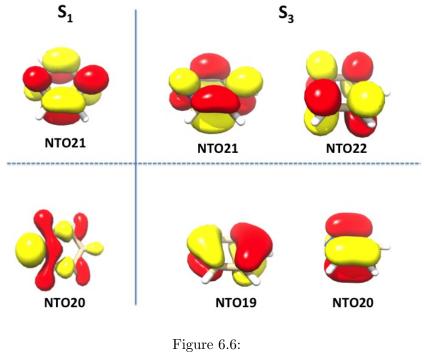
Figure 6.6:
fig:742

6.14.8 Computational Aspects

6.14.8.1 RI Approximation (AO-Basis)

If the SCF calculation used the RI approximation it will also be used in the TD-DFT calculation. The RI approximation saves a large amount of time while giving close to identical results (the errors will usually be ¡0.1 eV) and is generally recommended. If the functional is a hybrid functional the RI-approximation will only be applied to the Coulomb term while the exchange will be treated as before. In the SCF you can use this feature with the keyword (! RIJONX). It will then also be used in the TD-DFT calculation. Again, the RIJCOSX approximation can be used in TD-DFT and CIS calculations and leads to very large speedups at virtually no loss in accuracy.

6.14.8.2 RI Approximation (MO-Basis)

As an alternative to the direct AO-basis computation ORCA allows to use RI-integrals transformed to the MO basis to generate the CI matrix. This algorithm is more disk-intensive. However, for medium sized molecules we have observed speedups on the order of 15 or more with this method. It is particularly benefitial together with hybrid functionals.

In order to use this method you have to specify mode riints in the %tddft block and you also have to assign an auxiliary basis set (for example SV/C). There is a second algorithm of this kind that is labelled mode riints_disk

Note that the auxiliary basis set has to be valid for correlation treatments in case that you have a hybrid functional. Thus the basis sets developed for RI-MP2 are suitable (`def2-SVP/C`, `def2-TZVP/C` and `def2-TZVPP/C`). If you have a non-hybrid functional the normal RI-J auxiliary basis sets are fine.

An example that uses the B3LYP functional is given below:

```
! RKS B3LYP/G SV(P) SV/C TightSCF
%tddft mode riints    # or riints_disk (often faster but requires more disk space)
       nroots    8
       maxdim    64
       end
* int 0 1
    C  0 0 0  0.00    0.0    0.0
    O  1 0 0  1.20    0.0    0.0
    H  1 2 0  1.08 120.0    0.0
    H  1 2 3  1.08 120.0 180.0
*
```

NOTE:

- Do not forget to assign a suitable auxiliary basis set! If Hartree-Fock exchange is present (HF or hybrid-DFT) these are the auxiliary bases optimized for correlation while for non-hybrid functionals the standard RI-J bases are suitable.

- The standard auxiliary basis sets may not be suitable if you have diffuse functions present and want to study Rydberg states. You have to augment the axuliary basis with diffuse functions yourself in this case.

- Be prepared that the transformed integrals take up significant amounts of disk space.

6.14.8.3 Integral Handling

If the SCF calculation is carried out in an integral direct fashion this will also be done in the CIS/TD-DFT calculation. Thus, no bottlenecks arising from large integral transformations or large disk space requirement arise in the calculations. An exception is the MO based RI approximations described in the previous section.

6.14.8.4 Valence versus Rydberg States

For valence excited states the usual orbital basis sets are reasonable. Thus, with polarized double-zeta basis sets sensible results are obtained. Especially DFT calculations have the nice feature of not being overly basis set dependent.

If Rydberg states are desired you should make sure that diffuse functions are present in your basis set. These can be added to any "normal" basis set. For example, the following example provides a rather high quality basis for excited state calculations that is based on the Ahlrichs basis set:

```
%basis  basis    vtz            # standard triple-zeta
        pol      Ahlrichs_2df   # large (2d1f) polarization
        diff     _p             # plus 1 s,p diffuse set
        # augment the carbon basis set by even more
        # diffuse functions
        addgto 6
          s 1
             1 0.01 1.0
          p 1
             1 0.01 1.0
          d 1
             1 0.07 1.0
          end
        end
```

Smaller basis sets may also be sufficient. TIP

- If you want to augment a given basis set it is sensible to run a preliminary SCF calculation and use `%output print[p_basis] 2 end`. This will provide you with a detailed listing of basis functions and their exponents. You can then add additional s, p and perhaps d-functions with the `AddGTO` command as in the example above. It is sensible to decrease the exponent of the diffuse functions by roughly a factor of 3 from the smallest exponent in the original basis.

6.14.8.5 Restrictions for Range-Separated Density Functionals

Several restrictions apply for range-separated density functionals. They are currently only implemented to work with the AO-based algorithm within the RIJONX, RIJCOSX, and NORI integral schemes. Moreover, the calculation of triplet states has been disabled. Additionally, the asymptotic correction (section 6.14.8.6) has been disabled.

6.14.8.6 Asymptotatically Corrected Density Functionals

In studying Rydberg states one general shortcoming of the present day density functionals becomes particularly prominent. This is the too fast decay of the DFT *potential* in the long range limit. This causes the DFT orbital energies being too positive by several eV. Likewise the excitation energies to Rydberg states are underestimated by several eV. This can be fixed by using density functionals with improved long range behavior. One such functional is the `LB94` functional. However, the `LB94` has the disadvantage of being only defined through the potential and that this potential is not accurate in the bonding region. Therefore it is reasonable to only correct an existing functional (like the `BP` functional) in the long range with the `LB94` functional. In ORCA the so-called "gradient regular asymptotic correction" (GRAC) is implemented and can be applied with any "bulk" density functional in the bonding region. The energetics and occupied orbital energies are only affected to a minor extent while the virtual orbital energies are significantly changed and the excitation energies to Rydberg states are improved.

The following example shows how to use the asymptotic correction. The run almost exactly reproduces the results of Grüning *et al.* [314] who also developed the method implemented in ORCA.

```
    #
    # Gradient-regulated asymptotic correction calculation
    # on the excited states of the CO molecule.
    #
    # Reference: Grüning et al. (2001) JCP, vol. 114, p. 652
    #
    ! RKS BP NoRI Grid5 NoFinalGrid TightSCF
    %method xckernel     lda
            ldaopt       c_vwn5
            xckernellda  c_vwn5
            # *** turn on the asympt. correction
            gracLB  true
            # *** the input ionization potential
            # *** in eV !!!
            ip 14.01
            end
    %basis  basis    vtz
            pol      Ahlrichs_2df
            diff     _p
            addgto 6
             s 1
             1 0.01 1.0
             p 1
             1 0.01 1.0
             d 1
             1 0.07 1.0
            end
            addgto 8
             s 1
             1 0.02 1.0
             p 1
             1 0.02 1.0
             d 1
             1 0.1  1.0
            end
            end
    %tddft  nroots   10          # no of roots to determine
            tda      false       # Tamm-Dancoff approx
            maxdim   100         # max. dim of expansion space
            etol     -1          # energy tolerance
            rtol     1e-5        # residual tolerance
            triplets true        # generate triplets ?
            end
    * xyz 0 1
      C  0  0  0.00
```

```
    O  0  0  1.130
  *
```

From the example two additional inputs are necessary compared to a standard DFT calculation. The first is "`gracLB true`" in the `method` block to turn on the asymptotic correction. The second is "`ip 14.01`" also in the `method` block. *The second input is the ionization potential of the molecule in eV and is required for the correction to be meaningful. If there is no experimental value available you have to calculate the IP first by a ΔSCF calculation before you can run the TD-DFT!*

CAUTION:

- The corrected density functional *potentials* are not functional derivatives of well defined energy functionals. Therefore you should not rely on the total energies delivered by these functionals because these are not variational energies!

6.14.8.7 Potential Energy Surface Scans

ORCA allows the combination the scan feature with CIS or TD-DFT. This can be used to map out the excited state potential energy surfaces as a function of one- two- or three parameters. The output of the "trajectory" run automatically contains the excited state energies in addition to the ground state energy. For example consider the following simple job.

```
    ! def2-TZVPD
    %method scanguess pmodel  # this assignment forces a PModel guess at each step
                              # which is often better if diffuse functions are present
          end
    %cis  NRoots       7
          MaxDim       60
          end
    %paras rCO = 0.85,1.45,21;
          end
    * xyz 0 1
      O  0  0  0
      C  0  0  {rCO}
    *
```

The output file from this job contains the *total* energies (i.e. the ground state energy plus the excitation energy) for each excited state as a function of C-O bondlength as shown below. Howerver, the assignment of the individual states will change with geometry due to curve crossings. Thus, the state-to-state correlation must be worked out 'by hand'. These calculations are nevertheless very helpful in obtained at least a rough idea about excited state energy surfaces.
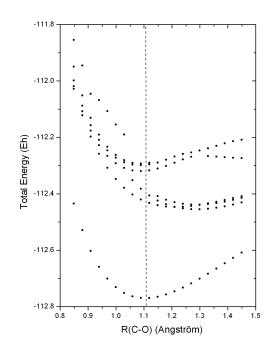
Figure 6.7: Result of a potential energy surface scan for the excited states of the CO molecule using the orca_cis module

6.14.8.8 Potential Energy Surface Scans along Normal Coordinates

The ground and excited state potential energy surfaces can also be mapped as a function of normal coordinates. The normal mode trajectory run is invoked by the keyword !MTR. In addition several parameters have to be specified in the block %mtr. The following example illustrates the use:

First you run a frequency job:

```
#
! BP86 def2-SV(P) def2-SVP/J TightScf AnFreq

* xyz 0 1
  C       0.000001     -0.000000     -0.671602
  C       0.000000      0.000000      0.671602
  H      -0.000000     -0.940772     -1.252732
  H      -0.000000     -0.940772      1.252732
  H      -0.000000      0.940772     -1.252732
  H      -0.000000      0.940772      1.252732
*
```

and then:

```
    ! BP86 def2-SV(P) def2-SVP/J TightScf MTR

    %tddft
      NRoots 3
      MaxDim 200
      triplets false
     end

    %mtr
        HessName "ethene.hess"
        modetype normal
        MList 9,13
        RSteps   4,5
        LSteps   4,5
        ddnc     1.0, 0.5
    end

    * xyz 0 1
      C        0.000001     -0.000000     -0.671602
      C        0.000000      0.000000      0.671602
      H       -0.000000     -0.940772     -1.252732
      H       -0.000000     -0.940772      1.252732
      H       -0.000000      0.940772     -1.252732
      H       -0.000000      0.940772      1.252732
    *
```

The `HessName` parameter specifies the name of the file which contains nuclear Hessian matrix calculated in the frequency run. The Hessian matrix is used to construct normal mode trajectories. The keyword `MList` provides the list of the normal modes to be scanned. The parameters `RSteps` and `LSteps` specify the number of steps in positive an negative direction along each mode in the list. In general, for a given set of parameters

```
    mlist m1,m2,...mn
    rsteps rm1,rm2,...rmn
    lsteps lm1,lm2,...lmn
```

the total number of the displaced geometries for which single point calculations will be performed is equal to $\prod_{m_i} (r_{m_i} + l_{m_i} + 1)$. Thus, in the present case this number is equal to $(4 + 4 + 1)(5 + 5 + 1) = 99$.

The ddnc parameter specifies increments δq_α for respective normal modes in the list in terms of dimensionless normal coordinates (DNC's). The trajectories are constructed so that corresponding normal coordinates are varied in the range from $-l_\alpha \delta q_\alpha$ to $r_\alpha \delta q_\alpha$. The measure of normal mode displacements in terms DNC's is appropriate choice since in spectroscopical applications the potential energy function U is usually expressed in terms of the DNC's. In particular, in the harmonic approximation $U(q_\alpha)$ has a very simple form around equilibrium geometry:

$$U = U_0 + \sum_{\alpha}^{3N-6} \frac{\hbar \omega_\alpha}{2} q_\alpha^2 \tag{6.220}$$

where ω_α is the vibrational frequency of the α-th mode.

Dimensionless normal coordinate q_α can be related to the vector of atomic Cartesian displacements $\delta \mathbf{X}$ as follows:

$$q_\alpha = \left(\frac{\omega_\alpha}{\hbar}\right)^{\frac{1}{2}} \sum_{k=1}^{3N} L_{k\alpha} \delta X_k \sqrt{M_k} \tag{6.221}$$

where $\{L_{k\alpha}\}$ is the orthogonal matrix obtained upon numerical diagonalization of the mass-weighted Hessian matrix, and $\mathbf{M}$ is the vector of atomic masses. Accordingly, the atomic Cartesian displacements corresponding to a given dimensionless normal coordinate q_α are given by:

$$\delta X_k = \left(\frac{\hbar}{\omega_\alpha}\right)^{\frac{1}{2}} L_{k\alpha} q_\alpha \left(M_k\right)^{-\frac{1}{2}} \tag{6.222}$$

Alternatively, it is possible to specify in the input the Cartesian increment for each normal mode. In such a case, instead of the **ddnc** parameter one should use the **dxyz** keyword followed by the values of Cartesian displacements, for example:

```
%mtr
   HessName "ethene.hess"
   modetype normal
   MList 9,13
   RSteps   4,5
   LSteps   4,5
   dxyz     0.01, 0.02 # increments in the Cartesian basis
                       # are given in angstrom units
end
```

For a given Cartesian increment $d_{X,\alpha}$ along the α–th normal mode the atomic displacements are calculated as follows:

$$\delta X_k = \frac{d_{X,\alpha}}{\|\mathbf{T}_\alpha\|} L_{k\alpha} \left(M_k\right)^{-\frac{1}{2}} \tag{6.223}$$

The vector $\mathbf{T}_\alpha$ in the Cartesian basis has components $T_{i\alpha} = L_{k\alpha} \left(M_k\right)^{-\frac{1}{2}}$ and length (norm) $\|\mathbf{T}_k\|$.

The increment length can also be selected on the basis of an estimate for the expected change in the total energy ΔE due to the displacement according to eq.6.118. The value of ΔE can be specified via the **EnStep** parameter:

```
%mtr
   HessName "ethene.hess"
   modetype normal
   MList 9,13
   RSteps   4,5
   LSteps   4,5
   EnStep   0.001, 0.001 # the values are given in Hartree
end
```

All quantum chemical methods have to tolerate a certain amount of numerical noise that results from finite convergence tolerances or other cutoffs that are introduced into the theoretical procedures. Hence, it is reasonable to choose ΔE such that it is above the characteristic numerical noise level for the given method of calculation.

At the beginning of the program run the following trajectory files which can be visualized in gOpenMol will be created:

- `BaseName.m9.xyz` and `BaseName.m13.xyz` contain trajectories alnong normal modes 9 and 13, respectively.

- `BaseName.m13s1.m9.xyz` - `BaseName.m13s5.m9.xyz` contain trajectories alnong normal mode 9 for different fixed displacements along mode 13, so that the file `BaseName.m13sn.m9.xyz` corresponds to the n-th step in the positive direction along mode 13.

- `BaseName.m13s-1.m9.xyz` - `BaseName.m13s-5.m9.xyz` contain trajectories alnong normal mode 9 for different fixed displacements along mode 13, so that the file `BaseName.m13s-n.m9.xyz` corresponds to the n-th step in the negative direction along mode 13.

- `BaseName.m9s1.m13.xyz` - `BaseName.m9s4.m13.xyz` contain trajectories alnong normal mode 13 for different fixed displacements along mode 9, so that the file `BaseName.m9sn.m13.xyz` corresponds to the n-th step in the positive direction along mode 9.

- `BaseName.m9s-1.m13.xyz` - `BaseName.m9s-4.m13.xyz` contain trajectories alnong normal mode 13 for different fixed displacements along mode 9, so that the file `BaseName.m9s-n.m13.xyz` corresponds to the n-th step in the negative direction along mode 9.

The results of energy single point calculations along the trajectories will be collected in files `BaseName.mtr. escf.S.dat` (for the SCF total energies) and files `BaseName.mtr.ecis.S.dat` (for the CIS/TDDFT total energies), where "S" in the suffix of `*.S.dat` filenames provides specification of the corresponding trajectory in the same way as it was done for the case of trajectory files `*.xyz` (e.g. S=''m9s-1.m13"). Likewise, the calculated along trajectories total energies will be collected in files `BaseName.mtr.emp2.S.dat` in the case of MP2 calculations, `BaseName.mtr.emdci.S.dat` (MDCI), `BaseName.mtr.ecasscf.S.dat` (CASSCF), `BaseName. mtr.emrci.S.dat` (MRCI).

Note, that in principle normal coordinate trajectories can be performed for an arbitrary number normal modes. This implies that in general trajectories will contain geometries which involve simultataneous displacement along several (>2) modes. However, trajectory files `*.xyz` and corresponding `*.dat` files will be generated only for the structures which are simultaneously displaced along not more than 2 normal coordinates.

6.14.8.9 Normal Mode Scan Calculations Between Different Structures

This type of job allows to map PES between two different structures as a function of normal coordinates. The H$_2$O molecule represent a trivial case which has formally 2 equivalent equilibrium structures which differ by angle H$_1$—O—H$_2$ ($103.5°$ and $256.5°$, respectively, as follows from the BP86/SV(P) calculations). In such a case the input for the nomal mode trajectory run would require the calculation of geometry difference between both structures in terms of the dimensionless normal coordinates. This can be done in `orca_vib` run as follows :

```
> orca_vib water.hess ddnc geom.xyz
```

The second parameter ddnc in the command line invokes the calculation of geometry difference in terms of the DNC's. Both structures are specified in the file `geom.xyz` which has a strict format:
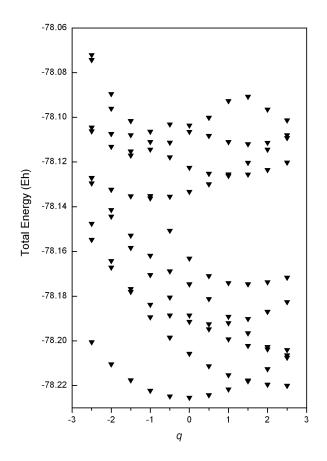
Figure 6.8: Result of a potential energy surface scan along C-C stretching normal coordinate (mode 13 in the present example) for the excited states of the ethene molecule using the orca_cis module.

```
   2   3
   0
             0.000000        0.000000         0.000000
             0.000000        0.607566         0.770693
             0.000000        0.607566        -0.770693
   1
             0.000000        0.000000         0.000000
             0.000000       -0.607566         0.770693
             0.000000       -0.607566        -0.770693
```

The first line of the input specifies the number of the structures and total number of atoms (2 and 3, respectively). Specification of each structure in sequence starts with a new line containing the number of the structure. The number 0 in the second line is used to denote the reference structure. Note that atomic coordinates should be given in angstrom units and in the same order as in the ORCA input for the frequency run from which the file `water.hess` was calculated.

At the end of the `orca_vib` run the file `geom.ddnc` is generated. It contains the geometry difference in terms of the dimensionless normal coordinates between the structures with nonzero numbers and the reference one in `geom.xyz` :

```
    1
    1 9
        0          0.000000
        1          0.000000
        2          0.000000
        3          0.000000
        4          0.000000
        5          0.000000
        6          9.091932
        7         -9.723073
        8          0.000000
```

The output file indicates that the structural difference occurs along 2 normal coordinates: 6 (bending mode) and 7 (totally symmetric O—H stretching mode). On the basis of the calculated displacement pattern the following input for the normal mode trajectory run between two structures can be designed:

```
    ! RKS BP86 SV(P) SV/J  RI TightScf MTR

    %mtr
       HessName "water.hess"
       modetype normal
       mlist 6,7
       rsteps   10,0
       lsteps   0, 10
       ddnc     0.9091932, 0.9723073
    end

    * xyz 0 1
    O        0.000000     0.000000     0.000000
    H        0.000000     0.607566     0.770693
    H        0.000000     0.607566    -0.770693
    *
```

Here the parameters RSteps, LSteps and ddnc are chosen in such a way that in the scan along modes 6 and 7 the corresponding dimensionless normal coordinates will be varied in the range $0 - 9.091932$ and $-9.723073 - 0$, respectively, in accordance with the projection pattern indicated in the file **geom.ddnc**. Note that normal modes are only defined up to an arbitrary choice of sign. Consequently, the absolute sign of the dimensionless displacements is ambiguous and in principle can vary in different **orca_vib** runs. It is important that the normal mode scan between different structures exemplified above is performed using the same sign of normal modes as in the calculation of normal mode displacements. This condition is fulfilled if the same normal modes are used in **orca_vib** run and trajectory calculation. Thus, since in **orca_vib** calculation normal modes are stored in **.hess** file it is necessary to use the same Hessian file in the trajectory calculation.

6.15 Excited States via ROCIS and DFT/ROCIS

The ORCA program package includes the *orca_rocis* module to perform configuration interaction with single excitations (CIS) calculations using a restricted open-shell Hartee-Fock (ROHF) reference function. It produces excitation energies, absorption energies and CD intensities. It was designed with the aim to reproduce and - even more importantly - reliably predict transition metal L-edges as observed in X-ray absorption spectroscopy (XAS).

6.15.1 General Use

In the present implementation the *orca_rocis* module is only able to perform CIS calculations on top of a high-spin ROHF reference function. All spins of the unpaired electrons have to be coupled ferrmoagnetically to give a total spin of S $= \frac{1}{2}$ N, where N is the number of unpaired electrons. Other ROHF functions such as Zerner's configuration averaged or spin averaged ROHF cannot be used as reference. The input for a high spin ROHF calculation is done in the `%scf` block.

```
%scf HFTyp ROHF #Flag for ROHF
     ROHF_Case HighSpin #selects the high-spin case
     ROHF_NEl[1] = 4 #the number of unpaired electrons
     end
```

In our experience ROHF calculations suffer a lot from convergence problems. UHF calculations generally exhibit better convergence properties. In most cases the quasi-restricted orbitals (qro's) of a UHF calculation resemble the ROHF orbitals. Thus the program features the ability to start a ROCIS calculation on top of a UHF calculation. It will automatically create the qro's and build the reference determinant with them. If one wants to avoid the (small) errors that are introduced by this procedure, one may take the qro's of a UHF calculation as starting orbitals for a subsequent ROHF calculation. Furthermore it is possible to invoke the *orca_rocis* module for closed shell molecules. The program will then perform a CI calculation with the provided RHF reference function. In this case it will yield the same result as the *orca_cis* program.

A number of basic variables in the `%rocis` block control the settings of the Davidson procedure that is used to solve the CI problem:

```
%rocis NRoots 6 #number of desired roots
       MaxDim 50 #maximum dimension of expansion #space
       ETol  1e-6 #energy convergence tolerance
       RTol  1e-6 #residual vector convergence  #tolerance
       MaxIter 35 #maxmimum number of iterations
       NGuessMat 500 #dimension of the guess matrix
       end
```

It is recommended to choose the `MaxDim` variable as a multiple of `NRoots`. The Default is the lowest possible choice. In general, by choosing `MaxDim` $\approx$ 5-10 times `NRoots` you will achieve favorable convergence by the cost of an increased disk space requirement. An increase of the `NGuessMat` variable to 1024 or 2048 will in most cases also lead to better convergence of the iterative CI procedure. The amount of output produced during the calculation is controlled via the `PrintLevel` variable

```
%rocis NRoots 3
       MaxDim 15
       PrintLevel 3
       end
```

Note, that this does not influence which spectra are calculated or printed. The absorption spectrum calculated on the basis of the pure dipole approximation for your calculation is always printed. In addition, it is possible to allow for electric quadrupole and magnetic dipole contributions to the absorption spectrum as well as to calculate the CD spectrum:

```
%rocis NRoots 6
       MaxDim 50
       DoQuad true  #invokes calculation of electric
                    #quadrupole and magnetic dipole
                    #contributions
       DoCD true    #invokes the calculation of the CD
                    #spectrum
       end
```

The printed spectra look like this:

```
-------------------------------------------------------------------------------
          ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS
-------------------------------------------------------------------------------
State   Energy  Wavelength    fosc        T2       TX       TY       TZ
        (cm-1)    (nm)                   (au**2)   (au)     (au)     (au)
-------------------------------------------------------------------------------
    1   2635.0   3795.1   0.000000001   0.00000   0.00001  -0.00001  0.00029
    2   4365.5   2290.7   0.000011416   0.00086   0.01200  -0.00864  0.02534
    3   4368.2   2289.3   0.000011174   0.00084  -0.02006   0.01442  0.01523
    4   5977.9   1672.8   0.000093897   0.00517  -0.04164  -0.05863  0.00000
    5  65245.3    153.3   0.027669631   0.13961  -0.20555  -0.31203 -0.00023


-------------------------------------------------------------------------------
          ABSORPTION SPECTRUM VIA TRANSITION VELOCITY DIPOLE MOMENTS
-------------------------------------------------------------------------------
State   Energy  Wavelength    fosc        P2       PX       PY       PZ
        (cm-1)    (nm)                   (au**2)   (au)     (au)     (au)
-------------------------------------------------------------------------------
    1   2635.0   3795.1   0.000000085   0.00000  -0.00000   0.00000 -0.00004
    2   4365.5   2290.7   0.001777771   0.00005  -0.00315   0.00223 -0.00618
    3   4368.2   2289.3   0.001850956   0.00006   0.00526  -0.00372 -0.00371
    4   5977.9   1672.8   0.003237195   0.00013   0.00667   0.00937  0.00000
    5  65245.3    153.3   0.057301314   0.02555   0.08779   0.13358  0.00010


-----------------------------------------------------------------
                         CD SPECTRUM
-----------------------------------------------------------------
State   Energy Wavelength     R        MX       MY       MZ
        (cm-1)    (nm)     (1e40*sgs)  (au)     (au)     (au)
```

```
-------------------------------------------------------------------
   1    2635.0    3795.1      0.00007  -0.00511  -0.01539   0.00021
   2    4365.5    2290.7     10.02484   0.57434  -0.40490   0.42899
   3    4368.2    2289.3    -10.03730   0.34432  -0.24269  -0.71470
   4    5977.9    1672.8      0.01537  -0.00033  -0.00032  -0.00286
   5   65245.3     153.3     -0.00865   0.00004   0.00003  -0.00005


-----------------------------------------------------------------------------------------------------------
            COMBINED ELECTRIC DIPOLE + MAGNETIC DIPOLE + ELECTRIC QUADRUPOLE SPECTRUM
-----------------------------------------------------------------------------------------------------------

State   Energy Wavelength   D2        M2        Q2        D2+M2+Q2          D2/TOT    M2/TOT    Q2/TOT
        (cm-1)  (nm)                  (*1e6)    (*1e6)
-----------------------------------------------------------------------------------------------------------
   1    2635.0    3795.1  0.00000   0.00011   0.00000   0.00000000080469   0.86010   0.13938   0.00052
   2    4365.5    2290.7  0.00001   0.47866   0.00000   0.00001189497194   0.95976   0.04024   0.00000
   3    4368.2    2289.3  0.00001   0.48629   0.00000   0.00001166062671   0.95830   0.04170   0.00000
   4    5977.9    1672.8  0.00009   0.00001   0.00001   0.00009389664707   1.00000   0.00000   0.00000
   5   65245.3     153.3  0.02767   0.00000   0.06183   0.02766969236508   1.00000   0.00000   0.00000



-----------------------------------------------------------------------------------------------------------
       COMBINED ELECTRIC DIPOLE + MAGNETIC DIPOLE + ELECTRIC QUADRUPOLE SPECTRUM (origin adjusted)
-----------------------------------------------------------------------------------------------------------

State   Energy Wavelength   D2        M2        Q2        D2+M2+Q2          D2/TOT    M2/TOT    Q2/TOT
        (cm-1)  (nm)                  (*1e6)    (*1e6)
-----------------------------------------------------------------------------------------------------------
   1    2635.0    3795.1  0.00000   0.00000   0.00000   0.00000000069409   0.99716   0.00016   0.00268
   2    4365.5    2290.7  0.00001   0.38277   0.00039   0.00001179947536   0.96753   0.03244   0.00003
   3    4368.2    2289.3  0.00001   0.36798   0.00045   0.00001154275975   0.96808   0.03188   0.00004
   4    5977.9    1672.8  0.00009   0.00000   0.00001   0.00009389663928   1.00000   0.00000   0.00000
   5   65245.3     153.3  0.02767   0.00003   0.06176   0.02766969232228   1.00000   0.00000   0.00000
```

If calculations on large molecules are conducted, the integral transformation will be the most time-consuming part. Therefore it is strongly recommended to use the resolution of the identity (RI) approximation in those cases. It effectively reduces the computational costs of the transformation step by only introducing minor errors to the calculation. It has to be kept in mind that in order to keep the introduced errors small, one has to provide a reasonable auxiliary basis sets along with your normal basis set input.

```
! TZVP TZV/C TightSCF SlowConv

%SCF  HFTyp ROHF
ROHF_Case HighSpin
ROHF_Nel[1] = 1
End

%ROCIS NROOTS 5
MaxDim 35
DoRI true # invokes the RI approximation
DoQuad true
end
```

```
    * xyz 0 2
    N 0 0 0
    O 0 0 1.15
    *
```

The *orca_rocis* module provides two ways of choosing the orbital excitation space: By orbital energy or orbital number. In the former case an energy window has to be specified and the program will then take all orbitals, whose orbital energies lie within this window, into account. Note, that one actually has to define two orbital windows: One for the donor and the second for the acceptor orbital. The input of the windows is done as an array: The first two numbers define the donor space while the last two numbers define the acceptor space.

```
    %rocis NRoots 3
           MaxDim 15
           EWin = -5,5,-5,5
           end
```

The default is to keep core orbitals and very high lying virtual orbitals out of their respective orbital excitation spaces. Since these orbitals span a space that is usually not reachable with regular UV/Vis spectroscopy, this is a reasonable approximation. One has to keep in mind that an orbital energy window makes only sense if the orbitals used in the calculation have a well-defined orbital energy. As a consequence one cannot use an orbital energy window for a calculation with localized orbitals. The second way to specify the excitation space is by orbital numbering.

```
    %rocis NRoots 3
           MaxDim 15
           OrbWin = 1,13,9,22
           end
```

In restricted calculations only one set of spatial orbitals is created. Hence it is not necessary to provide orbital windows for alpha and beta electrons separately. Of course, only doubly or singly occupied orbitals can act as donor orbitals and only singly and nonoccupied orbitals can act as acceptor orbitals. The program recognizes nonoccupied orbitals in the donor space and doubly occupied orbitals in the acceptor space and removes both.

The many-electron expansion space of a ROCIS calculation in ORCA is divided into five classes. Using second quantized replacement operators $E_p^q = \hat{a}_{q\alpha}^{\uparrow} \hat{a}_{p\alpha} + \hat{a}_{q\beta}^{\uparrow} \hat{a}_{p\beta}$ they take the form [315].

$$
\begin{aligned}
|\Phi_i^s\rangle &= E_i^s |0\rangle \\
|\Phi_s^a\rangle &= E_s^a |0\rangle \\
|\Phi_i^a\rangle &= \tfrac{1}{\sqrt{2}} E_i^a |0\rangle \\
|\Phi_{ti}^{as}\rangle &= E_t^a E_i^s |0\rangle \\
|\Phi_{ti}^{as}\rangle &= \tfrac{1}{\sqrt{6}} \left(E_i^a - 2 E_s^a E_i^s \right) |0\rangle
\end{aligned}
\tag{6.224}
$$

The orbital label 'i' denotes a doubly occupied orbital, 's' and 't' refer to singly occupied orbitals and orbital label 'a' corresponds to a virtual orbital. The form of the excitation classes ensures that all excited states are eigenfunctions of the $\hat{S}^2$-operator and have the same total spin S as the electronic ground state. Each of the five excitation classes can be switched on or off manually.

```
    %rocis NRoots 3
          MaxDim 15
          Do_is true #Include DOMO->SOMO excitations
          Do_sa true #Include SOMO->Virtual excitations
          Do_ia true #Include DOMO->Virtual excitations
          Do_ista true #Include DOMO->SOMO coupled to
    #SOMO->Virtual excitations with #s not equal t
          Do_isa true #Include DOMO->SOMO coupled to
    #SOMO->Virtual excitations with  #s = t
    #------------------------------
    #by default all switches for the
    #excitation classes are set to
    #'true'
    #------------------------------
    end
```

Formally, the $|\Phi_{ti}^{as}\rangle$ and $|\Phi_{ti}^{at}\rangle$ excitation classes can be regarded as double excitations. When the program finished the ROCIS calculation it gives the excitation energy together with the composition for each root. According to the number of labels of the respective functions $|\Phi\rangle$, contributions from excited configuration state functions belonging to the different excitation classes are given by two, three or four numbers.

```
    STATE   5   Exc. Energy: 297.279mEh   8.089eV          65245.3cm**-1
          47->50                 :   0.2196
          47->51                 :   0.0138
          37->50                 :   0.1165
          41->50                 :   0.0960
          38->46   ;   47->50    :   0.0103
          37->46  ->50           :   0.0150
          37->47  ->50           :   0.0938
          37->48  ->50           :   0.0179
          37->49  ->50           :   0.0179
          41->46  ->50           :   0.0174
          41->47  ->50           :   0.0585
          41->48  ->50           :   0.0213
          41->49  ->50           :   0.0211
```

Furthermore the *orca_rocis* module is able to calculate the effect of spin-orbit coupling (SOC) on the calculated ground and excited states. It introduces SOC in the framework of quasi-degenerate perturbation theory (QDPT). The SOC Hamiltonian is diagonalized in the basis of the calculated ROCIS states $|\Psi_I^{SM}\rangle$, where I is the root label and S and M are the spin and magnetic spin quantum numbers, respectively [316], [315].

```
    %rocis NRoots 3
          MaxDim 15
          OrbWin = 1, 3 ,9 ,22
```

```
        SOC true #invokes the calculation of  #SOC effects
        SOCTEMP 10  #temperature for SOC  #corrected spectra in Kelvin
        end
```

After the SOC calculation the program will produce additional spectra for the SOC corrected results. The spectra contain transitions from the 2S+1 lowest lying states into all excited states, where S is the spin quantum number of the electronic ground state. These 2S+1 lowest states may be split up in the order of 1-100 cm^{-1}. Due to the small magnitude of the splitting, all of the 2S+1 states can be significantly populated even at low temperatures. Experimentally, the intensity of a given transition is dependent on the population of the corresponding initial state. With the SOCTemp keyword the population of the theoretically calculated states can be manipulated by the varying the fictive temperature of the system. It has to be mentioned that the electric quadrupole transitions between spin-orbit coupled states are not well defined and are likely to give unreasonable results. Hence it is recommended to use the DoQuad keyword only for calculations that do not include SOC.

```
--------------------------------------------------------------------------------
SPIN ORBIT CORRECTED ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS
--------------------------------------------------------------------------------
States    Energy  Wavelength   fosc        T2        TX        TY        TZ
          (cm-1)    (nm)                  (au**2)    (au)      (au)      (au)
--------------------------------------------------------------------------------
 0  1       5.6       0.0    0.000000000  0.00000   0.00003   0.00002   0.00000
 0  2       6.2       0.0    0.000000000  0.00000   0.00000   0.00000   0.00005
 0  3      23.7  422287.3    0.000000000   0.00000   0.00000   0.00000   0.00000
 0  4      23.7  421562.8    0.000000000   0.00000   0.00018   0.00025   0.00000
 0  5    2621.7    3814.3    0.000000000  0.00000   0.00000   0.00001   0.00005
 0  6    2622.0    3813.9    0.000000000  0.00000   0.00003   0.00012   0.00000
 0  7    2634.7    3795.5    0.000000095  0.00002   0.00388   0.00273   0.00049
 0  8    2634.9    3795.2    0.000000103  0.00002   0.00039   0.00027   0.00495
 0  9    2639.5    3788.6    0.000000001  0.00000   0.00001   0.00001   0.00036
 0 10    4223.6    2367.6    0.000000103  0.00002   0.00043   0.00029   0.00390
 0 11    4223.9    2367.5    0.000000120  0.00002   0.00348   0.00236   0.00046
 0 12    4296.3    2327.6    0.000000696  0.00010   0.00562   0.00842   0.00000
 0 13    4357.6    2294.8    0.000000002  0.00000   0.00001   0.00001   0.00049
 0 14    4418.1    2263.4    0.000005778  0.00083   0.00653   0.00468   0.02762
 0 15    4422.1    2261.4    0.000005517  0.00079   0.02184   0.01559   0.00832
 0 16    4488.2    2228.0    0.000000001  0.00000   0.00004   0.00006   0.00038
 0 17    4524.2    2210.3    0.000000001  0.00000   0.00030   0.00018   0.00000
 0 18    4597.2    2175.2    0.000000027  0.00000   0.00023   0.00016   0.00191
 0 19    4597.4    2175.2    0.000000051  0.00001   0.00213   0.00153   0.00023
 0 20    6043.6    1654.6    0.000047989  0.00502   0.04104   0.05779   0.00000
 0 21    6049.5    1653.0    0.000000014  0.00000   0.00109   0.00057   0.00001
 0 22    6051.3    1652.5    0.000000021  0.00000   0.00001   0.00004   0.00150
 0 23    6069.7    1647.5    0.000000000  0.00000   0.00005   0.00007   0.00000
 0 24    6069.9    1647.5    0.000000028  0.00000   0.00098   0.00138   0.00000
 0 25   65281.7     153.2    0.014223474  0.13787   0.20423   0.31010   0.00023
 0 26   65281.7     153.2    0.000000035  0.00000   0.00032   0.00048   0.00011
 0 27   65281.7     153.2    0.000009000  0.00009   0.00522   0.00774   0.00001
 0 28   65281.7     153.2    0.000007207  0.00007   0.00460   0.00698   0.00000
 0 29   65281.7     153.2    0.000047448  0.00046   0.01179   0.01791   0.00001
 1  2       0.6       0.0    0.000000000  0.00000   0.00001   0.00001   0.00000
 1  3      18.1  553477.5    0.000000000   0.00000   0.00000   0.00000   0.00009
 1  4      18.1  552233.6    0.000000000   0.00000   0.00006   0.00004   0.00000
 1  5    2616.1    3822.5    0.000000063  0.00001   0.00006   0.00003   0.00261
```

```
    1  6    2616.4   3822.1   0.000000060   0.00001   0.00211   0.00144   0.00006
    1  7    2629.1   3803.6   0.000000143   0.00002   0.00225   0.00321   0.00003
    1  8    2629.3   3803.3   0.000000002   0.00000   0.00015   0.00025   0.00040
    1  9    2633.9   3796.7   0.000000271   0.00003   0.00011   0.00008   0.00538
    1 10    4218.0   2370.8   0.000000005   0.00000   0.00031   0.00046   0.00019
```

If the `PrintLevel` value is set to 3 or higher, the program will print out the composition of the SOC corrected states in the basis of states $\left| \Psi_I^{SM} \right\rangle$.

```
    Eigenvectors of SOC calculation:
    the threshold for printing is: 0.010000
          weight : Root Spin Mult
    State 0:       0.00 cm**-1     0.00000 eV
        0.378045  :  0    2    2
        0.235825  :  0    2    0
        0.378045  :  0    2   -2

    State 1:       5.61 cm**-1     0.00070 eV
        0.496236  :  0    2    2
        0.496236  :  0    2   -2

    State 2:       6.20 cm**-1     0.00077 eV
        0.496291  :  0    2    1
        0.496291  :  0    2   -1
```

Further details of the SOC calculation such as the procedure of SOC integral calculation can be controlled via the `%rel` block (section 6.12.

6.15.2 Transition Metal L-Edges with ROCIS or DFT/ROCIS

The *orca_rocis* program was designed to calculate transition metal L-edge spectra of large molecules as they are observed in X-ray absorption spectroscopy (XAS). An L-edge results when an electron is promoted from the 2p shell of a transition metal ion into the valence d-shell by an X-ray photon. Strong spin-orbit coupling in the 2p shell and p-d coupling phenomena complicate the interpretation and even more so the prediction of these spectra. It has to be kept in mind that the present program applies a variety of approximations which might lead to observable deviations from experimentally determined spectra. However, we believe that the results obtained from the program are in general qualitatively correct and in most cases accurate close to the experimental uncertainty. In cases where quantitative accuracy is not met, the provided results might still give some insight into the mechanisms of intensity distribution in the spectra.

The special input structure for orbital windows described in 6.15.1 allows the user to restrict the donor orbital space to the transition metal 2p shell. The acceptor orbital space is the same as in regular UV/Vis spectroscopy. It should include all singly occupied molecular orbitals and as many virtual orbitals as one can afford in the calculation. The number of roots should be chosen large enough so that at least all 2p-3d single excitations are calculated. In many cases even more roots are required since doubly excited or charge transfer states may become important. Moreover the strong SOC apparent in the 2p shell of transition metal ions necessitates the additional calculation of excited states with a total spin of S' = S + 1 and S' = S -1 where S is the total spin of the electronic ground state. Accordingly four

additional excitation classes introduce excited configuration state functions with a lower and higher spin multiplicity. They feature the second quantized spin raising and lowering operators $\hat{S}_{pq}^+ = \hat{a}_{q\alpha}^\uparrow \hat{a}_{p\beta}$, $\hat{S}_{pq}^- = \hat{a}_{q\beta}^\uparrow \hat{a}_{p\alpha}$.

$$
\left.
\begin{aligned}
\left| \Phi_i^{(t-)} \right\rangle &= \sqrt{\tfrac{2S'+1}{2S'+2}} S_{ti}^- \left| 0 \right\rangle - \sum_{u \neq t}^{SOMO} \tfrac{1}{\sqrt{2S'+1}} \tfrac{1}{\sqrt{2S'+2}} S_{uu}^- E_i^t \left| 0 \right\rangle \\
\left| \Phi_i^{(t-)} \right\rangle &= \sqrt{\tfrac{2S'+1}{2S'+2}} S_{ti}^- \left| 0 \right\rangle - \sum_{u \neq t}^{SOMO} \tfrac{1}{\sqrt{2S'+1}} \tfrac{1}{\sqrt{2S'+2}} S_{uu}^- E_i^t \left| 0 \right\rangle \\
\left| \Phi_i^{(a-)} \right\rangle &= \sqrt{\tfrac{2S'+1}{2S'+3}} S_{ai}^- \left| 0 \right\rangle - \sum_{t}^{SOMO} \sqrt{\tfrac{(S'+1)^2 - S'^2}{(S'+1)(2S'+3)}} \tfrac{1}{\sqrt{2(2S'+2)}} S_{tt}^- E_i^a \left| 0 \right\rangle \\
&+ \sum_{t,u \neq t}^{SOMO} \sqrt{\tfrac{2}{(2S'+2)(2S'+3)}} \sqrt{\tfrac{1}{(2S'+2)2(2S'+1)}} S_{tt}^- S_{uu}^- S_{ai}^+ \left| 0 \right\rangle
\end{aligned}
\right\} S' = S - 1
$$

$$
\left. \left| \Phi_i^{a+} \right\rangle = S_{ai}^+ \left| 0 \right\rangle \right\} S' = S + 1
$$

(6.225)

Inclusion of configuration state functions with higher or lower multiplicity is invoked with the keywords `DoLowerMult` and `DoHigherMult`, respectively.

```
%rocis NRoots 20
       MaxDim 50
       SOC true
       DoRI true
       PrintLevel 3
       DoLowerMult true #Invokes a CI calculation  #with S'=S-1
       DoHigherMult true #Invokes a CI calculation  #with S'=S+1
       OrbWin = 6,8,0,2000
       end
```

The program will conduct a separate Davidson procedure for each multiplicity. Subsequently it gives the excitation energies and compositions of the calculated excited states for all included multiplicities. After all CI calculations are finished, the program gives a list of all calculated roots with their excitation energies and their multiplicities. It is this number that will be referred to as label 'I' in the decomposition of spin-orbit coupled states in the basis $\left| \Psi_I^{SM} \right\rangle$. It is very important to note, that when states with different multiplicities are calculated this number might deviate from the number that appears in the respective CI part of the output. If one gets confused about the numbering of the states, the state energies might act as a guideline through the output of the program.

Without SOC the spin exclusion rule applies which means that only excited states with a total spin equal to the ground state spin (S' = S) give rise to non-vanishing intensities. Hence, only these transitions are listed in the spectra before SOC.

```
        -------------------------------------------------------------------------
        ROOT    Mult    Excitation energy[Eh]        [cm-1]        [eV]
        -------------------------------------------------------------------------
          0       5         0.00000000               0.00         0.000
          1       5        26.24822856           5760820.28     714.251
          2       5        26.24833619           5760843.90     714.254
          3       5        26.27159871           5765949.43     714.887
          4       5        26.27982129           5767754.08     715.110
          5       5        26.30321870           5772889.22     715.747
          6       5        26.30458669           5773189.46     715.784
          7       5        26.33143414           5779081.79     716.515
          8       5        26.33600432           5780084.83     716.639
```

```
   9        5            26.33865219                  5780665.97          716.711
  10        5            26.34522494                  5782108.52          716.890
  11        5            26.34577552                  5782229.36          716.905
  12        5            26.35183534                  5783559.34          717.070
  13        3            26.42121780                  5798787.03          718.958
  14        3            26.42122881                  5798789.45          718.958

...

  42        7            27.22926558                  5976133.02          740.946
  43        7            27.23201078                  5976735.52          741.021
  44        7            27.23280499                  5976909.83          741.042
  45        7            27.23594814                  5977599.67          741.128
  46        7            27.23865050                  5978192.77          741.201
  47        7            27.26590445                  5984174.32          741.943
  48        7            27.26597947                  5984190.78          741.945
  49        7            27.26604364                  5984204.87          741.947
  50        3            27.29447169                  5990444.10          742.720
  51        3            27.30121861                  5991924.88          742.904
  52        3            27.30655497                  5993096.08          743.049
  53        3            27.30685328                  5993161.55          743.057
  54        3            27.31274496                  5994454.62          743.218
  55        7            27.52164817                  6040303.58          748.902
  56        7            27.52433114                  6040892.42          748.975
  57        7            27.52448641                  6040926.50          748.979
  58        7            27.53903479                  6044119.50          749.375
  59        7            27.53935644                  6044190.10          749.384


-----------------------
ROCIS-EXCITATION SPECTRA
-----------------------


NOTE: At this point no SOC is included!!!
Hence only transitions to states with the same spin multiplicity
as the ground state are observed!!!


Center of mass = ( -0.0011, -0.0021,  0.0000)
Calculating the Dipole integrals                ... done
Transforming integrals                          ... done
Calculating the Linear Momentum integrals       ... done
Transforming integrals                          ... done


-------------------------------------------------------------------------------
        ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS
-------------------------------------------------------------------------------
State   Energy  Wavelength    fosc       T2        TX       TY       TZ
        (cm-1)    (nm)                  (au**2)    (au)     (au)     (au)
-------------------------------------------------------------------------------
  1 5760820.3     1.7   0.000985130   0.00006   0.00612  -0.00434  0.00011
  2 5760843.9     1.7   0.000777158   0.00004  -0.00008   0.00006  0.00666
  3 5765949.4     1.7   0.000000036   0.00000   0.00000   0.00001 -0.00004
  4 5767754.1     1.7   0.000007564   0.00000   0.00033   0.00057 -0.00000
  5 5772889.2     1.7   0.025379335   0.00145  -0.00031   0.00021 -0.03804
  6 5773189.5     1.7   0.026898175   0.00153   0.03203  -0.02254 -0.00039
  7 5779081.8     1.7   0.000000323   0.00000  -0.00006  -0.00009 -0.00008
  8 5780084.8     1.7   0.001711738   0.00010  -0.00572  -0.00805  0.00001
```

```
 9 5780666.0      1.7    0.113054940    0.00644   -0.04616   -0.06564   -0.00001
10 5782108.5      1.7    0.151287595    0.00861    0.00073   -0.00052    0.09281
11 5782229.4      1.7    0.147199895    0.00838    0.07488   -0.05266   -0.00088
12 5783559.3      1.7    0.000000026    0.00000    0.00001   -0.00001    0.00004
28 5960986.7      1.7    0.004292708    0.00024   -0.00881   -0.01263   -0.00000
29 5963084.1      1.7    0.001638281    0.00009   -0.00774    0.00553    0.00006
30 5963136.7      1.7    0.001369356    0.00008   -0.00005    0.00003   -0.00869
31 5963484.9      1.7    0.000935993    0.00005    0.00415    0.00587   -0.00000
32 5968477.0      1.7    0.000661255    0.00004    0.00493   -0.00349   -0.00007
33 5968705.6      1.7    0.000607238    0.00003    0.00006   -0.00004    0.00579
35 5970943.7      1.7    0.000000001    0.00000    0.00000    0.00000   -0.00001
```

After calculation of SOC in the basis of all calculated ROCIS roots, the program prints out the composition of the spin-orbit coupled states (if `PrintLevel` >2) and the corresponding absorption spectrum.

```
Eigenvectors of SOC calculation:
the threshold for printing is: 0.010000
        weight : Root Spin Mult
State 0:       0.00 cm**-1    0.00000 eV
    0.129027   :  0    2    2
    0.741116   :  0    2    0
    0.129027   :  0    2   -2
```

```
--------------------------------------------------------------------------------
SPIN ORBIT CORRECTED ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS
--------------------------------------------------------------------------------

States    Energy   Wavelength    fosc        T2        TX        TY        TZ
          (cm-1)   (nm)                     (au**2)    (au)      (au)      (au)
--------------------------------------------------------------------------------
0  1       0.0      0.0     0.000000000    0.00000   0.00000   0.00000   0.00000
0  2       0.8      0.0     0.000000000    0.00000   0.00000   0.00000   0.00000
0  3       0.8      0.0     0.000000000    0.00000   0.00000   0.00000   0.00000
0  4       1.0      0.0     0.000000000    0.00000   0.00000   0.00000   0.00000
0  5 5729330.4      1.7     0.000080556    0.00002   0.00013   0.00009   0.00464
0  6 5729330.4      1.7     0.000096984    0.00003   0.00415   0.00295   0.00013
0  7 5731365.3      1.7     0.000000001    0.00000   0.00001   0.00000   0.00000
0  8 5731365.4      1.7     0.000000000    0.00000   0.00000   0.00000   0.00001
0  9 5733452.5      1.7     0.000058329    0.00002   0.00323   0.00227   0.00004
0 10 5733477.2      1.7     0.000066389    0.00002   0.00003   0.00002   0.00421
0 11 5734964.4      1.7     0.000000034    0.00000   0.00005   0.00007   0.00004
0 12 5737151.2      1.7     0.000047769    0.00001   0.00208   0.00291   0.00000
```

With the aid of the orca_mapspc program it is possible to extract a `.plt` file from the printed spectra, which then can be used to generate a plot of the intensity vs the excitation energy. The orca_mapspc program applies Gaussian type lineshape functions to the calculated transitions with a user-defined FWHM. One has to provide several informations for the program such as the name of the output file, the type of spectrum you wish to plot, the energy range and the like. It is invoked in the command line and the parameters are given as arguments:

```
orca_mapspc FeIICl4.out socabs -eV -w1 -n3000 -x0710 -x1740
```

The first argument has to be the output file of your calculation followed by the type of spectrum that should be plotted. In the case of transition metal L-edges it is an absorption spectrum after the SOC correction. The arguments

'-eV' (use electron Volt as energy unit), '-w1' (FWHM = 1eV), '-n3000' (use 3000 grid points), '-x0710' and '-x1740' (energy range: 710 to 740 eV) have to be adapted to the specific calculation. As a result, one obtains a `.plt` and a `.stk` file. The `.plt` file contains five columns. In the first column one finds the energy and in the second the total intensity. Columns three to five contain the x-,y- and z-components of the transition moment. Note, that the distribution of the transition moment among its spatial components depends on the orientation of your molecular axis system. The `.stk` file contains a list of all transitions with their respective transition energy and intensity. A more detailed description of the orca_mapspc program and its usage can be found in chapter 6.24.1.
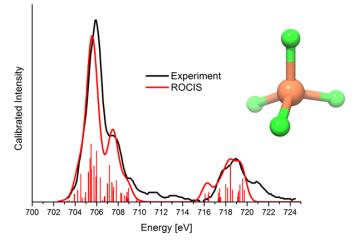
Figure 6.9: Comparison of the experimentally observed (black) and calculated ROCIS (red) Fe L-edge of $[FeCl_4]^{2-}$. The red bars highlight the contribution of individual states to the total spectrum. The calculation was performed using the TZVP basis set.

For many transition metal compounds the description of the electronic ground and excited states by Hartree-Fock theory and CIS is of rather poor quality. Especially covalency and relative spin state energetics are not reproduced correctly. This in turn might lead to wrong intensity distributions in the calculated L-edge spectra. In the majority of these cases the quality of the description and hence the predicted L-edge spectra can be significantly improved with the DFT/ROCIS method [315]. It features the usage of a restricted open-shell Kohn-Sham matrix as reference and also uses the DFT orbitals for setting up the excited configuration state functions in the CI expansion. The two electron integrals that include the DFT orbitals are scaled according to their nature and their position in the CI matrix by the parameters c_1, c_2 and c_3. They all lie in the interval [0;1]. Parameters c_1 and c_2 scale coulomb- and exchange- like terms in the diagonal part of the CI matrix, whereas c_3 reduces the size of all off-diagonal elements of the CI matrix. For example:

$$
\begin{aligned}
H_{ia,ia}^{DFT/ROCIS} &= F_{aa}^{C(KS)} - F_{ii}^{C(KS)} - c_1\,(ii|aa) + 2c_2\,(ia|ia) \\
H_{ia,jb}^{DFT/ROCIS} &= c_3 \left\{ \delta_{ij} F_{ab}^{C(KS)} - \delta_{ab} F_{ji}^{C(KS)} - (ij|ab) + 2\,(ia|jb) \right\}
\end{aligned}
\tag{6.226}
$$

The three default parameters $c_1 = 0.21$, $c_2 = 0.49$ and $c_3 = 0.29$ have been optimized for a test set of molecules and their excited states on a B3LYP/def2-TZVP(-f) level of theory but can be freely chosen [315]. It is most likely that for a different combination of test molecules, functional and basis set, a different set of parameters gives better results. Since the parameters are chosen with regard of a good "balance" between orbital energies, coulomb and exchange integrals, a new set of parameters should at least crudely resemble their relative proportions.

```
! def2-tzvp(-f) def2-TZV/C TightSCF B3LYP Grid4 NoFinalGrid
```

```
%ROCIS NRoots 20
       MaxDim 200
       DoRI true
       SOC true
       DoHigherMult true
       PrintLevel 3
       OrbWin = 5,7,50,60
       DoDFTCIS true #switches on the  #DFT/ROCISS method
       DFTCIS_c = 0.21, 0.49, 0.29 #Array input of the  #three parameters
       end
```
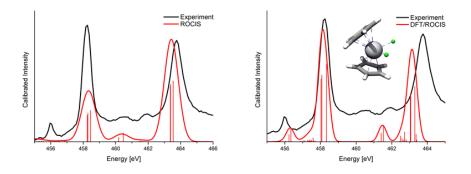

Figure 6.10: Comparison of the experimentally observed (black) and calculated (red) Ti L-edge of [Cp$_2$TiCl$_2$]. The red bars highlight the contribution of the individual states to the total spectrum. The pure ROCIS method (left) predicts a wrong L$_3$-L$_2$ intensity ratio and strongly overestimates the splitting of the satellite features to the main bands. Better results are obtained with the DFT/ROCIS method (right).

6.15.3 Keyword List

```
%rocis
#------------------------------------------------------------
# GENERAL KEYWORDS
#------------------------------------------------------------
       NRoots 3           #The number of desired roots
       MaxDim 15          #Maximum dimension of the
                          #expansion space in the Davidson procedure
       MaxIter 35         #Maximum CI Iterations
       NGuessMat 500      #The dimension of the guess matrix
       ETol 1e-6          #Energy convergence tolerance
       RTol 1e-6          #Residual Convergence tolerance
       MaxCore 2000       #Maximum memory used during the calculation in MB
       EWin= -5,5,-5,5    #Energy Window that defines orbital excitation space
       OrbWin=6,8,0,2000  #Orbital Window that defines orbital excitation space
                          #(overrides EWin)
       DoRI false         #Switch for the RI approximation
       DoLoc false        #Switch for localization of Donor orbital space
       LocMet PipekMezey  #chooses the localization method:
```

```
                                #PipekMezey or FosterBoys.
                                #Abbreviations "PM" and "FB"
                                #are equivalent to full names.
        SOC false               #Switch for inclusion of SOC
        SOCTemp 10              #The fictive temperature for the
                                #SOC corrected spectra
        DoDFTCIS false          #Switch for the DFT/ROCIS method
        DFTCIS_C = 0.21, 0.49, 0.29 #Array Input of the
                                #three DFT/ROCIS parameters


    #-----------------------------------------------------------
    # FLAGS FOR EXCITATION SPACES
    #-----------------------------------------------------------
        Do_is true              #Include DOMO->SOMO excitations
        Do_sa true              #Include SOMO->Virtual excitation
        Do_ia true              #Include DOMO->Virtual excitations
        Do_ista true            #Include DOMO->SOMO excitations
                                #coupled to SOMO->Virtual
                                #excitations with s not equal t
        Do_isa true             #Include DOMO->SOMO excitations
                                #coupled to SOMO->Virtual
                                #excitations with s = t


        DoLowerMult false       #Switch for excitation with S'=S-1
        Do_LM_is true           #Include DOMO->SOMO excitations
                                #with S'=S-1
        Do_LM_sa true           #Include SOMO->Virtual excitations
                                #with S'=S-1
        Do_LM_ia true           #Include DOMO->Virtual excitations
                                #with S'=S-1


        DoHigherMult false      #Switch for DOMO->Virtual
                                #excitations with S'=S+1

    #-----------------------------------------------------------
    OUTPUT KEYWORDS
    #-----------------------------------------------------------
        PrintLevel 3            #Controls the amount of output
                                #produced during the calculation
        DoCD false              #Invokes the calculation of CD spectra
        DoQuad false            #Invokes the calculation of
                                #electric quadrupole and magnetic
                                #dipole contributions to the
                                #calculated spectra
        PlotDiffDens = 1,2      #Array input for plotting
                                #difference densities of CI roots
                                #1 and 2 to the ground state.
        PlotSOCDiffDens=1,2     #Array input for plotting
                                #difference densities of SOC
                                #states 1 and 2 to the ground state
        Weight_Thresh 0.01      #Threshold for contributions to CI
                                #and SOC states to be printed
```

6.16 Excited States via EOM-CCSD

The EOM-CCSD routine is part of the *orca_mdci* module of the ORCA program package. It is called after a successful coupled cluster calculation if the appropriate flags and the number of roots have been set. In the following chapter the general program flow and all input parameters of the EOM routine will be described in detail (for typical use, see 5.6).

6.16.1 General Description

The EOM wave function is parametrized in the following manner

$$\mathcal{R}|\Psi_{CC}\rangle, \tag{6.227}$$

i.e. via the action of a linear excitation operator $\mathcal{R}$ on the coupled cluster ground state wave function Ψ_{CC}. In the beginning, the coupled cluster amplitudes are given from the previous CCSD calculation, and our task is to obtain $\mathcal{R}$. Note that since the CC Hamiltonian is nonsymmetric, a left hand solution ($\mathcal{L}$) would also be needed to evaluate properties. For excitation energies however, it is enough to obtain the right hand solutions ($\mathcal{R}$). In principle this is done by building the Hamiltonian and diagonalizing it in order to obtain energy expectation values.

In practice the size of the CCSD Hamiltonian matrix is prohibitively large, and thus various methods have been devised to obtain its lowest few eigenvalues and eigenstates. One of the most popular of these is the Davidson method, which can be summarized as follows:

- Construct an initial guess of orthogonal trial vectors, C.

- Evaluate the sigma vectors $\sigma = HC$.

- Build model Hamiltonian $\mathcal{H} = C^T \sigma$.

- Diagonalize $\mathcal{H}$: $\mathcal{E} = \mathcal{U}^T \mathcal{H} \mathcal{U}$.

- Compute Ritz vectors $X = C\mathcal{U}$.

- Compute residuals $R = X\mathcal{E} - \sigma\mathcal{U}$, check convergence: if yes, pass $X, \mathcal{E}$ as solutions.

- Preconditioning: $T = MR$ (many possible choices for the preconditioner M).

- Check if adding new trial vectors would exceed the maximum number of trial vectors:

  - if no, add T to C, and orthonormalize the united set

  - if yes, then set X as C (orthonormalize if H is nonsymmetric); then add T and orthonormalize

The advantage of the above method is that instead of the full Hamiltonian, only the sigma vectors have to be explicitly evaluated and stored. Below are all the parameters that influence the EOM routine. In the following sections, these parameters will be explained following the solver algorithm described above.

```
    %mdci
    #EOM parameters - defaults displayed
      DoEOM false          # whether to perform EOM
      UseEOMOptS true      # use optimized sigma routines for singles
      UseEOMOptD true      # use optimized sigma routines for doubles
      NDav 20              # maximum size of reduced space (i.e. 20*NRoots)
      CheckEachRoot true   # check convergence for each root separately
      RootHoming true      # apply root homing
      DoLanczos false      # use the Lanczos procedure rather than Davidson
      UseCISUpdate true    # use diagonal CIS for updating
      NInitS 0             # number of roots in the initial guess, if 0, use preset value
    #keywords which affect EOM parameters, but do not belong to the routine itself
      NRoots 9             # number of roots
      STol 1e-5            # default for CC residual threshold - for EOM 1e-8<=STol!
      OTol 1e-14           # orthogonalization threshold
      KCOpt KC_MO          # method for external exchange formation
      PrintLevel 3         # the amount of information to be printed
      MaxCore 500          # working memory
      MaxCoreIntAmp 500    # memory for integrals and amplitudes
    end
```

6.16.2 Memory Management

The most important data coming from the coupled cluster routine are the ground state energy and wave function, and the molecular integrals. The integrals are then used to create "dressed" integral containers, which allows for an efficient factorization of the EOM equations, since these dressed quantities do not change during the calculation. Most of these are written on disk, with the possible exception of the integral container which has three external labels. This, and the solver files may remain in core if enough memory is available. The program sequentially tries to allocate memory for the files in the order of their importance, and what cannot be kept in core, goes on disk. The order of allocation is residual vectors, Ritz vectors, three external integrals, sigma vectors, state (trial) vectors, as seen in the example below:

```
    -------------------------------
    AUTOMATIC CHOICE OF INCORE LEVEL
    -------------------------------

    Memory available                      ...    6512.00 MB
    Memory needed for Residual-vectors    ...      71.27 MB
    Memory needed for Ritz-vectors        ...      71.27 MB
    Memory needed for 3-ext integrals     ...      92.05 MB
    Memory needed for Sigma-vectors       ...    1425.31 MB
    Memory needed for State-vectors       ...    1425.31 MB
     -> Final InCoreLevel    ... 5
```

The sum of memory specified with keywords `MaxCore` and `MaxCoreIntAmp` is distributed among the five candidates, in the above case everything fits in. Note that these are only the largest contributors to memory consumption, and there should ideally be a safety margin when allocating memory.

In order to estimate the amount of necessary memory, it should be kept in mind that the memory requirements of the residual and Ritz vectors are proportional to $N_R N_P N_V^2$, the three external integrals to $N_R N_O N_V^3$ and the sigma and trial vectors to $N_D N_R N_P N_V^2$, where N_O and N_V are the number of occupied and virtual orbitals, $N_P = N_O(N_O + 1)$ is the number of occupied pairs, N_R is the number of roots, and N_D is the maximum size of the reduced space. The keyword `NRoots` sets N_R, while `NDav` determines N_D. Luckily, the contributions that in our experience are the most important to keep in memory, are also the ones that require the smallest amount of it.

6.16.3 Initial Guess

The present initial guess consists of constructing a CIS Hamiltonian of a certain dimension, and diagonalizing it. The roots are preselected based on the energetic order of the diagonal elements of the Hamiltonian. The number of roots in the initial guess is determined as 20 times the number of roots desired in EOM (`NRoots`) if `NDav` is 20 or smaller, otherwise it is set to `NDav` times the number of EOM roots. In the parameters `NInitS` is larger than zero, than the number of initial guess roots will be set to this parameter times `NRoots`. The maximum possible number of roots if the full CIS dimension, $N_O N_V$. One should keep in mind while increasing the number of initial guess vectors that it corresponds to diagonalizing a matrix of increasing dimensions. If for example `NRoots` is 10, then by default 200 roots are considered in the initial guess (unless it exceeds the size of the CIS space), or if `NInits` is set to 100, then there will be 1000 roots in the guess. In some cases, the roots calculated using EOM may not be the lowest ones, but a few of these may be replaced by some higher root "easier" to find. In such cases, it may help to increase `NRoots` or `NInitS` to converge to the proper roots.

6.16.4 Hamiltonian Construction

The Hamiltonian construction starts with calling the sigma routines. The logical variables `UseEOMOptS` and `UseEOMOptD` choose the routines to be used to evaluate the singles and doubles sigma vectors, respectively. If true, the optimized sigma routine using dressed integrals will be used. This should not be changed, the option is there mainly for debugging purposes. If set to false, an automatically generated, and much slower serial code will be used instead. The most time consuming part of the sigma vector construction, the formation of the external exchange, can be influenced via the CC keyword `KCOpt`. Currently, there are two options compatible with EOM: `KC_MO` and `KC_AOX` (see the MDCI documentation). In each early iteration N_R sigma vectors will be determined, except after restart, in which case the number of sigma vectors is $2N_R$. For further details on convergence, see 6.16.6 below.

Once the sigma vectors are available, they are multiplied with the trial vectors to yield the reduced space Hamiltonian. The Hamiltonian is built in a way that in each iteration only the new vector products are added to the "edge" of the old Hamiltonian, so that a full build is avoided. It should be clear that the parameter `NDav` plays an important role here, since it determines the maximum size of the Hamiltonian ($N_D N_R$), and also controls how much memory is needed for the trial and sigma vectors, as seen above. Since the choice of this parameter influences convergence properties, it will be discussed further in 6.16.6.

6.16.5 Solution of the (Nonsymmetric) Eigenproblem

Following the construction of the Hamiltonian, a nonsymmetric eigensolver is called. In this case, it is possible to have complex eigenvalues. In practice, this is rarely the case, and indicates a problem of some kind. A warning will be given if this happens, however, one may get away with this if it only happens in an isolated iteration step.

Once the eigenvectors are available, they are compared to those of the previous iteration, if root homing is turned on, i.e. if the `RootHoming` keyword is set to true. This means evaluating the overlap of the old and new eigenvectors, in order to keep track of the possible movement of the eigenvectors if root flipping occurs. If a global convergence

threshold is applied in a procedure where the lowest eigenvalues are desired, this is not necessary, since all intermediates are overwritten in each iteration, which automatically keeps track of such changes. However, if converged roots are removed from further iterations (see next section), it is important to keep track of changes in ordering, especially if a converged and a non-converged root is swapped. After diagonalization, the Ritz vectors and residuals can be evaluated.

6.16.6 Convergence, Restart, Preconditioning and Subspace Expansion

Convergence is signaled once a residual square norm based criteria is fulfilled. What this criteria is, is determined by the `CheckEachRoot` keyword. If it is true (default) the convergence of the residual square norm of each root is checked separately. This is due to the fact that different roots converge at a different rate. Once a root is converged, no new trial vectors will be generated belonging to that vector. This means that the EOM iterations will progressively become faster (until restart). Turning off rootwise convergence check is possible, but not recommended. In this case, the maximum of all residual square norms is checked for convergence, and iterations will take the same amount of time since no vectors are removed in any iteration. This procedure may however be numerically unstable, since the residuals of some roots may become very close to zero, and trying to generate new vectors orthogonal to these may lead to numerical disaster. In short, the recommended default is having both `CheckEachRoot` and `RootHoming` set to true. If `CheckEachRoot` is false, then so should be `RootHoming`, as it may cause problems if `NDav` is too small. The convergence threshold is set to 0.1 times the CC residual threshold, `STol`, except that it cannot be lower than 1e-8.

At this point it is worth discussing the role of the keyword `NDav`. This keyword determines at what point the Davidson algorithm should be restarted. If it is chosen too small, it may cause slow convergence. A too large value may result in overwhelming demands on memory/disk space requirements. The default value (20) is chosen with the hope that no, or maybe one restart will be required. It should only be changed if computational resources so demand it. At restart, Ritz vectors are copied as new trial vectors for all roots, which then will be orthonormalized, while new vectors will only be generated for the non-converged roots. This means that the step after the rebuilding of the expansion space will be 1-2 times as expensive as one of the initial steps.

New directions (trial vectors) are generated from the preconditioned residual vectors. If no preconditioning is applied (the preconditioner is taken to be a unit matrix), one falls back to the Lanczos algorithm, which the Davidson algorithm can be considered an improvement of. This happens if the keyword `DoLanczos` is true. This is not recommended, as the Lanczos algorithm converges several times slower than Davidson's, and is there for debugging mainly. The original Davidson preconditioner was the inverse of a diagonal matrix which contained the difference of the diagonal elements of the Hamiltonian and the current approximation to the eigenvalue belonging to the given root. If R_{ia} and R_{ijab} is an element of the singles and doubles amplitudes, respectively, than this means in practice that the updated vectors (T_{ia}, T_{ijab}) have the form

$$T_{ia} = \frac{R_{ia}}{D_{ia} + \mathcal{E}_R} \tag{6.228}$$

for singles, and

$$T_{ijab} = \frac{R_{ia}}{D_{ijab} + \mathcal{E}_R} \tag{6.229}$$

for doubles. Here, D_{ia} and D_{ijab} are related to and possibly approximations of the respective diagonal Hamiltonian elements. The simplest approximation can be just to construct these from diagonal Fock matrix elements (i.e. orbital energies) as $D_{ia} = \varepsilon_a - \varepsilon_i$ and $D_{ijab} = \varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j$. A slightly better preconditioning can be obtained as follows. For singles, take the exact CIS diagonal elements, $D_{ia} = \varepsilon_a - \varepsilon_i + \bar{g}_{iiaa}$, where the last term is the respective

antisymmetrized integral; and construct the doubles as $D_{ijab} = D_{ia} + D_{jb}$. This is the default, and can be changed back to the simple Fock matrix guess by setting `UseCISUpdate` to false.

Following the preconditioning step, the resulting vectors are orthogonalized to the previous set of trial vectors, and orthonormalized among themselves. Since until restart the trial vectors do not change once they are generated, only the new elements of the overlap matrix need to be generated for the orthonormalization. The threshold for numerical threshold during the inversion (and other division steps) is controlled by the parameter `OTol`. Finally, the amount of printed information can be controlled via the `PrintLevel` keyword. If not given or equal to 2, only basic iteration information will be printed. If set to 3, detailed iteration information will be printed (recommended if timing results for individual steps are required), while 4 or higher triggers additional (and very verbose) information from other subroutines as well.

6.17 The Multireference Correlation Module

6.17.1 General Description of the Correlation Module

The detailed documentation of all features of the MR-CI and MR-PT module is somewhat premature and at this point only a summary of keywords is given below. A thorough description of all technical and theoretical subtleties must wait for a later version of the manual.

```
%mrci
      # ----------------------------------------------------------
      # Orbital selection
      # NOTE: The orbitals are used as supplied. Thus, the ORDER of
      # orbitals is critical. Say you have
      #   nact    electrons in the active space
      #   nint    electrons in the internal space
      #   nfrozen electrons
      # * The first nfrozen/2 orbitals will not be included in the CI
      # * The next nint/2 orbitals will be doubly occupied in all
      #   references
      # * the nact electrons are distributed over the,say, mact
      #   orbitals according to the active space definitions.
      # The remaining orbitals are external.
      # IT IS YOUR RESPONSIBILITY THAT THE ORBITAL ORDERING MAKES
      # SENSE!
      # A sensible two-step procedure is:
      #  * generate some orbitals and LOOK AT THEM. Decide which ones
      #    to include in the CI.
      #  * re-read these orbitals with ! MORead NoIter. Perhaps use
      #    the "rotate" feature to reorder the MOs
      #    Then jump right into the CI which is defined in this se-
      #    cond job
      #
      # NOTE: the MRCI module respects the %method FrozenCore settings
      # ----------------------------------------------------------
      Loc  0,0,0
          # Localize orbitals in the internal (first flag), active
          # (second flag) and external space (third flag).
      UseIVOs  false
          # Use improved virtual orbitals in the CI
```

```
          # -------------------------------
          # Method selection
          # -------------------------------
          CIType    MRCI       # Multireference CI (default)
                    MRDDCI1    # Difference dedicated CI 1-degree of freedom
                    MRDDCI2    # Difference dedicated CI 2-degrees of freedom
                    MRDDCI3    # Difference dedicated CI 3-degrees of freedom
                    MRACPF     # Average coupled pair functional
                    MRACPF2    # Modified version of ACPF
                    MRACPF2a   # A slightly modified version of ACPF-2a
                    MRAQCC     # Average quadratic coupled cluster
                    MRCEPA_R   # Multireference CEPA due to Ruttink
                    MRCEPA_0   # CEPA-0 approximation
                    SORCI      # Spectroscopy oriented CI
                    SORCP      # Spectroscopy oriented couplet pair approx.
                    MRMP2      # Multireference Moeller-Plesset at second order
                    MRMP3      # Multireference Moeller-Plesset at third order
                    MRMP4      # Multireference Moeller-Plesset at fourth order
                               # but keeping only singles and doubles relative to
                               # the reference configurations.

          # ---------------------------------
          # Selection thresholds
          # ---------------------------------
          Tsel    1e-6   # Selection threshold for inclusion in the CI based
                         # 2nd order MP perturbation theory <0|H|I>/DE(MP)
          Tpre    1e-4   # Selection of configurations in the reference space
                         # after the initial diagonalization of the reference
                         # space only configurations with a weight large>Tpre
                         # to any root are included
          AllSingles    false
                         # include ALL SINGLES in the CI. Default is now TRUE!!!

          # perturbative estimate of the effect of the rejected configurations
          EunselOpt  0   # no correction
                     1   # based on the overlap with the 0th order wavefunction
                     2   # calculation with the relaxed reference space
                         # coefficients. This is the most accurate and only
                         # slightly more expensive

          # For CIType=MRCI,MRDDCI and SORCI the approximate correction for
          # higher excitations
          DavidsonOpt  Davidson1  # default
                       Davidson2  # modified version
                       Siegbahn   # Siegbahn's approximation
                       Pople      # Pople's approximation

          # For MRACPF,MRACPF2,MRAQCC and SORCP
          NelCorr      0
             # Number of electrons used for computing the average coupled
             # pair correction.
             # =0 : set equal to ALL electrons in the CI
             # =-1: set equal to all ACTIVE SPACE electrons
             # =-2: set equal to ACTIVE SPACE electrons IF inactive doubles
             #      are excluded (as in MRDDCI)
             # >0 : set equal to user defined input value
```

```
LinearResponse false
   # Use ground state correlation energy to compute the shift for
   # higher roots (not recommended)


# ---------------------------------
# Natural Orbital Iterations
# ---------------------------------
NatOrbIters   0 # default
   # number of average natural orbital iterations
Tnat        1e-4
   # cutoff of natural orbitals. NOs with an occupation number less
   # then Tnat will not be included in the next iteration
   # Also, orbitals with occupation number closer than Tnat to 2.0
   # will be frozen in the next iteration
Tnat2       -1
   # if chosen >0 then Tnat2 is the threshold for freezing the
   # almost doubly occupied orbitals. Otherwise it is set equal
   # to Tnat


# ---------------------------------
# Additional flags and algorithmic
# details
# ---------------------------------
PrintLevel   2  # default. Values between 1 and 4 are possible

DoDDCIMP2  false
   # for DDCI calculations: if set to true the program computes
   # a MP2 like correction for the effect of inactive double
   # excitations which are not explicitly included in the CI. This
   # is necessary if you compare molecules at different geometries
   # or compute potential energy surfaces.
# ---------------------------------
# The SORCP model
# ---------------------------------
CIType_in  # First step CIType
CIType_fi  # Second step CIType
Exc_in     # First step excitation scheme
Exc_fi     # Second step excitation scheme
Tsel_in    # First step Tsel
Tsel_fi    # Second step Tsel
Tpre_in    # First step Tsel
Tpre_fi    # Second step Tpre

  # Thus, the SORCI model corresponds to CIType=SORCP with
  # CIType_in MRCI  CIType_fi MRCI
  # Exc_in    DDCI2 Cexc_fi   DDCI3
  # Tsel_in   1e-5  Tsel_fi   1e-5
  # Tpre_in   1e-2  Tpre_fi   1e-2


# ---------------------------------
# Multirerence perturbation theory
# ---------------------------------
MRPT_b  0.02   # Intruder state avoidance PT after Hirao (default 0.0)
               # with this flag individual intruders are shifted away to
               # to some extent from the reference space
MRPT_shift 0.3 # Level shift introduced by Roos which shifts the entire
```

```
                    # excited manifold away in order to avoid intruder states.
                    # A correction is applied afterwards but results do depend
                    # on this (arbitrary) value to some extent.
H0Opt  projected  # use an off-diagonal definition of H0
       Diagonal   # use a diagonal definition of H0 (much faster but maybe
                    # a little less reliable
Partitioning MP   # Moeller plesset partitioning
             EN   # Epstein-Nesbet partitioning (not recommended)
Fopt  Standard    # Standard definition of MR Fock operators
      G3          # uses Anderson's g3 correction also used in CASPT2


# ====================================
# Definitions of blocks of the CI Matrix
# ====================================
NewBlock 2 *    # generate a Block with doublet(=2) multiplicity
  Nroots  1     # number of roots to be generated
  Excitations cis   # CI with single excitations
              cid   # CI with double excitations
              cisd  # CI with single and double excitations
              ddci1 # DDCI list with one degree of freedom
              ddci2 # DDCI list with two degrees of freedom
              ddci3 # DDCI list with three degrees of freedom
  Flags[_class_] 0 or 1
          # Turn excitation classes on or off individually
          # 's' stands for any SOMO, 'i','j' for internal orbitals and
          # 'a','b' for external orbitals
          #  Singles _class_ = ss, sa, is, ia
          #  Doubles _class_ = ijss, ijsa, ijab,
          #                     isss, issa, isab,
          #                     ssss, sssa, ssab
          # 'Flags' takes priority over 'Excitations'. In fact 'Excitations'
          # does nothing but to set 'Flags'. So, you can use 'Excitations'
          # to provide initial values for 'Flags' and then modify them
          # with subsequent 'Flags' assignments
  refs
        #
        # First choice - complete active space
        #
        CAS(nel,norb)  # CAS-CI reference with nel electrons in
                       # Norb orbitals
        #
        # Second choice - restricted active space
        #
        RAS(nel: m1 h/ m2 / m3 p)
                       # RAS-reference with nel electrons
            # m1= number orbitals in RAS-1
            # h = max. number of holes in RAS-1
            # m2= number of orbitals in RAS-2 (any number of
            #     electrons or holes)
            # m3= number of orbitals in RAS-3
            # p = max. number of particles in RAS-3
        #
        # Third choice - individually defined configurations
        #
         { 2 0 1 0}
         { 1 1 1 0}
```

```
                 etc.
                 # define as many configurations as you want. Doubly occupied MOs
                 # singly occupied MOs and empty MOs. Important notes:
                 #  a) the number of electrons must be the same in all references
                 #  b) the number of orbitals is determined from the number of
                 #     definitions. Thus, in the example above we have three active
                 #     electrons and four active orbitals despite the fact that the
                 #     highest orbital is not occupied in any reference.
                 #  The program determines the internal, active and external spaces
                 #  automatically from the number of active electrons and orbitals
              end
      end
      # there can be as many blocks as you want!!!


   # --------------------------------
   # Density matrix generation flags
   #  First Key= State densities <I|D|I>
   #     =0: none
   #     =1: Ground state only (lowest root of all blocks; Electron only)
   #     =2: Ground state only (Electron and spin density)
   #     =3: Lowest root from each block (Electron density)
   #     =4: Lowest root from each block (Electron and spin density)
   #     =5: All states (Electron density)
   #     =6: All states (Electron and spin density)
   #  Second Key= Transition densities <I|D|J>
   #     needed for all transition intensities, g-tensor etc
   #     =0: none
   #     =1: from the ground state into all excited states (el)
   #     =2: from the ground state into all excited states (el+spin)
   #     =3: from all lowest states into all excited states (el)
   #     =4: from all lowest states into all excited states (el+spin)
   #     =5: all state pairs (el)
   #     =6: all state pairs (el+spin)
   # Note that for perturbation theory the density is computed as
   # an expectation value over the first (second) order wavefunction.
   # which is renormalized for this purpose
   # --------------------------------
   Densities 1,1


   # --------------------------------
   # Algorithm for the solver
   # --------------------------------
   Solver   Diag   # Davidson like solver
            DIIS   # DIIS like solver
     # both solvers have their pros and cons. The DIIS may converge
     # better or use less time since it only recomputes the vectors that
     # have not yet converged; The DIIS may be less sensitive to root flipping
     # effects but occasionally it converges poorly and states of the same
     # symmetry are occasionally a little problematic
     # For perturbation theory DIIS is always used.
   # For both solvers
   MaxIter   100  # the maximum number of iterations
   Etol      1e-6 # convergence tolerance for energies in Hartree
   Rtol      1e-6 # convergence tolerance for residual


   # For Solver=Diag (Davidson solver)
```

```
        Otol      1e-16 # Orthogonality threshold for Schmidt process
        NguessMat     5 # The NGuess*Nroots lowest configurations will
                        # be used to compute the initial guess
        MaxDim       12 # Maximum number of expansion vectors. Should be
                        # a multiple of Nroots for best efficiency


        # For the Solver=DIIS. Particularly recommended for anything else but
        # straightforward CI and also for calculations in direct2 mode!
        MaxDIIS       5 # Maximum number of old vectors to be used in DIIS
        RelaxRefs  true # Relax reference space coefficients in the CI or
                        # freeze them to their zeroth order values
        LevelShift  0.4 # Level Shift for stabilizing the DIIS procedure


        # ----------------------------------
        # Integral storage, memory and files
        # ----------------------------------
        IntStorage    FloatVals
                      DoubleVals  (default)
           # store integrals with float (4 byte) or double (8 byte)
           # accuracy in main memory
        FourIndexInts false (default)
                      True
           # Store ALL four index integrals over Mos in main memory
           # only possible for relatively small systems, perhaps up
           # to 150-200 MOs included in the CI
        MaxMemInt   256
           # Maximum amount of core memory devoted to the storage of
           # integrals. If NOT all three index integrals fit into main
           # memory the program fails
        MaxMemVec    16
           # Maximum amount of memory used for section of the trial and
           # sigma vectors. This is not a particularly critical variable
        KeepFiles    false
           # Keep integrals and CI program input file (.mrciinp). Then
           # you can manually edit the .mrciinp file which is a standard
           # ASCII file and run the MRCI program directly. The only thing
           # you cannot change is the orbital window.
        end
```

6.17.2 The SOC Submodule of the MRCI Module

6.17.2.1 Calculation of the Spin-Spin Coupling and Spin-Orbit Coupling Effects

The spin-orbit coupling (SOC) and spin-spin coupling (SSC) effects can be calculated very accurately using a wavefunction obtained from a multiconfigurational calculation of a multi-reference type such as CASSCF, MRCI, or MRPT in the form of:

$$\left| \Psi_I^{SS} \right\rangle = \sum\nolimits_{\mu} C_{\mu l} \left| \Phi_\mu^{SS} \right\rangle \tag{6.230}$$

Here the upper indices SS stands for a wave function of the spin quantum number S and spin projection $M_S = S$. Since the BO Hamiltonian does not contain any complex-valued operator, the solutions $\left| \Psi_I^{SS} \right\rangle$ may be chosen to be real-valued.

The SOC and SSC effects along with the Zeeman interaction can be included by means of the quasi-degenerate perturbation theory (QDPT). In this approach the SOC, the SSC, and the Zeeman operators are calculated in the basis of pre-selected solutions of the BO Hamiltonian $\left\{ \Psi_I^{SM} \right\}$.

$$\left\langle \Psi_I^{SM} \left| \hat{H}_{BO} + \hat{H}_{SOC} + \hat{H}_{SSC} + \hat{H}_Z \right| \Psi_J^{S'M'} \right\rangle = \delta_{IJ}\delta_{SS'}\delta_{MM'}E_I^{(S)} + \left\langle \Psi_I^{SM} \left| \hat{H}_{SOC} + \hat{H}_{SSC} + \hat{H}_Z \right| \Psi_J^{S'M'} \right\rangle \quad (6.231)$$

Diagonalization of this matrix yields the energy levels and eigenvectors of the coupled states. These eigenvectors in fact represent linear combinations of the solutions of $\hat{H}_{BO}$ with complex coefficients.

The effective one-electron SOC operator in second quantized form can be written as:

$$\hat{H}_{SOMF} = \frac{1}{2} \sum_{pq} z_{pq}^- \hat{a}_p^\uparrow \hat{b}_q + z_{pq}^+ \hat{b}_p^\uparrow \hat{a}_q + z_{pq}^0 \left[\hat{a}_p^\uparrow \hat{a}_q - \hat{b}_p^\uparrow \hat{b}_q \right] \quad (6.232)$$

Here $\hat{a}_p^\uparrow$ and $\hat{b}_p^\uparrow$ stand for creation of α and β electrons respectively; $\hat{a}_p$ and $\hat{b}_p$ represent the corresponding annihilation operators. The matrix elements $z_{pq}^- = z_{pq}^x - iz_{pq}^y$, $z_{pq}^+ = z_{pq}^x + iz_{pq}^y$, and $z_{pq}^0 = z_{pq}^z$ (upper x, y, z indices denote the Cartesian components) are constructed from the matrix elements described in section 6.21.3.3.

The SSC Hamiltonian reads:

$$\hat{H}_{SS} = -\frac{3g_e^2\alpha^2}{8} \sum_{i \neq j} \sum_{m=0,\pm1,\pm2} \frac{(-1)^m}{r_{ij}^5} \left[\mathbf{r}_{ij} \times \mathbf{r}_{ij} \right]_{-m}^{(2)} \left[\mathbf{S}(i) \times \mathbf{S}(j) \right]_m^{(2)} \quad (6.233)$$

For matrix elements between states of the same multiplicity it can be simplified to

$$\left\langle aSM \left| \hat{H}_{SSC} \right| a'SM' \right\rangle = \frac{\sqrt{(S+1)(2S+3)}}{\sqrt{S(2S-1)}}$$
$$\times \sum_m (-1)^m \begin{pmatrix} S' & 2 & S \\ M' & m & M \end{pmatrix} \sum_{pqrs} D_{pqrs}^{(-m)} \left\langle aSS \left| Q_{pqrs}^0 \right| a'SS \right\rangle \quad (6.234)$$

Here

$$Q_{pqrs}^{(0)} = \frac{1}{4\sqrt{6}} \left\{ E_{pq}\delta_{sr} - S_{ps}^z S_{rq}^z + \frac{1}{2} \left(S_{pq}^z S_{rs}^z - E_{pq}E_{rs} \right) \right\} \quad (6.235)$$

represents the two-electron quintet density. The operators $E_{pq} = \hat{a}_p^\uparrow \hat{a}_q + \hat{b}_p^\uparrow \hat{b}_q$ and $S_{pq}^z = \hat{a}_p^\uparrow \hat{a}_q - \hat{b}_p^\uparrow \hat{b}_q$ symbolize here the one-electron density operator and the spin density operator accordingly. The spatial part

$$D_{pqrs}^{(0)} = \frac{1}{\sqrt{6}} \iint \varphi_p(\mathbf{r}_1) \varphi_r(\mathbf{r}_2) \frac{3r_{1z}r_{2z} - \mathbf{r}_1\mathbf{r}_2}{r_{12}^5} \varphi_q(\mathbf{r}_1) \varphi_s(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \quad (6.236)$$

denotes the two-electron field gradient integrals. These two-electron integrals can be evaluated using the RI approximation.

Finally, the Zeeman Hamiltonian is included in the form of:

$$\hat{H}_{Zeeman} = \mu_B \left(\hat{\mathbf{L}} + g_e\hat{\mathbf{S}} \right) \mathbf{B} \quad (6.237)$$

with $\hat{\mathbf{L}}$ representing the total orbital momentum operator, and $\hat{\mathbf{S}}$ being the total spin operator.

Since the both the energies and the wavefunction of the low-lying spin-orbit states are available, the effective Hamiltonian theory can be used to extract the full ZFS tensor. In this approach one starts by constructing a model Hamiltonian, which for the mononuclear complexes is:

$$\hat{H}_{mod} = \hat{\mathbf{S}}D\hat{\mathbf{S}} \tag{6.238}$$

provided that the ground state is non-degenerate. By applying this Hamiltonian on the basis of the model space, all the $|S, M_S\rangle$ components of the ground state, the interaction matrix is constructed.

The construction of effective Hamiltonian relies on the information contained in both the energies and the wavefunctions of the low-lying spin-orbit states. Following des Cloizeaux formalism, the effective Hamiltonian reproduces the energy levels of the "exact" Hamiltonian E_k and the wavefunctions of the low-lying states projected onto the model space $\tilde{\Psi}$:

$$\hat{H}_{eff}|\tilde{\Psi}_k\rangle = E_k|\tilde{\Psi}_k\rangle \tag{6.239}$$

These projected vectors are then symmetrically orthonormalized resulting in an Hermitian effective Hamiltonian, which can be written as:

$$\hat{H}_{eff}|\tilde{\Psi}\rangle = \sum_k |S^{-\frac{1}{2}}\tilde{\Psi}_k\rangle E_k \langle S^{-\frac{1}{2}}\tilde{\Psi}_k| \tag{6.240}$$

The effective interaction matrix obtained by expanding this Hamiltonian into the basis of determinants belonging to the model space, is the compared to the matrix resulted from expanding the model Hamiltonian. Based on a singular value decomposition procedure, all 9 elements of the ZFS tensor are extracted.

In the input file the relevant keys are located in the soc sub-block:

```
%mrci
  soc
    DoSOC true   # include the SOC contribution
    DoSSC true   # include the SSC contribution
    PrintLevel 2 # printing level
    TPrint 0.01  # threshold for printing eigenvalue contributions
  end
end
```

Firstly, the SSC contribution to ZFS is calculated. The output file for a CASCI calculation on top of a CASSCF(8,6) reference of an oxygen molecule with def2-SVP basis set and four singlets and four triplet states included into the QDPT treatment is presented as follows:

```
      --------------------
      SSC MATRIX GENERATION
      --------------------


      Blocks order:    1    0

      Ground state mult    3
      Ground state block   1
      Ground state root    0


      Calculated reduced SSC matrix elements

      BLOCKI(Mult) BLOCKJ(Mult)  Root I  Root J   dxx        dyy        dzz        dxy        dxz        dyz
                                                  1/cm       1/cm       1/cm       1/cm       1/cm       1/cm
      1( 3)  1( 3)     0    0   -0.316899  -0.316899   0.633797  -0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     1    0    0.000000   0.000000  -0.000000  -0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     1    1    0.251010   0.251010  -0.502020   0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     2    0   -0.000000  -0.000000   0.000000   0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     2    1   -0.000000   0.000000   0.000000   0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     2    2    0.251010   0.251010  -0.502020   0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     3    0   -0.000000   0.000000  -0.000000   0.000000  -0.000000  -0.000000
      1( 3)  1( 3)     3    1   -0.095105   0.095105  -0.000000  -0.010169  -0.000000  -0.000000
      1( 3)  1( 3)     3    2   -0.010169   0.010169  -0.000000   0.095105  -0.000000  -0.000000
      1( 3)  1( 3)     3    3    0.307060   0.307060  -0.614121  -0.000000  -0.000000  -0.000000
```

Next, the SSC matrix is constructed form the precalculated matrix elements and subjected to diagonalization. The full SSC matrix is available with the `PrintLevel` key set to 3. Then, the SSC energy levels are printed in cm^{-1} and eV along with the Boltzmann level populations.

```
      Energy levels (cm-1,eV) :     Boltzmann populations for T = :   300.000   K
          0 :        0.000    0.0000        3.34e-01
          1 :        0.951    0.0001        3.33e-01
          2 :        0.951    0.0001        3.33e-01
          3 :     7130.461    0.8841        4.71e-16
          4 :     7130.996    0.8841        4.69e-16
          5 :    13306.276    1.6498        6.45e-29
          6 :    46916.451    5.8169        6.38e-99
          7 :    48790.859    6.0493        7.95e-103
          8 :    48790.859    6.0493        7.95e-103
          9 :    48790.859    6.0493        7.95e-103
         10 :    48790.859    6.0493        7.95e-103
         11 :    48791.612    6.0494        7.93e-103
         12 :    48791.612    6.0494        7.93e-103
         13 :    49948.158    6.1928        3.09e-105
         14 :    49948.158    6.1928        3.09e-105
         15 :    49949.079    6.1929        3.08e-105
```

Additionally, the eigenvectors of the SSC matrix are printed if the `PrintLevel` key is set to 2:

```
      Eigenvectors :

      The threshold for printing is 0.0100000
```

```
                         Weight      Real        Image    : Block Root Spin Ms

   STATE   0 :    0.00

                         1.000000    0.999398   -0.034683 :  1    0    1  0

   STATE   1 :    0.95

                         0.490154   -0.700086   -0.005795 :  1    0    1  1
                         0.509846    0.714011    0.005862 :  1    0    1 -1

   STATE   2 :    0.95

                         0.509846    0.714010   -0.005997 :  1    0    1  1
                         0.490154    0.700085   -0.005927 :  1    0    1 -1

   STATE   3 : 7130.24

                         1.000000    1.000000   -0.000000 :  0    0    0  0

   STATE   4 : 7130.77

                         1.000000    1.000000   -0.000000 :  0    1    0  0
```

The SSC coupled states are counted here from the lowest one, and the energy of a state is printed in cm^{-1}. The 'Weight' column represents here squared absolute values of the complex contribution coefficients of the BO Hamiltonian states to the SSC coupled states. 'Real' and 'Image' are real and imaginary parts of complex coefficients. Accordingly, 'Root', 'Spin', and 'Ms' denote here the spin quantum number and the spin projection number of a particular contribution. The column 'Block' represent the symmetry block number. Note that the program re-aranges blocks from the oder given in the input file.

Next, the SSC contribution to the ZFS obtained with first-order perturbation theory is printed along with the D values and the E/D ratio.

```
      --------------------------------
          ZERO-FIELD SPLITTING
      (SPIN-SPIN COUPLING CONTRIBUTION)
      --------------------------------


   raw-matrix (cm**-1):
        -0.316896      -0.000000      -0.000000
        -0.000000      -0.316896      -0.000000
        -0.000000      -0.000000       0.633791
   diagonalized D-tensor (cm**-1):
        -0.316896      -0.316896       0.633791

         0.000000       1.000000       0.000000
         1.000000       0.000000       0.000000
         0.000000       0.000000       1.000000


   Direction X=1 Y=0 Z=2
   D   =     0.950687  cm**-1
   E/D =     0.000000
```

Here the eigenvalues are printed in the increasing order, and the column under the eigenvalue represents the eigenvector belonging to this eigenvalue. The axes for the coordinate system that diagonalizes the D tensor are chosen to fulfill the condition

$$0 \leqslant E/D \leqslant \frac{1}{3} \tag{6.241}$$

Next, the SOC part to the splitting is calculated. Firstly, the diagonal (with respect to the spin) matrix element type of $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^x S_{pq}^z |\Psi_J\rangle$, $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^y S_{pq}^z |\Psi_J\rangle$, $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^z S_{pq}^z |\Psi_J\rangle$ are evaluated between states of the same multiplicity and $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^x S_{pq}^+ |\Psi_J\rangle$, $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^y S_{pq}^+ |\Psi_J\rangle$, $\langle \Psi_I| \sum_{pq} \mathbf{z}_{pq}^z S_{pq}^+ |\Psi_J\rangle$ between states of different multiplicities.

```
      Calculated reduced SOC matrix elements

      BLOCKI(Mult)    BLOCKJ(Mult)    Root I   Root J   <I|Lx S |J> <I|Ly S |J> <I|Lz S |J>
                                                        1/cm        1/cm        1/cm


         0( 3)          0( 3)          0        0        -0.00        0.00       -0.00
         0( 3)          0( 3)          1        0        -0.00        0.00       -0.00
         0( 3)          0( 3)          1        1        -0.00       -0.00       -0.00
         0( 3)          0( 3)          2        0        -0.00       -0.00       -0.00
         0( 3)          0( 3)          2        1        -0.00        0.00     -289.98
         0( 3)          0( 3)          2        2        -0.00       -0.00       -0.00
         0( 3)          0( 3)          3        0         0.00       -0.00       -0.00
         0( 3)          0( 3)          3        1         0.00       -0.00        0.00
         0( 3)          0( 3)          3        2         0.00       -0.00        0.00
         0( 3)          0( 3)          3        3        -0.00       -0.00       -0.00
         0( 3)          1( 1)          0        0        -0.00        0.00       -0.00
         0( 3)          1( 1)          0        1        -0.00        0.00       -0.00
         0( 3)          1( 1)          0        2        -0.00        0.00     -225.16
         0( 3)          1( 1)          0        3         0.00        0.00        0.00
         0( 3)          1( 1)          1        0        -0.00        0.00        0.00
         0( 3)          1( 1)          1        1         0.00        0.00       -0.00
         0( 3)          1( 1)          1        2         0.00        0.00        0.00
         0( 3)          1( 1)          1        3        -0.00        0.00       -0.00
         0( 3)          1( 1)          2        0         0.00        0.00       -0.00
         0( 3)          1( 1)          2        1         0.00       -0.00       -0.00
         0( 3)          1( 1)          2        2        -0.00        0.00       -0.00
         0( 3)          1( 1)          2        3         0.00       -0.00        0.00
         0( 3)          1( 1)          3        0         0.00       -0.00        0.00
         0( 3)          1( 1)          3        1        -0.00       -0.00       -0.00
         0( 3)          1( 1)          3        2        -0.00       -0.00        0.00
         0( 3)          1( 1)          3        3         0.00       -0.00      205.22
```

The full SOC matrix elements can be printed if the `PrintLevel` key is set to 3. Similar to the SSC contribution, the SOC energies are printed next.

```
      Energy levels (cm-1,eV) :      Boltzmann populations for T =  :   300.000   K
         0 :         0.000     0.0000        3.35e-01
         1 :         1.905     0.0002        3.32e-01
         2 :         1.905     0.0002        3.32e-01
         3 :      7130.930     0.8841        4.71e-16
         4 :      7131.465     0.8842        4.70e-16
         5 :     13308.646     1.6501        6.39e-29
```

```
    6 :     46856.467      5.8095        8.53e-99
    7 :     48646.630      6.0314        1.59e-102
    8 :     48646.630      6.0314        1.59e-102
    9 :     48791.619      6.0494        7.95e-103
   10 :     48791.619      6.0494        7.95e-103
   11 :     48936.608      6.0674        3.97e-103
   12 :     48936.608      6.0674        3.97e-103
   13 :     49948.972      6.1929        3.09e-105
   14 :     49948.972      6.1929        3.09e-105
   15 :     49955.781      6.1937        2.99e-105
```

The eigenvectors of the SOC matrix are provided along with contributions from roots S, and M_s.

```
   Eigenvectors :

   The threshold for printing is 0.0100000

                          Weight       Real          Image      : Block Root Spin Ms

    STATE   0 :     0.00

                          0.999857    -0.000000    -0.999928 :    0     0    1  0

    STATE   1 :     1.90

                          0.464214    -0.228955     0.641711 :    0     0    1  1
                          0.535786     0.154270    -0.715533 :    0     0    1 -1

    STATE   2 :     1.90

                          0.535786     0.730198     0.050967 :    0     0    1  1
                          0.464214     0.680079    -0.041312 :    0     0    1 -1

    STATE   3 :  7130.93

                          1.000000     1.000000    -0.000000 :    1     0    0  0

    STATE   4 :  7131.46

                          1.000000    -1.000000     0.000000 :    1     1    0  0
```

Next, the SOC contribution to the ZFS obtained with second-order perturbation theory is printed along with the D values and the E/D ratio.

```
   --------------------------------
        ZERO-FIELD SPLITTING
   (SPIN-ORBIT COUPLING CONTRIBUTION)
   --------------------------------

   raw-matrix (cm**-1):
       0.000000     -0.000000     -0.000000
      -0.000000      0.000000      0.000000
      -0.000000      0.000000      1.905043
```

```
     diagonalized D-tensor (cm**-1):
          0.000000          0.000000          1.905043


          0.889614          0.456713         -0.000000
         -0.456713          0.889614          0.000000
          0.000000          0.000000          1.000000


     Direction X=1 Y=0 Z=2
     D   =    1.905043  cm**-1
     E/D =    0.000000
```

Independent state contribution to the ZFS is printed if `PrintLevel` is set to 3.

```
     Individual contributions to D tensor:


     block Mult root    D         E

        0    3    0       0.000       0.000
        0    3    1      -0.000       0.000
        0    3    2      -0.000      -0.000
        0    3    3      -0.000      -0.000
        1    1    0       0.000       0.000
        1    1    1       0.000      -0.000
        1    1    2       1.905       0.000
        1    1    3       0.000       0.000
```

From this example is readily seen that the whole contribution to the SOC part of D-value comes from the interaction between the ground state and the singlet third root.

The ZFS tensor based on the effective Hamiltonian theory is printed next.

```
     -----------------------------------------------------------
                    ZERO-FIELD SPLITTING
     (EFFECTIVE HAMILTONIAN SPIN-ORBIT COUPLING CONTRIBUTION)
     -----------------------------------------------------------


     raw-matrix (cm**-1):
         -0.000000          0.000000         -0.000000
          0.000000          0.000000         -0.000000
         -0.000000         -0.000000          1.904771
     diagonalized D-tensor (cm**-1):
         -0.000000          0.000000          1.904771


          0.756590         -0.653889         -0.000000
         -0.653889         -0.756590         -0.000000
          0.000000         -0.000000          1.000000


     Direction X=1 Y=0 Z=2
     D   =    1.904771  cm**-1
     E/D =    0.000000
```

Finally, all these steps described above are repeated for the combined SOC and SSC matrix.

The diagonal matrix elements of the SOC and SSC matrix which correspond to non-relativistic energies can be replaced by more accurate values. In the output of the program the corresponding transition energies printed in cm^{-1}.

```
    # In case that you want to run QDPT-SOC calculation with manually
    #adjusted diagonal energies you can copy the following part into
    #the %mrci soc block
    #and modify it as needed(energies are given in
    #wavenumbers relative to the lowest state)
    # NOTE: It is YOUR responsibility to make sur that the CAS-CI state
    #that you may want to dress with these energies correlate properly
    #with the energies printed here. The order of states or even the
    #identity of states may change with and without inclusion of
    #dynamic correlation In the case that dynamic correlation strongly
    #mixes different CAS-CI states there may not even be a proper
    #correlation!
    #
        EDiag[  0]          0.00  # root   0 of block 0
        EDiag[  1]      48328.40  # root   1 of block 0
        EDiag[  2]      48328.40  # root   2 of block 0
        EDiag[  3]      49334.96  # root   3 of block 0
        EDiag[  4]       7763.59  # root   0 of block 1
        EDiag[  5]       7763.59  # root   1 of block 1
        EDiag[  6]      11898.46  # root   2 of block 1
        EDiag[  7]      46754.23  # root   3 of block 1
```

Those transition energies can be substituted by a more accurate energies provided in the input file as follows:

```
    %soc
    dosoc true
    dossc true
        EDiag[  0]          0.00  # root   0 of block 0
        EDiag[  1]      48328.40  # root   1 of block 0
        EDiag[  2]      48328.40  # root   2 of block 0
        EDiag[  3]      49334.96  # root   3 of block 0
        EDiag[  4]       7763.59  # root   0 of block 1
        EDiag[  5]       7763.59  # root   1 of block 1
        EDiag[  6]      11898.46  # root   2 of block 1
        EDiag[  7]      46754.23  # root   3 of block 1
    end
```

Accurate diagonal energies generally improve the accuracy of the SOC and SSC splittings.

6.17.2.2 Magnetic Fields in MRCI

The inclusion of the Zeeman contribution into the QDPT procedure allows to obtain the splittings of the magnetic levels in an external magnetic field. The switch for this calculation and the magnetic field strength are defined in the soc subblock of the mrci block:

```
%mrci
  soc
    DoSOC true          #
    DoSSC true          #
    MagneticField true  # default false
    B 1,10,100,1000     # Strengh of the magnetic field in Gauss.
                        # 4000 is the default value
  end
end
```

Then, the output contains three sets of data of splittings of the magnetic levels with the magnetic field applied parallel to x, y, and z directions:

```
End B (Gauss)  Energy levels (cm-1) and populations for B || x

     1.0   -0.030   0.333   0.012   0.333   0.018   0.333
    10.0   -0.030   0.333   0.012   0.333   0.018   0.333
   100.0   -0.031   0.333   0.012   0.333   0.020   0.333
  1000.0   -0.102   0.333   0.012   0.333   0.091   0.333


B (Gauss)  Energy levels (cm-1) and populations for B || y

     1.0   -0.030   0.333   0.012   0.333   0.018   0.333
    10.0   -0.030   0.333   0.012   0.333   0.018   0.333
   100.0   -0.032   0.333   0.014   0.333   0.018   0.333
  1000.0   -0.105   0.334   0.018   0.333   0.087   0.333


B (Gauss)  Energy levels (cm-1) and populations for B || z

     1.0   -0.030   0.333   0.012   0.333   0.018   0.333
    10.0   -0.030   0.333   0.011   0.333   0.018   0.333
   100.0   -0.030   0.333   0.005   0.333   0.025   0.333
  1000.0   -0.079   0.333  -0.030   0.333   0.108   0.333
```

Here the number in a row represents the strength of the magnetic field (in Gauss), and the following pairs of numbers denote the energy of the magnetic level (in cm^{-1}) with its occupation number. This table can be readily plotted with any suitable graphical program.

6.17.2.3 g-Tensor Calculations

The orca_mrci program contains an option to calculate g-tensors using MRCI wavefunctions. For a system with an odd number of electrons, the doubly degenerate eigenvalues obtained from the QDPT procedure represent Kramers pairs and used to build the matrix elements of the total spin operator and the total angular momentum operator from the Zeeman Hamiltonian. Denoting Ψ as a solution and $\bar{\Psi}$ as its Kramers partner and using matrix element notations

$$\Phi_{11}^k = \langle \Psi | \hat{L}_k + g_e \hat{S}_k | \Psi \rangle , \ \Phi_{12}^k = \langle \Psi | \hat{L}_k + g_e \hat{S}_k | \Psi \rangle , \ k = x, y, z \tag{6.242}$$

The elements of g-matrix are obtained as:

$$g_{kz} = 2c\Phi_{11}^k, \; g_{ky} = -2cIm\left(\Phi_{12}^k\right), \; g_{kx} = 2cRe\left(\Phi_{12}^k\right) \tag{6.243}$$

Then, the true tensor G is built from g-matrices:

$$G = gg^T \tag{6.244}$$

G is subjected further to diagonalization yielding positive eigenvalues, the square roots of which give the principal values of g-matrix.

$$g_{xx} = \sqrt{G_{xx}}, \; g_{yy} = \sqrt{G_{yy}}, \; g_{zz} = \sqrt{G_{zz}} \tag{6.245}$$

A typical mrci block of the input file for a g-tensor calculation should look as the following:

```
%mrci    ewin -4,1000
         citype mrci
         cimode direct2
         intmode fulltrafo
         solver diis
         etol 1e-8
         rtol 1e-8
         tsel 1e-6
         tpre 1e-5
         soc
           PrintLevel 2
           GTensor true       # make g-tensor calculations
           NDoubGTensor 3     # number of Kramers doublets to account
                              # for every pair a separate
                              # calculation is performed
         end
         newblock 2 *
           excitations cisd
           nroots 10
           refs  cas(9,8) end
         end
   end
```

The result for the first Kramers pair is printed as follows:

```
     -------------------
      KRAMERS PAIR 1 :
     -------------------
     Matrix elements Re<1|S|1>    0.063898  -0.233970   0.970139
     Matrix elements Re<1|S|2>   -0.255627   0.935856   0.242538
     Matrix elements Im<1|S|2>    0.964660   0.263492   0.000010
     Matrix elements Re<1|L|1>   -0.000083   0.000305   0.000001
```

```
   Matrix elements Re<1|L|2>     0.000333  -0.001220   0.000000
   Matrix elements Im<1|L|2>    -0.001257  -0.000343   0.000000


   -------------------
   ELECTRONIC G-MATRIX
   -------------------


   raw-matrix g-matrix:
     -0.511181     1.871439     0.485640
     -1.929040    -0.526908    -0.000020
      0.127777    -0.467873     1.942530
   diagonalized g**2 matrix (sqrt of eigenvalues taken):
      1.999705     1.999707     2.002316


     -0.022115     0.999755    -0.000000
     -0.999755    -0.022115    -0.000000
     -0.000000     0.000000     1.000000


   g-Shifts in ppm (for completeness; 1e6*(g[i]-gel))
       g1 =    -2613.9
       g2 =    -2612.7
       g3 =       -3.0

   Isotropic g-factor :  2.000576
```

Here for the L and S matrix elements indexes 1 and 2 are assumed to denote Kramers partners, and three numbers in the first row stand for x, y, and z contributions.

6.17.2.4 Relativistic Picture Change in Douglas Kroll SOC and Zeeman Operators

The DKH correction to the SOC operator is implemented in ORCA as a correction to the one-electron part of the SOMF operator. The DKH transformation is performed up to the second order, and the two-electron part in our implementation is left untransformed. However, the electronic density employed for evaluating the SOMF matrix elements is obtained from a scalar relativistic calculation. The inclusion of the DKH correction is controlled by the picturechange key in the rel block:

```
%rel method DKH                # relativistic method
     picturechange 2           # include the DKH correction to SOC
     end
```

The 'picturechange' key can be set to 0, 1, and 2 for no picture change, the first order, and the second order DKH transformations of the SOC operator.

With 'picturechange' set to 1 or 2 the DKH correction are applied in the first order to the Zeeman operator. This correction has a visible effect on calculated g-tensors for molecules containing third-row and heavier atoms.

6.17.2.5 MCD and Absorption Spectra Calculation and Generation

The MRCI module of the ORCA program allows calculating MCD spectra and the SOC effects on absorption spectra. The formalism is described in detail by Ganyushin and Neese [317]. The approach is based on the direct calculation of the transition energies and transition probabilities between the magnetic levels. Namely, the differential absorption of lcp- and rcp photons for transitions from a manifold of initial states A to a manifold of final states J. Using Fermi's golden rule, the Franck-Condon approximation, assuming a pure electronic dipole mechanism and accounting for the Boltzmann populations of the energy levels, the basic equation of MCD spectroscopy may be written as (atomic units are used throughout):

$$\frac{\Delta \varepsilon}{E} = \gamma \sum_{a,j} \left(N_a - N_j \right) \left(\left| \langle \Psi_a \left| m_{LCP} \right| \Psi_j \rangle \right|^2 - \left| \langle \Psi_a \left| m_{RCP} \right| \Psi_j \rangle \right|^2 \right) f \left(E \right) \tag{6.246}$$

Here a and j label members of the initial and state manifold probed in the experiments.

$$N_a \left(B, T \right) = \frac{\exp \left(-E_a / kT \right)}{\sum_i \exp \left(-E_i / kT \right)} \tag{6.247}$$

denotes the Boltzmann population and if the a-th ground state sublevel at energy E_a, $f \left(E \right)$ stands for a line shape function, and γ denotes a collection of constants. The electric dipole operators are given by:

$$m_{LCP} \equiv m_x - i m_y \tag{6.248}$$

$$m_{RCP} \equiv m_x + i m_y \tag{6.249}$$

They represent linear combinations of the dipole moment operator:

$$\vec{m} = \sum_N Z_N \vec{R}_N - \sum_i \vec{r_i} \tag{6.250}$$

where N and i denotes summations of nuclei (at positions $\vec{R}_N$ with charges Z_N) and electrons (at positions $\vec{r_i}$) respectively. The calculated transition dipole moment are subjected to the space averaging over the Euler angles which is performed by a simple summation over three angular grids.

$$\left(\frac{\Delta \varepsilon}{E} \right)_{ev} = \frac{1}{8\pi^2} \int_{\psi=0}^{2\pi} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \left(\frac{\Delta \varepsilon}{E} \right) \sin \theta d\theta d\phi d\psi \approx \sum_{\mu\eta\tau} \left(\frac{\Delta \varepsilon}{E} \right)_{\mu\eta\tau} \sin \theta_\tau \tag{6.251}$$

Finally, every transition is approximated by a Gaussian curve with a definite Gaussian shape width parameter. Hence, the final calculated MCD spectrum arises from the superposition of these curves.

As an illustration, consider calculation of a classical example of MCD spectrum of $[\text{Fe(CN)}_6]^{3-}$. The mrci block of the input file is presented below.

```
%mrci   ewin -4,10000
        citype mrddci2
        intmode ritrafo
        Tsel  1e-6
        Tpre  1e-5
        etol 1e-8
        rtol 1e-8
        cimode direct2
        maxmemint 300
        solver diis
        davidsonopt 0
        nguessmat 150
        MaxIter 50
        LevelShift 0.5
        PrintLevel 3
        soc
        printlevel 3
        mcd true           # perform the MCD calculation
        NInitStates 24     # number of SOC and SSC state to account
                           # Starts from the lowest state
        NPointsTheta 10    # number of integration point for
        NPointsPhi 10      # Euler angles
        NPointsPsi 10      #
        B 43500            # experimental magnetic field strength
                           # in Gauss
        Temperature 299.0 # experimental temperature (in K)
        end
        newblock 2 *
         nroots 12
         excitations cisd
         refs cas(23,12) end
         end
        end
```

The parameters B and Temperature can be assigned in pairs, i.e. B = 1000, 2000, 3000..., Temperature = 4, 10, 300.... The program calculates the MCD and absorption spectra for every pair. Now for every point of the integration grid the program prints out the Euler angles, the orientation of the magnetic field in the coordinate system of a molecule, and the energy levels.

```
Psi = 36.000 Phi = 72.000 Theta = 20.000


Bx = 8745.0 By = 12036.5 Bz = 40876.6


Energy levels (cm-1,eV):Boltzmann populations for T = 299.000   K
```

```
 0 :          0.000      0.0000           4.53e-01
 1 :          3.943      0.0005           4.45e-01
 2 :        454.228      0.0563           5.09e-02
 3 :        454.745      0.0564           5.08e-02
 4 :       1592.142      0.1974           2.13e-04
 5 :       1595.272      0.1978           2.10e-04
 6 :      25956.363      3.2182           2.59e-55
 7 :      25958.427      3.2184           2.56e-55
 8 :      25985.656      3.2218           2.25e-55
 9 :      25987.277      3.2220           2.23e-55
10 :      26070.268      3.2323           1.49e-55
11 :      26071.484      3.2325           1.49e-55
12 :      31976.645      3.9646           6.78e-68
13 :      31979.948      3.9650           6.67e-68
14 :      32018.008      3.9697           5.56e-68
15 :      32021.074      3.9701           5.48e-68
16 :      32153.427      3.9865           2.90e-68
17 :      32157.233      3.9870           2.84e-68
18 :      42299.325      5.2444           1.81e-89
19 :      42303.461      5.2450           1.78e-89
20 :      42346.521      5.2503           1.45e-89
21 :      42348.023      5.2505           1.44e-89
22 :      42456.119      5.2639           8.53e-90
23 :      42456.642      5.2640           8.51e-90
```

In the next lines, ORCA calculates the strength of lcp and rcp transitions and prints the transition energies, the difference between lcp and rcp transitions (denoted as C), and sum of lcp and rcp transitions (denoted as D), and C by D ratio.

```
  dE       Na        C           D         C/D


   0 ->  1        3.943 4.53e-01  1.14e-13  8.13e-13   0.00e+00
   0 ->  2      454.228 4.53e-01  5.01e-09  9.90e-09   5.06e-01
   0 ->  3      454.745 4.53e-01 -4.65e-09  7.00e-09  -6.65e-01
   0 ->  4     1592.142 4.53e-01 -8.80e-08  1.02e-07  -8.67e-01
   0 ->  5     1595.272 4.53e-01 -2.29e-08  2.97e-08  -7.71e-01
   0 ->  6    25956.363 4.53e-01  1.22e+01  9.60e+01   1.27e-01
   0 ->  7    25958.427 4.53e-01  3.44e+01  3.52e+01   9.77e-01
   0 ->  8    25985.656 4.53e-01  3.83e+01  1.70e+02   2.25e-01
   0 ->  9    25987.277 4.53e-01 -7.73e+00  6.03e+01  -1.28e-01
   0 ->10    26070.268 4.53e-01 -6.11e+00  2.85e+01  -2.14e-01
   0 ->11    26071.484 4.53e-01  6.17e+00  9.21e+00   6.70e-01
   0 ->12    31976.645 4.53e-01  2.45e+01  6.21e+01   3.95e-01
   0 ->13    31979.948 4.53e-01 -6.58e+01  6.93e+01  -9.50e-01
```

```
    0 ->14      32018.008 4.53e-01  3.42e-01  1.07e+02   3.21e-03
    0 ->15      32021.074 4.53e-01 -6.16e+00  3.24e+01  -1.90e-01
    0 ->16      32153.427 4.53e-01 -4.73e+01  1.37e+02  -3.46e-01
    0 ->17      32157.233 4.53e-01 -1.02e+00  5.97e+01  -1.71e-02
    0 ->18      42299.325 4.53e-01  6.47e+00  2.11e+01   3.07e-01
    0 ->19      42303.461 4.53e-01 -2.59e+00  7.61e+00  -3.40e-01
    0 ->20      42346.521 4.53e-01  1.90e+01  8.99e+01   2.11e-01
    0 ->21      42348.023 4.53e-01  3.36e+00  3.55e+00   9.48e-01
    0 ->22      42456.119 4.53e-01  2.52e-01  4.86e-01   5.20e-01
    0 ->23      42456.642 4.53e-01 -2.01e+00  2.91e+00  -6.91e-01
    1 -> 2        450.285 4.45e-01  4.59e-09  6.87e-09   6.69e-01
    1 -> 3        450.802 4.45e-01 -4.96e-09  9.73e-09  -5.09e-01
```

All C and D values are copied additionally into the text files input.1.mcd, input.2.mcd..., for every pair of Temperature and B parameters. These files contain the energies and C and D values for every calculated transition. These files are used by the program orca_mapspc to calculate the spectra lines. The orca_mapspc program generates from the raw transitions data into spectra lines. The main parameters of the orca_mapspc program are described in section 7.18.1. A typical usage of the orca_mapspc program for MCD spectra calculation for the current example may look as the following:

```
    orca_mapspc input.1.mcd MCD -x020000 -x150000 -w2000
```

Here the interval for the spectra generation is set from 20000 cm^{-1} to 50000 cm^{-1}, and the line shape parameter is set to 2000 cm^{-1}.

Very often, it is desirable to assign different line width parameters to different peaks of the spectra to obtain a better fitting to experiment. orca_mapspc can read the line shape parameters from a simple text file named as input.1.mcd.inp. This file should contain the energy intervals (in cm^{-1}) and the line shape parameters for this energy interval in the form of:

```
    20000 35000 1000
    35000 40000 2000
    40000 50000 1000
```

This file should not be specified in the executing command; orca_mapspc checks for its presence automatically:

```
    orca_mapspc input.1.mcd MCD -x020000 -x150000
    Mode is MCD
    Number of peaks           ... 276001
    Start wavenumber [cm-1]    ...  20000.0
    Stop wavenumber [cm-1]     ...  50000.0
    Line width parameters are taken from the file:input.1.mcd.inp
    Number of points          ... 1024
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

447

Finally, the orca_mapspc program generates the output text file input.1.mcd.dat which contains seven columns of numbers: transition energies, intensities of MCD transitions (the MCD spectrum), intensities of absorption transitions (the absorption spectrum), the ratio between the MCD and absorption intensities, and the last three columns represent the 'sticks' of the corresponding transitions.

Energy	C	D	C/D	C	D	E/D
24310.8	0.6673	980.2678	0.0006	0.0000	0.0000	0.0000
24340.1	0.8471	1174.3637	0.0007	-0.0001	0.0129	-0.0112
24369.5	1.0664	1408.5788	0.0007	0.0001	0.0281	0.0033
24398.8	1.3325	1690.5275	0.0007	0.0000	0.0000	0.0000
24428.1	1.6542	2029.0152	0.0008	0.0000	0.0000	0.0000
24457.4	2.0416	2434.1699	0.0008	0.0000	0.0332	0.0003

Now the MCD and the absorption spectra can be plotted with a suitable graphical program, for instance with the Origin program.
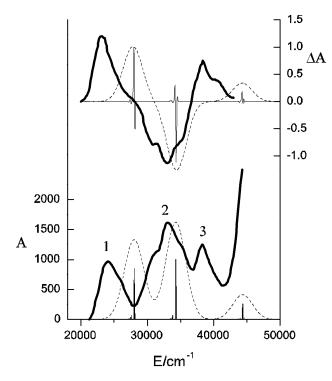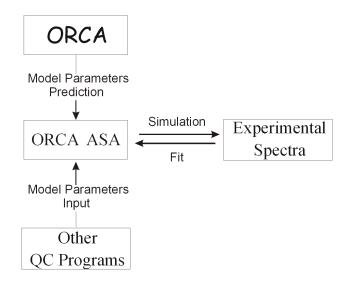
Figure 6.11: Calculated MCD and absorption spectra of $[Fe(CN)_6]^{3-}$ (dash lines) compared to experimental spectra (solid lines).

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

In this section various aspects of the simulation and fit of optical spectra, including absorption, fluorescence, and resonance Raman are considered. This part of the ORCA is fairly autonomous and can also be used in a data analysis

context, not only in a "quantum chemistry" mode. The program is called `orca_asa`, where ASA stands for "<u>A</u>dvanced <u>S</u>pectral <u>A</u>nalysis". The program was entirely designed by Dr. Taras Petrenko.

The general philosophy is as follows: An ORCA run produces the necessary data to be fed into the orca_asa program and writes an initial input file. This input file may be used to directly run orca_asa in order to predict an absorption, fluorescence or resonance Raman spectrum. Alternatively, the input file may be edited to change the parameters used in the simulations. Last – but certainly not least – the orca_asa program can be used to perform a fit of the model parameters relative to experimental data.

All examples below are taken from the paper:

Petrenko, T.; **Neese, F.** (**2007**) Analysis and Prediction of Absorption Bandshapes, Fluorescence Bandshapes, Resonance Raman Intensities and Excitation Profiles using the Time Dependent Theory of Electronic Spectroscopy. *J. Chem. Phys.*, <u>127</u>, 164319

Which must be cited if you perform any work with the orca_asa program!

6.18.1 General Description of the Program

The program input comprises the following information: (1) model and specification of the model parameters characterizing the electronic structure of a molecule, as well as lineshape factors; (2) spectral ranges and resolution for simulations; (3) specification of vibrational transitions for rR excitation profile and spectra generation; (4) certain algorithm-selecting options depending on the model; (5) fitting options.

All optional parameters (1)-(3) are given in the **%sim** block, and fitting options are in the **%fit** block. The model parameters are specified within various blocks that will be described below. The program *orca_asa* is interfaced to ORCA and inherits its input style. The input for *orca_asa* run can be also generated upon ORCA run.

The current implementation features so called "simple", "independent mode, displaced harmonic oscillator" (IMDHO), and "independent mode, displaced harmonic oscillator with frequency alteration" (IMDHOFA) models.

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

449

6.18.2 Spectral Simulation Procedures: Input Structure and Model Parameters

6.18.2.1 Example: Simple Mode

This model represents the simplest approach which is conventionally used in analysis of absorption spectra. It neglects vibrational structure of electronic transitions and approximates each individual electronic band by a standard lineshape, typically a Gaussian, Lorentzian or mixed (Voigt) function. This model can only make sense if vibrational progressions are not resolved in electronic spectra. Upon this approximation the intensity of absorption spectrum depends on the energy of the incident photon (E_L), the electronic transition energy (E_T), the e transition electric dipole moment (**M**,evaluated at the ground-state equilibrium geometry). Lineshape factors are specified by homogeneous linewidth Γ and standard deviation parameter Θ corresponding to Gaussian distribution of transition energies. The following example illustrates a simple input for simulation of absorption bandshapes using various intensity and lineshape parameters.

```
# example001.inp
#
# Input file to generate absorption spectrum consisting
# of 3 bands   with different lineshape factors:
#
# 1. Lorentzian centered at 18000cm**-1 (damping factor Gamma= 100 cm**-1)
# 2. Gaussian   centered at 20000cm**-1
#    (standard deviation Sigma= 100 cm**-1)
# 3. Mixed Gaussian-Lorentzian band representing Voight profile
#    centered at 21000 cm**-1

%sim
        Model Simple

        # Spectral range for absorption simulation:
        AbsRange     17000.0,   23000.0

        # Number of points to simulate absorption spectrum:
        NAbsPoints  2000

end



#------------------------------------------------------------------------
#  Transition        Gamma    Sigma  Transition Dipole Moment (atomic unit)
#  Energy (cm**-1)  (cm**-1) (cm**-1)     Mx         My         Mz
#------------------------------------------------------------------------

$el_states
3     # number of electronic states
   1  18000.0        100.00      0.0       1.0        0.0        0.0
   2  20000.0          0.00    100.0       1.0        0.0        0.0
   3  22000.0         50.00     50.0       1.0        0.0        0.0
```

The parameters of of the final electronic states reached by the respective transitions are specified in the `$el_states` block. The spectral range and resolution used in the calculation are defined by the `AbsRange` and `NAbsPoints` keywords in `%sim` block. The calculation of the absorption spectrum is automatically invoked if `NAbsPoints>1`. After the *orca_asa* run you will find in your directory file **example001.abs.dat** containing absorption spectrum in simple
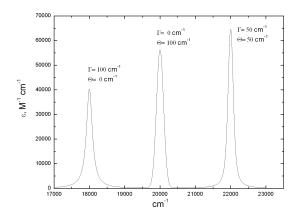
Figure 6.12: Absorption spectrum generated after orca_asa run on file `example001.inp.` Three bands have different lineshape pararameters. Note that although all transitions are characterized by the same transition electric dipole moment their intensities are scaled proportionally to the transition energies.

two-column ASCII format suitable to be plotted with any spreadsheet program. Absorption spectra corresponding to individual electronic transitions are stored in file `example001.abs.as.dat` (the suffix "`as`" stands for "All States").

The output of the program run also contains information about oscillator strengths and full-width-half-maximum (FWHM) parameters corresponding to each electronic band:

```
      ----------------------------------------------
      State      EV       fosc    Stokes shift
               (cm**-1)            (cm**-1)
      ----------------------------------------------
        1:    18000.00  0.054676     0.00
        2:    20000.00  0.060751     0.00
        3:    22000.00  0.066826     0.00



      ----------------------------------------
      BROADENING PARAMETETRS (cm**-1)
      ----------------------------------------
      State    Gamma    Sigma    FWHM
      ----------------------------------------
        1:      100.00     0.00   200.00
        2:        0.00   100.00   235.48
        3:       50.00    50.00   180.07
```

Note that although all three types of lineshape functions are symmetric this is not true for the overall shapes of individual absorption bands since the extinction coefficient (absorption cross-section) is also proportional to the incident photon energy. Therefore, if the linewidth is larger than 10% of the peak energy the asymmetry of the electronic band can be quite noticeable.

6.18.2.2 Example: Modelling of Absorption and Fluorescence Spectra within the IMDHO Model

The IMDHO model is the simplest approach that successfully allows for the prediction of vibrational structure in electronic spectra as well as rR intensities for a large variety of real systems. This model assumes:

1. harmonic ground- and excited-state potential energy surfaces;

2. origin shift of the excited-state potential energy surface relative to the ground-state one;

3. no vibrational frequency alteration or normal mode rotation occurs in the excited state;

4. no coordinate dependence of the electronic transition dipole moment.

In addition to the parameters that enter the "Simple model" defined above it requires some information about the vibrational degrees of freedom. The required information consists of the ground-state vibrational frequencies $\{\omega_{gm}\}$ and (dimensionless) origin shifts $\{\Delta_{mi}\}$, where i and m refer to electronic states and normal modes respectively. Δ is expressed in terms of dimensionless normal coordinates. Accordingly, for the IMDHO model one has to specify the following blocks

- The $el_states block contains the parameters E_T, Γ,Θ, $\mathbf{M}$ for each electronic state. By default E_T is assumed to be adiabatic minima separation energy. Alternatively, it can be redefined to denote for the vertical transition energy.This is achieved by specifiying the keyword EnInput=EV in the %sim block.

- A $vib_freq_gs block specifies ground-state vibrational frequencies.

- A $sdnc block contains parameters $\{\Delta_{mi}\}$ in matrix form such that the i-th column represents the dimensionless displacements along all normal modes for the i-th excited-state PES.

The file example002.inp provides the input for simulation of absorption and fluorescence spectra of a system characterized by significant displacements of the excited-state origin along 5 normal coordinates.

```
# example002.inp
#
# Input file for simulation  of vibrational structure
# in absorption and fluorescence spectra assuming
# origin shift of excited PES along 5 normal coordinates.
# The simulated spectra closely reproduce the experimental
# optical bandshapes for the tetracene molecule.
#
%sim
     Model IMDHO

     # spectral range for absorption simulation (cm**-1)
     AbsRange    20000.0, 27000.0
     NAbsPoints  2000              # number of points in absorption spectrum


     # spectral range for simulation of fluorescence (cm**-1)
     FlRange     22000.0, 16000.0
     NFlPoints   2000      # number of points in fluorescence spectrum


     # the following options require the spectra to be normalized
     # so that their maxima are equal to 1.0
     AbsScaleMode Rel
```

```
        FlScaleMode  Rel    # default  for fluorescence


        # for absorption spectrum the default option is AbsScaleMode= Ext
        # which stands for extinction coefficient


   end


   #-----------------------------------------------------------------------
   #   Transition        Gamma     Sigma   Transition Dipole Moment (atomic unit)
   #   Energy (cm**-1)  (cm**-1) (cm**-1)      Mx          My         Mz
   #-----------------------------------------------------------------------


   $el_states
   1
      1  21140.0           50.00    100.0        1.0         0.0         0.0



   # Block specifying Stokes Shift parameter for each electronic state
   # This information is optional
   $ss
   1   # number of excited states
     1   300.0       # the Stokes shift for the 1st electronic transition


   # Block providing the values of VIBrational FREQuencies
   # for 5 Ground-State normal modes.
   # Obligatory for IMDHO and IMDHOFA models.
   $vib_freq_gs
   5
        1       310.0
        2      1193.0
        3      1386.0
        4      1500.0
        5      1530.0



   # Block specifying origin Shift of the excite-state PES
   # along each normal mode in terms of the ground-state
   # Dimensionless Normal Coordinates
   # Obligatory for IMDHO and IMDHOFA models.
   $sdnc
   5 1
                     1
        1        0.698
        2       -0.574
        3        0.932
        4       -0.692
        5        0.561
```

The calculation of absorption and fluorescence spectra is automatically invoked if the parameters `NAbsPoints`>1 and `NFlPoints`>1. The input file also contains the optional block `$ss` which specifies the Stokes shift λ for each electronic transition. This parameter is equal to the energy separation between the 0-0 vibrational peaks in the absorption and fluorescence spectra as shown in Figure 6.13 . In general λ accounts for solvent induced effects as well as unresolved vibrational structure corresponding to low-frequency modes that are not specified in the input. Note that we have specified parameters `AbsScaleMode=Rel` and `FlScaleMode=Rel` in `%sim` block in order to ensure that the simulated spectra are normalized to unity. The calculated absorption and fluorescence spectra are stored in `example002.abs.dat`

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

453

and `example002.fl.dat` files, respectively.
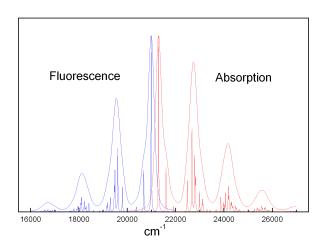
Figure 6.13: Absorption and fluorescence spectra generated after orca_asa run on the file `example002.inp.` If the homogeneous broadening is set to be $\Gamma = 10cm^{-1}$ one can resolve underlying vibrational structure and identify various fundamental and combination transitions.

6.18.2.3 Example: Modelling of Absorption and Fluorescence Spectra within the IMDHOFA Model

IMDHOFA (Independent Mode Displaced Harmonic Oscillators with Frequency Alteration) is based on the same assumptions as the IMDHO model except for vibrational frequency alteration in excited state can take place. The file `example003.inp` features almost the same input parameters as `example002.inp`. The IMDHOFA model is invoked by the keyword `Model=IMDHOFA` in the `%sim` block. Additionally, one has to provide the obligatory block `$vib_freq_es`. It contains the excited-state vibrational frequencies $\{\omega_{emi}\}$ in matrix form such that the i-th column represents the vibrational frequencies of all normal modes for the i-th excited-state PES.

```
# Block providing the values of VIBrational FREQuencies
# for 5 Excited-State normal modes.
# Obligatory for IMDHOFA model.

$vib_freq_es
5  1           # number of modes and number of excited states
             1
      1      410.0
      2     1293.0
      3     1400.0
      4     1600.0
      5     1730.0
```
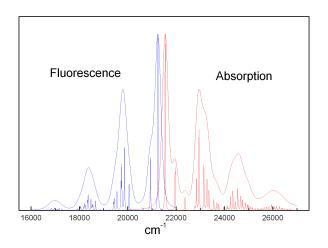
Figure 6.14: Absorption and fluorescence spectra generated after orca_asa run on the file `example003.inp`. Also, the high-resolution spectra corresponding to homogeneous broadening $\Gamma = 10cm^{-1}$ are shown.

6.18.2.4 Example: Modelling of Effective Broadening, Effective Stokes Shift and Temperature Effects in Absorption and Fluorescence Spectra within the IMDHO Model

For the IMDHO model the *orca_asa* is capable to model absorption and emission spectra in the finite-temperature approximation. While the keyword `Model=IMDHO` assumes the zero-temperature approximation, the value of `Model=IMDHOT` invokes the calculation of the spectra for the finite temperature which is specified by the paramter `TK` in the block `%sim`:

```
# example004.inp
#
#
%sim
     Model IMDHOT
     TK 300 # temperature (in Kelvin)

     # spectral range for absorption simulation (cm**-1)
     AbsRange    18000.0, 35000.0
     NAbsPoints  5000                # number of points in absorption spectrum

     # spectral range for simulation of fluorescence (cm**-1)
     FlRange     22000.0, 10000.0
     NFlPoints   5000     # number of points in fluorescence spectrum

     # the following options require the spectra to be normalized
     # so that their maxima are equal to 1.0
     AbsScaleMode Rel
     FlScaleMode  Rel   # default  for fluorescence
end
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

455

```
#-------------------------------------------------------------------------
#   Transition        Gamma    Sigma  Transition Dipole Moment (atomic unit)
#   Energy (cm**-1)  (cm**-1) (cm**-1)    Mx          My          Mz
#-------------------------------------------------------------------------

$el_states
1
   1  21140.0        50.00    100.0       1.0         0.0         0.0



# Block specifying Stokes Shift parameter for each electronic state
$ss
1   # number of excited states
   1   300.0       # the Stokes shift for the 1st electronic transition

# Block providing the values of VIBrational FREQuencies
# for 10 Ground-State normal modes.
$vib_freq_gs
10
     1        30.0
     2        80.0
     3       100.0
     4       120.0
     5       130.0
     6       140.0
     7       160.0
     8       200.0
     9       310.0
    10      1300.0

# Block specifying origin Shift of the excite-state PES
# along each normal mode in terms of the ground-state
# Dimensionless Normal Coordinates
$sdnc
 10  1
              1
     1       2.5
     2       2.0
     3       1.8
     4       1.9
     5       1.5
     6       1.9
     7       2.4
     8       1.9
     9       2.5
    10       0.9
```

This example illustrates a typical situation in large molecules which feature a number of low frequency modes with significant values of dimensionless displacements for a given excited-state PES. In the case of high density of vibrational states with frequencies below or comparable to the intrincic value of FWHM (determined by Γ and Θ) the vibrational progression is unresolved, whereby the spectra become very diffuse and show large separation between the maxima of absorption and emission spectra (Figure 6.14). Besides , upon the condition $h\nu_i \leqslant kT$ the effective bandwidths and positions of maxima in the spectra can be strongly subject to temperature effects.

The effective Stokes shift and linewidth parameters which are evaluated in the simple self-consistent procedure are given in the output of the *orca_asa* run:
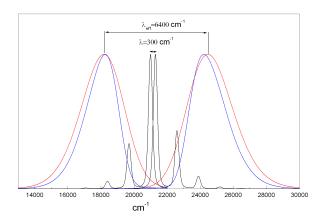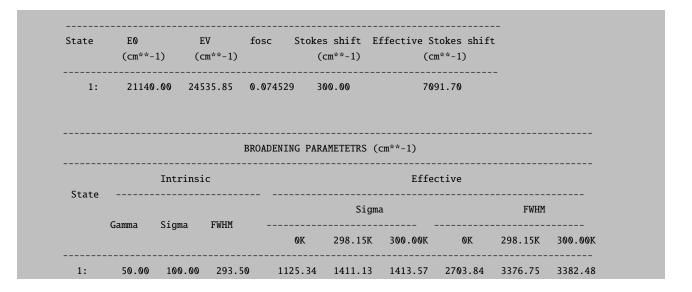
Figure 6.15: Absorption and fluorescence spectra for T=0 K (blue) and T=300 K (red) generated after orca_asa run on the file `example004.inp`. Black lines show spectra corresponding to the case where all low-frequency modes were excluded from the calculation.

```
-------------------------------------------------------------------------------
State      E0          EV        fosc     Stokes shift  Effective Stokes shift
          (cm**-1)    (cm**-1)              (cm**-1)          (cm**-1)
-------------------------------------------------------------------------------
  1:      21140.00   24535.85   0.074529    300.00            7091.70



----------------------------------------------------------------------------------
                          BROADENING PARAMETETRS (cm**-1)
----------------------------------------------------------------------------------
              Intrinsic                                Effective
State   --------------------------   --------------------------------------------------

                                           Sigma                        FWHM
        Gamma   Sigma    FWHM      --------------------------   --------------------------
                                    0K     298.15K  300.00K     0K     298.15K  300.00K
-------------------------------------------------------------------------------------
  1:     50.00   100.00   293.50   1125.34  1411.13  1413.57  2703.84  3376.75  3382.48
```

Note that the evaluation of the effective parameters is rather approximate and these values can noticeable deviate from those which can be directly deduced from the calculated spectra. However, such an information usually provides the proper order of magnitude of the effective vibronic broadening and Stokes shift. As indicated in the program output above, the effective bandshape has predominantly a Gaussian character which varies with the temperature so that $\Theta = 1125 cm^{-1}$ (T=0 K) and $\Theta = 1414 cm^{-1}$ (T=300 K). Indeed, as shown in Figure 6.16 the absorption spectrum at T=300 K can be well fitted using Gaussian lineshape with $\Theta = 1388 cm^{-1}$ (FWHM=3270 cm^{-1}). One can see that at higher temperatures the deviation between the spectrum and its Gauss fit becomes even smaller.

In molecules the normal distribution of the electronic transition energies in the ensemble would give rise to a Gaussian bandshape of the absorption band. However, the corresponding standard deviation is expected to be of the order of 100 cm^{-1}, whereby a typical Gaussian bandwidth of the order of 1000 cm^{-1} appears to result from unresolved vibronic progression. In general, this statement is supported by quantum chemical calculation of the model parameters. In principle the effective bandwidth parameters can also be used for characterization and assignement of individual electronic bands.

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program
457
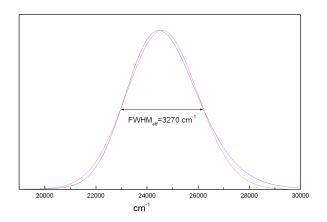
Figure 6.16: Absorption spectrum (blue) for T=300 K generated after orca_asa run on the file
`example004.inp`. Red line represents the Gauss-fit of the calculated spectrum.

6.18.2.5 Example: Modelling of Absorption and Resonance Raman Spectra for the $1\text{-}^1A_g \to$ 1-^1B_u Transition in trans-1,3,5-Hexatriene

The hexatriene molecule is characterized by 9 totally-symmetric normal modes which dominate vibrational structure in absorption and are active in rR spectra corresponding to the strongly dipole-allowed $1 -^1 A_g \to 1 -^1 B_u$ transition around 40000 cm^{-1} . Except for some peculiarities related to the neglect of normal mode rotations in the excited state the optical spectra are quite satisfactorily described by the IMDHO model.

The following input exemplifies simulation of absorption spectrum and rR spectra for an arbitrary predefined number of excitation energies.

```
#
# example005.inp
#
# input for simulation of absorption and resonance Raman spectra
# using experimental values of transition energy and displacement
# parameters corresponding to the strongly allowed 1-1Ag 1-1Bu transition
# in trans-1,3,5-hexatriene
#
%sim
   Model IMDHO

   AbsRange     38000.0,   48000.0
   NAbsPoints   2000
   AbsScaleMode Rel

   # resonance Raman intensities will be calculated
   # for all vibrational states with excitation number
   # up to RamanOrder:
   RamanOrder  4

   # excitation energies (cm**-1) for which rR spectra will be calculated:
   RRSE 39500, 39800, 41400

   # full width half maximum of Raman bands in rR spectra (cm**-1):
```

```
    RRS_FWHM   10

    RSRange    0, 5000   # spectral range for simulation of rR spectra (cm**-1)
    NRRSPoints  5000     # number of points to simulate rR spectra (cm**-1)

  end


$el_states
1
   1  39800.0        150.00     0.0     1.0    0.0    0.0


$vib_freq_gs
9
      1        354.0
      2        444.0
      3        934.0
      4       1192.0
      5       1290.0
      6       1305.0
      7       1403.0
      8       1581.0
      9       1635.0

$sdnc
9 1
                     1
       1        0.55
       2        0.23
       3        0.23
       4        0.82
       5        0.485
       6        0.00
       7        0.085
       8        0.38
       9        1.32
```

After the *orca_asa* run the following files will be created:

- `example005abs.dat` contains the simulated absorption spectrum. It is shown in Figure 6.17.

- `example005.o4.rrs.39500.dat`, `example005.o4.rrs.39800.dat` and `example005.o4.rrs.41400.dat` contain the simulated rR spectra for excitation energies at 39500, 39800 and 41400 cm^{-1}, respectively. The suffix "**o4**" stands for the order of Raman scattering specified in the input by keyword `RamanOrder=4`. The rR specta are shown in Figure 6.18.

- `example005.o4.rrs.39500.stk`, `example005.o4.rrs.39800.stk` and `example005.o4.rrs.41400.stk` provide Raman shifts and intensities for each vibrational transition. Corresponding vibrational states are specified by the quantum numbers of excited modes.

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

459
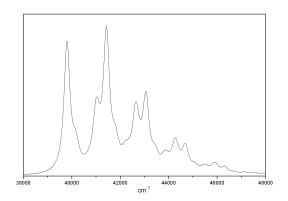
Figure 6.17: Absorption spectrum corresponding to $1 -^1 A_g \rightarrow 1 -^1 B_u$ transition in *trans*-1,3,5-hexatriene generated after orca_asa run on the file `example005.inp`.
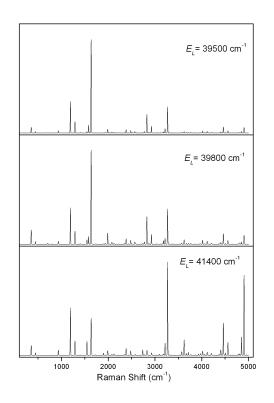
Figure 6.18: Resonance Raman spectra for 3 different excitation energies which fall in resonance with $1 -^1 A_g \rightarrow 1 -^1 B_u$ transition in *trans*-1,3,5-hexatriene.

NOTE

- By default the program provides rR spectra on an arbitrary scale since only relative rR intensities within a single rR spectrum are of major concern in most practical cases. However, one can put rR spectra corresponding to different excitation energies on the same intensity scale by providing the keyword `RSISM=ASR` in `%sim` block

(RSISM – Raman Spectra Intensity Scaling Mode; ASR – All Spectra Relative). By default `RSISM=SSR` (SSR – Single Spectrum Relative) for which each rR spectrum is normalized so that the most intense band in it has intensity 1.0. The relative intensities of bands in rR spectra measured for different excitation energies can be compared if they are appropriately normalized relative to the intensity of a reference signal (e.g. Raman band of the solvent). We also keep in mind the possibility to extend our methodology in order to provide the absolute measure of rR intensities in terms of the full or differential cross-sections.

- Within the harmonic model, for a single electronic state neither relative rR intensities nor absorption bandshapes in the case of `AbsScaleMode=Rel` do depend on the values of the electronic transition dipole moment (unless it is precisely zero).

In the example above resonance Raman spectra have been generated for all vibrational transitions with total excitation number up to the value specified by the parameter `RamanOrder`. Its is also possible to make explicit specification of vibrational states corresponding to various fundamental, overtone and combination bands via the `$rr_vib_states` block. In such a case rR spectra involving only these vibrational transitions will be generated separately.

```
$rr_vib_states 5 # total number of vibrational transitions
 1
    modes 1
    quanta 1; # final vibrational state for the fundamental band corresponding to mode 1
 2
    modes 9
    quanta 1; # final vibrational state for the fundamental band corresponding to mode 9
 3
    modes 3, 4
    quanta 1, 1; # final vibrational state for the combination band involving single
                 # excitations in modes 3 and 4
 4
    modes 5
    quanta 3; # final vibrational state for the second overtone band corresponding to
              # mode 5
 5
     modes 1, 5,9
     quanta 1,2, 1; # final vibrational state for the combination band involving single
                    # excitations in modes 1 and 2, and double excitation in mode 5
```

Each vibrational transition is specified via the subblock which has the following structure:

```
    k
            modes m1,m2,...mn
            quanta q1,q2,...qn;
```

This means that the k-th transition is characteriezed by excitation numbers q_i for modes m_i so that corresponding Raman shift is equal to $\nu = \sum q_i \nu_i$, where ν_i is vibrational frequency of the mode m_i.

After the *orca_asa* run the following files will be created in addition:

- `example005.us.rrs.39500.dat`, `example005.us.rrs.39800.dat` and `example005.us.rrs.41400.dat` contain the simulated rR spectra involving only vibrational transitions specified in the `$rr_vib_states` block, for excitations energies at 39500, 39800 and 41400 cm^{-1}, respectively. The suffix "us" stands for "User specified vibrational States".

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

461

- `example005.us.rrs.39500.stk`, `example005.us.rrs.39800.stk` and `example005.us.rrs.41400.stk` provide Raman shifts and intensities for each vibrational transition specified in the `$rr_vib_states` block.

6.18.2.6 Example: Modelling of Absorption Spectrum and Resonance Raman Profiles for the $1\text{-}^1\mathbf{A}_g \to 1\text{-}^1\mathbf{B}_u$ Transition in trans-1,3,5-Hexatriene

The following example illustrates an input for simulation of absorption bandshape and resonance Raman profiles (RRP):

```
#
# example006.inp
#
# input for simulation of absorption and resonance Raman profiles
# using experimental values of transition energy and displacement
# parameters corresponding to the strongly allowed 1-1Ag 1-1Bu transition
# in trans-1,3,5-hexatriene
#


%sim
     Model IMDHO

     AbsRange      38000.0,    48000.0
     NAbsPoints   2000
     AbsScaleMode Rel


     RRPRange    38000.0, 48000.0 # spectral range for simulation of
                                  # rR profiles (cm**-1)

     NRRPPoints    2000 # number of points for simulation of rR profiles
     CAR 0.8

     RamanOrder 2

end


$el_states
1
   1  39800.0      150.00    0.0     1.0   0.0   0.0



$vib_freq_gs
9
     1      354.000000
     2      444.000000
     3      934.000000
     4     1192.000000
     5     1290.000000
     6     1305.000000
     7     1403.000000
     8     1581.000000
     9     1635.000000
```

```
    $sdnc
    9 1
                    1
          1        0.55
          2        0.23
          3        0.23
          4        0.82
          5        0.485
          6        0.00
          7        0.085
          8        0.38
          9        1.32


    $rr_vib_states 5 # total number of vibrational transitions
     1
        modes 1
        quanta 1;
     2
        modes 9
        quanta 1;
     3
        modes 3, 4
        quanta 1, 1;
     4
        modes 5
        quanta 3;
     5
         modes 1, 5,9
         quanta 1,2, 1;
```

The keyword `RamanOrder=2` will invoke generation of rR profiles for all vibrational transitions with total excitation number up to 2 in the range of excitation energies specified by the keywords `RRPRange` and `NRRPPoints`. Likewise, rR profiles for the vibrational states given in the `$rr_vib_states` block will be generated separately. Since in most cases only relative rR intensities are important, and one would be interested to compare absorption bandshape and shapes of individual rR profiles, the keyword `CAR = 0.8` is used to scale rR profiles for all vibrational transitions by a common factor in such a way that the ratio of the maximum of all rR intensities and the maximum of absorption band is equal to 0.8.

After the *orca_asa* run the following files will be created:

- `example006.abs.dat` contains the simulated absorption spectrum (Figure 6.19).

- `example006.o1.rrp.dat` and `example006.o2.rrp.dat` contain rR profiles for vibrational transitions with total excitation numbers 1 and 2, respectively. RR profiles for all fundamental bands (from the file `example006.o1.rrp.dat`) are shown in Figure 6.19.

- `example006.o1.info` and `example006.o1.info` contain specification of vibrational transitions with total excitation numbers 1 and 2, respectively, as well as corresponding Raman shifts.

- `example006.us.rrp.1.dat`--`example006.us.rrp.5.dat` contain rR profiles for vibrational transitions 1–5 specified in the `$rr_vib_states` block.

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

463
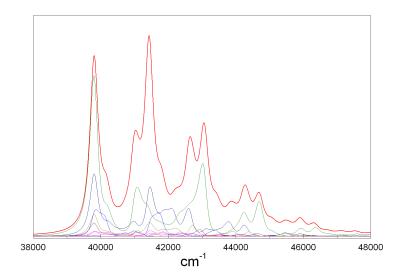
Figure 6.19: Absorption spectrum and resonance Raman profiles of fundamental bands corresponding to $1 -^1 A_g \rightarrow 1 -^1 B_u$ transition in *trans*-1,3,5-hexatriene.

6.18.3 Fitting of Experimental Spectra

6.18.3.1 Example: Gauss-Fit of Absorption Spectrum

An absorption spectrum basically consists of a number of absorption bands. Each absorption band corresponds to a transition of the ground electronic state to an excited electronic state. In molecules such transitions are usually considerably broadened. In many cases there will be overlapping bands and one would need to deconvolute the broad absorption envelope into contributions from individual transitions. Within the "Simple model" the *orca_asa* program enables fit of an absorption spectrum with a sum of standard lineshape functions (Gaussian, Lorentzian) or more general Voigt functions. In most cases, one simply performs a "Gauss-Fit". That is, it is assumed that the shape of each individual band is that of a Gaussian function. Then one applies as many (or as few) Gaussians as are necessary for an accurate representation of the absorption envelope. In order to explain the fitting procedures within the "Simple model" let us consider an experimental absorption spectrum in Figure 6.20:
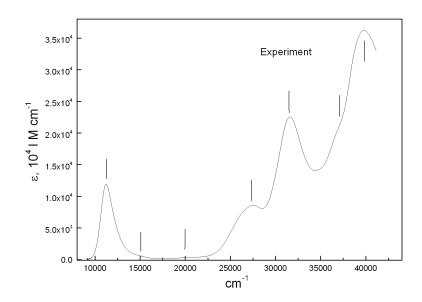
Figure 6.20: Experimental absorption spectrum. Bars indicate transition energies which were used for the initial guess in the input for spectral fitting.

As shown in Figure 6.20 one can identify roughly 7 electronic bands. The initial estimates of transition energies corresponding to the maxima and shoulders in the absorption spectrum (indicated by bars in Figure 6.20) and rather approximate values of inhomogeneous broadening and transition dipole moment components are specified in the $el_states block of the input file for the spectral fitting:

```
# example007.inp
#
# Input file for fitting of experimental absorption spectrum
#

%sim
        model Simple
end


%fit
        Fit true              # Global flag to turn on the fit
        AbsFit true           # Flag to include absorption into the fit
        method Simplex
        WeightsAdjust true

        AbsRange 0.0, 100000.0 # absorption spectral range to be included in the fit;
                              # in the present case all experimental points
                              # will be included



        AbsName "absexp.dat"  # name of the file containing experimental
                              # absorption spectrum in a simple two-column
                              # ASCII format

        ExpAbsScaleMode Ext   # This keyword indicates that the experimental
                              # absorption intensity is given in terms of
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

465

```
                                    # the extinction coefficient. This is important
                                    # for the proper fitting of transition dipole
                                    # moments and oscillator strengths

          NMaxFunc 10000    # maximum number of function evaluations in simplex
                            # algorithm

          MWADRelTol 1e-5   # Relative Tolerance of the Mean Weighted Absolute
                            # Difference  (MWAD) function which specifies the
                            # convergence criterion


          E0Step 500.00     # initial step for the transition energies
                            # in the simplex fitting

          TMStep 0.5        # initial step for the transition dipole moments
                            # in the simplex fitting

          E0SDStep 500.0    # initial step for the inhomogeneous linewidth (Sigma)
                            # in the simplex fitting


      end

    # ! Parameters specified in the $el_states block
    #   are used as initial guess in the fit


    #-------------------------------------------------------------------------
    # Transition        Gamma   Sigma  Transition Dipole Moment (atomic unit)
    # Energy (cm**-1)  (cm**-1) (cm**-1)     Mx         My        Mz
    #-------------------------------------------------------------------------

    $el_states
    7
        1 11270           0.0     1000.00    1.0000     0.0000    0.0000
        2 15100           0.0     1000.00    1.0000     0.0000    0.0000
        3 20230           0.0     1000.00    1.0000     0.0000    0.0000
        4 27500           0.0     1000.00    1.0000     0.0000    0.0000
        5 31550           0.0     1000.00    1.0000     0.0000    0.0000
        6 37070           0.0     1000.00    1.0000     0.0000    0.0000
        7 39800           0.0     1000.00    1.0000     0.0000    0.0000


    # the integer values specified in $el_states_c block indicate parameters
    # in the $el_states block to be varied
    $el_states_c
    7
        1    1           0           1           1          0          0
        2    2           0           2           2          0          0
        3    3           0           3           3          0          0
        4    4           0           4           4          0          0
        5    5           0           5           5          0          0
        6    6           0           6           6          0          0
        7    7           0           7           7          0          0
```

The functionality of the constraint block $el_states_c should be understood as follows: 1) 0 flag indicates that the

corresponding parameter in the `$el_state` block will not be varied in the fitting; 2) if the number corresponding to a certain parameter coincides with the number of the corresponding electronic state this parameter will be varied independently. Thus, the block `$el_states_c` in the input indicates that all transition energies, inhomogeneous linewidths and x-components of the transition electric dipole moment will be varied independently, while homogeneous linewidths, y- and z-components of the transition dipole moment will be fixed to their initial values.

The following considerations are important:

- Since in conventional absorption spectroscopy one deals with the orientationally averaged absorption cross-section, the signal intensity is proportional to the square of the transition electric dipole moment $|\mathbf{M}|^2$. Thus, the intensities do not depend on the values of the individual components of $\mathbf{M}$ as long as $|\mathbf{M}|^2 = const$. Therefore, we have allowed to vary only M_x components. Otherwise there can be problems in convergence of the fitting algorithm.

- The sum of the weights of experimental points which enter the mean absolute difference function employed in the the minimization is always kept equal to the number of experimental points. In the case of equidistant experimental photon energies all weights are assumed to be equal. However, in experimental electronic spectra the density of spectral points can increase significantly upon going from high- to low-energy spectral regions, which is due to the fact that experimental absorption spectra are initially acquired on the wavelength scale. In such a case the quality of the fit can be noticeably biased towards low-energy spectral region. Therefore, it is advisable to adjust relative weights of experimental points according to the their density which is controlled by the keyword `WeightsAdjust` in the `%fit` block. Although this parameter is not crucial for the present example, in general, it will provide a more balanced fit.

- The parameters `E0Step`, `TMStep`, `E0SDStep` in the `%fit` block specify the initial dimension of the simplex in the space of E_T, $\mathbf{M}$, Θ and should roughly correspond to the expected uncertainty of initial guess on these parameters in the `$el_states` block relative to their actual values. The quality of the fit can noticeably deteriorate if the parameters specifying initial steps are too low or too high.

The fit run of *orca_asa* on file `example007.inp` will converge upon approximately 3600 function evaluations (for `MWADRelTol=1e-5`). The results of the fit will be stored in file `example007.001.inp` which has the same structure as the input file `example007.inp`. Thus, if the fit is not satisfactory and/or it is not fully converged it can be refined in a subsequent *orca_asa* run upon which file `example007.002.inp` will be created, and so on. Some model parameters in intermediate files can be be additionally modified and/or some constraints can be lifted or imposed if so desired. The output file `example007.001.inp` will contain fitted model parameters stored in the `$el_states` block:

```
$el_states
7
    1 11368.24     0.00  732.50      1.6290   0.0000   0.0000
    2 15262.33     0.00  495.17     -0.2815   0.0000   0.0000
    3 19500.08     0.00 1023.39      0.2300   0.0000   0.0000
    4 26969.01     0.00 1832.30      1.4089   0.0000   0.0000
    5 31580.41     0.00 1440.87      1.8610   0.0000   0.0000
    6 35769.07     0.00 1804.02      1.5525   0.0000   0.0000
    7 39975.11     0.00 1909.38      2.4745   0.0000   0.0000
```

The overall quality of the fit is determined by the parameter MWAD which upon convergence reaches the value of ≈0.009 (MWAD stands for Mean Weighted Absolute Difference).

After the *orca_asa* run files `absexp.fit.dat` and `absexp.fit.as.dat` will be created. Both files contain the experimental and fitted spectra which are shown in Figure 6.21 . In addition, the file `absexp.fit.as.dat` will contain individual contributions to the absorption spectrum corresponding to different excited states.

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

467
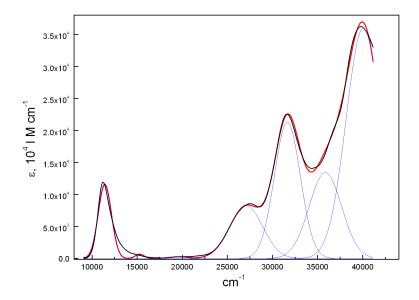

Figure 6.21: Comparison of the experimental (black curve) and fitted (red) absorption spectra corresponding to the fit run of orca_asa on the file `example007.inp`. Blue curves represent individual contributions to the absorption spectrum from each state.

Since there is a noticeable discrepancy between the fitted and experimental spectra around 1300 cm^{-1} (Figure 6.21) it is worthwhile to refine the fit after adding parameters for a new state in the file `example007.001.inp`:

```
$el_states
8
   1 11368.24     0.00  732.50     1.6290   0.0000   0.0000
... ... ...
   8 13280.00     0.00  1000.00     1.000   0.0000   0.0000


$el_states_c
8
   1     1        0        1        1        0        0
... ... ...
   8     8        0        8        8        0        0
```

Actually, the character of the discrepancy in the present case is very similar to that in Figure 6.18 (section 6.18.2.4) where a vibronically broadened absorption spectrum was fitted with a Gaussian lineshape. Thus, the poor fit in the region around 1300 cm^{-1} is most likely due to the essentially asymmetric character of the vibronic broadening rather than to the presence of another electronic band.

As shown in Figure 6.22 the refined fit leads to much better agreement between the experimental and fitted absorption spectra (MWAD=0.0045).

Due to some peculiarities of the simplex algorithm for function minimization, you can still refine the fit by rerunning *orca_asa* on the file `example007.002.inp`! This leads to an even lower value of the parameter MWAD= 0.0038, and therefore to better agreement of experimental and fitted spectra (even though the previous run has been claimed to be converged).
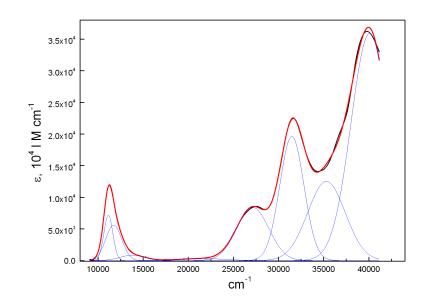
Figure 6.22: Comparison of the experimental (black) and fitted (red) absorption spectra corresponding to the fit run of orca_asa on the file **example007.001.inp**. Blue curves represent individual contributions to the absorption spectrum from each state.

It is also possible to perform a fit using the same value of inhomogeneous linewidth for all electronic states. For this purpose one needs to choose as a guess the same linewidth parameters in the $el_states block:

```
        $el_states
        8
            1 11118.58     0.00   1000.0      1.0687    0.0000    0.0000
            2 13673.38     0.00   1000.0     -0.5530    0.0000    0.0000
            3 21267.40     0.00   1000.0      0.3675    0.0000    0.0000
            4 27024.71     0.00   1000.0      1.4041    0.0000    0.0000
            5 31414.74     0.00   1000.0      1.7279    0.0000    0.0000
            6 35180.77     0.00   1000.0      1.6246    0.0000    0.0000
            7 39985.52     0.00   1000.0      2.5708    0.0000    0.0000
            8 11665.01     0.00   1000.0      1.2332    0.0000    0.0000
```

In addition the constraint block should be modified as follows:

```
        $el_states_c
        8
            1      1       0       1       1       0       0
            2      2       0       1       2       0       0
            3      3       0       1       3       0       0
            4      4       0       1       4       0       0
            5      5       0       1       5       0       0
            6      6       0       1       6       0       0
            7      7       0       1       7       0       0
            8      8       0       1       8       0       0
```

The constraint parameters for the inhomogeneous broadening were chosen to be 1, which means that formally Θ_1 corresponding to the first state is varied independently while the linewidths $\{\Theta_i\}$ for other bands are varied in such a

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

469

way that the ratios Θ_i/Θ_1 are kept fixed to their initial values, whereby the same linewidth parameter will be used for all states.
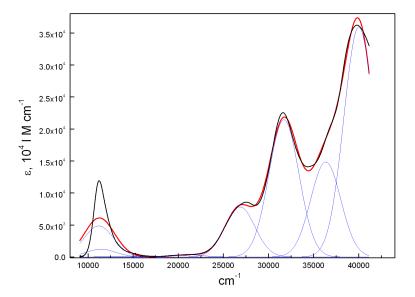
Figure 6.23: Comparison of the experimental (black) and fitted (red) absorption spectra corresponding to the fit run of orca_asa on the file `example007.002.inp` in which equal broadening was assumed for all electronic bands. Blue curves represent individual contributions to the absorption spectrum from each state.

One can see (Figure 6.23) that the assumption of equal linewidths for all electronic bands leads to a rather pronounced deterioration of the quality of the fit in the low-energy spectral range (MWAD=0.017). Apparantly, this discrepancy can be fixed assuming more electronic states at higher energies.

NOTE

- The homogeneous linewidth parameters can also be included in the fit in a similar way. However, one can see that in most cases they appear to be much smaller than corresponding Gaussian linewidth parameters.

- Gauss-fit of absorption spectra is coventionally performed assuming the same linewidth parameters for all bands. However, since a large portion of Gaussian broadening is mainly due to the unresolved vibronic structure in the spectra which can significantly vary depending on the nature of transition, the assumption of unequal Gaussian bandwidths seems to be a physical one.

6.18.3.2 Example: Fit of Absorption and Resonance Raman Spectra for $1\text{-}^1A_g \rightarrow 1\text{-}^1B_u$ Transition in trans-1,3,5-Hexatriene

Below we provide an example of the fit of the lineshape parameters and $\{\Delta_m\}$ corresponding to the strongly dipole-allowed $1\text{-}^1A_g \rightarrow 1\text{-}^1B_u$ transition in hexatriene. It is known that the most intense bands in rR spectra correspond to the most vibronically active in absorption spectrum. For the IMDHO model this correlation is determined by the values of $\{\Delta_m\}$. Thus, the larger Δ, the larger is the rR intensity of a given mode and the more pronounced is the progression in the absorption spectrum corresponding to this mode. In principle, if all vibrational transitions in absorption are well resolved it is possible to determine $\{\Delta_m\}$ by a fit of the absorption spectrum alone. In practice this task is ambiguous due to the limited resolution of the experimental absorption spectra. The observation of a rR spectrum enables the identification of the vibrational modes that are responsible for the progression in the absorption

spectrum, as well as a quantitative analysis in terms of $\{\Delta_m\}$. The file `example006.inp` provides a brute-force example on how to approach the fit employing the minimal possible experimental information: 1) An absorption spectrum; 2) relative rR intensities of fundamental bands for a given excitation energy. The rR spectrum upon the excitation in resonance with the 0-0 vibronic band at 39809 cm^{-1} is shown in Figure 6.12.
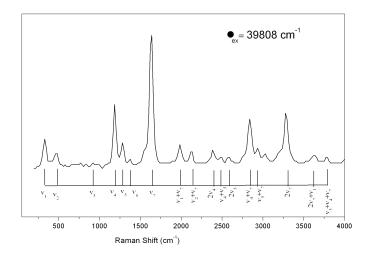
Figure 6.24: Experimental Resonance Raman spectrum corresponding to 1-^{1}A$_g$ → 1-^{1}B$_u$ transition in *trans*-1,3,5-hexatriene.

The experimental rR spectrum has enabled the identification of seven vibrational modes that give rise to the most intense resonance Raman bands. Therefore, they are expected to have the largest excited-state displacements and the most pronounced effect on the vibrational structure of the absorption spectrum. Their vibrational frequencies have been entered as input for the fit as shown below:

```
#
# example008.inp
#
# Input for fit of absorption and resonance Raman spectra
# corresponding to the strongly allowed 1-1Ag 1-1Bu transition
# in 1,3,5 trans-hexatriene.
#
# Parameters to be varied:
# 1) adiabatic minima transiton energy
# 2) homogeneous linewidth (Gamma)
# 3) dimensionless normal coordinate displacements of the
#    excited-state origin
#

%sim
      Model IMDHO
end


%fit

      Fit true     # boolean parameter to switch on the fit
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

471

```
        # boolean parameter to include experimental absorption
        # spectrum in the fit:
        AbsFit true

        # boolean parameter to include experimental rR spectra
        # specified in $rrs_exp block in the fit:
        RRSFit true

        AbsExpName "hex-abs.dat" # name of the file with experimental absorption
                                 # spectrum

        # the following value of keyword ExpAbsScaleMode
        # indicates that only the shape of absorption band
        # but not its total intensity will be accounted in the fit:
        ExpAbsScaleMode Rel

        # the weight of absorption relative to the total weight of
        #  rR intensities in the difference function to be minimized:
        CWAR  5.0

        NMaxFunc  1000   # maximum number of function evaluations in simplex
                         # algorithm

        MWADRelTol 1e-4  # Relative Tolerance of the Mean Weighted Absolute
                         # Difference  (MWAD) function which specifies the
                         # convergence criterion
        SDNCStep  1.0

    end


    # The values specified in $el_states block serve as initial guess in the fit
    $el_states
    1
       1  40000.0      200.00    0.0     1.0   0.0   0.0

    # the integer values specified in $el_states_c block indicate parameters
    # in $el_states block to be varied
    $el_states_c
    1
       1    1          1        0       0     0     0

    # 7 totally symmetric vibrations which give rise to the most
    # intense bands in the rR spectra are included into analysis.
    # Experimental values of vibration frequencies are given:
    $vib_freq_gs
    7
        1      354.0
        2      444.0
        3      934.0
        4     1192.0
        5     1290.0
        6     1403.0
        7     1635.0

    # Initial guess for the values of dimensionless normal
```

```
# coordinate displacements of the excited-state origin
$sdnc
7 1
                1
        1        0.0
        2        0.0
        3        0.0
        4        0.0
        5        0.0
        6        0.0
        7        0.0


# the integer values specified in $sdnc_c block indicate parameters
# in $sdnc block to be varied
$sdnc_c
7 1
                1
        1        1
        2        2
        3        3
        4        4
        5        5
        6        6
        7        7



# specification  of vibrational transitions and their intensities
# in experimental rR spectra:
$rrs_exp
1         # number of rR spectra
1 1       # start of the block specifying the 1st rR spectrum
    Ex 39809.0 # excitation energy for the first rR spectrum
    NTr  7     # number of vibrational transitions for which intensities are
               # provided
    1
        int   10.0  1.0
        modes  1
        quanta 1;
    2
        int   5.0  1.0
        modes  2
        quanta 1;
    3
        int   1.5  1.0
        modes  3
        quanta 1;
    4
        int   21.0  1.0
        modes  4
        quanta 1;
    5
        int   7.5  1.0
        modes  5
        quanta 1;
    6
        int   2.0  1.0
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

473

```
         modes  6
         quanta 1;
      7
         int   46.0  1.0
         modes  7
         quanta 1;
```

The input of rR intensities for an arbitrary number of excitation energies follows the keyword **$rrs_exp** block:

```
  $rrs_exp
  1          # number of rR spectra
  1 1
```

The first "1" in the last line denotes the number of the rR spectrum for which specification starts below. If the second number is the same as the number of the spectrum, then it means that only relative intensities for the 1st rR spectrum are meaningful in the fit. If several spectra are given in the input then the second number may have a different value, e.g.:

```
  $rrs_exp
  3          # number of rR spectra
  1 2
  ...
```

This input is to be interpreted as indicating that 3 rR spectra are provided and the relative intensities for the 1st spectrum are given on the same scale as the 2nd one that will be accounted for in the fit. The value of the excitation energies and the number of vibrational transitions specified are indispensable within the blocks specifying intensities for each rR spectrum.

Following the number of vibrational transitions given by the keyword **NTr** one has to specify each vibrational transition and its intensity. Thus, in the present case there are seven subblocks with the following structure:

```
  k    int   I  W
           modes m1,m2,...mn
           quanta q1,q2,...qn;
```

This means that the k-th transition has intensity I and weight W in the mean absolute difference function that is used for the minimization (W is an optional parameter). The following 2 lines specify the vibrational transitions by providing excitation numbers q_i for modes m_i so that the corresponding Raman shift is equal to $\nu = \sum_{q_i \nu_i}$, where ν_i is vibrational frequency of the mode m_i.

The parameters that are to be varied are specified within the constraint blocks **$el_states_c** and **$sdnc_c**. Both blocks have the same structure and number of parameters as **$el_states** and **$sdnc**, respectively. A parameter from the **$el_states** block is supposed to be independently varied if its counterpart from the **$el_states_c** block is equal to the number of the electronic state. Likewise, a parameter from the **$sdnc** block is supposed to be independently varied if its counterpart from the **$sdnc_c** block is equal to the number of the normal mode. Model parameters that are set to 0 in the corresponding constraint blocks are not varied in the fit. The values of the following parameters may be important for the quality of the fit:

- CWAR in the %fit block specifies the weight of absorption relative to the weight of rR intensities in the difference function to be minimized. If this parameter was not specified the fit would be almost insensitive to the rR intensities in the input, since typically the number of experimental absorption points is much larger than the number of rR transitions in the input. In most cases the value of CWAR in the range 1.0–5.0 is a good choice since the error in the measured experimental intensity is expected to be much smaller for absorption than for resonance Raman.

- SDNCStep in the %fit block specifies the initial dimension of the simplex in the space of $\{\Delta_m\}$ and should roughly correspond to the expected uncertainty of initial guess on $\{\Delta_m\}$ in the $sdnc block compared to their actual values. You can notice in the present example that if this parameter is too large (>2.0) or too small (<0.4) the quality of the fit may significantly deteriorate

- Although the default initial dimensions of the simplex have reasonable values for different types of parameters it may turn out to be helpful in some cases to modify the default values:

```
FREQGStep 10.0  # ground-state vibrational frequencies
        FREQEStep 10.0  # excited-state vibrational frequencies
        E0Step    300.0 # transition energies
        SSStep     20.0 # Stokes shift
        TMStep      0.5 # electronic transition dipole moment
        GammaStep  50.0 # homogeneous linewidth
        E0SDStep   50.0 # inhomogeneous linewidth
        SDNCStep 1.0 # origin shift along dimensionless normal coordinate
```

The fit run of *orca_asa* on the file example008.inp will converge upon approximately 700 function evaluations (for MWADRelTol=1e-4). The results of the fit will be stored in file example008.001.inp which has the same structure as the input file example008.inp. Thus, if the fit is not satisfactory and/or it is not fully converged it can be refined in subsequent *orca_asa* run upon which file example008.002.inp will be created, and so on. Some model parameters in intermediate files can be be additionally modified and/or some constraints can be lifted if so desired. The output file example008.001.inp will contain fitted displacement parameters $\{\Delta_m\}$stored in the $sdnc block:

```
$sdnc
7 1
                 1
     1      0.675000
     2     -0.194484
     3     -0.217527
     4      0.811573
     5      0.529420
     6     -0.149991
     7      1.314915
```

In the present example, these parameters are actually in very close agreement with those published for the hexatriene molecule!

The overall quality of the fit is determined by the parameter MWAD which upon convergence reaches the value of ≈ 0.027. The fitted rR intensities are presented in the commented lines next to the experimental rR intensities in file example008.001.inp:

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

475

```
$rrs_exp
1
   1   1 3.495285e+001
   Ex    39809.00
   NT   7
      1
        Int 10.0 1.0  # simulated intensity: 1.000982e+001
        modes  1
        quanta 1;
      2
        Int 5.0 1.0   # simulated intensity: 8.976285e-001
        modes  2
        quanta 1;
      3
        Int 1.5 1.0   # simulated intensity: 1.255880e+000
        modes  3
        quanta 1;
      4
        Int 21.0 1.0  # simulated intensity: 1.761809e+001
        modes  4
        quanta 1;
      5
        Int 7.5 1.0   # simulated intensity: 7.499749e+000
        modes  5
        quanta 1;
      6
        Int 2.0 1.0   # simulated intensity: 6.014466e-001
        modes  6
        quanta 1;
      7
        Int 46.0 1.0  # simulated intensity: 4.600071e+001
        modes  7
        quanta 1;
```

The file hex-abs.fit.dat will contain the experimental and fitted absorption spectra in ASCII format which can be plotted in order to visualize the quality of absorption fit (Figure 6.25).

NOTE

- The more experimental rR intensities are included in the analysis the more reliable is the fit. In principle it is possible to obtain fully consistent results even if only a limited number of vibrational transitions is provided. However, in such a case it is desirable to include into analysis at least a single Raman transition involving the mode for which Δ is to be determined.

- The quality of the fit can be improved if the IMDHOFA model is invoked and excited-state vibrational frequencies are allowed to vary.

- Due to the initial guess and dimension of the simplex, as well as some peculiarities of the simplex algorithm for function minimization, you can still refine the fit by rerunning *orca_asa* on file example008.001.inp that may lead to an even lower value of the parameter MWAD = 0.021, and therefore to better agreement of experimental and fitted spectra (even though the previous run has been claimed to be converged).

- In this respect it appears to be wise to perform the fit in 3 steps:
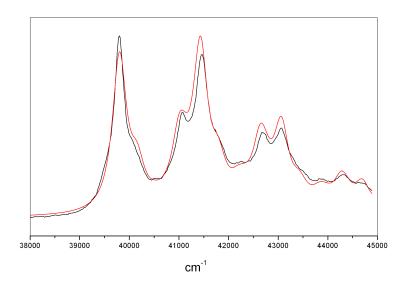
Figure 6.25: Experimental (black) and fitted (red) absorption spectrum corresponding to $1\text{-}^1A_g \rightarrow$ 1-^1B_u transition in 1,3,5 *trans*-hexatriene.

1. Fit the preresonance region below the 0-0 vibronic band with a single Lorentzian band, from which the adiabatic transition energy E_0, and homogeneous linewidth Γ are obtained. The range for fit of the absorption spectrum can be specified by the `AbsRange` keyword in the `%fit` block.

2. Fix E_0 and Γ, and optimize $\{\Delta_m\}$ fitting the entire spectral range and rR intensities.

3. Lift constraints on E_0 and Γ, and reoptimize simultaneously all parameters.

6.18.3.3 Example: Single-Mode Fit of Absorption and Fluorescence Spectra for $1\text{-}^1A_g \rightarrow$ 1-^1B_{2u} Transition in Tetracene

In this section we provide an example and discuss the most important aspects of joint fit of fluorescence and absorption spectra. Figure 6.26 displays the experimental emission and absorption spectra corresponding to $1\text{-}^1A_g \rightarrow 1\text{-}^1B_{2u}$ transition in tetracene.

Both spectra show pronounced effective vibrational progressions that are dominated by 3 and 5 peaks, respectively. As can be shown on the basis of quantum chemical calculations this progression has essentially multimode character. However, the experimental spectra can be well fitted under the assumption of a single vibronically active mode. The input has the following structure:

```
#
# example009.inp
#
# Parameters to be varied:
# 1) adiabatic minima transition energy
# 2) homogeneous and inhomogeneous linewidths
# 3) normal mode frequency and corresponding dimensionless displacement of the
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program
477
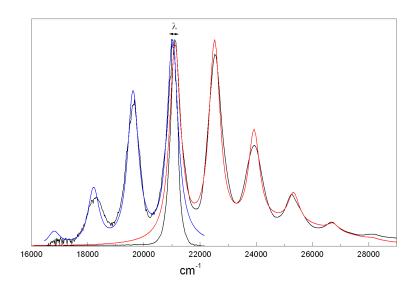
Figure 6.26: Deconvoluted absorption (red) and fluorescence (blue) spectra of tetracene in cyclo-
hexane upon the assumption of a single vibronically active mode. The black solid lines
represent experimental spectra.

```
#    excited-state origin
#

%sim
     Model IMDHO
     EnInput E0 # we assume adiabatic minima separation energies
end

%fit
    Fit true      # global flag to turn on the fit
    AbsFit true   # flag to include absorption spectrum into the fit
    FlFit true    # flag to include fluorescence spectrum into the fit


    WeightsAdjust true

    AbsRange 19000.0, 28000.0 # spectral range for absorption
                              # which will be included into the fit

    FlRange  17800.0, 22300.0 # spectral range for absorption
                              # which will be considered in the fit

    AbsName "absexp.dat" # name of the file containing experimental
                         # absorption spectrum in a simple two-column
                         # ASCII format
    FlName  "flexp.dat"  # name of the file with experimental fluorescence spectrum

    ExpAbsScaleMode Rel  # flags indicating that only relative shapes of the
    ExpFlScaleMode Rel   # absorption and fluorescence bands will be fitted.


    CWAF   1.000 # important parameter to have a balanced relative quality of fit
```

```
                    # of fluorescence and absorption

    NMaxFunc 10000  # maximum number of function evaluations in simplex
                    # algorithm

    MWADRelTol= 0.0001  # Relative Tolerance of the Mean Weighted Absolute
                        # Difference  (MWAD) function which specifies the
                        # convergence criterion

    TMStep 0.5       # initial step for the transition dipole moments
                     # in the simplex fitting

    E0SDStep 500.0   # initial step for the inhomogeneous linewidth (Sigma)

    FREQGStep 100.00 # initial step for the vibrational frequencies

    E0Step 1000.0    # initial step for the transition energies

    SSStep 10.0      # initial step for the Stokes shift

    GammaStep 100    # initial step for the homogeneous linewidth

    SDNCStep 0.5     # initial step for the displacement parameter
end

$el_states
2
    1 21100.00     100.00    100.00     1.0000   0.0000   0.0000
    2 24000.00     100.00   1000.00     1.0000   0.0000   0.0000

$el_states_c
2
    1     1         1         1          0        0        0
    2     2         2         2          2        0        0

$abs_bool
2
  1   1
  2   1

$fl_bool
2
  1   1
  2   0

$ss
2
    1       100.000000
    2         0.000000

$ss_c
2
    1             1
    2             0
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

479

```
    $vib_freq_gs
    1
        1        1500.0



    $vib_freq_gs_c
    1
        1                1



    $sdnc
    1 2
                        1           2
            1        2.0000000    0.000000



    $sdnc_c
    1 2
                        1           2
            1            1           0
```

The parameter `CWAF=1.0` in the `%fit` block specifies the weight of absorption relative to the weight of fluorescence in the difference function to be minimized. If this parameter was not specified the quality of the fit would be biased towards the spectrum with a larger number of experimental points. In some typical situations where the error in the measured experimental intensity is expected to be smaller for absorption than for emission it is desirable to choose the value of `CWAF` to be more than 1.0.

In order to account for a broad featureless background signal in the absorption spectrum above 24000 cm^{-1}, the second band was included into the analysis and approximated with a Voigt lineshape which means also that the corresponding frequency in the `$vib_freq_gs` block and displacement parameter in the `$sdnc` block are fixed to zero in the fit. Thus, the `$el_states` block contains an initial guess on the transition energies, transition electric dipole moments and linewidth parameters for 2 states:

```
    $el_states
    2
        1 21100.00    100.00    100.00     1.0000    0.0000    0.0000
        2 24000.00    100.00    1000.00    1.0000    0.0000    0.0000
```

The initial value of the adiabatic minima separation energy for the first state was approximated by the energy corresponding to the first vibronic peak in the absorption spectrum (21100 cm^{-1}). The transition energies and linewidth parameters are varied independently as indicated in the `$el_states_c` block. Since we allow to fit only bandshapes, but not the overall intensities of the spectra, only relative absolute values of the transition electric dipole moments of two bands are important. Therefore it is reasonable to fix all components of the transition moment for the first state and vary only M_x component for the second one:

```
    $el_states_c
    2
        1      1        1        1        0        0        0
        2      2        2        2        2        0        0
```

Since we assume the absorption by both states and emission only from the first one, it is necessary to include Boolean arrays $abs_bool and $fl_bool which specify states which will be included in the treatment of the absorption and fluorescence spectra, respectively:

```
    $abs_bool
    2
      1  1 # 1 indicates that the corresponding state will be included in the calculation of
      2  1 # absorption


    $fl_bool
    2
      1  1
      2  0 # 0 indicates that the corresponding state will be excluded from the calculation
             # of emission spectrum
```

We need also to vary the value of vibrational frequency of the mode which determines separation of vibrational peaks in the spectra. This is done via the constraint block $vib_freq_gs_c:

```
    $vib_freq_gs_c
    1
           1              1
```

Note that it is meaningless to include into the treatment the Stokes shift for the second state which give rise to the background signal in the absorption since the corresponding emission is not present. Therefore λ for the second state is fixed to zero as indicated in the $ss block and its constraint counterpart $ss_c:

```
    $ss
    2
      1        100.000000 # initialization of the Stokes shift for the 1st electronic state
      2          0.000000

    $ss_c
    2
      1             1 # the Stokes shift for the 1st electronic state will be varied in the fit
      2             0 # the Stokes shift for the 2nd electronic state will be fixed in the fit
```

The fit run of *orca_asa* on file `example009.inp` will converge upon approximately 700 function evaluations (for `MWADRelTol=1e-4`). The file `example009.001.inp` will contain the fitted effective values of the vibrational frequency and dimensionless displacement: $\omega = 1404 cm^{-1}$, $\Delta = 1.35$. One can notice that the fit is rather poor in the low- and high-energy edges of the absorption and fluorescence spectra, respectively (Figure 6.26). The source of this discrepancy is the single-mode approximation which was employed here. The quality of the fit can be significantly improved assuming several modes with non-zero displacement parameters. Note that in such a case the proper guess on the number of active modes and corresponding dimensionless displacements can be deduced from quantum chemical calculations.

6.18.4 Quantum-Chemically Assisted Simulations and Fits of Optical Bandshapes and Resonance Raman Intensities

In this section we finally connect the spectra simulation algorithms to actual quantum chemical calculations and outline a detailed approach for the analysis of absorption, fluorescence and resonance Raman spectra within the IMDHO model. Our procedure becomes highly efficient and nearly automatic if analytical excited state derivatives

with respect to nuclear displacements are available. However, this availability is not mandatory and hence, spectral predictions may as well be achieved by means of normal mode scan calculations for high-level electronic structure methods for which analytic gradients have not been implemented.

6.18.4.1 Example: Quantum-Chemically Assisted Analysis and Fit of the Absorption and Resonance Raman Spectra for $1\text{-}^1A_g \rightarrow 1\text{-}^1B_u$ Transition in trans-1,3,5-Hexatriene

The following input file for an ORCA run invokes the calculation of the excited-state origin displacements along all normal modes by means of energy and excited state gradient calculations at the ground-state equilibrium geometry. The method is valid for the IMDHO model for which the excited-state energy gradient along a given normal mode and corresponding origin shift are related in a very simple way.

```
#
# example010.inp
#
# TDDFT BHLYP Normal Mode Gradient Calculation
#
#   The keyword NMGrad invokes the normal mode gradient calculation
#
! RKS BHandHLYP TightSCF SV(P) NMGrad

%cis    NRoots 1
        triplets false
end

%rr
     # the nuclear Hessian must have been calculated before - for example by a
     # DFT calculation.
     HessName= "hexatriene.hess"

     states 1    # Perform energy-gradient calculations for the 1st
                 # excited state.
     Tdnc 0.005  # Threshold for dimensionless displacements to be
                 # included in the input file for spectral simulations
                 # generated at the end of the program run.
                 # By default Tdnc= 0.005

     ASAInput true # Generate the input file for spectra simulations
   end

* xyz   0 1
   C    -0.003374    0.678229    0.00000
   H    -0.969173    1.203538    0.00000
   C     1.190547    1.505313    0.00000
   H     2.151896    0.972469    0.00000
   C     1.189404    2.852603    0.00000
   H     0.251463    3.423183    0.00000
   H     2.122793    3.426578    0.00000
   C     0.003374   -0.678229    0.00000
   H     0.969172   -1.203538    0.00000
   C    -1.190547   -1.505313    0.00000
   H    -2.151897   -0.972469    0.00000
   C    -1.189404   -2.852603    0.00000
   H    -0.251463   -3.423183     0.00000
```

```
    H    -2.122793   -3.426578    0.00000
*
```

In the ORCA run the TDDFT excited state gradient calculations are performed on top of a TDDFT calculation. Note, that the numbers of the excited-states which have to be included into analysis and input file for spectral simulations must be specified after the `States` keyword in the `%rr` block. They should also be consistent with the required number of roots in the `%tddft` block. The 1-^1B_u excited state appears to be the 1st root in the TDDFT calculation. Therefore, NRoots=1 in the `%tddft` block, and States=1 in the `%rr` block. One should also provide the name of the file containing the nuclear Hessian matrix via the `HessName` keyword in the `%rr` block. Here we used the `.hess` file obtained in a frequency calculation at the BHLYP/SV(P) level of theory.

After the ORCA calculation you will find in your directory a file called `example010.asa.inp` that is appropriate to be used together with the orca_asa program as defined in the preceding sections.

```
#
# example010.asa.inp
#
# ASA input
#
%sim
        model IMDHO
        method Heller
        AbsRange    5000.0,  100000.0
        NAbsPoints   0

        FlRange     5000.0,  100000.0
        NFlPoints    0

        RRPRange    5000.0,  100000.0
        NRRPPoints   0

        RRSRange       0.0,    4000.0
        NRRSPoints 4000

        RRS_FWHM 10.0


        AbsScaleMode Ext
        FlScaleMode Rel
        RamanOrder  0

        EnInput E0

        CAR   0.800

    end

    %fit
        Fit false
        AbsFit false
        FlFit false
        RRPFit fsalse
        RRSFit false
        method Simplex
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

483

```
         WeightsAdjust true

         AbsRange        0.0,   10000000.0
         FlRange         0.0,   10000000.0
         RRPRange        0.0,   10000000.0
         RRSRange        0.0,   10000000.0
         AbsName ""
         FlName  ""

         ExpFlScaleMode  Rel
         ExpAbsScaleMode Rel

         CWAR  -1.000
         CWAF  -1.000

         NMaxFunc 100
         MWADRelTol= 1.000000e-004

         SFRRPSimStep= 1.000000e+002
         SFRRSSimStep 1.000000e+002
         FREQGStep 1.000000e+001
         FREQEStep 1.000000e+001
         E0Step 3.000000e+002
         SSStep 2.000000e+001
         TMStep 5.000000e-001
         GammaStep 5.000000e+001
         E0SDStep 5.000000e+001
         SDNCStep 4.000000e-001
  end

  $el_states
  1
     1 42671.71   100.00    0.00     1.0725   3.3770  -0.0000

  $vib_freq_gs
  12
        1        359.709864
        2        456.925612
        3        974.521651
        4       1259.779018
        5       1356.134238
        6       1370.721341
        7       1476.878592
        8       1724.259894
        9       1804.572974
       10       3236.588264
       11       3244.034359
       12       3323.831066


  $sdnc
  12 1
                  1
        1      -0.594359
        2       0.369227
        3      -0.132430
```

```
     4        -0.727616
     5         0.406841
     6        -0.105324
     7         0.177617
     8        -0.090105
     9        -1.412258
    10         0.048788
    11         0.021438
    12         0.008887
```

This input file can be used to construct theoretical absorption and rR spectra. **In order to compare experimental and theoretical rR spectra, it is necessary to use in both cases excitation energies that are approximately in resonance with the same vibrational transitions in the absorption spectrum. Therefore, in the case of the absorption spectrum with resolved or partially resolved vibrational structure it is necessary to modify the transition energies in the %el_states such that they coincide with the experimentally observed 0-0 vibrational peaks.** It is also desirable to roughly adjust homogeneous and, possibly, inhomogeneous linewidth parameters such that the experimental and calculated absorption spectra show similar slopes in the preresonance region (below the 0-0 transition). Then the assignment of experimental rR spectra can be done on the basis of comparison with the theoretical rR spectra calculated for the corresponding experimental excitation energies. For the sake of consistency and simplicity it is better to use those excitation energies which fall into the preresonace region and/or are in resonance with the 0-0 transition. **In the case of diffuse absorption spectra (i.e. those not showing resolved vibrational structure) it is also necessary to adjust the theoretical transition energies and linewidth parameters such that experimental and calculated positions of absorption maxima roughly coincide, and corresponding slopes below the maxima have a similar behavior.** According to above mentioned considerations one needs to modify the %el_states block in the file example010.asa.inp:

```
    $el_states
    1
       1 39808.0   150.00    0.00  1.0725   3.3770  -0.0000
```

The calculated absorption spectrum obtained by providing `AbsScaleMode= Rel`, `AbsRange= 39000`, `49000` and `NAbsPoints= 2000` is shown in Figure 6.27. Upon comparison with the experimental spectrum one can notice that the BHLYP functional gives relatively small discrepancies with somewhat lower intensity in the low-frequency edge and larger intensity on the high-energy side of the spectrum. Besides, there is a noticeable mismatch in the separation between individual vibronic peaks which is due to overestimation of vibrational frequencies by the BHLYP functional (typically by $\approx 10\%$).

You can arbitrarily vary various normal coordinate displacements in `%sdnc` block within 10–30% of their values in order to observe modifications of the calculated spectrum. This will tell you how these parameters influence the spectrum and probably it will be possible to obtain better initial guesses for the fit. In the present example you will find that reduction of the absolute value of the displacement parameter corresponding to the 9th mode by $\approx 10\%$, and reduction of vibrational frequencies by $\approx 10\%$ can noticeably improve the spectral envelope. Such a quick analysis suggests that experimentally observed peaks in the absorption spectrum represent different vibrational transitions corresponding to a single electronically excited state rather than to different electronic excitations. This conclusion will be confirmed upon establishing the fact that the absorption and rR spectra can be successfully fitted based on the assumption of a single electronic transition.

In order to calculate the rR spectrum for experimental excitation energies you need to specify its value through `RRSE` keyword in `%sim` block as well as possibly to modify the parameters related to the spectral range and linewidth of rR bands which are suitable for comparison with the experimental rR spectrum:

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

485
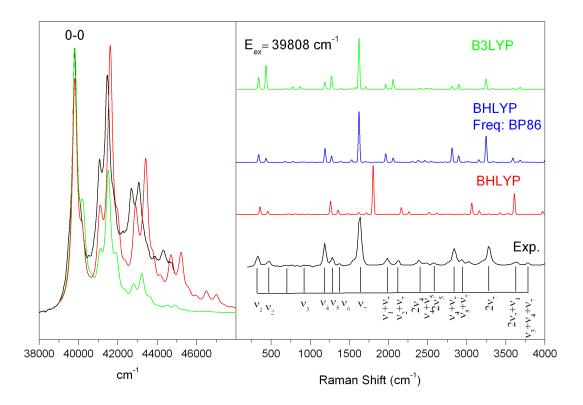
Figure 6.27: Experimental and calculated at the BHLYP/SV(P) and B3LYP/SV(P) levels of theory absorption (left panel) and rR spectra (right panel) corresponding to $1\text{-}^1A_g \rightarrow 1\text{-}^1B_u$ transition in *trans*-1,3,5-hexatriene.

```
# excitation energies (cm**-1) for which rR spectra will be calculated:
RRSE 39808

# full width half maximum of Raman bands in rR spectra (cm**-1):
RRS_FWHM  20

RSRange   0, 4000   # spectral range for simulation of rR spectra (cm**-1)
NRRSPoints  4000    # number of points to simulate rR spectra (cm**-1)

# resonance Raman intensities will be calculated
# for all vibrational states with excitation number
# up to RamanOrder:
RamanOrder 3
```

The calculated rR spectrum is shown in Figure 6.27. In the input we have invoked the calculation of rR intensities for the transitions with up to 3 vibrational quanta in the final vibrational state (`RamanOrder = 3`). Make sure that the rR intensity pattern in the given spectral range does not change noticeably upon further increase of this parameter. Typically, the larger are the normal coordinate displacements the greater order of Raman scattering is required in the calculation to account for all the most intense transitions in the rR spectrum. The inclusion of vibrational transitions beyond the fundamentals is a particular feature of the orca_asa program.

Comparison of the calculated and experimental rR spectra (Figure 6.27) mainly shows discrepancies in the values of the Raman shifts that are mainly related to the low accuracy of the vibrational frequencies obtained at the BHLYP

level (typically overestimated by $\approx 10\%$). However, the intensity patterns of the calculated and experimental rR spectra show very nice agreement with experiment that is already sufficient to assign the experimental peaks to individual vibrational transitions. This can be done upon examination of file `example010.asa.o3.rrs.39808.stk` which provides intensity, Raman shift, and specification for each vibrational transition. It is actually one of the most consistent procedures that enables one to identify different fundamental, overtone and combination bands in the experimentally observed rR spectrum. Such an assignment is a necessary prerequisite for the fit. The current example is relatively straightforward since the spectral region 1–1700 cm^{-1} is actually dominated by fundamental bands while the most intense overtone and combination transitions occur at higher frequencies. However, in many cases even the low-frequency spectral range is characterized by significant contributions from overtone and combination bands that sometimes are even more intense than fundamental transitions! Thus, quantum chemical calculations can greatly facilitate the assignment of experimental rR bands.

After having performed the assignment it is advisable to discard those modes from the analysis that are not involved in any of the experimentally observed fundamental, overtone, or combination rR bands with noticeable intensities. In the present example these are the modes 6, 8, 10–12 from the input file given above. For these modes it is implied that the fitted displacement parameters are zero. You will find that the calculated displacement values are rather small indeed. Also it is advisable to change the ground-state vibrational frequencies in the **$vib_freq_gs** block to their experimental values.

Below is the modified input file for the fit run:

```
#
# example010-01.asa.inp
#
# ASA input
#
%sim
        model IMDHO
        method Heller
end


%fit

    Fit true
    AbsFit true
    RRSFit true
    AbsExpName "hex-abs.dat"
    ExpAbsScaleMode Rel
    CWAR  5.0

    NMaxFunc  1000

    SDNCStep  0.5

end

$el_states
1
    1 39808.0    150.00     0.00     -0.8533  -3.3690  -0.0000

$el_states_c
1
    1    1        1        0         0        0        0
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

487

```
$vib_freq_gs
7
    1        354.0
    2        444.0
    3        934.0
    4       1192.0
    5       1290.0
    6       1403.0
    7       1635.0




$sdnc
7 1
                    1
        1      -0.594359
        2       0.369227
        3      -0.132430
        4      -0.727616
        5       0.406841
        6       0.177617
        7      -1.412258

$sdnc_c
7 1
               1
        1        1
        2        2
        3        3
        4        4
        5        5
        6        6
        7        7

$rrs_exp
1
1 1
    Ex 39809.0
    NTr  11
    1
       int   10.0  1.0
       modes  1
       quanta 1;
    2
       int    5.0  1.0
       modes  2
       quanta 1;
    3
       int    1.5  1.0
       modes  3
       quanta 1;
    4
       int   21.0  1.0
       modes  4
       quanta 1;
```

```
        5
          int   7.5  1.0
          modes  5
          quanta 1;
        6
          int   2.0  1.0
          modes  6
          quanta 1;
        7
          int   46.0  1.0
          modes  7
          quanta 1;

        8
          int  6.8 1.0
          modes  1, 7
          quanta 1, 1;

        9
          int  4.0 1.0
          modes  2, 7
          quanta 1, 1;

       10
          int  2.0 1.0
          modes  3, 7
          quanta 1, 1;

       11
          int 17.0 1.0
          modes  7
          quanta 2;
```

In addition to the experimental intensities of fundamental bands the input file also contains the information about some overtone and combination transitions. Note that it is not really necessary to include all of them them into the fit, in particular if some of the rR bands are strongly overlapping with each other.

Fitted normal coordinate displacements of the excited-state origin show nice agreement with the published values:

```
    $sdnc
    7 1
                      1
        1     -0.638244
        2      0.455355
        3     -0.229126
        4     -0.854357
        5      0.501219
        6      0.197679
        7     -1.292997
```

NOTE

- It is not really important to employ the BHLYP/SV(P) method in the frequency calculations in order to obtain the `.hess` file (this was merely done to be consistent with the TDDFT/BHLYP/SV(P) method for the

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation Profiles and Spectra with the orca_asa Program

489

excited-state model parameters calculation). The frequency calculations can for example be carried out at the BP86/TZVP or RI-SCS-MP2/TZVP level of theory. This will provide displacements pattern very similar to that of the BHLYP/SV(P) method, but much more accurate vibrational frequencies which will further facilitate the assignment of rR spectra (Figure 6.27). However, such a procedure can be inconsistent if the two methods give noticeably different normal mode compositions and/or vibrational frequencies. From our experience it can lead to significant overestimation of the excited-state displacements for some low-frequency modes.

- It is known that predicted dimensionless normal coordinate displacements critically depend on the fraction of the "exact" Hartree-Fock exchange (EEX) included in hybrid functionals. In general no universal amount of EEX exists that provides a uniformly good description for all systems and states. Typically, for a given molecule either the BHLYP/TZVP (50% of EEX) or B3LYP/TZVP (20% of EEX) methods yields simulated spectra that compare very well with those from experiment if vibrational frequencies are appropriately scaled.

6.18.4.2 Important Notes about Proper Comparison of Experimental and Quantum Chemically Calculated Resonance Raman Spectra

In order to compare experimental and theoretical rR spectra, **it is necessary to use in both cases excitation energies that are approximately in resonance with the same vibrational transitions in the absorption spectrum**. Therefore, in the case of diffuse absorption spectra (i.e. those not showing resolved vibrational structure) one needs to adjust the transition energies and linewidth paramters in the %el_states block such that the envelopes of the experimental and theoretical spectra rouhgly coincide, and then to employ experimental values of excitation energies to construct theoretical rR spectra. Typically in the case of diffuse absorption spectra rR profiles are rather smooth. Therefore, even though excitation energies are not in resonance with the same vibrational transition in the absorption spectrum, the rR spectra are not expected to vary significantly in the case of such mismatch.

In the case of the absorption spectrum with resolved or partially resolved vibrational structure it is necessary to modify the transition energies in the %el_states block such that the calculated and experimentally observed 0-0 vibrational peaks coincide, and modify linewidth parameters so that the low-energy slopes in the calculated and experimental spectra have a similar behavior.

Consider a single-mode model system for which "experimental" and calculated absorption spectra are shown in Figure 6.28.

Comaprison of the calculated and experimental spectra shows that some adjustment of the linewidth parameters is neceassy before construction of theoretical rR spectra. One can directly compare calculated and experimental rR spectra upon the excitation at 16200 cm^{-1} which is in resonance with the 0-0 vibronic band. However, it is not consisitent to use experimental value of the excitation energy in the calculation of rR spectrum which is in resonance with one of the other vibronic bands since the separation between vibartional peaks in the experimental and calculated spectra is different whereby positions of the peaks in both spectra do not coincide. Instead **one should use the excitaition energy which corresponds to the same vibronic peak in the calculated absorption spectrum as in the experimental one**. Alternatively, one can adjust theoretical value of vibrational frequency such that positions of corresponding vibronic peaks in the spectra coincide, and then use experimental values of excitation energies for the calculation of rR spectra.

6.18.4.3 Example: Normal Mode Scan Calculations of Model Parameters for 1-^{1}A$_g$ → 1-^{1}B$_u$ Transition in trans-1,3,5-Hexatriene

If excited state gradients are not available (which is the case for many of the electronic structure methods supported by ORCA), you have to resort to a more laborious procedure – single point calculations at geometries that are displaced
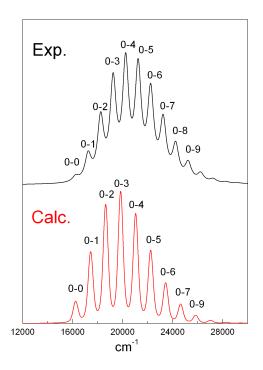
Figure 6.28: Experimental and theoretical absorption spectra for a single-mode model system. The
calculated spectrum is adjusted such that the position of 0-0 peak coincide with the
experimental one.

along the various normal modes of the system. This roughly corresponds to taking numerical derivatives – however, once this extra effort is invested more information can be obtained from the calculation than what would be possible from a analytic derivative calculation.

The present example illustrates the application of normal mode scan calculations for the evaluation of excited state harmonic parameters that are necessary to simulate optical spectra within the IMDHO model. This method can be applied with any method like CIS, CASSCF, MRCI or TD-DFT.

The reference wavefunctions for the multireference calculations reported below are of the state-averaged CASSCF (SA-CASSCF) type. The complete active space CAS(6,6) includes all 6 valence shell $.\pi$-orbitals. The average is taken over the first four states which was found necessary in order to include the ground state and the strongly allowed 1^1B_u state.

```
#
# example011.inp
#
# CASSCF normal mode scan calculations
#

# first do single point RHF calculation
! RHF TZVP TightSCF

* xyz 0 1
  C      -0.002759    0.680006    0.000000
  H      -0.966741    1.204366    0.000000
  C       1.187413    1.500920    0.000000
```

6.18 Simulation and Fit of Vibronic Structure in Electronic Spectra, Resonance Raman Excitation
Profiles and Spectra with the orca_asa Program

491

```
    H        2.146702     0.969304     0.000000
    C        1.187413     2.850514     0.000000
    H        0.254386     3.420500     0.000000
    H        2.116263     3.422544     0.000000
    C        0.002759    -0.680006     0.000000
    H        0.966741    -1.204366     0.000000
    C       -1.187413    -1.500920     0.000000
    H       -2.146702    -0.969304     0.000000
    C       -1.187413    -2.850514     0.000000
    H       -0.254386    -3.420500     0.000000
    H       -2.116263    -3.422544     0.000000
*



# perform SA-CASSCF calculation upon appropriate rotation of MOs
$new_job
! RHF   TZVP TightSCF

%scf
    rotate {23,27,90} end
end

%casscf
        nel 6
        norb  6
        mult  1
        nroots  4
end

* xyz 0 1
  C       -0.002759     0.680006     0.000000
  H       -0.966741     1.204366     0.000000
  C        1.187413     1.500920     0.000000
  H        2.146702     0.969304     0.000000
  C        1.187413     2.850514     0.000000
  H        0.254386     3.420500     0.000000
  H        2.116263     3.422544     0.000000
  C        0.002759    -0.680006     0.000000
  H        0.966741    -1.204366     0.000000
  C       -1.187413    -1.500920     0.000000
  H       -2.146702    -0.969304     0.000000
  C       -1.187413    -2.850514     0.000000
  H       -0.254386    -3.420500     0.000000
  H       -2.116263    -3.422544     0.000000
*



# do normal mode scan calculations
# to map CASSCF ground and excited-state PESs
$new_job
! RHF  TZVP TightSCF  NMScan

%casscf
        nel 6
        norb  6
        mult  1
```

```
        nroots  4
    end

%rr
    HessName "hexatriene_bp86.hess"
    NMList 10,11,18,24,26,28,29,31,32
    NSteps 6
    FreqAlter true
    EnStep 0.0001
    State 3
end

* xyz 0 1
  C       -0.002759    0.680006    0.000000
  H       -0.966741    1.204366    0.000000
  C        1.187413    1.500920    0.000000
  H        2.146702    0.969304    0.000000
  C        1.187413    2.850514    0.000000
  H        0.254386    3.420500    0.000000
  H        2.116263    3.422544    0.000000
  C        0.002759   -0.680006    0.000000
  H        0.966741   -1.204366    0.000000
  C       -1.187413   -1.500920    0.000000
  H       -2.146702   -0.969304    0.000000
  C       -1.187413   -2.850514    0.000000
  H       -0.254386   -3.420500    0.000000
  H       -2.116263   -3.422544    0.000000
*
```

The file containing the hessian matrix (`"hexatriene_bp86.hess"`) was obtained from the BP86/TZVP frequency calculations. The keyword `NMList` provides the list of the normal modes to be scanned. These should be only the totally symmetric vibrations, since only they can be significant for absorption and resonance Raman spectra within the constraints of the IMDHO model. The `FreqAlter` flag indicates whether frequency alterations are assumed in the post-scan potential surface fit. The Parameter `EnStep` is used to select the appropriate step during the scan calculations. The value is chosen such that the average energy change (in hartree) in both directions is not less than this parameter.

6.19 The COSMO Solvation Model

The conductor like screening model developed by Klamt is – within the restrictions posed by a dielectric theory – an efficient and elegant way to calculate the energies, structures and properties of molecules in solution. The solvent is represented as a dielectric polarizable continuum. In the COSMO approach it is initially assumed to be a perfect conductor which completely shields ('screens') the charge density of the solute. The interaction is then scaled to a finite dielectric constant with the simple factor:

$$f\left(\varepsilon\right) = \frac{\varepsilon - 1}{\varepsilon + x} \tag{6.252}$$

where $x = \frac{1}{2}$ has been found most useful. Hence the name of the method COSMO. The critical step is to construct a molecular surface which embeds the solute. A lot of experience indicates that 1.2 times the Van der Waals surface is a good choice. This cavity is then disretized into segments and the charge density on each segment is assumed to be

constant. If the discretization is sufficiently smooth one obtains a good approximation to the continuum limit. The principal idea is that the charge density of the solute (nuclear and electronic) creates compensating charges on the surface which then, in turn, interact with the molecular charge density. Thus, an iterative process is necessary in order to find self-consistent values of the screening charges and the molecular charge density. This is readily incorporated into the self-consistent field process with reasonable computational cost (30-40% increase in the computation time typically). Due to the mathematical simplicity of the model analytic gradients can be formulated and implemented thus opening the way for efficient geometry optimization in solution. Likewise, the first order response to external properties can be calculated in a rather similar way (using the response density) and finally solvent effects on transition energies can also be calculated.

The use of the COSMO model is rather simple. One can either use the simple-input lines such as:

```
! COSMO(solvent)
```

where `solvent` is one of the standard solvents (epsilon,refrac)

- Water (80.4, 1.33), Acetonitrile (36.6, 1.344), Acetone (20.7, 1.359), Ammonia (22.4, 1.33), Ethanol (24.3, 1.361), Methanol (32.63, 1.329), CH2Cl2 (9.08, 1.424), CCl4 (2.24, 1.466), DMF (38.3, 1.430), DMSO (47.2, 1.479), Pyridine (12.5, 1.510), THF (7.25, 1.407), Chloroform (4.9,1.45), Hexane (1.89, 1.375), Toluene (2.4, 1.497).

Alternatively, the model is invoked by giving a `%cosmo` block and specifying the solvent dielectric constant and refractive index (the latter is used only for excited state calculations using the CIS or MRCI modules).

```
%cosmo epsilon 80
       refrac 1.33
       end
```

A number of technical parameters can be set. Most important, the radius used for cavity construction is accessed by:

```
%cosmo Radius[AtomicNumber] Value # in Angstroem!
       end
```

Reasonable defaults are available for the whole periodic table. In particular, the values for H,C,N,O,F,Cl,Br are optimized values and should probably not be changed. ORCA implements three technical variants for the calculation of the potential on the surface:

```
%cosmo PotOpt  analytic # default, exact analytic integrals
               numeric  # use numeric integration
               semianalytic
                 # Potential done numerically
                 # Operator part done analytically
       end
```

The numeric integration is particularly efficient in the framework of DFT for the potential and only slightly less accurate than the analytic integration. It might be worth a try (in particular since you have a numerical integration noise anyways in the calculation of the exchange-correlation energy)! For the Fock matrix update the numeric integration produces a somewhat larger error which may add up to a millihartree or so for larger molecules. Since the analytic Fock matrix update is fast the semianalytic option may be an alternative to the fully analytic treatment. However, at the present level of code optimization there is not so much difference in computation times.

Other technical parameters with their default values are:

```
%cosmo rsolv   1.3
       routf   0.85
       disex   10.0
       nppa    1082
       nspa     92
       # additional expert flags
       PolScal  1.0   # scaling factor for self-consistent
                      # COSMO-RS calculations
       DoOuter  true  # turn the outlying charge correction on or
                      # off.
       end
```

These values influence the cavity construction and are explained in detail in the original publications. They should rarely be touched.

After carrying out the calculation one has a modified total energy and in the printout of the total energy one additional term appears:

```
    Total Energy       :        -114.23417993 Eh          -3108.39200 eV


    Components:
    Nuclear Repulsion  :          31.37089639 Eh            853.62405 eV
    Electronic Energy  :        -145.60507632 Eh          -3962.01605 eV
    COSMO(ediel)       :          -0.00510803 Eh             -0.13899 eV
```

The `COSMO(ediel)` value is the stabilization of the molecular energy by the salvation (note – ETOT-E(diel) is *not* the gas phase energy since the molecular charge density is self-consistenly polarized in the COSMO calculation). The dielectric energy is already contained in the value of the total energy printed which needs not be corrected by COSMO(ediel) further. Note that COSMO(ediel) is half the interaction energy of the solute and the solvent since the other half is spent to polarize the solvent!

One subtlety is the presence of a little bit of molecular charge outside the cavity which is inconsistent with the model. This is handled in COSMO by the "outlying charge correction" which is computed at the end of the SCF calculation

```
    Total Energy after outlying charge correction =   -114.234043011 Eh
```

If you use the value of the total energy for example to compute relative energies this value should be used. The value of the outlying charge correction itself is usually small. It is not added to the SCF energy since it is added non-variationally and would be inconsistent with geometry optimization.

6.20 SMD solvent model

SMD is an implementation of SMD CDS algorithm from the Crammer and Truhlar group citeCrammer2009JPCB.

6.20.1 Method

SMD solvation model bases on the quantum mechanical charge density of a solute molecule interacting with a continuum description of the solvent. In the model the full solute electron density is used without defining partial atomic charges and the solvent is not represented explicitly but rather as a dielectric medium with the surface tension at the solute-solvent boundary. SMD is a universal solvation model for its applicability to any charged or uncharged solute in any solvent or liquid medium for which a few key descriptors are known (in particular, dielectric constant, refractive index, bulk surface tension, and acidity and basicity parameters). Neglected the concentration contribution, the model separates the observable solvation free energy into two main components,

$$\Delta G_{\mathrm{S}} = \Delta G_{\mathrm{ENP}} + \Delta G_{\mathrm{CDS}}. \tag{6.253}$$

In ORCA, the first component is the bulk electrostatic contribution arising from a self-consistent reaction field treatment that involves the electrostatic interaction using the Conductor-like Screening Model (COSMO). However the radii are set to "intrinsic atomic Coulomb radii". The second component called the cavity-dispersion solvent-structure (CDS) term is the contribution resulting from short-range interactions between the solute and solvent molecules in the first solvation shell. This contribution is a sum of terms that are proportional (with geometry-dependent proportionality constants called atomic surface tensions) to the solvent-accessible surface areas of the individual atoms of the solute. The CDS contribution to the free energy of solvation is given by

$$\Delta G_{\mathrm{CDS}} = \sum_{k}^{\mathrm{atoms}} \sigma_k A_k(\mathbf{R}, R_{Z_k} + r_s) + \sigma^{[\mathrm{M}]} \sum_{k}^{\mathrm{atoms}} A_k(\mathbf{R}, R_{Z_k} + r_s), \tag{6.254}$$

where σ_k and $\sigma^{[\mathrm{M}]}$ are the atomic surface tension of atom k and the molecular surface tension, respectively, and A_k is the solvent accessible surface area (SASA). The SASA depends on the geometry $\mathbf{R}$, the set R_{Z_k} of all atomic van der Waals radii, and the solvent radius r_s, which is added to each of the atomic van der Waals radii. In the program Bondi's radii are used for CDS contribution. More details could be found in the SMD paper.

6.20.2 How to cite

We would appreciate if you cite the following papers in publications resulting from the use of SMD:

- A. V. Marenich, C. J. Cramer and D. G. Truhlar, *J. Phys. Chem. B* **113**, 6378, (2009).

6.20.3 Input and output BLOCK

SMD can be employed in single point calculations and geometry optimizations, using in single-determine SCF (HF and DFT) and CASSCF methods. In post SCF the result has been corrected in the reference wave function. User should only specify the keyword "smd" as "true" and the solvent property. In most of cases solvents can be found from the library so that one can add its name in the cosmo block. Rarely user can specify the 6 descriptors of solvents for SMD calculation.

```
%cosmo    smd true      # turn on smd option
          solvent "DMF" # specify the name of solvent from the list
          aqueous false # depends on if the solvent is water
          soln          # index of refraction at optical frequencies at 293 K
          soln25        # index of refraction at optical frequencies at 298 K
          sola          # Abraham's hydrogen bond acidity
          solb          # Abraham's hydrogen bond basicity
          solg          # relative macroscopic surface tension
          Solc          # aromaticity, fraction of non-hydrogenic solvent atoms
                        # that are aromatic carbon atoms
          Solh          # electronegative halogenicity, fraction of non-hydrogenic
                        # solvent atoms that are F, Cl, or Br
   end
```

Before SCF start, it prints the SMD information.

```
--------------------
COSMO INITIALIZATION
--------------------

Epsilon                              ...   37.2190
Refractive Index                     ...    1.0000
Potential Evaluation                 ...   ANALYTIC
Technical COSMO parameters:
  rsolv                              ...    1.3000
  routf                              ...    0.8500
  disex                              ...   10.0000
  nppa                               ... 1082
  nspa                               ...   92
Solvent:                             ... DMF
  Non-Aqueous solution
SMD-CDS solvent descriptors:
  Soln                               ...    1.4305
  Soln25                             ...    1.4280
  Sola                               ...    0.0000
  Solb                               ...    0.7400
  Solg                               ...   49.5600
  Solc                               ...    0.0000
  Solh                               ...    0.0000
Radii:
 Radius for Fe used is    3.6661 Bohr (=   1.9400 Ang.)
 Radius for C  used is    3.2125 Bohr (=   1.7000 Ang.)
 Radius for H  used is    2.2677 Bohr (=   1.2000 Ang.)
```

After SCF is done, the output file shows the SMD contribution to the total energy.

```
          *****************************************************
          *                    SUCCESS                       *
          *           SCF CONVERGED AFTER  14 CYCLES         *
          *****************************************************
```

```
Old exchange energy                        =    -22.029826261 Eh
New exchange energy                        =    -22.029833040 Eh
Exchange energy change after final integration =    -0.000006779 Eh
Total energy after final integration       =  -1664.968094721 Eh
Final COS-X integration done in            =     258.813 sec
      1.76169954764045        -2.99192151761839         5.55983672704991
      1.969904463244464E-003  0.167822602012468        1.173801645083608E-002
Total Energy after outlying charge correction =  -1664.968881930 Eh


SMD CDS free energy correction energy :               -1.58900     Kcal/mol
Total Energy after SMD CDS correction =             -1664.970626953 Eh


----------------
TOTAL SCF ENERGY
----------------


Total Energy       :        -1664.97062695 Eh           -45306.15408 eV


COSMO(ediel)       :           -0.00985444 Eh               -0.26815 eV
SMD CDS (Gcds)     :           -0.00253223 Eh               -0.06891 eV
```

6.20.4 Solvent list

At present there are 179 solvents in the library.

	solvent	short name		solvent	shortname	
1	1,1,1-TRICHLOROETHANE		91		CYCLOPENTANE	
2	1,1,2-TRICHLOROETHANE		92		CYCLOPENTANOL	
3	1,2,4-TRIMETHYLBENZENE		93		CYCLOPENTANONE	
4	1,2-DIBROMOETHANE		94		DECALIN (CIS/TRANS MIXTURE)	
5	1,2-DICHLOROETHANE		95		CIS-DECALIN	
6	1,2-ETHANEDIOL		96		N-DECANE	
7	1,4-DIOXANE		97		DIBROMOMETHANE	
8	1-BROMO-2-METHYLPROPANE		98		DIBUTYLETHER	
9	1-BROMOOCTANE		99		O-DICHLOROBENZENE	
10	1-BROMOPENTANE		100		E-1,2-DICHLOROETHENE	
11	1-BROMOPROPANE		101		Z-1,2-DICHLOROETHENE	
12	1-BUTANOL		102		DICHLOROMETHANE	
13	1-CHLOROHEXANE		103		DIETHYL ETHER	
14	1-CHLOROPENTANE		104		DIETHYL SULFIDE	
15	1-CHLOROPROPANE		105		DIETHYLAMINE	
16	1-DECANOL		106		DIIODOMETHANE	
17	1-FLUOROOCTANE		107		DIISOPROPYL ETHER	
18	1-HEPTANOL		108		CIS-1,2-DIMETHYLCYCLOHEXANE	
19	1-HEXANOL		109		DIMETHYL DISULFIDE	
20	1-HEXENE		110		N,N-DIMETHYLACETAMIDE	
21	1-HEXYNE		111		N,N-DIMETHYLFORMAMIDE	DMF
22	1-IODOBUTANE		112		DIMETHYLSULFOXIDE	DMSO
23	1-IODOHEXADECANE		113		DIPHENYLETHER	
24	1-IODOPENTANE		114		DIPROPYLAMINE	
25	1-IODOPROPANE		115		N-DODECANE	
26	1-NITROPROPANE		116		ETHANETHIOL	
27	1-NONANOL		117		ETHANOL	
28	1-OCTANOL		118		ETHYL ETHANOATE	
29	1-PENTANOL		119		ETHYL METHANOATE	
30	1-PENTENE		120		ETHYL PHENYL ETHER	
31	1-PROPANOL		121		ETHYLBENZENE	
32	2,2,2-TRIFLUOROETHANOL		122		FLUOROBENZENE	
33	2,2,4-TRIMETHYLPENTANE		123		FORMAMIDE	
34	2,4-DIMETHYLPENTANE		124		FORMIC ACID	
35	2,4-DIMETHYLPYRIDINE		125		N-HEPTANE	
36	2,6-DIMETHYLPYRIDINE		126		N-HEXADECANE	
37	2-BROMOPROPANE		127		N-HEXANE	
38	2-BUTANOL		128		HEXANOIC ACID	
39	2-CHLOROBUTANE		129		IODOBENZENE	
40	2-HEPTANONE		130		IODOETHANE	

41	2-HEXANONE		131	IODOMETHANE	
42	2-METHOXYETHANOL		132	ISOPROPYLBENZENE	
43	2-METHYL-1-PROPANOL		133	P-ISOPROPYLTOLUENE	
44	2-METHYL-2-PROPANOL		134	MESITYLENE	
45	2-METHYLPENTANE		135	METHANOL	
46	2-METHYLPYRIDINE		136	METHYL BENZOATE	
47	2-NITROPROPANE		137	METHYL BUTANOATE	
48	2-OCTANONE		138	METHYL ETHANOATE	
49	2-PENTANONE		139	METHYL METHANOATE	
50	2-PROPANOL		140	METHYL PROPANOATE	
51	2-PROPEN-1-OL		141	N-METHYLANILINE	
52	E-2-PENTENE		142	METHYLCYCLOHEXANE	
53	3-METHYLPYRIDINE		143	N-METHYLFORMAMIDE (E/Z MIXTURE)	
54	3-PENTANONE		144	NITROBENZENE	phNO2
55	4-HEPTANONE		145	NITROETHANE	
56	4-METHYL-2-PENTANONE		146	NITROMETHANE	MeNO2
57	4-METHYLPYRIDINE		147	O-NITROTOLUENE	
58	5-NONANONE		148	N-NONANE	
59	ACETIC ACID		149	N-OCTANE	
60	ACETONE		150	N-PENTADECANE	
61	ACETONITRILE	MeCN	151	PENTANAL	
62	ACETOPHENONE		152	N-PENTANE	
63	ANILINE		153	PENTANOIC ACID	
64	ANISOLE		154	PENTYL ETHANOATE	
65	BENZALDEHYDE		155	PENTYLAMINE	
66	BENZENE		156	PERFLUOROBENZENE	
67	BENZONITRILE		157	PROPANAL	
68	BENZYL ALCOHOL		158	PROPANOIC ACID	
69	BROMOBENZENE		159	PROPANONITRILE	
70	BROMOETHANE		160	PROPYL ETHANOATE	
71	BROMOFORM		161	PROPYLAMINE	
72	BUTANAL		162	PYRIDINE	
73	BUTANOIC ACID		163	TETRACHLOROETHENE	
74	BUTANONE		164	TETRAHYDROFURAN	THF
75	BUTANONITRILE		165	TETRAHYDROTHIOPHENE-S,S-DIOXIDE	
76	BUTYL ETHANOATE		166	TETRALIN	
77	BUTYLAMINE		167	THIOPHENE	
78	N-BUTYLBENZENE		168	THIOPHENOL	
79	SEC-BUTYLBENZENE		169	TOLUENE	
80	TERT-BUTYLBENZENE		170	TRANS-DECALIN	
81	CARBON DISULFIDE		171	TRIBUTYLPHOSPHATE	
82	CARBON TETRACHLORIDE	CCl4	172	TRICHLOROETHENE	
83	CHLOROBENZENE		173	TRIETHYLAMINE	
84	CHLOROFORM		174	N-UNDECANE	
85	A-CHLOROTOLUENE		175	WATER	
86	O-CHLOROTOLUENE		176	XYLENE (MIXTURE)	
87	M-CRESOL		177	M-XYLENE	
88	O-CRESOL		178	O-XYLENE	
89	CYCLOHEXANE		179	P-XYLENE	
90	CYCLOHEXANONE				

6.21 Calculation of Properties

6.21.1 Electric Properties

For the calculation of first order (electric dipole and quadrupole moments) and second order (polarizabilities) electric properties, the `%elprop` module was implemented. The second order properties can be calculated through the solution of the CP-SCF equations. Details are shown below:

```
%elprop
  Dipole      true
  Quadrupole  true
  Polar         1  # analytic calculation
                2  # numeric differentiation of dipole
```

```
                    # moments
                 3  # fully numeric second derivatives
   Solver   CG     # CG(conjugate gradient)
                   # other options: DIIS or POPLE(default)
   MaxDIIS  5    # max. dimension of DIIS method
   Shift    0.2  # level shift used in DIIS solver
   Tol      1e-8 # Convergence of the CP-SCF equations
                # (norm of the residual)
   MaxIter 64    # max. number of iterations in CPSCF
   PrintLevel 2
   EField  1e-4 # electric field step (a.u.) used in the
                # numeric calculation of the polarizabilities
   Origin 1     # Center of mass
          2      # Center of nuclear charge
   end
```

The most efficient and accurate way to calculate the polarizability analytically is to use the coupled-perturbed SCF method. The most time consuming and least accurate way is the numerical second derivative of the total energy. This is, for example, required for MP2 calculations because these do not yield a correlated density in the present version 2.9 of ORCA. Note that the numerical differentiation requires: (a) tightly or even very tightly converged SCF calculations and (b) carefully chosen field increments. If the field increment is too large then the truncation error will be large and the values will be unreliable. On the other hand, if the field increment is too small the numerical error associated with the finite difference differentiation will get unacceptably large up to the point where the whole calculation becomes useless.

6.21.2 The Spin-Orbit Coupling Operator

Starting from version 2.4.00 we have implemented several variants of spin-orbit-coupling operators, which are used in the property calculations. These are based on effective potential and mean-field approaches. The options are selected as follows:

```
%rel SOCType 0  # none
              1  # effective nuclear charge
              3  # Effective potential/mean-field approach
                 # (default)
     SOCFlags 1,1,1,1 # Flags for construction of potential
      # Operative only for SOCType 3
      # ----------------------------
      # Flag 1 = 0 - do not include 1-electron terms
      #        = 1 - do include 1-electron terms
      # Flag 2 = 0 - do not include Coulomb terms
      #        = 1 - compute Coulomb terms fully numeric
      #        = 2 - compute Coulomb term seminumeric
      #        = 3 - compute Coulomb term with RI approx.
```

```
#            = 4 - compute Coulomb term exactly
# Flag 3 = 0 - do not include exchange terms
#            = 1 - do include local X-alpha exchange
#   (note: the X-Alpha parameter can be chosen via
#            %rel Xalpha 0.7 (default)
#            = 2 - same as 1 but with sign reversed!
#            = 3 - Exchange via one-center exact
#                  integrals including the spin-other
#                  orbit interaction
#            = 4 - All exchange terms full analytic
#                  (this is expensive)
# Flag 4 = 0 - do not include DFT local correlation
#                  terms
#            = 1 - do include local DFT correlation (here
#                  this is done with VWN5)
SOCMaxCenter 4 # max. number of centers to include in
                 # the integrals (not fully consistently
                 # implemented yet; better leave at 4)
# Preferred choices:
#  SOCFlags 1,3,3,1 = "complete mean field" with
#     efficient Coulomb terms via RI. This is the
#     most accurate one available
#  SOCFlags 1,2,3,1 = the same but to be preferred if
#     no fitting basis is available for RI
#  SOCFlags 1,3,1,1 = "DFT-Veff"
#  SOCFlags 1,3,2,1 = "DFT-Veff but with estimate of
#     spin-other orbit interaction. Almost as accurate
#     as the first variant
end
```

The Breit-Pauli spin-orbit coupling operator is given by:

$$\hat{H}_{SOC} = \hat{H}_{SOC}^{(1)} + \hat{H}_{SOC}^{(2)}$$

with the one- and two-electron contributions

$$\hat{H}_{SOC}^{(1)} = \frac{\alpha^2}{2} \sum_i \sum_A Z_A \frac{(\mathbf{r}_i - \mathbf{R}_A \times \mathbf{p}_i)}{|\mathbf{r}_i - \mathbf{R}_A|^3} \hat{s}_i \equiv \frac{\alpha^2}{2} \sum_i \sum_A Z_A r_{iA}^{-3} \hat{\mathbf{l}}_{iA} \hat{\mathbf{s}}_i \tag{6.255}$$

$$\hat{H}_{SOC}^{(2)} = -\frac{\alpha^2}{2} \sum_i \sum_{j \neq i} \frac{(\mathbf{r}_i - \mathbf{r}_j \times \mathbf{p}_i)}{|\mathbf{r}_i - \mathbf{r}_j|^3} (\hat{\mathbf{s}}_i + 2\hat{\mathbf{s}}_j) \tag{6.256}$$

$$\equiv -\frac{\alpha^2}{2} \sum_i \sum_{j \neq i} \hat{\mathbf{l}}_{ij} r_{ij}^{-3} (\hat{\mathbf{s}}_i + 2\hat{\mathbf{s}}_j) \tag{6.257}$$

This operator would be hard to handle exactly; therefore it is common to introduce mean field and/or effective potential approaches in which the operator is written as an effective one-electron operator:

$$\hat{H}_{SOC} \cong \sum_i \hat{\mathbf{h}}_i^{(eff)} \hat{\mathbf{s}}_i \qquad (6.258)$$

The simplest approximation is to simply use the the one-electron part and regard the nuclear charges as adjustable parameters. Reducing their values from the exact nuclear charge is supposed to account in an average way for the screening of the nuclear charge by the electrons. In our code we use the effective nuclear charges of Koseki et al. This approximation introduces errors which are usually smaller than 10% but sometimes are larger and may approach 20% in some cases. The approximation is best for first row main group elements and the first transition row (2p and 3d elements). For heavier elements it becomes unreliable.

A much better approximation is to take the two-electron terms into account precisely. Without going into details here – the situation is as in Hartree-Fock (or density functional) theory and one gets Coulomb, exchange and correlation terms. The correlation terms (evaluated in a local DFT fashion) are negligible and can be safely neglected. They are optionally included and are not expensive computationally. The Coulomb terms is (after the one-electron term) the second largest contribution and is expensive to evaluate exactly. The situation is such that in the Coulomb-part the spin-other orbit interaction (the second term in the two-electron part) does not contribute and one only has to deal with the spin-own-orbit contribution. The exact evaluation is usually too expensive to evaluate. The RI and seminumeric approximation are much more efficient and introduce only minimal errors (on the order of usually not more than 1 ppm in g-tensor calculations for example) and are therefore recommended. The RI approximation is computationally more efficient. The one-center approximation to the Coulomb term introduces much larger errors. The fully numeric method is both slower and less accurate and is not recommended.

The exchange term has contributions from both the spin-own-orbit and spin-other-orbit interaction. These are taken both into account in the mean-field approximation which is accessed by `Flag 3 = 3`. Here a one-center approximation is much better than for the Coulomb term since both the integrals and the density matrix elements are short ranged. Together with the Coulomb term this gives a very accurate SOC operator which is recommended. The DFT-Veff operator suffers from not treating the spin-other-orbit part in the exchange which gives significant errors (also, local DFT underestimates the exchange contributions from the spin-same-orbit interaction by some 10% relative to HF but this is not a major source of error). However, it is interesting to observe that in the precise analytical evaluation, the spin-other-orbit interaction is exactly -2 times the spin-own-orbit interation. Thus, in the DFT framework one gets a *much* better SOC operator if the sign of the DFT exchange term is simply reversed! This is accessed by `Flag 3 = 2`.

Altogether the new SOC operators are a sigifcant step forward and are also more accurate than those used in other programs which either neglect some contributions or make more approximations. For compatibility reasons the default is still the old effective nuclear charge model und therefore the SOC operator actually wanted should be directly specified.

6.21.3 The EPR/NMR Module

Starting from release 2.4.30 the module `orca_eprnmr` replaces the earlier separate modules for calculating g-tensors and hyperfine couplings (`orca_dftg` and `orca_hf`). The use of the EPR/NMR module is rather simple. The individual flags are given below.

```
%eprnmr
  # Calculate the g-tensor using CP-KS theory
  GTensor 1


  # Calculate the D-tensor
  DTensor so      # spin orbit part
          ss      # spin-spin part (not implemented)
          ssandso # both parts
  DSOC    qro     # quasi-restricted method for SOC part
                  # NOTE: MUST BE DONE WITH KEYWORD ! UNO
          pk       # Pederson/Kortus method.
     # NOTE: both approximations are only valid for
     # pure functionals (no HF exchange)


  PrintLevel n  # Amount of output (default 2)


  # For the solution of the CP-SCF equations:
  Solver   Pople  # Pople Solver (default)
           CG     # conjugate gradient
           DIIS   # DIIS type solver
  MaxIter  32     # Maximum number of iterations
  MaxDIIS  12     # Max. Number of DIIS vectors (only DIIS)
  Tol      1e-6   # Convergence tolerance
  LevelShift 0.2  # LevelShift for DIIS (ignored otherwise)
  EWin    -1000,1000  # orbital window included in CP-SCF
                      # in particular for NMR these should be
                      # all MOs (respects settings in
                      # %method block)
  # For localization:
  locmet   pm     # Pipek-Mezey
           fb     # Foster-Boys


  Ori      CenterOfMass
           CenterOfNucCharge
           CenterOfElCharge  #(default)
           OwnNuc            # for NMR chemical shifts take
                             # the own nucleus as gauge
                             # origin; this removes the
                             # origin dependence but the
                             # basis set convergence may
                             # still be very slow)
           IGLO   # (IGLO treatment; only rigorous if there
                  # is no HF exchange)
           GIAO   # (not yet implemented)
           N      # (number of the atom to put the origin)
```

```
            X,y,z  # (explicit position of the origin)



     Nuclei = all type { flags }
       # Calculate nuclear properties. Here the properties
       # for all nuclei of 'type' are calculated ('type' is
       # an element name, for example Cu.)
       # Flags can be the following:
       #   aiso  - calculate the isotropic part of the HFC
       #   adip  - calculate the dipolar part of the HFC
       #   aorb  - 2nd order contribution to the HFC from SOC
       #   fgrad - calculate the electric field gradient
       #   rho   - calculate the electron density at the
       #           nucleus
       #   shift - NMR chemical shift tensor


     # For example:
     # calculates the hyperfine coupling for all nitrogen atoms
     Nuclei = all N { aiso, adip, fgrad, rho};


     # In addition you can change several parameters:
     Nuclei = all N { PPP=39.1, QQQ=0.5, III=1.0 };
         # PPP : the proportionality factor for this nucleus
         #       = ge*gN*betaE*betaN
         # QQQ : the quadrupole moment for this nucleus
         # III : the spin for this nucleus


     # You can also calculate the hfc's for individual atoms
     # as in the following example:
     Nuclei = 1,5,8 { aiso, adip};

       # NOTE: counting starts with atom 1!!!


     end
```

A little background information might be in order:

6.21.3.1 Hyperfine and Quadrupole Couplings

The hyperfine coupling has three contributions:

(a) The isotropic Fermi contact term that arises from the finite spin density on the nucleus under investigation. It is calculated for nucleus N from:

$$a_{iso}(N) = \left(\frac{4}{3}\pi \langle S_z \rangle^{-1} \right) g_e g_N \beta_e \beta_N \rho \left(\vec{R}_N \right) \tag{6.259}$$

Here, $\langle S_z \rangle$ is the expectation value of the z-component of the total spin, g_e and g_N are the electron and nuclear g-factors and β_e and β_N are the electron and nuclear magnetons respectively. $\rho\left(\vec{R}_N\right)$ is the spin density at the nucleus. The proportionality factor $P_N = g_e g_N \beta_e \beta_N$ is commonly used and has the dimensions MHz bohr^3 in ORCA.

(b) The spin dipole part that arises from the magnetic dipole interaction of the magnetic nucleus with the magnetic moment of the electron. It is also calculated as an expectation value over the spin density as:

$$A_{\mu\nu}^{dip}(N) = P_N \sum kl\rho_{kl} \left\langle \phi_k \left| r_N^{-5}\left(3\vec{r}_{N\mu}\vec{r}_{N\nu} - \delta_{\mu\nu}r_N^2\right) \right| \phi_l \right\rangle \tag{6.260}$$

where ρ is the spin density matrix and $\vec{r}_N$ is a vector of magnitude r_N that points from the nucleus in question to the electron ($\{\phi\}$ is the set of basis functions).

(c) The second order contribution that arises from spin-orbit coupling. Presently ORCA can calculate all these contributions. The first two are calculated as simple expectation values of the appropriate operators over the self-consistent spin density, but the second order contribution requires the solution of the coupled-perturbed SCF equations and is consequently computationally more demanding. The contribution can be written:

$$A_{\mu\nu}^{orb}(N) = -\frac{1}{2S}P_N \sum_{kl} \frac{\partial\rho_{kl}}{\partial I_\mu} \left\langle \phi_k \left| h_\nu^{SOC} \right| \phi_l \right\rangle \tag{6.261}$$

The derivative of the spin density is computed from solving the coupled-perturbed SCF equations with respect to the nucleus-orbit coupling as perturbation. The nucleus-orbit coupling is represented by the operator

$$h_\nu^{NOC}(A) = \sum_i r_{iA}^{-3} l_{i,\nu}^{(A)} \tag{6.262}$$

where the sum is over electrons and A is the nucleus in question.

The **field gradient tensor** is closely related to the dipole contribution to the hyperfine coupling. The main differences are that the electron instead of the spin density enters its calculation and that it contains a nuclear contribution due to the surrounding nuclei. It is calculated from

$$\begin{aligned}
V_{\mu\nu}(N) = &-\sum_{kl} P_{kl} \left\langle \phi_k \left| r_N^{-5}\left(3\vec{r}_{N\mu}\vec{r}_{N\nu} - \delta_{\mu\nu}r_N^2\right) \right| \phi_l \right\rangle \\
&+ \sum_{A \neq N} Z_A \vec{R}_{AN}^{-5}\left(3\vec{R}_{AN\mu}\vec{R}_{AN\nu} - \delta_{\mu\nu}R_{AN}^2\right)
\end{aligned} \tag{6.263}$$

with Z_A as the nuclear charge of nucleus A and $\vec{R}_{AN}$ as a vector of magnitude R_{AN} that points from nucleus A to nucleus N. $\mathbf{P}$ is the first order density matrix.

NOTE:

- Hyperfine and quadrupole couplings are properties where the standard basis sets that have been designed for geometry optimization and the like may not be entirely satisfactory (especially for atoms heavier than Ne). You should probably look into tailoring the basis set according to your needs. While it is likely that a later release will provide one or two special basis sets for 'core-property' calculations at this time you have to make sure yourself that the basis set has enough flexibility in the core region, for example by uncontracting core basis functions and adding s-primitives with large exponents (or using the "decontraction feature", section 6.3). If you add these tight functions and use DFT make sure that the numerical integration is still satisfactory. Use the 'SpecialGrid' feature to enlarge grids for individual atoms without increasing the computational effort too drastically.

- For very heavy nuclei you may not get satisfactory results because scalar relativistic effects may become large. These are available at the ZORA and DKH levels in ORCA if the flag `PictureChange` is set to true in the `%rel` block. The situation is such that for DKH the implementation is only valid for the field gradient while the ZORA methodology is consistent throughout. The IORA method is also available but the results will be approximate. In my experience some nice results have been obtained with ZORA for HFCs and very nice results for EFGs with DKH. In these quasi-relativistic calculations with DFT one has to be extremely cautious with numerical integration accuracy and values of `intacc` of `9.0` and higher have been used for heavier (transition metal) nuclei. Much lower values just produce noise!

Second order HFCs require the calculation of the spin-orbit coupling contributions which in turn requires the calculation of the coupled perturbed SCF equations. These effects can be quite significant for heavier nuclei and should definitely be included for transition metal complexes. The spin-orbit coupling treatment used is the same as described under 6.21.2.

6.21.3.2 The g-Tensor

The EPR g-tensor is a property that can be calculated as a second derivative of the energy and it is implemented as such in ORCA for the SCF methods, e.g. HF and DFT. At the SCF level four contributions arise:

$$g_{\mu\nu}^{(SZ)} = \delta_{\mu\nu} g_e \tag{6.264}$$

$$g_{\mu\nu}^{(RMC)} = -\frac{\alpha^2}{S} \sum_{k,l} P_{kl}^{\alpha-\beta} \left\langle \phi_k \left| \hat{T} \right| \phi_l \right\rangle \tag{6.265}$$

$$g_{\mu\nu}^{(DSO)} = \frac{1}{2S} \sum_{k,l} P_{kl}^{\alpha-\beta} \left\langle \phi_k \left| \sum_A \xi\left(r_A\right) \left[\mathbf{r}_A \mathbf{r}_O - \mathbf{r}_{A,\mu} \mathbf{r}_{O,\nu}\right] \right| \phi_l \right\rangle \tag{6.266}$$

$$g_{\mu\nu}^{(PSO)} = \sum_{k,l} \frac{\partial P_{kl}^{\alpha-\beta}}{\partial B_\mu} \left\langle \phi_k \left| h_\nu^{SOC} \right| \phi_k \right\rangle \tag{6.267}$$

Here, g_e is the free-electron g-value ($=2.002319...$), S is the total spin, α the fine structure constant, $P^{\alpha-\beta}$ is the spin density matrix, $\{\phi\}$ is the basis set, $\hat{T}$ is the kinetic energy operator, $\xi\left(r_A\right)$ an approximate radial operator, h^{SOC} the spatial part of an effective one-electron spin-orbit operator and B_μ is a component of the magnetic field. The derivative of the spin-density matrix is calculated from coupled-perturbed SCF theory with respect to a magnetic field perturbation (orbital Zeeman term).

Accuracy. G-tensor calculations at the SCF level are not highly demanding in terms of basis set size. Basis sets that give reliable SCF results (at least valence double zeta plus polarization) usually also give reliable g-tensor results. For many molecules the Hartree-Fock approximation will give reasonable predictions. In a number of cases, however, it breaks down completely. DFT is more robust in this respect and the number of molecules where it fails is much smaller. Among the density functionals, the hybrid functionals seem to be the most accurate. In my hands PBE0 is perhaps the best although PWP1 and B3LYP are not much worse. The GGA functionals such as BP, PW91, BLYP or PBE are equally good for small radicals but are significantly inferior to their hybrid counterparts for transition metal complexes.

Gauge dependence. Unfortunately, the g-tensor is a gauge dependent property, that is, the results depend on where the origin is chosen within the molecule. Unless fully invariant procedures (such as IGLO or GIAO) are used, this undesirable aspect is always present in the calculations. GIAO will be implemented in later releases but is not yet

available in ORCA. However, if the choice of gauge origin is not outrageously poor, the gauge dependence is usually so small that it can be ignored for all practical purposes, especially if large basis sets are used. ORCA gives you considerable freedom in the choice of gauge origin. It can either be the center of mass, the center of nuclear charge, the center of electronic charge (recommended), a special atom or a user-defined point in space. It is wise to check the sensitivity of the results with respect to the choice of origin, especially when small g-shifts on the order of only a few hundred ppm are calculated. The IGLO method should be a much better treatment of the origin dependence but there is presently little experience with it. At least for NMR chemical shifts (in the absence of HF exchange) it is the method of choice.

Spin-orbit coupling operator. In previous versions of the code, the g-tensor module used the parameterization of Koseki *et al.* [318–320] for the spin-orbit operator. This is expected to be a reasonable approximation for the 2p and 3d elements and less satisfactory for heavier main group or transition metal containing systems. Thus, the main target molecules with the simple operators are radicals made of light atoms and first row transition metal complexes. More accurate SOC operators (at only moderately increased computational cost) have now been implemented and are described in section 6.21.2. With theses operators there are fewer restrictions. However, for very heavy elements they will suffer from the shortcomings of the Breit-Pauli approximation and future releases will modify these operators to take into account the ZORA or DKH corrections to the SOC.

6.21.3.3 Zero-Field-Splitting

It is well known that the ZFS consists of a first order term arising from the direct spin-spin interaction [321]:

$$D_{KL}^{(SS)} = \frac{1}{2}\frac{\alpha^2}{S(2S-1)}\left\langle 0SS\left|\sum_i\sum_{j\neq i}\frac{r_{ij}^2\delta_{KL}-3\left(\mathbf{r}_{ij}\right)_K\left(\mathbf{r}_{ij}\right)_L}{r_{ij}^5}\left\{2\hat{s}_{zi}\hat{s}_{zj}-\hat{s}_{xi}\hat{s}_{xj}-\hat{s}_{yi}\hat{s}_{yj}\right\}\right|0SS\right\rangle \tag{6.268}$$

($K,L=$x,y,z). Here α is the fine structure constant ($\approx 1/137$ in atomic units), $\mathbf{r}_{ij}$ is the electronic distance vector with magnitude r_{ij} and $\hat{s}_i$ is the spin-vector operator for the i'th electron. $|OSS\rangle$ is the exact ground state eigenfunction of the Born-Oppenheimer Hamiltonian with total spin S and projection quantum number $M_S = S$. Since the spin-spin interaction is of first order, it presents no particular difficulties. The more complicated contribution to the **D**-tensor arises from the spin-orbit interaction, which gives a second order contribution. Under the assumption that the spin-orbit coupling (SOC) operator can to a good approximation be represented by an effective one-electron operator ($\hat{H}_{SOC} = \sum_i \hat{\mathbf{h}}_i^{SOC}\hat{\mathbf{s}}_i$), ref [322] has derived the following sum-over-states (SOS) equations for the SOC contribution to the ZFS tensor:

$$D_{KL}^{SOC-(0)} = -\frac{1}{S^2}\sum_{b(S_b=S)}\Delta_b^{-1}\left\langle 0^{SS}\left|\sum_i\hat{h}_i^{K;SOC}\hat{s}_{i,0}\right|b^{SS}\right\rangle\left\langle b^{SS}\left|\sum_i\hat{h}_i^{L;SOC}\hat{s}_{i,0}\right|0^{SS}\right\rangle \tag{6.269}$$

$$D_{KL}^{SOC-(-1)} = -\frac{1}{S(2S-1)}\sum_{b(S_b=S-1)}\Delta_b^{-1}\left\langle 0^{SS}\left|\sum_i\hat{h}_i^{K;SOC}\hat{s}_{i,+1}\right|b^{S-1S-1}\right\rangle\left\langle b^{S-1S-1}\left|\sum_i\hat{h}_i^{L;SOC}\hat{s}_{i,-1}\right|0^{SS}\right\rangle \tag{6.270}$$

$$D_{KL}^{SOC-(+1)} = -\frac{1}{(S+1)(2S+1)}\cdot$$
$$\sum_{b(S_b=S+1)}\Delta_b^{-1}\left\langle 0^{SS}\left|\sum_i\hat{h}_i^{K;SOC}\hat{s}_{i,-1}\right|b^{S+1S+1}\right\rangle\left\langle b^{S+1S+1}\left|\sum_i\hat{h}_i^{L;SOC}\hat{s}_{i,+1}\right|0^{SS}\right\rangle \tag{6.271}$$

Here the one-electron spin-operator for electron i has been written in terms of spherical vector operator components $s_{i,m}$ with $m = 0, \pm 1$ and $\Delta_b = E_b - E_0$ is the excitation energy to the excited state multiplet $\left| b^{SS} \right\rangle$ (all M_S components are degenerate at the level of the BO Hamiltonian).

One attractive possibility is to represent the SOC by the spin-orbit mean-field (SOMF) method developed by Hess et al., [323] widely used in the AMFI program by Schimmelpfennig [324, 325] and discussed in detail by Berning et al. [326] as well as in ref [327]. In terms of an (orthonormal) one-electron basis, the matrix elements of the SOMF operator are:

$$
\begin{aligned}
h_{rs}^{K;SOC} &= \left(p \left| \hat{h}_K^{1el-SOC} \right| q \right) \\
&+ \sum_{rs} P_{rs} \left[\left(pq \left| \hat{g}_K^{SOC} \right| rs \right) - \tfrac{3}{2} \left(pr \left| \hat{g}_K^{SOC} \right| sq \right) - \tfrac{3}{2} \left(sq \left| \hat{g}_K^{SOC} \right| pr \right) \right]
\end{aligned}
\tag{6.272}
$$

and:

$$
\hat{h}_k^{1el-SOC} \left(\mathbf{r}_i \right) = \frac{\alpha^2}{2} \sum_i \sum_A Z_A r_{iA}^{-3} \hat{l}_{iA;k}
\tag{6.273}
$$

$$
\hat{g}_k^{SOC} \left(\mathbf{r}_i, \mathbf{r}_j \right) = -\frac{\alpha^2}{2} \hat{l}_{ij;k} r_{ij}^{-3}
\tag{6.274}
$$

$\hat{l}_{iA} = (\hat{\mathbf{r}}_i - \mathbf{R}_A) \times \hat{\mathbf{p}}_i$ is the angular momentum of the i'th electron relative to nucleus A. The vector $\hat{\mathbf{r}}_{iA} = \hat{\mathbf{r}}_i - \mathbf{R}_A$ of magnitude r_{iA} is the position of the i'th electron relative to atom A. Likewise, the vector $\hat{\mathbf{r}}_{ij} = \hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j$ of magnitude r_{ij} is the position of the i'th electron relative to electron j and $\hat{l}_{ij} = (\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j) \times \hat{\mathbf{p}}_i$ is its angular momentum relative to this electron. $\mathbf{P}$ is the charge density matrix of the electron ground state ($P_{pq} = \left\langle 0SS \left| E_q^p \right| 0SS \right\rangle$ with $E_q^p = a_{p\beta}^+ a_{q\beta} + a_{p\alpha}^+ a_{q\alpha}$ where $a_{p\sigma}^+$ and $a_{q\sigma}$ are the usual Fermion creation and annihilation operators).

6.21.3.4 General Treatment

The zero-field splitting (ZFS) is typically the leading term in the Spin-Hamiltonian (SH) for transition metal complexes with a total ground state spin $S > 1/2$ (for reviews and references see chapter 8). Its net effect is to introduce a splitting of the $2S + 1$ M_S levels (which are exactly degenerate at the level of the Born-Oppenheimer Hamiltonian), even in the absence of an external magnetic field. Thus, an analysis and interpretation of the ZFS is imperative if the information content of the various physical methods that are sensitive to ZFS effects.

In 2007, we have developed a procedure that makes the ZFS calculation compatible with the language of analytic derivatives (FN (**2007**) *J. Chem. Phys.*, 127, 164112). Perhaps the most transparent route is to start from the exact solutions of the Born-Oppenheimer Hamiltonian. To this end, we look at the second derivative of the ground state energy ($E = \left\langle 0^{SS} \left| \hat{H} \right| 0^{SS} \right\rangle$) with respect to a spin-dependent one-electron operator of the general form:

$$
\hat{h}^{K;(m)} = x_K^{(m)} \sum_{pq} h_{pq}^K \hat{S}_{pq}^{(m)}
\tag{6.275}
$$

Where h_{pq}^K is the matrix of the K'th component of the spatial part of the operator (assumed to be imaginary Hermitian as is the case for the spatial components of the SOC operator) and $\hat{S}_{pq}^{(m)}$ is the second quantized form of the spin vector operator ($m = 0, \pm 1$). The quantity $x_K^{(m)}$ is a formal perturbation parameter. Using the exact eigenfunctions of the BO operator, the first derivative is:

$$\frac{\partial E}{\partial x_K^{(m)}}\bigg|_{x_K^{(m)}=0} = \sum_{pq} h_{pq}^K P_{pq}^{(m)} \tag{6.276}$$

With the components of the spin-density:

$$P_{pq}^{(m)} = \left\langle 0^{SS} | \hat{S}_{pq}^{(m)} | 0^{SS} \right\rangle \tag{6.277}$$

The second derivative with respect to a second component for $m' = -m$ is:

$$\frac{\partial^2 E}{\partial x_K^{(m)} \partial x_L^{(-m)}}\bigg|_{x_K^{(m)}=x_L^{(-m)}=0} = \sum_{pq} h_{pq}^K \frac{\partial P_{pq}^{(m)}}{x_L^{(-m)}} \tag{6.278}$$

The derivative of the spin-density may be written as:

$$\frac{\partial P_{pq}^{(m)}}{x_L^{(-m)}} = \left\langle 0_L^{SS(-m)} | \hat{S}_{pq}^{(m)} | 0^{SS} \right\rangle + \left\langle 0^{SS} | \hat{S}_{pq}^{(m)} | 0_L^{SS(-m)} \right\rangle \tag{6.279}$$

Expanding the perturbed wavefunction in terms of the unperturbed states gives to first order:

$$\left| 0_L^{SS(-m)} \right\rangle = -\sum n \neq 0 \Delta_n^{-1} |n\rangle \left\langle n \left| \hat{h}^{L;(-m)} \right| 0^{SS} \right\rangle \tag{6.280}$$

Where $|n\rangle$ is any of the $\left| b^{S'M'} \right\rangle$. Thus, one gets:

$$\frac{\partial^2 E}{\partial x_K^{(m)} \partial x_L^{(-m)}} = \sum_{pq} h_{pq}^K \frac{\partial P_{pq}^{(m)}}{x_L^{(-m)}} \tag{6.281}$$

$$= -\sum_{n \neq 0} \Delta_n^{-1} \left[\left\langle 0^{SS} | \hat{h}^{L;(-m)} | n \right\rangle \left\langle n | \hat{h}^{K;(m)} | 0^{SS} \right\rangle + \left\langle 0^{SS} | \hat{h}^{K;(m)} | n \right\rangle \left\langle n | \hat{h}^{L;(-m)} | 0^{SS} \right\rangle \right] \tag{6.282}$$

The equality holds for exact states. For approximate electronic structure treatments, the analytic derivative approach is more attractive since an infinite sum over states can never be performed in practice and the calculation of analytic derivative is computationally less demanding than the calculation of excited many electron states.

Using eq 6.281, the components of the SOC-contribution to the **D**-tensor are reformulated as

$$D_{KL}^{SOC-(0)} = \frac{1}{2S^2} \sum_{pq} h_{pq}^{K;SOC} \frac{\partial P_{pq}^{(0)}}{\partial x_L^{(0)}} \tag{6.283}$$

$$D_{KL}^{SOC-(-1)} = \frac{1}{S(2S-1)} \sum_{pq} h_{pq}^{K;SOC} \frac{\partial P_{pq}^{(+1)}}{\partial x_L^{(-1)}} \tag{6.284}$$

$$D_{KL}^{SOC-(+1)} = \frac{1}{(S+1)(2S+1)} \sum_{pq} h_{pq}^{K;SOC} \frac{\partial P_{pq}^{(-1)}}{\partial x_L^{(+1)}} \tag{6.285}$$

These are general equations that can be applied together with any non-relativistic or scalar relativistic electronic structure method that can be cast in second quantized form. Below, the formalism is applied to the case of a self-consistent field (HF, DFT) reference state.

For DFT or HF ground states, the equations are further developed as follows:

The SCF energy is:

$$E_{SCF} = V_{NN} + \langle \mathbf{Ph}^+ \rangle + \frac{1}{2} \int \int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 - \frac{1}{2} a_x \sum_{\mu\nu\kappa\tau\sigma} P^\sigma_{\mu\kappa} P^\sigma_{\nu\tau} (\mu\nu|\kappa\tau) + c_{DF} E_{XC}[\rho_\alpha, \rho_\beta] \qquad (6.286)$$

Here V_{NN} is the nuclear repulsion energy and $h_{\mu\nu}$ is a matrix element of the one-electron operator which contains the kinetic energy and electron-nuclear attraction terms ($\langle \mathbf{ab} \rangle$ denotes the trace of the matrix product $\mathbf{ab}$). As usual, the molecular spin-orbitals ψ^σ_p are expanded in atom centered basis functions ($\sigma = \alpha, \beta$):

$$\psi^\sigma_p(\mathbf{r}) = \sum_\mu c^\sigma_{\mu p} \phi_\mu(\mathbf{r}) \qquad (6.287)$$

with MO coefficients $c^\sigma_{\mu p}$. The two-electron integrals are defined as:

$$(\mu\nu|\kappa\tau) = \int \int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1) r_{12}^{-1} \phi_\kappa(\mathbf{r}_2)\phi_\tau(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \qquad (6.288)$$

The mixing parameter a_x controls the fraction of Hartree-Fock exchange and is of a semi-empirical nature. $E_{XC}[\rho_\alpha, \rho_\beta]$ represent the exchange-correlation energy. The parameter c_{DF} is an overall scaling factor that allows one to proceed from Hartree-Fock theory ($a_x = 1, c_{DF} = 0$) to pure DFT ($a_x = 0, c_{DF} = 1$) to hybrid DFT ($0 < a_x < 1, c_{DF} = 1$). The orbitals satisfy the spin-unrestricted SCF equations:

$$F^\sigma_{\mu\nu} = h_{\mu\nu} + \sum_{\kappa\tau} P_{\kappa\tau}(\mu\nu|\kappa\tau) - a_x P^\sigma_{\kappa\tau}(\mu\kappa|\nu\tau) + c_{DF}(\mu|V^\alpha_{XC}|\nu) \qquad (6.289)$$

With $V^\sigma_{XC} = \frac{\delta E_{XC}}{\delta \rho_\sigma(\mathbf{r})}$ and $P_{\mu\nu} = P^\alpha_{\mu\nu} + P^\beta_{\mu\nu}$ being the total electron density. For the SOC perturbation it is customary to regard the basis set as perturbation independent. In a spin-unrestricted treatment, the first derivative is:

$$\frac{\partial E_{SCF}}{\partial x_K^{(m)}} = \sum_{i_\alpha} \left(i_\alpha | h^K s_m | i_\alpha \right) + \sum_{i_\beta} \left(i_\beta | h^K s_m | i_\beta \right) = 0 \qquad (6.290)$$

For the second derivative, the perturbed orbitals are required. However, in the presence of a spin-dependent perturbation they can no longer be taken as pure spin-up or spin-down orbitals. With respect to the L'th component of the perturbation for spin-component m, the orbitals are expanded as:

$$\psi^{\alpha;(m)L}_i(\mathbf{r}) = \sum_{a_\alpha} U^{(m);L}_{a_\alpha i_\alpha} \psi^\alpha_a(\mathbf{r}) + \sum_{a_\beta} U^{(m);L}_{a_\beta i_\alpha} \psi^\beta_a(\mathbf{r}) \qquad (6.291)$$

$$\psi^{\beta;(m)L}_i(\mathbf{r}) = \sum_{a_\alpha} U^{(m);L}_{a_\alpha i_\beta} \psi^\alpha_a(\mathbf{r}) + \sum_{a_\beta} U^{(m);L}_{a_\beta i_\beta} \psi^\beta_a(\mathbf{r}) \qquad (6.292)$$

Since the matrix elements of the spin-vector operator components are purely real and the spatial part of the SOC operator has purely complex matrix elements, it follows that the first order coefficients are purely complex. The second derivative of the total SCF energy becomes:

$$\frac{\partial^2 E_{SCF}}{\partial x_K^{(m)} \partial x_L^{(-m)}} = \sum_{i_\alpha a_\alpha} \left\{ U_{a_\alpha i_\alpha}^{(-m);L*} \left(a_\alpha | h^K s_m | i_\alpha \right) + U_{a_\alpha i_\alpha}^{(-m);L} \left(i_\alpha | h^K s_m | a_\alpha \right) \right\}$$
$$+ \sum_{i_\alpha a_\beta} \left\{ U_{a_\beta i_\alpha}^{(-m);L*} \left(a_\beta | h^K s_m | i_\alpha \right) + U_{a_\beta i_\alpha}^{(-m);L} \left(i_\alpha | h^K s_m | a_\beta \right) \right\}$$
$$+ \sum_{i_\beta a_\alpha} \left\{ U_{a_\alpha i_\beta}^{(-m);L*} \left(a_\alpha | h^K s_m | i_\beta \right) + U_{a_\alpha i_\beta}^{(-m);L} \left(i_\beta | h^K s_m | a_\alpha \right) \right\}$$
$$+ \sum_{i_\beta a_\beta} \left\{ U_{a_\beta i_\beta}^{(-m);L*} \left(a_\beta | h^K s_m | i_\beta \right) + U_{a_\beta i_\beta}^{(-m);L} \left(i_\beta | h^K s_m | a_\beta \right) \right\} \tag{6.293}$$

Examination of the three cases $m = 0, \pm 1$ leads to the following equations for the **D**-tensor components:

$$D_{KL}^{(0)} = -\frac{1}{4S^2} \sum_{\mu\nu} \frac{\partial P_{\mu\nu}^{(0)}}{\partial x_L^{(0)}} \left(\mu | h^{K;SOC} | \nu \right) \tag{6.294}$$

$$D_{KL}^{(+1)} = \frac{1}{2(S+1)(2S+1)} \sum_{\mu\nu} \left(\mu | h^{K;SOC} | \nu \right) \frac{\partial P_{\mu\nu}^{(-1)}}{\partial x_L^{(+1)}} \tag{6.295}$$

$$D_{KL}^{(-1)} = \frac{1}{2S(2S-1)} \sum_{\mu\nu} \left(\mu | h^{K;SOC} | \nu \right) \frac{\partial P_{\mu\nu}^{(+1)}}{\partial x_L^{(-1)}} \tag{6.296}$$

Where a special form of the perturbed densities has been chosen. They are given in the atomic orbital basis as:

$$\frac{\partial P_{\mu\nu}^{(0)}}{\partial x_L^{(0)}} = \sum_{i_\alpha a_\alpha} U_{a_\alpha i_\alpha}^{(0);L} c_{\mu i}^\alpha c_{\nu a}^\alpha + \sum_{i_\beta a_\beta} U_{a_\beta i_\beta}^{(0);L} c_{\mu i}^\beta c_{\nu a}^\beta \tag{6.297}$$

$$\frac{\partial P_{\mu\nu}^{(+1)}}{\partial x_L^{(-1)}} = \sum_{i_\alpha a_\beta} U_{a_\beta i_\alpha}^{(-1);L} c_{\mu i}^\alpha c_{\nu a}^\beta - \sum_{i_\beta a_\alpha} U_{a_\alpha i_\beta}^{(-1);L} c_{\mu a}^\alpha c_{\nu i}^\beta \tag{6.298}$$

$$\frac{\partial P_{\mu\nu}^{(-1)}}{\partial x_L^{(+1)}} = -\sum_{i_\alpha a_\beta} U_{a_\beta i_\alpha}^{(+1);L} c_{\mu a}^\beta c_{\nu i}^\alpha + \sum_{i_\beta a_\alpha} U_{a_\alpha i_\beta}^{(+1);L} c_{\mu i}^\beta c_{\nu a}^\alpha \tag{6.299}$$

The special form of the coupled perturbed equations are implemented in ORCA run as follows: The perturbation is of the general form $h^K \hat{s}_m$. The equations 6.294-6.299 and 6.300-6.305 below have been written in such a way that the spin-integration has been performed but that the spin-dependent factors have been dropped from the right-hand sides and included in the prefactors of eqs 6.294-6.296. The explicit forms of the linear equations to be solved are:

$m = 0$:

$$\left(\varepsilon_{a_\alpha}^{(0)} - \varepsilon_{i_\alpha}^{(0)} \right) U_{a_\alpha i_\alpha}^{K(0)} + a_x \sum_{j_\alpha b_\alpha} U_{b_\alpha j_\alpha}^{K(m)} \left\{ (b_\alpha i_\alpha | a_\alpha j_\alpha) - (j_\alpha i_\alpha | a_\alpha b_\alpha) \right\} = - \left(a_\alpha | h^K | i_\alpha \right) \tag{6.300}$$

$$\left(\varepsilon_{a_\beta}^{(0)} - \varepsilon_{i_\beta}^{(0)} \right) U_{a_\beta i_\beta}^{K(0)} + a_x \sum_{j_\beta b_\beta} U_{b_\beta j_\beta}^{K(m)} \left\{ (b_\beta i_\beta | a_\beta j_\beta) - (j_\beta i_\beta | a_\beta b_\beta) \right\} = - \left(a_\beta | h^K | i_\beta \right) \tag{6.301}$$

$m = +1$:

$$\left(\varepsilon_{a_\alpha}^{(0)} - \varepsilon_{i_\beta}^{(0)} \right) U_{a_\alpha i_\beta}^{K(+1)} + a_x \sum_{j_\alpha b_\beta} U_{b_\beta j_\alpha}^{K(+1)} (b_\beta i_\beta | a_\alpha j_\alpha) - a_x \sum_{b_\alpha j_\beta} U_{b_\beta j_\alpha}^{K(+1)} (j_\beta i_\beta | a_\alpha b_\alpha) = - \left(a_\alpha | h^K | i_\beta \right) \tag{6.302}$$

$$\left(\varepsilon_{a_\beta}^{(0)} - \varepsilon_{i_\alpha}^{(0)}\right) U_{a_\beta i_\alpha}^{K(+1)} + a_x \sum_{j_\beta b_\alpha} U_{b_\alpha j_\beta}^{K(+1)} \left(b_\alpha i_\alpha | a_b j_\beta\right) - a_x \sum_{b_\beta j_\alpha} U_{b_\beta j_\alpha}^{K(+1)} \left(j_\alpha i_\alpha | a_\beta b_\beta\right) = 0 \tag{6.303}$$

$m = -1$:

$$\left(\varepsilon_{a_\beta}^{(0)} - \varepsilon_{i_\alpha}^{(0)}\right) U_{a_\beta i_\alpha}^{K(-1)} + a_x \sum_{j_\beta b_\alpha} U_{b_\alpha j_\beta}^{K(-1)} \left(b_\alpha i_\alpha | a_b j_\beta\right) - a_x \sum_{b_\beta j_\alpha} U_{b_\beta j_\alpha}^{K(-1)} \left(j_\alpha i_\alpha | a_\beta b_\beta\right) = -\left(a_\beta | h^K | i_\alpha\right) \tag{6.304}$$

$$\left(\varepsilon_{a_\alpha}^{(0)} - \varepsilon_{i_\beta}^{(0)}\right) U_{a_\alpha i_\beta}^{K(-1)} + a_x \sum_{j_\alpha b_\alpha} U_{b_\beta j_\alpha}^{K(-1)} \left(b_\beta i_\beta | a_\alpha j_\alpha\right) - a_x \sum_{b_\alpha j_\beta} U_{b_\beta j_\alpha}^{K(-1)} \left(j_\beta i_\beta | a_\alpha b_\alpha\right) = 0 \tag{6.305}$$

Note that these coupled-perturbed (CP) equations contain no contribution from the Coulomb potential or any other local potential such as the exchange-correlation potential in DFT. Hence, in the absence of HF exchange, the equations are trivially solved:

$$U_{a_\alpha i_\alpha}^{K(0)} = -\frac{\left(a_\alpha | h^K | i_\alpha\right)}{\varepsilon_{a_\alpha}^{(0)} - \varepsilon_{i_\alpha}^{(0)}} \tag{6.306}$$

$$U_{a_\beta i_\beta}^{K(0)} = -\frac{\left(a_\beta | h^K | i_\beta\right)}{\varepsilon_{a_\beta}^{(0)} - \varepsilon_{i_\beta}^{(0)}} \tag{6.307}$$

$$U_{a_\alpha i_\beta}^{K(+1)} = -\frac{\left(a_\alpha | h^K | i_\beta\right)}{\varepsilon_{a_\alpha}^{(0)} - \varepsilon_{i_\beta}^{(0)}} \tag{6.308}$$

$$U_{a_\beta i_\alpha}^{K(+1)} = 0 \tag{6.309}$$

$$U_{a_\beta i_\alpha}^{K(-1)} = -\frac{\left(a_\beta | h^K | i_\alpha\right)}{\varepsilon_{a_\beta}^{(0)} - \varepsilon_{i_\alpha}^{(0)}} \tag{6.310}$$

$$U_{a_\alpha i_\beta}^{K(-1)} = 0 \tag{6.311}$$

It is interesting that the "reverse spin flip coefficients" $U_{a_\beta i_\alpha}^{K(+1)}$ and $U_{a_\alpha i_\beta}^{K(-1)}$ are only nonzero in the presence of HF exchange. In a perturbation expansion of the CP equations they arise at second order ($V^2/\Delta\varepsilon^2$) while the other coefficients are of first order ($V/\Delta\varepsilon$; V represents the matrix elements of the perturbation). Hence, these contributions are of the order of α^4 and one could conceive dropping them from the treatment in order to stay consistently at the level of α^2. These terms were nevertheless kept in the present treatment.

Equations 6.300-6.305 are referred to as CP-SOC (coupled-perturbed spin-orbit coupling) equations. They can be solved by standard techniques and represent the desired analogue of the CP-SCF magnetic response equations solved for the determination of the g-tensor and discussed in detail earlier [328]. It is readily confirmed that in the absence of HF exchange, eqs 6.306-6.311 inserted into eqs 6.294-6.299 lead back to a modified Pederson-Khanna type treatment of the SOC contributions to the **D**-tensor [329]. In the framework of the formalism developed above, the Pederson-Khanna formula can be re-written in the form:

$$D_{KL}^{(SOC;PK)} = \frac{1}{4S^2} \sum_{i_\beta, a_\beta} \left(\psi_i^\beta \left| h^{K;SOC} \right| \psi_a^\beta \right) U_{a_\beta i_\beta}^{L;(0)} + \frac{1}{4S^2} \sum_{i_\alpha, a_\alpha} \left(\psi_i^\alpha \left| h^{K;SOC} \right| \psi_a^\alpha \right) U_{a_\alpha i_\alpha}^{L;(0)}$$
$$- \frac{1}{4S^2} \sum_{i_\alpha, a_\beta} \left(\psi_i^\alpha \left| h^{K;SOC} \right| \psi_a^\beta \right) U_{a_\beta i_\alpha}^{L;(-1)} - \frac{1}{4S^2} \sum_{i_\beta, a_\alpha} \left(\psi_i^\alpha \left| h^{K;SOC} \right| \psi_a^\alpha \right) U_{a_\alpha i_\beta}^{L;(+1)} \tag{6.312}$$

This equation was derived from second-order non-self-consistent perturbation theory without recourse to spin-coupling. For the special case of no Hartree-Fock exchange, the main difference to the treatment presented here is that the correct prefactors from eqs 6.283-6.285 occur in front of the spin-flip contributions rather than $\pm 1/(4S^2)$ in eq 6.312. In the presence of HF exchange it is suggested that the consistent generalization of the PK method are eqs 6.294-6.296 with the $\pm 1/(4S^2)$ prefactors and this way the method has been implemented as an option into the ORCA program.

For completeness, the evaluation of the spin-spin term in the SCF case proceeds conveniently through:

$$D_{KL}^{(SS)} = \frac{g_e^2}{4} \frac{\alpha^2}{S(2S-1)} \sum_{\mu\nu} \sum_{\kappa\tau} \left\{ P_{\mu\nu}^{\alpha-\beta} P_{\kappa\tau}^{\alpha-\beta} - P_{\mu\kappa}^{\alpha-\beta} P_{\nu\tau}^{\alpha-\beta} \right\} \left\langle \mu\nu \left| r_{12}^{-5} \left\{ 3r_{12,K} r_{12,L} - \delta_{KL} r_{12}^2 \right\} \right| \kappa\tau \right\rangle \tag{6.313}$$

as derived by McWeeny and Mizuno and discussed in some detail by Sinecker and FN (2006). In this reference it was found that DFT methods tend to overestimate the spin-spin contribution if the calculations are based on a spin-unrestricted SCF treatment. A much better correlation with experiment was found for open-shell spin restricted calculations. The origin of this effect proved to be difficult to understand but it was shown in ref [330] that in the case of small spin-contamination, the results of ROKS calculations and of those that are obtained on the basis of the spin-unrestricted natural orbital (UNO) determinant are virtually indistinguishable. It is therefore optionally possible in the ORCA program to calculate the spin-spin term on the basis of the UNO determinant.

6.22 Population Analyses and Control of Output

At present ORCA knows three different ways of analyzing the computed SCF wavefunction that will be described below. All of these methods can produce a tremendous amount of output. However, this output can be precisely controlled by the user to his or her individual needs.

In general there is one compound key called `PrintLevel` which is there to choose reasonable amounts of output. All that `PrintLevel` does is to set certain flags in the array `Print` which holds the details about what to print and what not.

6.22.1 Controlling Output

The array `Print` allows the control of output. The general way of assigning elements of `Print` is:

```
%output
  PrintLevel    Normal
  Print[ Flag ] 0  # turn print off
                1  # turn print on
                n  # some flags are more sophisticated
  end
```

The compound key `PrintLevel` can be used to select certain default settings for the print array. Specifying `Print` after `PrintLevel` can be used to modify these defaults.

```
%output
   PrintLevel      Nothing
                   Mini
      Small

                   Normal
                   Maxi
      Large

                   Huge
                   Debug
      end
```

`Print` has presently the following elements that can be user controlled:

Flag	Action
P_InputFile	Echo the input file
P_Cartesian	Print the cartesian coordinates
P_Internal	Print the internal coordinates
P_Basis	$= 1$: Print the basis set information
	$= 2$: Also print the primitives in input format
P_OneElec	Print of the one electron matrix
P_Overlap	Print the overlap matrix
P_KinEn	Print the kinetic energy matrix
P_S12	Print the $S^{-1/2}$ matrix
P_GuessOrb	Print the initial guess orbitals
P_OrbEn	Print Orbital Energies
P_Mos	Print the MO coefficients on convergence
P_Density	Print the converged electron density
P_SpinDensity	Print the converged spin density
P_EHDetails	Print initial guess extended Hückel details
P_SCFInfo	Print the SCF input flags
P_SCFIterInfo	$= 1$: print short iteration information
	$= 2$: print longer iteration information
	$= 3$: in a direct SCF also print integral progress
P_Fockian	Print Fockian matrix
P_DIISMat	Print DIIS matrix
P_DIISError	Print DIIS error
P_Iter_P	Print Density
P_Iter_C	Print MO coefficients
P_Iter_F	Print Fock matrix
P_Mayer	Print Mayer population analysis. Default = on.
P_NatPop	Print Natural population analysis. Default = off.
P_Hirshfeld	Print Hirshfeld population analysis. Default = off.
P_Mulliken	Print Mulliken population analysis. Default = on
P_AtCharges_M	Print Mulliken atomic charges
P_OrbCharges_M	Print Mulliken orbital charges

P_FragCharges_M	Print Mulliken fragment charges
P_FragBondOrder_M	Print Mulliken fragment bond orders
P_BondOrder_M	Print Mulliken bond orders
P_ReducedOrbPop_M	Print Mulliken reduced orb. charges
P_FragPopMO_M	Print Mulliken fragment population for each MO
P_FragOvlMO_M	Print Mulliken overlap populations per fragment pair
P_AtPopMO_M	Print Mulliken atomic charges in each MO
P_OrbPopMO_M	Print Mulliken orbital population for each MO
P_ReducedOrbPopMO_M	Print Mulliken reduced orbital population for each MO
P_Loewdin	Print Loewdin population analysis. Default = on.
P_AtCharges_L	Print Loewdin atomic charges
P_OrbCharges_L	Print Loewdin orbital charges
P_FragCharges_L	Print Loewdin fragment charges
P_FragBondOrder_L	Print Loewdin fragment bond orders
P_BondOrder_L	Print Loewdin bond orders
P_ReducedOrbPop_L	Print Loewdin reduced orb. charges
P_FragPopMO_L	Print Loewdin fragment population for each MO
P_FragOvlMO_L	Print Loewdin overlap populations per fragment pair
P_AtPopMO_L	Print Loewdin atomic charges in each MO
P_OrbPopMO_L	Print Loewdin orbital population for each MO
P_ReducedOrbPopMO_L	Print Loewdin reduced orbital population for each MO
P_NPA	Natural population analysis
P_NBO	Natural bond orbital analysis
P_Fragments	Print fragment information
P_GUESSPOP	Print initial guess populations
P_UNO_FragPopMO_M	Print Mulliken fragment population per UNO
P_UNO_OrbPopMO_M	Print Mulliken orbital pop. per UNO
P_UNO_AtPopMO_M	Print Mulliken atomic charges per UNO
P_UNO_ReducedOrbPopMO_M	Print Mulliken reduced orbital pop. per UNO
P_UNO_FragPopMO_L	Print Loewdin fragment population per UNO
P_UNO_OrbPopMO_L	Print Loewdin orbital pop. per UNO
P_UNO_AtPopMO_L	Print Loewdin atomic charges per UNO
P_UNO_ReducedOrbPopMO_L	Print Loewdin reduced orbital pop. per UNO
P_UNO_OccNum	Print occupation numbers per UNO
P_AtomExpVal	Print atomic expectation values
P_AtomBasis	Print atomic basis
P_AtomDensFit	Print electron density fit
P_Symmetry	Symmetry basic information
P_Sym_Salc	Symmetry process printing
P_DFTD	Print info on Grimme's dispersion correction
	print mini = 0
	print small = 1
	print normal = 1
	print maxi = 2
	print huge = 2

P_DFTD_GRAD	Print gradient info on Grimme's dispersion correction
	print mini = 0
	print small = 0
	print normal = 0
	print maxi = 1
	print huge = 2
P_Unpaired_T	Print Mulliken and Loewdin analysis for Takatsuka's eff. unpaired electron density.
P_Unpaired_HG	Print Mulliken and Loewdin analysis for Head-Gordon's eff. unpaired electron density

The various choices for PrintLevel have the following defaults:

PrintLevel	Print settings	
Mini	P_OrbEn	= 1
	P_Cartesian	= 1
	P_InputFile	= 1
	P_SCFIterInfo	= 1
Small	all the previous plus	
	P_SCFInfo	= 1
	P_Mayer	= 1
	P_MULLIKEN	= 1
	P_AtCharges_M	= 1
	P_ReducedOrbPop_M	= 1
	P_Loewdin	= 1
	P_AtCharges_L	= 1
	P_ReducedOrbPop_L	= 1
	P_Fragments	= 1
	P_FragCharges_M	= 1
	P_FragBondOrder_M	= 1
	P_FragCharges_L	= 1
	P_FragBondOrder_L	= 1
Normal	all the previous plus	
	P_Internal	= 1
	P_BondOrder_L	= 1
	P_BondOrder_M	= 1
	P_FragPopMO_L	= 1
	P_ReducedOrbPopMO_L	= 1
	P_SCFIterInfo	= 2
Maxi	all the previous plus	
	P_GuessOrb	= 1
	P_MOs	= 1
	P_Density	= 1
	P_SpinDensity	= 1
	P_Basis	= 1
	P_FragOVLMO_M	= 1
	P_OrbPopMO_M	= 1
	P_OrbCharges_M	= 1

Huge	All the previous plus	
	P_OneElec	= 1
	P_Overlap	= 1
	P_S12	= 1
	P_AtPopMO_M	= 1
	P_OrbPopMO_M	= 1
	P_AtPopMO_L	= 1
	P_EHTDetails	= 1
Debug	print everything	

6.22.2 Mulliken Population Analysis

The Mulliken population analysis [331] is, despite all its known considerable weaknesses, the standard in most quantum chemical programs. It partitions the total density using the assignment of basis functions to given atoms in the molecules and the basis function overlap. If the total charge density is written as $\rho(\vec{r})$ and the total number of electrons is N we have:

$$\int \rho(\vec{r}) \, d\vec{r} = N \tag{6.314}$$

and from the density matrix $\mathbf{P}$ and the basis functions $\{\phi\}$:

$$\rho(\vec{r}) = \sum_{\mu\nu} P_{\mu\nu} \phi_\mu(\vec{r}) \phi_\nu(\vec{r}) \tag{6.315}$$

therefore:

$$\int \rho(\vec{r}) \, d\vec{r} = \sum_{\mu\nu} P_{\mu\nu} \underbrace{\int \phi_\mu(\vec{r}) \phi_\nu(\vec{r}) \, d\vec{r}}_{S_{\mu\nu}} \tag{6.316}$$

$$= \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} \tag{6.317}$$

After assigning each basis function to a given center this can be rewritten:

$$= \sum_A \sum_B \sum_\mu{}^A \sum_\nu{}^B P_{\mu\nu}^{AB} S_{\mu\nu}^{AB} \tag{6.318}$$

$$= \sum_A \sum_\mu{}^A \sum_\nu{}^A P_{\mu\nu}^{AA} S_{\mu\nu}^{AA} + 2 \sum_A \sum_{B<A} \sum_\mu{}^A \sum_\nu{}^B P_{\mu\nu}^{AB} S_{\mu\nu}^{AB} \tag{6.319}$$

Mulliken proposed to divide the second term equally between each pair of atoms involved and define the number of electrons on center A, N_A, as:

$$N_A = \sum_\mu{}^A \sum_\nu{}^A P_{\mu\nu}^{AA} S_{\mu\nu}^{AA} + \sum_{B \neq A} \sum_\mu{}^A \sum_\nu{}^B P_{\mu\nu}^{AB} S_{\mu\nu}^{AB} \tag{6.320}$$

such that $\sum_A N_A = N$. The charge of an atom in the molecule is then:

$$Q_A = Z_A - N_A \tag{6.321}$$

where Z_A is the core charge of atom A. The cross terms between pairs of basis functions centered on different atoms is the overlap charge and is used in ORCA to define the Mulliken bond order:

$$B_{AB} = 2 \sum_\mu{}^A \sum_\nu{}^B P_{\mu\nu}^{AB} S_{\mu\nu}^{AB} \tag{6.322}$$

The Mulliken population analysis is turned on by using:

```
%output
  Print[ P_Mulliken ] 1  # default = on
  end
```

A number of additional options can be specified to control the details of the Mulliken population analysis. By default the Mulliken population analysis is turned on.

```
%output
  Print[ P_AtCharges_M  ] 1   # Print atomic charges
  Print[ P_OrbCharges_M ] 1   # Print orbital charges
  Print[ P_FragCharges_M] 1   # Print fragment charges
  Print[ P_BondOrder_M  ] 1   # Print bond orders
  Print[ P_FragBondOrder_M ] 1# Print fragment b.o.
  Print[ P_ReducedOrbPop_M ] 1# Print reduced orb. Charges
  Print[ P_AtPopMO_M    ] 1   # Print atomic charges in
                              # each MO
  Print[ P_OrbPopMO_M   ] 1   # Print orbital populaiton
                              # for each MO
  Print[ P_ReducedOrbPopMO_M] 1 # Print reduced orbital
                                # pop for each MO
  Print[ P_FragPopMO_M  ] 1     # Print the fragment
                                # population for for each MO
  end
```

These options allow to get very detailed information about the computed wavefunctions and is much more convenient than to look at the MOs directly. A 'reduced orbital population' is a population per angular momentum type. For example the sum of populations of each p_z orbital at a given atom is the reduced orbital population of the p_z function.

6.22.3 Löwdin Population Analysis

The Löwdin analysis [96] is somewhat more straightforward than the Mulliken analysis. In the Löwdin method one changes to a basis where all overlap integrals vanish. This is accomplished via Löwdins symmetric orthogonalization matrix $\mathbf{S}^{-1/2}$. Using this transformation matrix the new basis functions are multicentered but are in a least square sense as similar as possible to the original, strictly localized, atomic basis functions. The similarity of the transformed functions and original functions is explored in the population analysis. The density matrix transforms as:

$$\mathbf{P}^L = \mathbf{S}^{1/2} \mathbf{P} \mathbf{S}^{1/2} \tag{6.323}$$

Then the atomic populations are:

$$N_A = \sum_\mu {}^A P^L_{\mu\mu} \tag{6.324}$$

The bond order is defined from the Wiberg index [332] that was first used in the context of semiempirical methods (that are formulated in the Löwdin basis right from the start):

$$B_{AB} = \sum_\mu {}^A \sum_\nu {}^B \left(P^L_{\mu\nu} \right)^2 \tag{6.325}$$

The output for the Löwdin population analysis (that I personally prefer over the Mulliken analysis) is closely similar. By default the Löwdin population analysis is turned on and provides some more detail than the Mulliken analysis.

```
%output
  Print[ P_Loewdin ] 1  # default = on
  end
```

The flags to regulate the details are almost identical:

```
%output
  Print[ P_AtCharges_L  ] 1   # Print atomic charges
  Print[ P_OrbCharges_L ] 1   # Print orbital charges
  Print[ P_FragCharges_L] 1   # Print fragment charges
  Print[ P_BondOrder_L  ] 1   # Print bond orders
  Print[ P_FragBondOrder_L ] 1# Print fragment b.o.
  Print[ P_ReducedOrbPop_L ] 1# Print reduced orb. Charges
  Print[ P_AtPopMO_L    ] 1   # Print atomic charges in
                              # each MO
  Print[ P_OrbPopMO_L   ] 1   # Print orbital population
                              # for each MO
  Print[ P_ReducedOrbPopMO_L] 1 # Print reduced orbital
                                # pop for each MO
  Print[ P_FragPopMO_L  ] 1    # Print the fragment
                               # population for each MO
  end
```

6.22.4 Mayer Population Analysis

Mayers bonding analysis [333–336] is another creative attempt to define chemically useful indices. The Mayer atomic charge is identical to the Mulliken charge. The Mayer bond order is defined as:

$$B_{AB} = \sum_{\mu}^{A} \sum_{\nu}^{B} (\mathbf{PS})_{\mu\nu} (\mathbf{PS})_{\nu\mu} + (\mathbf{RS})_{\mu\nu} (\mathbf{RS})_{\nu\mu} \tag{6.326}$$

Here $\mathbf{P}$ is the total electron density matrix and $\mathbf{R}$ is the spin density matrix. These Mayer bond orders are very useful. Mayer's total valence for atom A is defined as:

$$V_A = 2N_A - \sum_{\mu}^{A} \sum_{\nu}^{A} (\mathbf{PS})_{\mu\nu} (\mathbf{PS})_{\nu\mu} \tag{6.327}$$

In normal bonding situations and with normal basis sets V_A should be reasonably close to the valence of atom A in a chemical sense (i.e. close to four for a carbon atom). The bonded valence is given by:

$$X_A = V_A - \sum_{B \neq A} B_{AB} \tag{6.328}$$

and finally the free valence (a measure of the ability to form further bonds) is given by:

$$F_A = V_A - X_A \tag{6.329}$$

The Mayer population analysis is turned on by:

```
%output
  Print[ P_Mayer ] 1  # default = on
  end
```

The output is rather simple and short and can not be further controlled. By default the Mayer population analysis is turned on.

6.22.5 Natural Population Analysis

A popular and useful method for population analysis is the natural population analysis due to Weinhold and co-workers. It is implemented in the **gennbo** program which is distributed in older versions via the CCL list and in newer versions via the university of Wisconsin/Madison. Information about the NBO program can be found at http://www.chem.wisc.edu/~nbo5. In order to use it together with ORCA you need a version of the stand-alone executable in some directory that is known via the standard path. Starting with version 3.0.1 ORCA can be used with either NBO5 or NBO6. To interface NBO5: set environment variables NBOVERSION=5 and NBOEXE=/full/name/of/nbo5-executable. For use of NBO6: set environment variables GENEXE=/full/name/of/gennbo-executable and NBOEXE=/full/name/of/nbo6-executable.

ORCA features only two keywords to interface the **gennbo** program: `NPA` and `NBO`. `NPA` simply requests a NPA analysis without potentially large output from the NBO analysis whereas `NBO` invokes the NPA as well as the NBO analysis. The following example illustrates the use:

```
#
# Test the interface to the GENNBO program
#
! RHF SVP NBO

* xyz 0 1
  C     0.000000     0.000000     0.000000
  O     1.200000     0.000000     0.000000
  H    -0.550000     0.952628     0.000000
  H    -0.550000    -0.952628    -0.000000
*
```

This produces the following output:

```
********************************* NBO 5.0 *********************************
         N A T U R A L   A T O M I C   O R B I T A L   A N D
       N A T U R A L   B O N D   O R B I T A L   A N A L Y S I S
***************************************************************************
 (c) Copyright 1996-2002 Board of Regents of the University of Wisconsin System
     on behalf of the Theoretical Chemistry Institute.  All Rights Reserved.

        Cite this program as:

        NBO 5.0.  E. D. Glendening, J. K. Badenhoop, A. E. Reed,
        J. E. Carpenter, J. A. Bohmann, C. M. Morales, and F. Weinhold
        (Theoretical Chemistry Institute, University of Wisconsin,
        Madison, WI, 2001); http://www.chem.wisc.edu/~nbo5

     /NPA   / : Natural Population Analysis
     /NBO   / : Natural Bond Orbital Analysis
     /AONBO / : Print the AO to NBO transformation
     /FILE  / : Set to Test-NPA-H2CO

 Job title:  *** NBO job generated by orca_scf

 Storage needed:      7426 in NPA,       5920 in NBO (    2000000 available)


 NATURAL POPULATIONS:  Natural atomic orbital occupancies

  NAO Atom No lang   Type(AO)   Occupancy
 -------------------------------------------
   1   C  1  s      Cor( 1s)    1.99966
   2   C  1  s      Val( 2s)    1.01124
   3   C  1  s      Ryd( 3s)    0.00705
   4   C  1  px     Val( 2p)    0.81950
   5   C  1  px     Ryd( 3p)    0.01290
   6   C  1  py     Val( 2p)    1.09794
   7   C  1  py     Ryd( 3p)    0.00054
   8   C  1  pz     Val( 2p)    0.65995
   9   C  1  pz     Ryd( 3p)    0.00281
  10   C  1  dxy    Ryd( 3d)    0.00576
```

```
11     C   1   dxz     Ryd( 3d)      0.00375
12     C   1   dyz     Ryd( 3d)      0.00000
13     C   1   dx2y2   Ryd( 3d)      0.00337
14     C   1   dz2     Ryd( 3d)      0.00114

15     O   2   s       Cor( 1s)      1.99976
16     O   2   s       Val( 2s)      1.70418
17     O   2   s       Ryd( 3s)      0.00171
18     O   2   px      Val( 2p)      1.62361
19     O   2   px      Ryd( 3p)      0.00082
20     O   2   py      Val( 2p)      1.91529
21     O   2   py      Ryd( 3p)      0.00382
22     O   2   pz      Val( 2p)      1.32995
23     O   2   pz      Ryd( 3p)      0.00011
24     O   2   dxy     Ryd( 3d)      0.00213
25     O   2   dxz     Ryd( 3d)      0.00340
26     O   2   dyz     Ryd( 3d)      0.00000
27     O   2   dx2y2   Ryd( 3d)      0.00406
28     O   2   dz2     Ryd( 3d)      0.00119

29     H   3   s       Val( 1s)      0.88722
30     H   3   s       Ryd( 2s)      0.00305
31     H   3   px      Ryd( 2p)      0.00030
32     H   3   py      Ryd( 2p)      0.00159
33     H   3   pz      Ryd( 2p)      0.00002

34     H   4   s       Val( 1s)      0.88722
35     H   4   s       Ryd( 2s)      0.00305
36     H   4   px      Ryd( 2p)      0.00030
37     H   4   py      Ryd( 2p)      0.00159
38     H   4   pz      Ryd( 2p)      0.00002


Summary of Natural Population Analysis:


                                 Natural Population
                   Natural    ------------------------------------------------
 Atom No           Charge        Core      Valence     Rydberg     Total
--------------------------------------------------------------------
   C   1          0.37439      1.99966     3.58863     0.03732     5.62561
   O   2         -0.59002      1.99976     6.57303     0.01723     8.59002
   H   3          0.10782      0.00000     0.88722     0.00496     0.89218
   H   4          0.10782      0.00000     0.88722     0.00496     0.89218
====================================================================
* Total *  0.00000            3.99942    11.93611     0.06447    16.00000

                          Natural Population
--------------------------------------------------------------
  Core                      3.99942 ( 99.9854% of    4)
  Valence                  11.93611 ( 99.4676% of   12)
  Natural Minimal Basis    15.93553 ( 99.5970% of   16)
  Natural Rydberg Basis     0.06447 (  0.4030% of   16)
--------------------------------------------------------------


   Atom No        Natural Electron Configuration
-----------------------------------------------------------------------------
```

```
        C  1        [core]2s( 1.01)2p( 2.58)3s( 0.01)3p( 0.02)3d( 0.01)
        O  2        [core]2s( 1.70)2p( 4.87)3d( 0.01)
        H  3               1s( 0.89)
        H  4               1s( 0.89)


   NATURAL BOND ORBITAL ANALYSIS:

                        Occupancies      Lewis Structure    Low   High
                Occ.    -------------------  -----------------  occ   occ
     Cycle   Thresh.   Lewis   Non-Lewis    CR  BD  3C  LP   (L)   (NL)   Dev
     ====================================================================
      1(1)    1.90    15.89605  0.10395      2   4   0   2    0     0    0.04
     --------------------------------------------------------------------


   Structure accepted: No low occupancy Lewis orbitals


     --------------------------------------------------------
       Core                    3.99942 ( 99.985% of   4)
       Valence Lewis          11.89664 ( 99.139% of  12)
      =================        ===========================
       Total Lewis            15.89605 ( 99.350% of  16)
     --------------------------------------------------------
       Valence non-Lewis       0.07823 (  0.489% of  16)
       Rydberg non-Lewis       0.02571 (  0.161% of  16)
      =================        ===========================
       Total non-Lewis         0.10395 (  0.650% of  16)
     --------------------------------------------------------



   NHO DIRECTIONALITY AND BOND BENDING (deviations from line of nuclear centers)

          [Thresholds for printing:  angular deviation  >  1.0 degree]
                                     p- or d-character  > 25.0%
                                     orbital occupancy  >  0.10e

                      Line of Centers      Hybrid 1            Hybrid 2
                      ---------------  -------------------  -------------------
               NBO      Theta   Phi     Theta   Phi   Dev    Theta   Phi   Dev
     ====================================================================
      1. BD ( 1) C 1- O 2   90.0   0.0    0.7   0.0  89.3     5.7  180.0  84.3
      7. LP ( 1) O 2        --     --     90.0  180.0  --     --     --    --
      8. LP ( 2) O 2        --     --     90.0  269.9  --     --     --    --



   The NBO description array
   -------------------------------------------
   NBO   OCC       TYPE        ATOMS
   -------------------------------------------
     0   1.99966 : CORE        0C
     1   1.99976 : CORE        1O
     2   1.99996 : BOND        0C    1O
     3   1.99977 : BOND        0C    1O
     4   1.99557 : BOND        0C    2H
```

```
 5   1.99557 : BOND        0C    3H
 6   1.98820 : LONE-PAIR   1O
 7   1.91756 : LONE-PAIR   1O
 8   0.00000 : ANTIBOND    0C    1O
 9   0.00003 : ANTIBOND    0C    1O
10   0.03910 : ANTIBOND    0C    2H
11   0.03910 : ANTIBOND    0C    3H
12   0.01025 : RYDBERG     0C
13   0.00517 : RYDBERG     0C
14   0.00001 : RYDBERG     0C
15   0.00000 : RYDBERG     0C
16   0.00000 : RYDBERG     0C
17   0.00000 : RYDBERG     0C
18   0.00000 : RYDBERG     0C
19   0.00001 : RYDBERG     0C
20   0.00000 : RYDBERG     0C
21   0.00368 : RYDBERG     1O
22   0.00019 : RYDBERG     1O
23   0.00000 : RYDBERG     1O
24   0.00000 : RYDBERG     1O
25   0.00000 : RYDBERG     1O
26   0.00000 : RYDBERG     1O
27   0.00000 : RYDBERG     1O
28   0.00001 : RYDBERG     1O
29   0.00000 : RYDBERG     1O
30   0.00314 : RYDBERG     2H
31   0.00003 : RYDBERG     2H
32   0.00002 : RYDBERG     2H
33   0.00001 : RYDBERG     2H
34   0.00314 : RYDBERG     3H
35   0.00003 : RYDBERG     3H
36   0.00002 : RYDBERG     3H
37   0.00001 : RYDBERG     3H
```
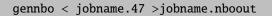
Thus, in this example the NBO analysis shows that a single Lewis structure is dominant with single bonds between C and H, a double bond between C and O and two lone pairs at the oxiygen – just as ordinary chemical arguments would imply. In addition, the program produces the four corresponding valence antibonds (as expected). The remaining components of the basis set span the "Rydberg" space and lead to semilocalized, orthogonal orbitals that are assigned to single atoms (Note the nomenclature: BD = bond, BD* = antibond, LP = lone pair, CR = core orbital, RY= Rydberg orbital). The NPA analysis shows a patially negative oxygen and partially positive carbon and hydrogen atoms – all matching the expectations.

NOTE:

- ORCA reorders the NBOs compared to the **gennbo** program. The ORCA ordering is: core orbitals – bond orbitals – lone pairs – antibonds – Rydberg orbitals.

- ORCA suppresses a lot of NBO related output. In order to get the full **gennbo** output you have to run the program in the standalone mode using the file that ORCA leaves on disk for you.

- ORCA leaves a `jobname.nbo` file on disk – This file is a standard GBW file and can be used to visualize the NBOs. Note that in this case the ORCA ordering of NBO's is used.

The **gennbo** program has many additional features and analysis tools. Since these are subject to changes or improvements over time I have decided to provide only a superficial interface to the basic features and leave a file on

disk that can be used to access the more fancy features of the **gennbo** program manually. This file will be called `jobname.47.` This is an ascii file that can be edited with a text editor. Add or remove the keywords in the `$NBO` block as needed and call the gennbo program like

```
gennbo < jobname.47 >jobname.nboout
```


(a) π (b) $\pi - LP$ (c) π^*

(d) σ (e) $\sigma - LP$ (f) σ^*
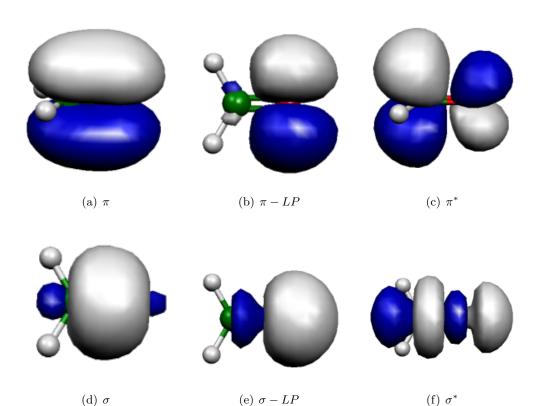
Figure 6.29: Six NBOs of the H_2CO molecule. Shown are the occupied bonding π and σ orbitals (left) for C and O, the two oxygen lone-pairs (middle) and the two π and σ antibonding orbitals (right).

This will produce the desired output in the file `jobname.nboout`. The `jobname.47` file looks like:

```
$NBO                 !  add keywords to this block
  FILE=Test-NPA-H2CO
  NPA
  NBO
$END
! the whole rest is wavefunction, basis, geometry info.
! generated by ORCA. It is similar in content to the GBW
! file. Thus you could also use the .47 file to interface
! your own programs.
$GENNBO
  natoms       4
```

```
    nbas        38
    bodm
    upper
    bohr
  $END

  $COORD
   *** NBO job generated by orca_scf
       6    6     0.000000        0.000000        0.000000
       8    8     2.267715        0.000000        0.000000
       1    1    -1.039369        1.800241        0.000000
       1    1    -1.039369       -1.800241        0.000000
  $END

  $BASIS
    etc. etc. (refer to the NBO manual for a
              detailed description)
```

If you have no need for this (rather large) file, then you have to delete it manually!

6.22.6 Density of the Effectively Unpaired Electrons

Investigating open-shell molecules, it is common practice to consult the spin-population analysis i.e. the Mulliken spin-population using the spin-density matrix. The trace of spin-density corresponds to difference in the number of alpha and beta electrons. For an open-shell system in high-spin configuration, the trace matches the number of unpaired electrons. However, the spin-density is not able to discriminate anti-ferromagnetic coupled electrons from electron pairs. Thus, open shell singlets appear to have no unpaired electrons. There are many approaches in the literature to define a density matrix for unpaired electrons that correctly describe low-spin configurations. For a CASSCF-type wavefunction, two common effective unpaired electron densities (eff. UED) are available in ORCA. Namely, Taksuka's [337] and Head-Gordon's [338] formulations. If the one-particle density matrix is denoted with γ, its eigenvalues $\mathbf{n}$ and vectors $\mathbf{U}$ satisfy

$$\gamma = \mathbf{U}\mathbf{n}\mathbf{U}^{\dagger}. \tag{6.330}$$

The eff. unpaired electron density matrix $\mathbf{UED}$ can be expressed as a function of the natural orbital occupation

$$\mathbf{UED} = \mathbf{U}f(\mathbf{n})\mathbf{U}^{\dagger}. \tag{6.331}$$

In case of Takatsuka's definition we have

$$f(\mathbf{n}) = 2\mathbf{n} - \mathbf{nn} \tag{6.332}$$

and for Head-Gordon the function is modified towards

$$f(\mathbf{n}) = 1 - |1 - \mathbf{n}|. \tag{6.333}$$

As an example, let us recompute the example in [338] - two oxygen atoms at of 10Å distance in the triplet state. The active space contains all the 2p orbitals and electrons. Since the ground-state is 9-fold degenerate, we must take into account all 9 roots.

```
    !svp
%maxcore 2000
%casscf nel 8
        norb 6
        mult 3
        nroots 9
        actorbs locorbs # localized active orbitals
end
%output
  print[p_unpaired_t] 1   # Takatsuka   UED
  print[p_unpaired_hg] 1  # Head-Gordon UED
end

*xyz 0 3
 O 0 0 0
 O 0 0 10.0
*
```

According to the CASSCF wavefunction, there should be 4 unpaired electrons coupled to a triplet state.

```
    ------------------------------------------------
    CAS-SCF STATES FOR BLOCK  1 MULT= 3 NROOTS= 9
    ------------------------------------------------

 ROOT   0:  E=     -149.4274934720 Eh
       0.49867 [    31]: 121211
       0.48600 [    50]: 211112
       0.01400 [    51]: 211121
 ROOT   1:  E=     -149.4274934720 Eh   0.000 eV
       0.49867 [    22]: 112211
       0.48599 [    51]: 211121
       0.01400 [    50]: 211112


       ...
```

As expected, the spin-density is not sensitive for the two-antiferromagnetically coupled electrons

```
    -------------------
    SPIN-DENSITY MATRIX
    -------------------
```

```
              0          1          2          3          4          5
   0       0.333333   0.000000  -0.000000  -0.000000  -0.000000   0.000000
   1       0.000000   0.333333  -0.000000   0.000000   0.000000   0.000000
   2      -0.000000  -0.000000   0.333333  -0.000000   0.000000   0.000000
   3      -0.000000   0.000000  -0.000000   0.333333  -0.000000   0.000000
   4      -0.000000   0.000000   0.000000  -0.000000   0.333333   0.000000
   5       0.000000   0.000000   0.000000   0.000000   0.000000   0.333333
Trace of the spin density:  2.000000
```

The eff. unpaired density matrix by Head-Gordon on the other hand is able to reproduce the correct picture here.

```
     ------------------------------------------------------------------------------
            Head-Gordon eff. unpaired electron density
     ------------------------------------------------------------------------------
              0          1          2          3          4          5
   0       0.666667   0.000000  -0.000000  -0.000000   0.000000  -0.000000
   1       0.000000   0.666667   0.000000   0.000000   0.000000   0.000000
   2      -0.000000   0.000000   0.666667  -0.000000   0.000000   0.000000
   3      -0.000000   0.000000  -0.000000   0.666667  -0.000000   0.000000
   4       0.000000   0.000000   0.000000  -0.000000   0.666667  -0.000000
   5      -0.000000   0.000000   0.000000   0.000000  -0.000000   0.666667


Number of eff. unpaired electrons  ... 4.000


   ******************************
   * MULLIKEN POPULATION ANALYSIS *
   *          (EFF. UED)          *
   ******************************


   ------------------------------------------
   MULLIKEN ATOMIC EFF. UED
   ------------------------------------------
   0 O :    2.000000
   1 O :    2.000000
   Sum of atomic contributions:    4.0000000


     --------------------------------------------------
          MULLIKEN REDUCED ORBITAL EFF. UED
     --------------------------------------------------
SPIN
0 O s      :     0.000000  s :     0.000000
pz      :      0.666667  p :     2.000000
px      :      0.666667
py      :      0.666667
```

```
    dz2      :       0.000000   d :      0.000000
    dxz      :       0.000000
    dyz      :       0.000000
    dx2y2    :       0.000000
    dxy      :       0.000000
    1 0 s        :       0.000000   s :       0.000000
    pz       :       0.666667   p :       2.000000
    px       :       0.666667
    py       :       0.666667
    dz2      :       0.000000   d :      0.000000
    dxz      :       0.000000
    dyz      :       0.000000
    dx2y2    :       0.000000
    dxy      :       0.000000
```

The full printing will contain both eff. unpaired electron densities their Mulliken and Loewdin analysis. Extensions for single-reference UHF/UKS wavefunction exist and might find their way into ORCA in the future.

6.22.7 Local-Spin Analysis

There exists a vast literature on the computation of local spin-square value for atomic centers. [339] Starting from localized orbitals, these values can be used to estimate exchanges coupling of magnetic centers. [340] There are ambiguities in the formulation that basically arise from:

- The lack of true spin-eigenfunctions

- Having approximate two-body density matrix

- Only poorly localized orbitals

Local spin analysis is available for CASSCF and DMRG wavefunctions. Both methods are properly spin-adapted and the two-body density matrix is readily available. So, the remaining source of ambiguity is the handling of poorly localized orbitals. In our implementation, orbitals are split in a Mulliken type fashion. The working equations are presented in the work of D.R. Alcoba et. al. [341]. The total spin-square expectation value is expressed in terms of the one- and two-body density matrices γ and Γ

$$\left\langle S^2 \right\rangle = \frac{3}{4} \sum_p^{Act} \gamma_p^p - \sum_{pq}^{Act} 2\Gamma_{pq}^{pq} - \Gamma_{qq}^{pp}. \tag{6.334}$$

The contribution to the total spin can then be divided into a sum of atomic contributions and a spin-correlation matrix (di-atomic)

$$\left\langle S^2 \right\rangle = \sum_A \left\langle S_A^2 \right\rangle + \frac{1}{2} \sum_{A \neq B} \left\langle S_{AB}^2 \right\rangle, \tag{6.335}$$

where A and B are atomic centers. Multiplying equation 6.334 with Mulliken derived atom-weights for the molecular orbitals, the local spin and spin-correlation matrices in 6.335 are readily indentified.

The atomic centers of interest are set in the output-block of ORCA, where the numbering corresponds to the atom number in the coordinate block i.e.

```
!svp  noautostart
%casscf nel 2
norb 2
mult 1
actorbs locorbs # localizes the active orbitals
end

%output
localspin 0,1
end

*xyz 0 1
H 0 0 0
H 0 0 10.0
*
```

In this example we want to compute the two singlet-coupled hydrogen atoms. The keyword `actorbs locorbs` ensures localized orbitals at the end of the CASSCF procedure. At the beginning of the population analysis, we find the desired local spin-square values.

```
--------------------
Local Spin Analysis
--------------------
Input particle densities        ... h2.casp.tmp
Active orbitals                 ... 0 - 1

  Weighting matrix
            0           1
0       1.000000   0.000000
1       0.000000   1.000000

Atom number 0 (H ): <S**2> = 0.750000
Atom number 1 (H ): <S**2> = 0.750000

  Spin Correlation Matrix
            0           1
0        0.000000  -0.750000
1       -0.750000   0.000000
 Total recombined <S**2> = -0.000000
```

Since the atoms are well separated, we have perfectly localized orbitals. Consequently, the weighting matrix reduces to values of one and zero. We obtain local spin-square values of $\frac{3}{4}$ corresponding two the exact values for single electrons.

For less localized orbitals the quality of the anlysis is less rigorous.

6.22.8 UNO Orbital Printing

The analysis of UNO's can be controlled similarly. The flags together with their default values are shown below:

```
%output
  Print[ P_UNO_OccNum      ] = 1; # Occupation numbers
  Print[ P_UNO_AtPopMO_M  ] = 0; # Mulliken atom pop.
                                  # per UNO
  Print[ P_UNO_OrbPopMO_M] = 0;  # Mulliken orbital pop.
                                  # per UNO
  Print[ P_UNO_ReducedOrbPopMO_M] = 0;
                                  # Mulliken reduced orbital
                                  # pop. per UNO
  Print[ P_UNO_AtPopMO_L  ] = 0; # Loewdin atom pop.
                                  # per UNO
  Print[ P_UNO_OrbPopMO_L] = 0;  # Loewdin orbital pop.
                                  # per UNO
  Print[ P_UNO_ReducedOrbPopMO_L] = 0;
                                  # Loewdin reduced orbital
                                  # pop. per UNO
end
```

6.23 Orbital and Density Plots

There are two types of graphics output possible in ORCA - two dimensional contour plots and three dimensional surface plots. The quantities that can be plotted are the molecular orbitals, the total electron density or the total spin density. The graphics is controlled through the block %plots.

6.23.1 Contour Plots

The contour plots are controlled via the following variables

```
%plots
  #*** the vectors defining the cut plane
  v1 0, 0, 0  # pointer to the origin
  v2 1, 0, 0  # first direction
  v3 0, 1, 0  # second direction
  #*** alternative to defining vectors. Use atom coordinates
  at1 0 # first atom defining v1
  at2 2 # second atom defining v2
```

```
    at3 4 # third atom defining v3
    #*** resolution of the contour
    dim1  45   # resolution in v2-direction
    dim2  45   # resolution in v3-direction
    #*** minimum and maximum values along v2 and v3
    min1  -7.0 # min value along v2 in bohr
    max1   7.0 # min value along v2 in bohr
    min2  -7.0 # min value along v3 in bohr
    max2   7.0 # max value along v3 in bohr
    #***
    UseCol   true    # Use color in the plot (blue=positive,
                     # red=negative)
    Skeleton true    # Draw Skeleton of the molecule of those
                     # atoms that are in or close to the cut
                     # plane
    Atoms    true    # Draw the atoms that are in the plane as
                     # circles
    NCont    200     # Number of contour levels.
    ICont    0       # Draw NCont equally space contours
             1       # Start with 1/NCont and the double the
                     # value for each additional contour
    #*** the format of the output file
    Format   Origin  # straight ascii files
             HPGL    # plotter language files
    #*** the quantities to plot
    MO("MyOrbital-15xy.plt",15,0);  # orbital to plot
    v3= 0, 0, 1                     # change cut plane
    MO("MyOrbital-16xz.plt",16,0);  # orbital to plot
    ElDens("MyElDens.plt");         # Electron density
    SpinDens("MySpinDens");         # Spin density
    end
```

The input was:

```
v1 = 0, 0, 0;
v2 = 1, 0, 0;
v3 = 0, 1, 0;
min1= -8; max1= 8;
min2= -8; max2= 8;
dim1= 50; dim2=50;
Format = HPGL;
NCont = 200;
Icont = 1;
Skeleton= true;
Atoms = true;
MO("Test-DFT-H2CO+-MO7xy.plt",7,1);
```
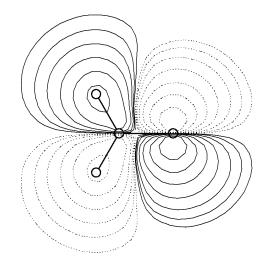
Figure 6.30: Contour plot of the lowest unoc-
cupied spin down orbital of the
H_2CO^+ cation radical in the x,y
plane.

NOTE:

- The command `MO(''MyOrbital-15xy.plt'',15,0);` is to be interpreted as follows: `MO` means that a MO is to be plotted. "`MyOrbital-15xy.plt`" is the file to be created. `15` is the number of the MO to be drawn (remember: counting starts at orbital 0!) and `0` is the operator the orbital belongs to. For a RHF (or RKS) calculation there is only one operator which has number 0. For a UHF (or UKS) calculation there are two operators - the spin-up orbitals belong to operator 0 and the spin-down orbitals belong to operator 1. For ROHF calculations there may be many operators but at the end all orbitals will be collected in one set of vectors. Thus the operator is always =0 in ROHF.

- The `ELDENS` (plot of the total electron density) and `SPINDENS` (plot of the total spin density) commands work analogous to the `MO` with the obvious difference that there is no MO or operator to be defined.

- The `UNO` option plots natural orbitals of the UHF wavefunction (if they are available). No operator can be given for this command because there is only one set of UHF-NOs.

- If the program cannot find the plot module ('Bad command or filename') try to use `ProgPlot=''orca_plot.exe''` in the `[METHOD]` block or point to the explicit path.

- The defining vectors `v2` and `v3` are required to be orthonormal. The program will use a Schmidt orthonormalization of `v3` with respect to `v2` to ensure orthonormality. If you don't like this make sure that the input vectors are already orthogonal.

- `at1`, `at2` and `at3` can be used instead of `v1`, `v2` and `v3`. In this case say `v1` is taken as the coordinates of atom `at1`. Mixed definitions where say `v2` is explicitly given and say `v3` is defined through `at3` are possible. A value of -1 for `at1`, `at2` and `at3` signals that `at1`, `at2` and `at3` are not to be used. This type of definition may sometimes be more convenient.

- Variables can be assigned several times. The 'actual' value a variable has is stored together with the command to generate a plot (`MO`, `ELDENS` or `SPINDENS`). Thus after each plot command the format or orientation of the plot can be changed for the next one.

- The `Origin` format produces a straightforward ASCII file with x,y and z values that can be read into your favorite contour plot program or you could write a small program that reads such files and converts them to whatever format is more appropriate for you.

- I usually use Word for Windows to open the `HPGL` files which appears to work fine. Double clicking on the graphics will allow modification of linewidth etc. For some reason that is not clear to me some graphics programs do not like the HPGL code that is produced by ORCA. If you are an HPGL expert and you have a suggestion - let me know.

6.23.2 Surface Plots

6.23.2.1 General Points

Surface plots can, for example, be created through an interface to Leif Laaksonen's **_gOpenMol_** program. This program can be obtained free of charge over the internet (http://laaksonen.csc.fi/gopenmol). It runs on a wide variety of platforms, is easy to use, produces high quality graphics and is easy to interface[18] - thank you Leif for making this program available!

The relevant `[PLOTS]` section looks like this:

```
%output
  XYZFile   true
  end


%plots
  dim1  45    # resolution in x-direction
  dim2  45    # resolution in y-direction
  dim3  45    # resolution in z-direction
  min1  -7.0 # x-min value in bohr
  max1   7.0 # x-min value in bohr
  min2  -7.0 # y-min value in bohr
  max2   7.0 # y-max value in bohr
  min3  -7.0 # z-min value in bohr
  max3   7.0 # z-max value in bohr
  Format  gOpenMol_bin        # binary *.plt file
          gOpenMol_ascii      # ascii *.plt file
          Gaussian_Cube       # Gaussian-cube format
                              # (an ASCII file)
  MO("MyOrbital-15.plt",15,0);  # orbital to plot
  MO("MyOrbital-16.plt",16,0);  # orbital to plot
  UNO("MyUNO-48.plt",48);       # UHF-NO to plot
  ElDens("MyElDens.plt");       # Electron density
  SpinDens("MySpinDens.plt");   # Spin density
  end
```

---

[18] There were some reports of problems with the program on Windows platforms. Apparently it is better to choose the display settings as 'true color 32 bit' rather than 'high 16 bit'. Thanks to Thomas Brunold!
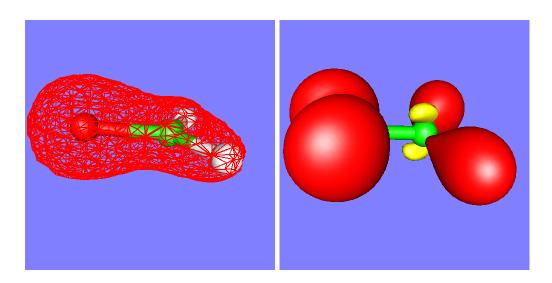
Figure 6.31: The total electron density (shown as a mesh) and the spin density (shown with solid contours) of the H_2CO^+ cation radical as calculated by the RI-BP/VDZP method. Note the small negative spin density at the carbon atom. The spin density was calculated at 120x120x120 resolution which takes much longer than the DFT calculation itself.

NOTE:

- it is admittedly inconvenient to manually input the dimension of the cube that is used for plotting. If you do nothing such that `min1 = max1 = min2 = max2 = min3 = max3=0` then the program will try to be smart and figure out a good cube size by itself. It will look at the minimum and maximum values of the coordinates and then add 7 bohrs to each dimension in the hope to properly catch all wavefunction tails.

Sometimes you will want to produce orbital plots after you looked at the output file and decided which orbitals you are interested in. In this case you can also run the `orca_plot` program in a crude interactive form by invoking it as:

```
orca_plot  MyGBWFile.gbw  -i
```

This will only provide you with a small subset of the capabilities of this program but may already be enough to produce the plots you want to look at. Note that for the name of the GBW-file you may as well input files that result from natural orbitals (normally `*.uno`), corresponding orbitals (normally `*.uco`) or localized orbitals (normally `*.loc`).

6.23.2.2 Interface to gOpenMol

Here is a short summary of how to produce these plots with gOpenMol:

- First of all the molecular geometry must be save by choosing `XYZFile=true` in the `[OUTPUT]` block. This will produce a straightforward ascii file containing the molecular geometry (or simply `!  XYZFile`).

- After having produced the plot files start **gOpenMol** and choose **File-Import-Coords**. In the dialog choose the *XYZ* format and select the file. Then press **apply** and **dismiss**. The molecule should now be displayed in the graphics window.
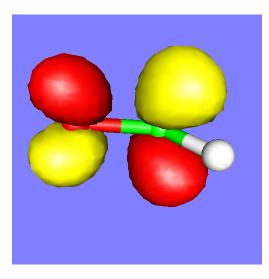
Figure 6.32: The π^* orbital of H_2CO as calculated by the RI-BP/VDZP method.

- You can change the appearance by choosing **View-Atom type**.

- The color of the background can be changed with **Colour-Background**.

- After having done all this choose **Plot-Contour** and select the **Browse** button to select the appropriate file. Then press **Import File** to read it in. NOTE: you can only directly read file that were produced in gOpenMol_bin format. If you have chosen gOpenMol_ascii you must first use the **gOpenMol** file conversion utility under **Run-Pltfile (conversion)** to produce the binary plt file.

- After having read the plt file choose the appropriate isocontour value (one for the positive and one for the negative lobes of an orbital) and select suitable colors via **Colour(n)** to the right of the isocontour value. The **Details** button allows you to choose between solid and mesh representation and other things.

- Once the plot looks the way you like, use **File-Hardcopy** to produce a publication quality postscript or bitmap picture that can be imported into any word processing or graphics software.

6.23.2.3 Interface to Molekel

The Molekel program by Stefan Portmann (http://www.cscs.ch/molekel/) is another beautiful and easy-to-use graphics tool that is recommended in combination with ORCA. You may even find it a little easier to use than gOpenMol but this may be a matter of personal taste. In order to produce plots with Molekel follow the following procedure:

- Produce Gaussian-Cube files (and optionally also an XYZ file) with ORCA as described above.

- Start Molekel and use the right mouse button to obtain the **Load** menu.

- Choose the format **xyz** to load your coordinates

- Use the right mouse button again to select the **Surface** menu

- Choose the format "Gaussian Cube" and click **Load Surface**

- Click on **Both Signs** if you visualize an orbital or spin density

- Select a suitable contour value in the **Cutoff** box.

- Click on **Create Surface**. That's it!

- In the **Color** menu (also available via the right mouse button) you can adjust the colors and in the main menu the display options for your molecule. Default settings are in a startup file that you can modify to suit your taste. More details are in the Molekel manual – check it out; it can do many other useful things for you too!

6.24 Utility Programs

6.24.1 orca_mapspc

This utility program is used to turn calculated spectra into a format that can be plotted with standard graphics programs. The usage is simple (for output examples see for example sections 6.14.2, 5.9.3.1 and 5.9.3.2):

```
orca_mapspc  file spectrum options

  file    = name of an ORCA output file
            name of an ORCA hessian file (for IR and Raman)


  spectrum= abs    - Absorption spectra
            cd     - CD spectra
            ir     - IR spectra
            raman  - Raman spectra


  options   -x0value: Start of the x-axis for the plot
            -x1value: End of the x-axis for the plot
            -wvalue : Full-width at half-maximum height in
                      cm**-1 for each transition
            -nvalue : Number of points to be used
```

NOTE:

- The input to this program can either be a normal output file from an ORCA calculation or a ORCA .hess file if ir or Raman spectra are desired

- The lineshape is always assumed to be a gaussian

- There will be two output files:

    - `Input-file.spc.dat` (spc=abs or cd or ir or raman): This file contains the data to be plotted

    - `Input-file.spc.stk`: This file contains the individual transitions (wavenumber and intensity)

- The absorption plot has five columns: The first is the wavenumber in reciprocal centimeters, the second the total intensity and the third to fifth are the individual polarizations (i.e. assuming that the electric vector of the incoming beam is parallel to either the input x-, or y- or z-axis respectively). The last three columns are useful for interpreting polarized single crystal spectra.

6.24.2 orca_chelpg

This program calculates CHELPG atomic charges according to Breneman and Wiberg [342]. The atomic charges are fitted to reproduce the electrostatic potential on a regular grid around the molecule, while constraining the sum of all atomic charges to the molecule's total charge.

The program works with default values in the following way:

```
orca_chelpg   MyJob.gbw
```

First Argument : The gbw file containing the correct geometry and basis set.

The program uses three adjustable parameters, which can also be set in a separate chelpg input block

```
%chelpg
   GRID 0.3 # Spacing of the regular grid in Ang
   RMAX 2.8 # Maximum distance of all atoms to any
   #  gridpoint in Ang
VDWRADII COSMO # VDW Radii of Atoms, COSMO default
BW # Breneman, Wiberg radii
end
```

In this case the ORCA automatically calculates the CHELPG charges at the end of the calculation. Automatic calculation of CHELPG charges using the default values can also be achieved by specifying

```
! CHELPG
```

In the simple input section. By default the program uses the COSMO VDW radii for the exclusion of gridpoints near the nuclei, as these are defined for all atoms. The BW radii are similar, but only defined for very few atom types.

6.24.3 orca_pltvib

This program is used in conjunction with gOpenMol (or xmol) to produce animations or plots of vibrational modes following a frequency run. The usage is again simple and described in section 5.9.3.5 together with a short description of how to produce these plots in gOpenMol.

6.24.4 orca_vib

This is a small "standalone" program to perform vibrational analysis. The idea is that the user has some control over things like the atomic masses that enter the prediction of vibrational frequencies but are independent of the electronic structure calculation as such.

The program takes a ".hess" file as input and produces essentially the same output as follows the frequency calculation. The point is that the ".hess" is a user-editable textfile that can be manually changed to achieve isotope shift predictions and the like. The usage together with an example is described in section 5.9.3.6. If you pipe the output from the screen into a textfile you should also be able to use orca_mapspc to plot the modified IR, Raman and NRVS spectra.

6.24.5 orca_loc

Localization of the orbitals can be requested at the end of any ORCA calculation. Details are set in the %loc block.

```
LocMet PM # Localization method e.g. Pipek-Mezey
 FB # Foster-Boys
 PrintLevel 2 # Amount of printing
 MaxIter 64 #  Max number of iterations
 Tol 1e-6 # Convergence thresh
 T_Bond 0.85 # Thresh that classifies orbitals in bond-like at the printing
 T_Strong 0.9 # Threhs that classifies orbitals into strongly-localizud at the printing
 OCC true # Localize the occupied space
 VIRT false # Localize the virtual space
end
```

Setting the flags OCC/VIRT to true will request a localization of the subspace. If both flags are set, two consecutive localization are performed. The localized orbitals are stored in the form of a standard GBW file named *.loc*. Keep in mind that the localization of the occupied orbitals might change the total energy depending on what type of calculation you want to perform thereafter. For RHF and UHF there shouldnt be any problems, but for CASSCF the keyword OCC is not sufficient. **CASSCF is not invariant to rotation of *all* the occupied orbitals.**
Calling the **orca_loc** program directly from shell, a localization with a customized orbital ranges is possible. The orca_loc program requires an input of its own. The input is a texfile containing the necessary parameters. If no input is specified, **orca_loc** returns a help-file with a description of the necessary input-parameters. You need to specifiy in/output gbw-files, along with orbital ranges and the localization method to be used. A source of confusion is the operator line op (alpha = 0 or beta = 1). For CASSCF, this should be set to zero in example

```
InitialOrbitals.gbw      # input orbitals
LocalizedOrbitals.gbw # output orbitals
0  # operator:alpha here, 1 for beta
10 # orbital window: first orbital to be localized e.g. first active
15    # orbital window: last orbital to be localized e.g. last active
120  # maxiter
1e-6  # localization thresh
0.95  # printing thresh to call an orbital strongly localized
0.85  # printing thresh to call an orbital bond-like
1 # localization method: 1=Pipek-Mezey  2=Foster-Boys
```

If the input file is called myloc.inp, running "orca_loc myloc.inp" will produce the LocalizedOrbitals.gbw file containing the localized orbitals.

6.24.6 orca_plot

The use of this program is described more fully in section 6.23. It is used to create three dimensional graphics data for visualization. It is also possible to run this program interactively:

```
orca_plot gbwfile -i
```

You will then get a simple, self-explaining menu that will allow you to generate `.plt` files directly from the `.gbw` files without restarting or running a new job.

It is possible to use the module to create difference densities between the ground and excited states from CIS or TD-DFT calculations. This is implemented as an extra interactive menu point that is (hopefully) self explaning.

6.24.7 orca_2mkl: Old Molekel as well as Molden inputs

This little utility program can be used to convert gbw files into mkl files which are of ASCII format. This is useful since molekel can read these files and use them for plotting and the like. The contents of the mkl file is roughly the same as the gbw file (except for the internal flags of ORCA) but this is an ASCII file which can also be read for example by your own programs. It would therefore be a good point for developing an interface. It is likely that this functionality will be further expanded in the future.

```
orca_2mkl BaseName
      (will produce BaseName.mkl from BaseName.gbw)
orca_2mkl BaseName -molden
      writes a file in molden format
orca_2mkl BaseName -mkl
      writes a file in MKL format
```

6.24.8 orca_2aim

This utility program converts a `gbw` file into a `wfn` file that can be used for topological analysis of the electron density by other programs. Currently this works only for closed-shell wave functions. The usage is very simple – just type **AIM** in the simple input line or use

```
orca_2aim BaseName
        (will produce BaseName.wfn from BaseName.gbw)
```

6.24.9 orca_vpot

This program calculates the electrostatic potential at a given set of user defined points. It takes four arguments:

```
orca_vpot  MyJob.gbw  MyJob.scfp.tmp MyJob.vpot.xyz MyJob.vpot.out
```

First : The gbw file containing the correct geometry and basis set

Second : The desired density matrix in this basis (perhaps use the KeepDens keyword)

Third : an ASCII file with the target positions, e.g.

```
6(number_of_points)
  5.0 0.0  0.0 (XYZ coordinates)
 -5.0 0.0  0.0
  0.0 5.0  0.0
  0.0-5.0  0.0
  0.0 0.0  5.0
  0.0 0.0 -5.0
```

Fourth : The target file which will then contain the electrostatic potential. E.g.

```
6 (number of points)
 VX1  VY1  VZ1     (potential for first point)
 VX2  VY2  VZ2     (potential for second point)
  Etc.
```

It should be straightforward for you to read this file and use the potential for whatever purpose.

6.24.10 orca_euler

This utility program is used to calculate the relative orientation between calculated hyperfine coupling (HFC)/nuclear quadrupole coupling (NQC) tensors and a reference tensor (the calculated molecular g-/D-tensor). The orca_euler program is run by default in an ORCA job after the calculation of HFCs or NQCs, if g- or D-tensor are also calculated in the same job. The utility program can also be run as a stand-alone program. In this case the .prop file of a previous NQC/HFC- and D- or g-tensor calculation must be available.

The orientation between the tensors is calculated in terms of a 3x3 rotational matrix R. This is parametrized by the three so-called Euler angles α, β and γ. These angles define the relative orientation between two tensors A and B by three successively applied rotations around different axes in order to align A with B. In the commonly used z-y-z convention these three rotations are:

- Rotate $\mathbf{A}_{xyz}$ counterclockwise around its z axis by α to give $\mathbf{A}_{x'y'z'}$.

- Rotate $\mathbf{A}_{x'y'z'}$ counterclockwise around its y' axis by β to give $\mathbf{A}_{x''y''z''}$.

- Rotate $\mathbf{A}_{x''y''z''}$ counterclockwise around its z" axis by γ to align with B.

```
orca_euler prop-file options

  file    = name of an ORCA .prop file

  options
   -refg/-refD:  Reference tensor (g-tensor or D-tensor, default is -refg)
   -conv zyz/-conv zxz:  Euler rotation convention (default is zyz)
   -order:  Ordering of the reference tensor (x, y, z) with respect to
   ORCA output (min, mid, max)
   -plotA: plot the HFC-tensors
```

```
    -plotQ: plot the NQC-tensors
    -detail: print detailed information
```

NOTE:

- By default the D-tensor is used as reference tensor only if S $>\frac{1}{2}$ and if |D|>0.3 cm^{-1}; in all other cases the g-tensor is used as reference tensor. The user can manually select the reference tensor – if the information is available in the prop-file – by using –**refg** or -**refD**.

- By default the Euler rotation in the z-y-z convention is used. The z-x-z convention can be selected manually by using the option –**conv zxz**.

- By default the axes of the g- or D-tensor are assigned depending on their magnitude. $g_{min} \to g_x$, $g_{mid} \to g_y$, $g_{max} \to g_z$ (similarly for D). This ordering can be modified manually when running the standalone program as shown in the following examples:

$$
\begin{aligned}
\text{-order 3 2 1:} \quad &\text{min} \to \text{z} \\
&\text{mid} \to \text{y} \\
&\text{max} \to \text{x} \\
\text{-order 1 -2 3:} \quad &\text{min} \to \text{x} \\
&\text{mid} \to \text{y (flipped in the orientation)} \\
&\text{max} \to \text{z}
\end{aligned}
$$

- The nuclear hyperfine and quadrupole coupling tensors can be plotted (in the xyz-file format) by the orca_euler program using –**plotA** or –**plotQ**. The HFC tensor for atom 3 (counting starts at zero) is e.g. stored in the file **prop-file.3.A.xyz**, the respective NQC tensor is stored in **prop-file.3.Q.xyz**. In these xyz files the position of four atoms (He, Ne, Ar, Kr) is given. The x-, y- and z-direction of the tensor are in the direction of the vectors between He-Ne, He-Ar and He-Kr.

- The actual definition of the used rotation matrix and more information on the relative orientation can be printed by using the option –**detail**.

7 Some Tips and Tricks

7.1 Cost versus Accuracy

A difficult but important subject in electronic structure theory is to balance the price/accuracy ratio of the calculations. This ratio is governed by: (a) the method used, (b) the basis set used and (c) the cutoffs and tolerances used. There are certainly differing opinions among scientists and I merely quote a few general, subjective points:

- Calculations with minimal basis sets are always unreliable and are only good for explorations. This is also true for small split-valence basis sets like 3-21G, 3-21GSP and perhaps also 4-22G. These basis sets are significantly more reliable than their minimal basis counterparts but are not capable of delivering quantitatively reliable results. They may, however, be the only choice if very large molecules are targeted.

- In our own research we almost exclusively use the basis sets of the kralsruhe group. They have been updated to the 'def2' set that is more consistent than the older basis sets.

- Def2-SV(P) is the smallest and computationally efficient split-valence basis set and is largely identical to the old SV(P), except for the transition metals which have more consistent polarization sets.

- Def2-TZVP is substantially different from the old TZVP. It has been realized that if one invests into an accurate triple-zeta description of the valence region it makes limited sense to only employ a single polarization function. The accuracy is then limited by the polarization set and is not much better than what one gets from SV(P). Hence, def2-TZVP contains a single p-set for hydrogens but is otherwise very similar to the old TZVPP basis set, e.g. it contains 2d1f polarization for main group elements and much more extensive polarization sets for transition metals. The highest polarization function (f for main group) does add substantially to the computational effort. Hence, we often use def2-TZVP(-f). Together with RI or RIJCOSX this, or the more complete def2-TZVP, is still computationally economic enough for most studies.

- Def2-TZVPP is a fully consistent triple-zeta basis set that provides excellent accuracy for SCF calculations (HF and DFT) and is still pretty good for correlated calculations. It is a good basis set to provide final single point energies.

- Def2-QZVPP is a high accuracy basis set for all kinds of calculations. It provides SCF energies near the basis set limit and correlation energies that are also excellent. It is computationally expensive but with RI and RIJCOSX in conjunction with parallelization it can often still be applied for final single-point energy calculations. In conjunctions with such large basis sets one should also increase the accuracy of the integration grids in DFT and RIJCOSX — it would be a shame to limit the accuracy of otherwise very accurate calculations by numerical noise due to the grid.

- Correlation consistent basis sets prpvide good correlation energies but poor to very poor SCF energies. For the same size, the ano-PVDZ basis sets are much more accurate but are also computationally more expensive. Except for systematic basis set extrapolation we see little reason to use the cc bases.

- Pople basis sets are somewhat old fashioned and also much less consistent across the periodic table than the basis from the Karlsruhe group. Hence, we generally prefer the latter.

- For scalar relativistic calculations we strongly recommend to use the SARC bases in conjunction with the recontractions of the Kralsruhe bases. They are automatically loaded for ZORA and DKH. They are also flexible enough in the core region for general purpose and spectroscopic applications.

- Effective core potentials lead to some savings (but not necessarily spectacular ones) compared to all-electron relativistic calculations. For accurate results, small core ECPs should be used. They are generally available for Karlsruhe type basis sets for elements past the first transition row. We prefer Stuttgart–Dresden ECPs over LANL ones. For the first transition row, the choices are more meager. Here Karlsruhe basis sets do not exist in conjunction with ECPs and you are bound to either SDD or LANL of which we recommend the former. Geometries and energies are usually good from ECPs, for property calculations we recommend to switch to all electron scalar relativistic calculations using ZORA (magnetic properties) or DKH (electric properties).

- You can take advantage of a built-in basis set (`Print[P_Basis]=2`) and then modify it by uncontracting primitives, adding steeper functions etc. (fully uncontracted bases are generated via `uncontract` in `%basis`) Alternatively some basis sets exist that are of at least double-zeta quality In the core region including the DZP and Dunning basis sets. For higher accuracy you may want to consider the CAUG series of basis sets.

- Likewise, if you are doing calculations on anions in the gas phase it is advisable to include diffuse functions in the basis set. Having these diffuse functions around, however, makes things much more difficult as the locality of the basis set is significantly reduced. If these functions are included it is advisable to choose a small value for `Thresh` (10^{-12} or lower).

- In cases of charged molecules the COSMO feature is really useful in providing more realistic charge distributions and energetics.

- The integration grids (in DFT) used should be viewed together with the basis set. If large basis set calculations are converged to high accuracy it is advisable to also used large DFT integration grids like (`Grid=5`) or even (`Grid=6`). For 'unlimited' accuracy (i.e. benchmark calculations) it is probably best to use product grids (`Grid=0`) with a large value for `IntAcc` (perhaps around 6.0). The default grids have been chosen such that they provide adequate accuracy at the lowest possible computational cost. You should be aware that for large molecules the exchange-correlation integration is usually *not* the dominating factor (not even in combination with RI-J).

- Similarly important is the value of `Thresh` that will largely determine the tunaround time for direct SCF calculations. It may be possible to go to values of 10^{-6}–10^{-8} which will result in large speed-ups. However, the error in the final energy may then be 3 orders of magnitude larger than the cutoff or, sometimes, your calculation will fail to converge, due to the limited integral accuracy. In general it will not be possible to converge a direct SCF calculation to better than `Thresh` (the program will also not allow this). For higher accuracy values of maybe 10^{-10}–10^{-12} may be used with larger molecules requiring smaller cutoffs. In cases where the SCF is almost converged but then fails to finally converge (which is very annoying) decreasing `Thresh` (and possibly switch to `NRSCF`) may be a good idea. In general, `TCut` should be around `0.01`×`Thresh` in order to be on the safe side.

- DFT calculations have many good features and in many cases they produce reliable results. In particular if you study organic molecules it is nevertheless a good idea to check on your DFT results using MP2. MP2 in the form of RI-MP2 is usually affordable and produces reliable results (in particular for weaker interactions where DFT is less accurate). In case of a large mismatch between the MP2 and DFT results the alarm clock rings — in many such cases MP2 is the better choice. Remember that SCS-MP2 (RI-SCS-MP2) will usually produce more accurate results than MP2 itself.

- Coupled cluster calculations become more and more feasible and should be used whenever possible. The LPNO-CCSD calculations are available for single-point closed shell molecules and provide accurate results. However, a coupled cluster study does require careful study of basis set effects because convergence to the basis

set limit is very slow. The established basis set extrapolation schemes may be very helpful here. For open-shell molecules and in particular for transition metals one cannot be careful enough with the reference. You have to carefully check that the Hartree-Fock calculation converged to the desired state in order to get coupled cluster results that are meaningful. Orbital optimized MP2, CASSCF or DFT orbitals may help but we have often encountered convergence difficulties in the coupled cluster equations with such choices.

- Generally speaking, CEPA is often better than CCSD and approaches the quality of CCSD(T). It is, however, also a little less robust than CC methods because of the less rigorous treatment of the single excitations in relation to electronic relaxation.

- Don't forget: "Computers don't solve problems – people do". Not denying the importance and desire to obtain accurate numbers: don't forget that in the end it is the molecule and its chemistry or spectroscopy that we want to learn something about. The fact that you may be able to compute one or the other number a little more accurate doesn't mean that this helps us understanding the physics and chemistry of our target system any better. The danger of getting locked into technicalities and miss the desired insight is real!

7.2 Converging SCF Calculations

Despite all efforts you may still find molecules where SCF convergence is poor. These are almost invariably related to open shell situations and the answer is almost always to provide 'better' starting orbitals. Here is my standard strategy to deal with this (assuming a DFT calculation):

- Perform a small basis set (`SV`) calculation in using the `LSD` or `BP` functional and RI approximation with a cheap (e.g. `DeMon_J`) auxiliary basis set. Set `Convergence=Loose` and `MaxIter=200` or so. Turn the FinalGrid off (`NoFinalGrid`). The key point is to use a large damping factor and damp until the DIIS comes into a domain of convergence. This is accomplished by `SlowConv` or even `VerySlowConv`. If you have an even more pathological case you may need to set `DampFac` larger and `DampErr` smaller than chosen by these defaults. This calculation is quite crude and may take many cycles to converge. It will however be rather quick in terms of wall clock time. If the DIIS gets stuck at some error 0.001 or so the `SOSCF` (or even better `NRSCF`) could be put in operation from this point on.

- Use the orbitals of this calculation and `GuessMode=CMatrix` to start a calculation with the target basis set. In DFT I normally use a pure functional (e.g. `Functional=BP`). This calculation normally converges relatively smoothly.

- Use the target functional, grid etc. to get the final calculation converged. In many cases this should converge fairly well now.

Here are a few other things that can be tried:

- Try to start from the orbitals of a related closed shell species. In general closed shell MO calculations tend to converge better. You then hope to reach the convergence radius of another converger for the open shell case.

- Try to start from the orbitals of a more positive cation. Cation calculations tend to converge better.

- If you are running a CASSCF job, try to start from the natural orbitals of a MP2 or simple selected CI calculation. That will usually lead to a much better starting point (of course this does not apply if the MP2 calculation is too expensive; but it rarely will be if you are courageous enough to try CASSCF ☺).

- Try to start from a calculation with a smaller basis set. Smaller basis sets converge better. Then you have the choice of `GuessMode=CMatrix` or `GuessMode=FMatrix` which will affect the convergence behavior.

- Use large level shifts. This increases the number of iterations but stabilizes the converger. (`shift shift 0.5 erroff 0 end`)

- If your calculations converge to, say 10^{-4} in the DIIS error and then "creep" it is usually a good idea to invoke the second order (`NRSCF`) converger. Close to the solution it is incredibly strong and will usually converge in 1 or 2 macro-iterations. Each of these cost perhaps 5–7 or maybe even 10 normal SCF cycles but after the DIIS has brought you into the radius of the convergence of the NR procedure its convergence properties are excellent.

- If you are doing DFT calculations try to start from a Hartree-Fock solution for your molecule. HF calculations tend to converge somewhat better because they have a larger HOMO-LUMO gap (there are of course exceptions).

- Carefully look at the starting orbitals (`Print[P_GuessOrb]=1`) and see if they make sense for your molecule. Perhaps you have to reorder them (using `Rotate`) to obtain smooth convergence.

- Most of the time the convergence problems come from "unreasonable" structures. Did you make sure that your coordinates are in the correct units (Angström or Bohrs?) and have been correctly recognized as such by the program?

- If you have trouble with UHF calculations try ROHF (especially SAHF or CAHF) first and then go to the UHF calculation.

- Fool around with `Guess=Hueckel`, `PAtom` or even `HCore`.

- It may sometimes be better to converge to an undesired state and then take the orbitals of this state, reorder them (using `Rotate`) and try to converge to the desired state.

- Try to start the calculation with a large damping factor (`DampFac=0.90`; or even larger) and specify a relatively small DIIS error to turn damping off (say `DampErr=0.02`;). This will increase the number of cycles but may guide you into a regime were the calculation actually converges.

- If nothing else helps, stop, grab a friend and go to the next pub (you can also send me an unfriendly e-mail but this will likely not make your calculation converge any quicker; ⌣).

7.3 Choice of Theoretical Method

The array of available functionals makes it perhaps difficult to decide which one should be used. While this is a matter of ongoing research and, in the end, can only be answered by experimentation and comparison to experimental results or high-level *ab initio* calculations, I may attempt to give some guidelines.

The simplest density functionals (and in general the least accurate) are the local functionals (`Functional=LSD`). Although several variants of the local DFT exist in ORCA there is little to choose among them – they give more or less the same result.

The gradient corrected functionals are (very slightly) more expensive because the gradient of the electron density at each point in space must be computed but they are also significantly more accurate for structures and energetics of molecules. The various gradient corrected functionals (=GGA functionals) are generally similar in their behavior. The BP functional is probably the most widely used in transition metal chemistry. The BLYP, PBE or PW91 functionals may also be considered. PWP has been shown to be rather good for hyperfine coupling predictions of light nuclei in radicals. In addition, since no Hartree-Fock exchange is used you have the ability to speed up the calculation by a factor of 4-40 if the RI approximation is employed. This approximation is really advisable for the LSD and GGA functionals since it leads to very little or no loss in accuracy while giving large speedups. It is, in fact, automatically chosen to be operative when you use pure functionals.

In addition, meta-GGA's (TPSS) are available in ORCA and may provide superior results for certain properties compared to standard GGA's. They are somewhat but not much more expensive to evaluate than standard GGA's.

For many properties (but not necessarily for geometries), more accurate results are usually given by the hybrid density functionals that incorporate part of the HF exchange. The computational effort for these is higher than for the GGA functionals because the HF exchange needs to be computed exactly. Very large speedups result if this is done via the RIJCOSX approximation. Nevertheless for energetics, properties and for predictions of charge and spin densities the hybrids appear to be the best choice. The prototype functional of this kind is B3LYP, which has been very widely used throughout chemistry and is successful for a wide range of molecular properties. Other hybrids have been less well tested but maybe a good choice in specific situations, for example the PBE0 functional has been advertised for NMR chemical shift predictions and other properties. From my personal experience I can also recommend PBE0 and PWP1 as two functionals that give good predictions for EPR g-values and hyperfine couplings. The TPSSh meta-GGA hybrid is also very succesful in this area.[1]

Together with DFT, it is often observed that the atom-pairwise dispersion correction of Stefan Grimme substantially improves the results at no extra cost.

Don't forget that in present days the MP2 method becomes affordable for molecules of significant size and there are quite a number of instances where MP2 (and particularly SCS-MP2) will do significantly better than DFT even if it takes a little longer (the RI approximation is also highly recommended here). The perturbatively corrected functionals (B2PLYP) may also be a very good choice for many problems (at comparable cost to MP2; note that even for large molecules with more than 1000 basis functions the MP2 correction only takes about 10-20% of the time required for the preceeding SCF calculation if the RI approximation is invoked).

Beyond DFT and (SCS)MP2 there are coupled cluster methods and their implementation in ORCA is efficient. With the local pair natural orbital methods you can even study molecules of substantial size and with appealing turnaround times.

When to go to multireference methods is a more complicated question. Typically, this will be the case if multiplets are desired, pure spin functions for systems with several unpaired electrons, in bond breaking situations or for certain classes of excited states (loosely speaking: whenever there are weakly interacting electrons in the system). However, whenever you decide to do so, please be aware that this require substantial insight into the physics and chemistry of the problem at hand. An uneducated use of CASSCF or MRCI/MRPT method likely yields numbers that are nonsensical and that at tremendous computational cost. Here, there is no substitute for experience (and patience ☺).

---

[1] Some researchers like to adjust the amount of Hartree-Fock exchange according to their needs or what they think is "better" than the standard. This increases the semiempirical character of the calculations and may represent fixes that only work for a given class of compounds and/or properties while worsening the results for others. With this caveat in mind it is one of the things that you are free to try if you like it. However, we do not recommend it since it will deteriorate the comparability of your results with those of other workers the vast majority of which use standard functionals. An alternative to changing the amount of HF exchange could be to simply construct a linear regression for a number of known cases and then use linear regression.

8 Publications Related to ORCA

The generic reference for ORCA is:

Neese, F. (**2012**) The ORCA program system, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, <u>2</u>, 73–78.

Please do not only cite the above generic reference, but also cite in addition the original papers that report the development and ORCA implementation of the methods you have used in your studies! The following publications describe functionality implemented in ORCA. We would highly appreciate if you cite them when you use the program.

Multireference CI Module and its application to EPR properties and optical spectra:

1. Ganyushin, D.; Neese, F. (**2013**) A fully variational spin-orbit coupled complete active space self-consistent field approach: Application to electron paramagnetic resonance g-tensors, *J. Chem. Phys.*, <u>138</u>, 104113.

2. Atanasov, M.; Zadrozny, J. M.; Long, J. R.; Neese, F. (**2013**) A theoretical analysis of chemical bonding, vibronic coupling, and magnetic anisotropy in linear iron(II) complexes with single-molecule magnet behavior, *Chem. Sci.*, <u>4</u>, 139–156.

3. Atanasov, M.; Ganyushin, D.; Pantazis, D. A.; Sivalingam, K.; Neese, F. (**2011**) Detailed Ab Initio First-Principles Study of the Magnetic Anisotropy in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes, *Inorg. Chem,* <u>50</u>, 7460–7477.

4. Neese, F.; Pantazis, D. A. (**2011**) What is not required to make a single molecule magnet, *Faraday Discussions*, <u>148</u>, 229–238.

5. Duboc, C.; Ganyushin, D.; Sivalingam, K.; Collomb, M. N.; Neese, F. (**2010**) Systematic Theoretical Study of the Zero-Field Splitting in Coordination Complexes of Mn(III). Density Functional Theory versus Multireference Wave Function Approaches, *J. Phys. Chem. A*, <u>114</u>, 10750–10758.

6. Sundararajan, M.; Ganyushin, D.; Ye, S.; Neese, F. (**2009**) Multireference *ab initio* studies of Zero-Field Splitting and Magnetic Circular Dichroism Spectra of Tetrahedral Co(II) Complexes, *Dalton Trans.*, <u>30</u>, 6021–6036.

7. Ganyushin, D.; Neese, F. (**2008**) First principles calculation of magnetic circular dichroism spectra, *J. Chem. Phys.*, <u>128</u>, 114117.

8. Petrenko, T.; Neese, F. (**2007**) A general efficient quantum chemical method for predicting absorption bandshapes, resonance Raman spectra and excitation profiles for larger molecules. *J. Chem. Phys.*, <u>127</u>, 164319.

9. Neese, F. (**2007**) Analytic Derivative Calculation of Electronic g-Tensors based on Multireference Configuration Interaction Wavefunctions. *Mol. Phys.* (honorary issue for Prof. Peter Pulay), <u>105</u>, 2507–2514.

10. Neese, F.; Petrenko, T.; Ganyushin, D.; Olbrich, G. (**2007**) Advanced Aspects of *ab initio* Theoretical Spectroscopy of Open-Shell Transition Metal Ions. *Coord. Chem. Rev.*, <u>205</u>, 288–327.

11. Neese, F. (**2006**) Importance of Direct Spin-Spin Coupling and Spin-Flip Excitations for the Zero-Field Splittings of Transition Metal Complexes: A Case Study, *J. Am. Chem. Soc.*, <u>128</u>, 10213–10222.

12. Chalupský, J.; Neese, F.; Solomon, E.I.; Ryde, U.; Rulíšek, L. (**2006**) Identification of intermediates in the reaction cycle of multicopper oxidases by quantum chemical calculations of spectroscopic parameters, *Inorg. Chem.*, <u>45</u>, 11051–11059.

13. Neese, F. (**2006**) Theoretical spectroscopy of model-nonheme $[Fe^{IV}O(NH_3)_5]^{2+}$ complexes with triplet and quintet ground states using multireference *ab initio* and density functional theory methods. *J. Inorg. Biochem.* (special issue on high-valent Fe(IV)), 716–726.

14. Ganyushin, D.; Neese, F. (**2006**) First Principle Calculation of Zero-Field Splittings, *J. Chem. Phys.*, <u>125</u>, 024103.

15. Ray, K.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2005**) Electronic Structure of Square-Planar Bis(benzene-1,2-dithiolate)metal Complexes $[M(L)_2]^z$ ($z = 2-, 1-, 0$; M = Ni, Pd, Pt, Cu, Au): An experimental, Density Functional and Correlated *ab initio* Study. *Inorg. Chem.*, <u>44</u>, 5345–5360.

16. Schöneboom, J.; Neese, F.; Thiel, W. (**2005**) Towards Identification of the Compound I Reactive Intermediate in Cytochrome P450 Chemistry: A QM/MM Study of its EPR and Mössbauer Parameters, *J. Am. Chem. Soc.*, <u>127</u>, 5840–5853.

17. Wanko, M.; Hoffmann, M.; Strodel, P.; Thiel, W.; Neese, F.; Frauenheim, T.; Elstner, M. (**2005**) Calculating Absorption Shifts for Retinal Proteins: Computational Challenges *J. Phys. Chem. B*, <u>109</u>, 3606–3615.

18. Neese, F. (**2004**) Sum Over States Based Multireference *ab initio* Calculation of EPR Spin Hamiltonian Parameters for Transition Metal Complexes. A Case Study *Mag. Res. Chem.*, <u>42</u>, S187–S198.

19. Neese, F. (**2003**) Correlated *ab Initio* Calculation of Electronic g-Tensors Using a Sum Over States Formulation. *Chem. Phys. Lett.*, <u>380</u>/5–6, 721–728.

20. Neese, F. (**2003**) A Spectroscopy Oriented Configuration Interaction Procedure, *J. Chem. Phys.*, <u>119</u>, 9428–9443.

21. Neese, F. (**2001**) Configuration Interaction Calculation of Electronic g-Tensors in Transition Metal Complexes, *Int. J. Quant. Chem.*, <u>83</u>, 104–114.

22. Neese, F.; Solomon, E.I. (**1998**) Calculation of Zero-Field Splittings, *g*-values and the Relativistic Nephelauxetic Effect in Transition Metal Complexes. Application to High Spin Ferric Complexes. *Inorg. Chem.*, <u>37</u>, 6568–6582.

Pair Natural Orbital Local Correlation Methods:

23. Riplinger, C.; Neese, F. (**2013**) An efficient and near linear scaling pair natural orbital based local coupled cluster method, *J. Chem. Phys.*, <u>138</u>, 034106.

24. Liakos, D. G.; Neese, F. (**2012**) Improved Correlation Energy Extrapolation Schemes Based on Local Pair Natural Orbital Methods, *J. Phys. Chem. A*, <u>116</u>, 4801–4816.

25. Huntington, L. M. J.; Hansen, A.; Neese, F.; Nooijen, M. (**2012**) Accurate thermochemistry from a parameterized coupled-cluster singles and doubles model and a local pair natural orbital based implementation for applications to larger systems, *J. Chem. Phys.*, <u>136</u>, 064101.

26. Izsak, R.; Hansen, A.; Neese, F. (**2012**) The resolution of identity and chain of spheres approximations for the LPNO-CCSD singles Fock term, *Mol. Phys.*, <u>110</u>, 2413–2417.

27. Liakos, D. G.; Hansen, A.; Neese, F. (**2011**) Weak Molecular Interactions Studied with Parallel Implementations of the Local Pair Natural Orbital Coupled Pair and Coupled Cluster Methods, *J. Chem. Theory Comput.*, <u>7</u>, 76–87.

28. Hansen, A.; Liakos, D. G.; Neese, F. (**2011**) Efficient and accurate local single reference correlation methods for high-spin open-shell molecules using pair natural orbitals, *J. Chem. Phys.*, <u>135</u>, 214102.

29. Kollmar, C.; Neese, F. (**2011**) An orbital-invariant and strictly size extensive post-Hartree-Fock correlation functional, *J. Chem. Phys.*, <u>135</u>, 084102.

30. Kollmar, C.; Neese, F. (**2011**) The relationship between double excitation amplitudes and Z vector components in some post-Hartree-Fock correlation methods, *J. Chem. Phys.*, <u>135</u>, 064103.

31. Neese, F.; Liakos, D.; Hansen, A. (**2009**) Efficient and accurate local approximations to the coupled cluster singles and doubles method using a truncated pair natural orbital basis *J. Chem. Phys.*, <u>131</u>, 064103.

32. Neese, F.; Wennmohs, F.; Hansen, A. (**2009**) Efficient and accurate local approximations to coupled electron pair approaches. An attempt to revive the pair-natural orbital method *J. Chem. Phys.*, <u>130</u>, 114108.

Coupled-Cluster and Coupled Pair Implementation (MDCI module):

33. Kollmar, C.; Neese, F. (**2010**) The coupled electron pair approximation: variational formulation and spin adaptation, *Mol. Phys.*, <u>108</u>, 2449–2458.

34. Neese, F.; Wennmohs, F.; Hansen, A.; Grimme, S. (**2009**) Accurate Theoretical Chemistry with Coupled Electron Pair Models *Acc. Chem. Res.*, <u>42</u>(5), 641–648.

35. Wennmohs, F.; Neese, F. (**2008**) A Comparative Study of Single Reference Correlation Methods of the Coupled-Pair Type, *Chem. Phys.* (70th birthday issue for Prof. Peyerimhoff), <u>343</u>, 217–230.

ANO basis sets:

36. Neese, F.; Valeev, E. F. (**2011**) Revisiting the Atomic Natural Orbital Approach for Basis Sets: Robust Systematic Basis Sets for Explicitly Correlated and Conventional Correlated ab initio Methods?, *J. Chem. Theory Comput.*, <u>7</u>, 33–43.

Orbital Optimized MP2:

37. Sandhoefer, B.; Kossmann, S.; Neese, F. (**2013**) Derivation and assessment of relativistic hyperfine-coupling tensors on the basis of orbital-optimized second-order Moller-Plesset perturbation theory and the second-order Douglas–Kroll–Hess transformation, *J. Chem. Phys.*, <u>138</u>, 104102.

38. Kossmann, S.; Neese, F. (**2010**) Correlated ab Initio Spin Densities for Larger Molecules: Orbital-Optimized Spin-Component-Scaled MP2 Method, *J. Phys. Chem. A*, <u>114</u>, 11768-11781.

39. Neese, F.; Schwabe, T.; Kossmann, S.; Schirmer, B.; Grimme, S. (**2009**) Assessment of Orbital Optimized, Spin-Component Scaled Second Order Many Body Perturbation Theory for Thermochemistry and Kinetics. *J. Chem. Theory Comput.*, <u>5</u>, 3060-3073.

The Split-J, Split-RI-J, RIJCOSX and RI-JK methods:

40. Izsak, R.; Neese, F. (**2011**) An overlap fitted chain of spheres exchange method, *J. Chem. Phys.*, <u>135</u>, 144105.

41. Kossmann, S.; Neese, F. (**2010**) Efficient Structure Optimization with Second-Order Many-Body Perturbation Theory: The RIJCOSX-MP2 Method, *J. Chem. Theory Comput.*, <u>6</u>, 2325-2338.

42. Kossmann, S.; Neese, F. (**2009**) Comparison of Two Efficient Approximate Hartree–Fock Approaches. *Chem. Phys. Lett.*, <u>481</u>, 240-243.

43. Neese, F.; Wennmohs, F.; Hansen, A.; Becker, U. (**2009**) Efficient, approximate and parallel Hartree–Fock and hybrid DFT calculations. A 'chain-of-spheres' algorithm for the Hartree–Fock exchange, *Chem. Phys.*, <u>356</u>, 98–109.

44. Neese, F. (**2003**) An Improvement of the Resolution of the Identity Approximation for the Calculation of the Coulomb Matrix, *J. Comp. Chem.,* <u>24</u>, 1740–1747.

DFT/Hartree–Fock Theory of EPR Parameters:

45. Sandhoefer, B.; Neese, F. (**2012**) One-electron contributions to the g-tensor for second-order Douglas–Kroll–Hess theory, *J. Chem. Phys.*, <u>137</u>, 094102.

46. Ganyushin, D.; Gilka, N.; Taylor, P. R.; Marian, C. M.; Neese, F. (**2010**) The resolution of the identity approximation for calculations of spin-spin contribution to zero-field splitting parameters, *J. Chem. Phys.*, <u>132</u>, 144111.

47. Duboc, C.; Ganyushin, D.; Sivalingam, K.; Collomb, M. N.; Neese, F. (**2010**) Systematic Theoretical Study of the Zero-Field Splitting in Coordination Complexes of Mn(III). Density Functional Theory versus Multireference Wave Function Approaches, *J. Phys. Chem. A*, <u>114</u>, 10750–10758.

48. Neese, F. (**2009**) First principles approach to Spin-Hamiltonian Parameters, invited chapter in Misra, S.K. Multifrequency EPR: Theory and Applications, Wiley-VCH, pp. 297–326.

49. Pantazis, D. A.; Orio, M.; Petrenko, T.; Messinger, J.; Lubitz, W.; Neese, F. (**2009**) A new quantum chemical approach to the magnetic properties of oligonuclear transition metal clusters: Application to a model for the tetranuclear manganese cluster of Photosystem II *Chem. Eur. J.*, <u>15</u>(20), 5108–5123.

50. Riplinger, C.; Kao, J.P.Y.; Rosen, G.M.; Kathirvelu, V.; Eaton, G.R.; Eaton, S.S.; Kutateladze, A.; Neese F. (**2009**) Interaction of Radical Pairs Through-Bond and Through-Space: Scope and Limitations of the Point-Dipole Approximation in Electron Paramagnetic Resonance Spectroscopy, *J. Am. Chem. Soc.*, <u>131</u>, 10092–10106.

51. Cirera, J.; Ruiz, E.; Alvarez, S.; Neese, F.; Kortus, J. (**2009**) How to Build Molecules with Large Magnetic Anisotropy. *Chem. Eur. J.*, <u>15</u>(16), 4078–4087.

52. Neese, F. (**2008**) Spin Hamiltonian Parameters from First Principle Calculations: Theory and Application. In Hanson, G.; Berliner, L. (Eds), *Biological Magnetic Resonance*, pp. 175–232.

53. Koßmann, S.; Kirchner, B.; Neese, F. (**2007**) Performance of modern density functional theory for the prediction of hyperfine structure: meta-GGA and double hybrid functionals, *Molec. Phys.* (Arthur Schweiger memorial issue), <u>105</u>, 2049–2071.

54. Neese, F. (**2007**) Calculation of the Zero-Field Splitting Tensor Using Hybrid Density Functional and Hartree-Fock Theory. *J. Chem. Phys.*, <u>127</u>, 164112.

55. Neese, F. (**2006**) Importance of Direct Spin-Spin Coupling and Spin-Flip Excitations for the Zero-Field Splittings of Transition Metal Complexes: A Case Study, *J. Am. Chem. Soc.*, <u>128</u>, 10213–10222.

56. Sinnecker, S.; Neese, F. (**2006**) Spin-Spin Contributions to the Zero-Field Splitting Tensor in Organic Triplets, Carbenes and Biradicals – A Density Functional and *ab initio* Study. *J. Phys. Chem. A*, 110, 12267–12275.

57. Sinnecker, S.; Rajendran, A.; Klamt, A.; Diedenhofen, M.; Neese, F. (**2006**) Calculation of Solvent Shifts on Electronic G-Tensors with the Conductor-Like Screening Model (COSMO) and its Self-Consistent Generalization to Real Solvents (COSMO-RS), *J. Phys. Chem. A*, 110, 2235–2245.

58. Neese, F.; Wolf, A.; Reiher, M.; Fleig, T.; Hess, B.A. (**2005**) Higher Order Douglas-Kroll Calculation of Electric Field Gradients, *J. Chem. Phys.*, 122, 204107.

59. Neese, F. (**2005**) Efficient and Accurate Approximations to the Molecular Spin-Orbit Coupling Operator and their use in Molecular g-Tensor Calculations, *J. Chem. Phys.*, 122, 034107.

60. Ray, K.; Begum, A; Weyhermüller, T.; Piligkos, S.; van Slageren, J.; Neese, F.; Wieghardt, K. (**2005**) The Electronic Structure of the Isoelectronic, Square Planar Complexes $[Fe^{II}(L)_2]^{2-}$ and $[Co^{III}(L^{Bu})_2]^-$ (L^{2-} and $(L^{Bu})^{2-}$ = benzene-1,2-dithiolates): an Experimental and Density Functional Theoretical Study, *J. Am. Chem. Soc.*, 127, 4403–4415.

61. Neese, F. (**2003**) Metal and Ligand Hyperfine Couplings in Transition Metal Complexes. The Effect of Spin-Orbit Coupling as Studied by Coupled Perturbed Kohn-Sham Theory and Hybrid Density Functionals, *J. Chem. Phys.*, 117, 3939–3948.

62. Neese, F. (**2001**) Prediction of Electron Paramagnetic Resonance g-values by Coupled Perturbed Hartree-Fock and Kohn-Sham Theory. *J. Chem. Phys.*, 115, 11080–11096.

63. Neese, F. (**2001**) Theoretical Study of Ligand Superhyperfine Structure. Application to Cu(II) Complexes. *J. Phys. Chem. A*, 105, 4290–4299.

Corresponding Orbital Transformation:

64. Neese, F. (**2004**) Definition of Corresponding Orbitals and the Diradical Character in Broken Symmetry DFT Calculations on Spin Coupled Systems. *J. Phys. Chem. Solids*, 65, 781–785.

TD-DFT/CIS Module and Resonance Raman Spectra:

65. Petrenko, T.; Kossmann, S.; Neese, F. (**2011**) Efficient time-dependent density functional theory approximations for hybrid density functionals: Analytical gradients and parallelization, *J. Chem. Phys.*, 134, 054116.

66. Petrenko, T.; Krylova, O.; Neese, F.; Sokolowski, M. (**2009**) Optical Absorption and Emission Properties of Rubrene: Insight by a Combined Experimental and Theoretical Study. *New J. Phys.*, 11, 015001.

67. Grimme, S.; Neese, F. (**2007**) Double Hybrid Density Functional Theory for Excited States of Molecules, *J. Chem. Phys.*, 127, 154116.

68. Petrenko, T.; Ray, K.; Wieghardt, K.; Neese, F. (**2006**) Vibrational Markers for the Open-Shell Character of Metal bis-Dithiolenes: An Infrared, resonance Raman and Quantum Chemical Study. *J. Am. Chem. Soc.*, 128, 4422–4436.

69. Neese, F.; Olbrich, G. (**2002**) Efficient use of the Resolution of the Identity Approximation in Time-Dependent Density Functional Calculations with Hybrid Functionals, *Chem. Phys. Lett.*, 362, 170–178.

Absorption, Resonance Raman and Fluorescence Spectra:

70. Petrenko, T.; Neese F. (**2012**) Efficient and automatic calculation of optical band shapes and resonance Raman spectra for larger molecules within the independent mode displaced harmonic oscillator model, *J. Chem. Phys.*, 137, 234107.

71. Petrenko, T.; Neese, F. (**2007**) A general efficient quantum chemical method for predicting absorption bandshapes, resonance Raman spectra and excitation profiles for larger molecules. *J. Chem. Phys.*, 127, 164319.

72. Petrenko, T.; Krylova, O.; Neese, F. Sokolowski, M. (**2009**) Optical Absorption and Emission Properties of Rubrene: Insight by a Combined Experimental and Theoretical Study. *New J. Phys.*, 11, 015001.

Magnetic Circular Dichroism Spectra:

73. Westphal, A.; Broda, H.; Kurz, P.; Neese, F.; Tuczek, F. (**2012**) Magnetic Circular Dichroism Spectrum of the Molybdenum(V) Complex (Mo(O)Cl$_3$dppe): C-Term Signs and Intensities for Multideterminant Excited Doublet States, *Inorg. Chem.*, 51, 5748–5763.

74. van Slageren, J.; Piligkos, S.; Neese, F. (**2010**) Magnetic circular dichroism spectroscopy on the Cr(8) antiferromagnetic ring, *Dalton Trans.*, 39, 4999–5004.

75. Sundararajan, M.; Ganyushin, D.; Ye, S.; Neese, F. (**2009**) Multireference *ab initio* studies of Zero-Field Splitting and Magnetic Circular Dichroism Spectra of Tetrahedral Co(II) Complexes, *Dalton Trans.*, 30, 6021–6036.

76. Piligkos, S.; Slep, L.; Weyhermüller, T.; Chaudhuri, P.; Bill, E.; Neese, F. (**2009**) Magnetic Circular Dichroism Spectroscopy of weakly exchange coupled dimers. A model study. *Coord. Chem. Rev.*, 253, 2352–2362.

77. Ganyushin, D.; Neese, F. (**2008**) First principles calculation of magnetic circular dichroism spectra, *J. Chem. Phys.*, 128, 114117.

78. Neese, F.; Solomon, E.I. (**1999**) MCD *C*-term Signs, Saturation Behavior and Determination of Band Polarizations in Randomly Oriented Systems with Spin $S \geqslant 1/2$. Applications to $S = 1/2$ and $S = 5/2$. *Inorg. Chem.*, 38, 1847–1865.

Mössbauer Isomer shifts:

79. Römelt, M.; Ye, S.; Neese, F. (**2009**) Calibration of Mössbauer Isomer Shift Calculations for Modern Density Functional Theory: meta-GGA and Double Hybrid Functionals *Inorg. Chem.*, 48, 784–785.

80. Sinnecker, S.; Slep, L.; Bill, E.; Neese, F. (**2005**) Performance of Nonrelativistic and Quasirelativistic Hybrid DFT for the Prediction of Electric and Magnetic Hyperfine Parameters in ^{57}Fe Mössbauer Spectra, *Inorg. Chem.*, 44, 2245–2254.

81. Neese, F. (**2002**) Prediction and Interpretation of Isomer Shifts in ^{57}Fe Mössbauer Spectra by Density Functional Theory. *Inorg. Chim. Acta* (special Karl Wieghardt honorary issue), 337C, 181–192.

Nuclear Resonance Vibrational Spectra:

82. Petrenko T,; Sturhahn, W.; Neese, F. (**2008**) First principles calculation of Nuclear Resonance Vibrational Spectra, *Hyperfine Interactions*, 175, 165–174.

83. DeBeer-George, S.; Petrenko, T.; Aliaga-Alcade, N.; Bill, E.; Mienert, B.; Sturhan, W.; Ming, Y.; Wieghardt, K.; Neese, F. (**2007**) Characterization of a Genuine Iron(V)Nitrido Species by Nuclear Resonant Vibrational Spectroscopy Coupled to Density Functional Calculations, *J. Am. Chem. Soc.*, 129, 11053–11060.

X-Ray Absorption and X-Ray Emission Spectra:

84. Roemelt, M.; Beckwith, M. A.; Duboc, C.; Collomb, M.-N.; Neese, F.; DeBeer, S. (**2012**) Manganese K-Edge X-Ray Absorption Spectroscopy as a Probe of the Metal-Ligand Interactions in Coordination Compounds, *Inorg. Chem.*, <u>51</u>, 680–687.

85. Chandrasekaran, P.; Stieber, S. C. E.; Collins, T. J.; Que, L.; Neese, F.; DeBeer, S. (**2011**) Prediction of high-valent iron K-edge absorption spectra by time-dependent Density Functional Theory, *Dalton Trans.*, <u>40</u>, 11070–11079.

86. Beckwith, M. A.; Roemelt, M.; Collomb, M. N.; Duboc, C.; Weng, T. C.; Bergmann, U.; Glatzel, P.; Neese, F.; DeBeer, S. (**2011**) Manganese K beta X-ray Emission Spectroscopy As a Probe of Metal-Ligand Interactions, *Inorg. Chem,* <u>50</u>, 8397–8409.

87. Lee, N.; Petrenko, T.; Bergmann, U.; Neese, F.; DeBeer, S. (**2010**) Probing Valence Orbital Composition with Iron K beta X-ray Emission Spectroscopy, *J. Am. Chem. Soc.*, <u>132</u>, 9715–9727.

88. DeBeer-George, S.; Neese, F. (**2010**) Calibration of Scalar Relativistic Density Functional Theory for the Calculation of Sulfur K-Edge X-ray Absorption Spectra, *Inorg. Chem,* <u>49</u>, 1849–1853.

89. DeBeer-George, S.; Petrenko, T.; Neese, F. (**2008**) Prediction of Iron- K-edge Absorption Spectra using Time-Dependent Density Functional Theory, *J. Phys. Chem. A.*, <u>112</u>, 12936–12943.

90. DeBeer-George, S.; Petrenko, T.; Neese, F. (**2008**) Time-dependent density functional calculations of ligand K-edge X-ray absorption spectra, *Inorg. Chim. Acta* (60th birthday issue of Prof. E.I. Solomon), <u>361</u>, 965–972.

Double hybrid density functionals:

91. Grimme, S.; Neese, F. (**2007**) Double Hybrid Density Functional Theory for Excited States of Molecules, *J. Chem. Phys.*, <u>127</u>, 154116.

92. Neese, F.; Schwabe, T.; Grimme, S. (**2007**) Analytic Derivatives for Perturbatively Corrected 'Double Hybrid' Density Functionals, *J. Chem. Phys.*, <u>126</u>, 124115.

93. Koßmann, S.; Kirchner, B.; Neese, F. (**2007**) Performance of modern density functional theory for the prediction of hyperfine structure: meta-GGA and double hybrid functionals, *Mol. Phys.* (Arthur Schweiger memorial issue), <u>105</u>, 2049–2071.

Dispersion Corrections to DFT:

(not originally implemented in ORCA but the ORCA implementation is based on the code described in these papers)

94. Grimme, S. (**2004**) *J. Comput. Chem.*, <u>25</u>, 1463–1476.

95. Grimme, S. (**2006**) *J. Comput. Chem.*, <u>27</u>, 1787–1799.

96. Grimme, S.; Antony, J. Ehrlich, S. Krieg, H. (**2010**) *J. Chem. Phys.*, <u>132</u>, 154104.

97. Grimme, S.; Ehrlich S.; Goerigk L. (**2011**) *J. Comput. Chem.*, <u>32</u>, 1456-1465.

gCP correction to HF and DFT:

(ORCA implementation is based upon the code used in this paper)

98. Kruse, H; Grimme S. (**2012**) *J. Chem. Phys.*, <u>126</u>, 154101.

HF-3c method:

99. Sure, R; Grimme S. (**2013**) *J. Comput. Chem.*, <u>accepted</u> .

COSMO Implementation:

100. Sinnecker, S.; Rajendran, A.; Klamt, A.; Diedenhofen, M.; Neese, F. (**2006**) Calculation of Solvent Shifts on Electronic G-Tensors with the Conductor-Like Screening Model (COSMO) and its Self-Consistent Generalization to Real Solvents (COSMO-RS), *J. Phys. Chem. A*, <u>110</u>, 2235–2245.

SOSCF Method:

101. Neese, F. (**2000**) Approximate Second Order Convergence for Spin Unrestricted Wavefunctions. *Chem. Phys. Lett.*, <u>325</u>, 93–98.

Relativity and SARC Basis Sets:

102. Pantazis, D. A.; Neese, F. (**2012**) All-Electron Scalar Relativistic Basis Sets for the 6p Elements, *Theor. Chem. Acc.*, <u>131</u>, 1292.

103. Pantazis, D. A.; Neese, F. (**2011**) All-Electron Scalar Relativistic Basis Sets for the Actinides, *J. Chem. Theory Comput.*, <u>7</u>, 677–684.

104. Pantazis, D. A.; Neese, F. (**2009**) All-Electron Scalar Relativistic Basis Sets for the Lanthanides, *J. Chem. Theory Comput.*, <u>5</u>, 2229–2238.

105. Bühl, M.; Reimann, C.; Pantazis, D. A.; Bredow, T.; Neese, F. (**2008**) Geometries of Third-row Transition-Metal Complexes from Density-Functional Theory *J. Chem. Theory Comput.*, <u>4</u>, 1449–1459.

106. Pantazis, D. A.; Chen, X.-Y.; Landis, C.R.; Neese, F. (**2008**) All Electron Scalar Relativistic Basis Sets for Third Row Transition Metal Atoms. *J. Chem. Theory Comput.*, <u>4</u>, 908–919.

QM/MM calculations with ORCA:

107. Sundararajan, M.; Neese, F. (**2012**) Detailed QM/MM study of the Electron Paramagnetic Resonance Parameters of Nitrosyl Myoglobin, *J. Chem. Theory Comput.*, <u>8</u>, 563–574.

108. Riplinger, C.; Neese, F. (**2011**) The reaction mechanism of Cytochrome P450 NO Reductase: A Detailed Quantum Mechanics/Molecular Mechanics Study, *ChemPhysChem*, <u>12</u>, 3192–3203.

109. Radoul, M.; Sundararajan, M.; Potapov, A.; Riplinger, C.; Neese, F.; Goldfarb, D. (**2010**) Revisiting the nitrosyl complex of myoglobin by high-field pulse EPR spectroscopy and quantum mechanical calculations, *Phys. Chem. Chem. Phys.*, <u>12</u>, 7276–7289.

110. Sundararajan, M.; Riplinger, C.; Orio, M.; Wennmohs, F.; Neese, F. (**2009**) Spectroscopic Properties of Protein-Bound Cofactors: Calculation by Combined Quantum Mechanical/Molecular Mechanical (QM/MM) Approaches, *Encyc. Inorg. Chem.*, DOI: 10.1002/9781119951438.eibc0371.

111. Altun, A.; Kumar, D.; Neese, F.; Thiel, W. (**2008**) Multi-reference Ab Initio QM/MM Study on Intermediates in the Catalytic Cycle of Cytochrome P450cam, *J. Phys. Chem.*, <u>112</u>, 12904–12910.

112. Chalupský, J.; Neese, F.; Solomon, E.I.; Ryde, U.; Rulíšek, L. (**2006**) Identification of intermediates in the reaction cycle of multicopper oxidases by quantum chemical calculations of spectroscopic parameters, *Inorg. Chem.*, <u>45</u>, 11051–11059.

113. Sinnecker, S.; Neese, F. (**2006**) QM/MM Calculations with DFT for Taking into Account Protein Effects on the EPR and Optical Spectra of Metalloproteins. Plastocyanin as a Case Study. *J. Comp. Chem. (Special issue on Theoretical Bioinorganic Chemistry)*, <u>27</u>, 1463–1475.

114. Wanko, M.; Hoffmann, M.; Strodel, P.; Thiel, W.; Neese, F.; Frauenheim, T.; Elstner, M. (**2005**) Calculating Absorption Shifts for Retinal Proteins: Computational Challenges *J. Phys. Chem. B*, <u>109</u>, 3606–3615.

115. Schöneboom, J.; Neese, F.; Thiel, W. (**2005**) Towards Identification of the Compound I Reactive Intermediate in Cytochrome P450 Chemistry: A QM/MM Study of its EPR and Mössbauer Parameters, *J. Am. Chem. Soc.*, <u>127</u>, 5840–5853.

Applications that make use of ORCA include the following:

116. Krewald, V.; Neese, F.; Pantazis, D. A. (**2013**) On the magnetic and spectroscopic properties of high-valent Mn_3CaO_4 cubanes as structural units of natural and artificial water oxidizing catalysts, *J. Am. Chem. Soc.*, <u>135</u>, 5726–5739.

117. Kampa, M.; Pandelia, M.-E.; Lubitz, W.; van Gastel, M.; Neese, F. (**2013**) A Metal-Metal Bond in the Light-Induced State of [NiFe] Hydrogenases with Relevance to Hydrogen Evolution, *J. Am. Chem. Soc.*, <u>135</u>, 3915–3925.

118. Pandelia, M.-E.; Bykov, D.; Izsak, R.; Infossi, P.; Giudici-Orticoni, M.-T.; Bill, E.; Neese, F.; Lubitz, W. (**2013**) Electronic structure of the unique [4Fe-3S] cluster in O_2-tolerant hydrogenases characterized by Fe-57 Mossbauer and EPR spectroscopy, *Proc. Natl. Acad. Sci. USA*, <u>110</u>, 483–488.

119. Atanasov, M.; Surawatanawong, P.; Wieghardt, K.; Neese, F. (**2013**) A theoretical study of zero-field splitting in $Fe(IV)S_6$ ($S = 1$) and $Fe(III)S_6$ ($S = 1/2$) core complexes, $[Fe^{IV}(Et_2dtc)_{3-n}(mnt)_n]^{(n-1)-}$ and $[Fe^{III}(Et_2dtc)_{3-n}(mnt)_n]^{n-}$ ($n = 0, 1, 2, 3$): The origin of the magnetic anisotropy, *Coord. Chem. Rev.*, <u>257</u>(1), 27–41.

120. Retegan, M.; Collomb, M.-N.; Neese, F.; Duboc, C. (**2013**) A combined high-field EPR and quantum chemical study on a weakly ferromagnetically coupled dinuclear Mn(III) complex. A complete analysis of the EPR spectrum beyond the strong coupling limit, *Phys. Chem. Chem. Phys.*, <u>15</u>, 223–234.

121. Zadrozny, J. M.; Atanasov, M.; Bryan, A. M.; Lin, C. Y.; Rekken, B. D.; Power, P. P.; Neese, F.; Long, J. R. (**2013**) Slow magnetization dynamics in a series of two-coordinate iron(II) complexes, *Chem. Sci.*, <u>4</u>, 125–138.

122. Weber, K.; Krämer, T.; Shafaat, H. S.; Weyhermüller, T.; Bill, E.; van Gastel, M.; Neese, F.; Lubitz, W. (**2012**) A Functional [NiFe]-Hydrogenase Model Compound That Undergoes Biologically Relevant Reversible Thiolate Protonation, *J. Am. Chem. Soc.*, <u>134</u>, 20745–20755.

123. Kampa, M.; Lubitz, W.; van Gastel, M.; Neese, F.; (**2012**) Computational study of the electronic structure and magnetic properties of the Ni-C state in [NiFe] hydrogenases including the second coordination sphere, *J. Biol. Inorg. Chem.*, <u>17</u>, 1269–1281.

124. Atanasov, M.; Comba, P.; Helmle, S.; Müller, D.; Neese, F. (**2012**) Zero-Field Splitting in a Series of Structurally Related Mononuclear Ni^{II}-Bispidine Complexes, *Inorg. Chem.*, <u>51</u>, 12324–12335.

125. Shafaat, H. S.; Weber, K.; Petrenko, T.; Neese, F.; Lubitz, W. (**2012**) Key Hydride Vibrational Modes in [NiFe] Hydrogenase Model Compounds Studied by Resonance Raman Spectroscopy and Density Functional Calculations, *Inorg. Chem.*, 51, 11787–11797.

126. Argirevic, T.; Riplinger, C.; Stubbe, J.; Neese, F.; Bennati, M. (**2012**) ENDOR Spectroscopy and DFT Calculations: Evidence for the Hydrogen-Bond Network Within α2 in the PCET of *E. coli* Ribonucleotide Reductase, *J. Am. Chem. Soc.*, 134, 17661–17670.

127. Albrecht, C.; Shi, L. L.; Perez, J. M.; van Gastel, M.; Schwieger, S.; Neese, F.; Streubel, R. (**2012**) Deoxygenation of Coordinated Oxaphosphiranes: A New Route to P=C Double-Bond Systems, *Chem. Eur. J.*, 18, 9780–9783.

128. Maganas, D.; Krzystek, J.; Ferentinos, E.; Whyte, A. M.; Robertson, N.; Psycharis, V.; Terzis, A.; Neese, F.; Kyritsis, P. (**2012**) Investigating Magnetostructural Correlations in the Pseudooctahedral trans-[Ni$^{\text{II}}${(OPPh$_2$) (EPPh$_2$)N}$_2$(sol)$_2$] Complexes (E = S, Se; sol = DMF, THF) by Magnetometry, HFEPR, and ab Initio Quantum Chemistry, *Inorg. Chem.*, 51, 7218–7231.

129. Ye, S. F.; Neese, F. (**2012**) How Do Heavier Halide Ligands Affect the Signs and Magnitudes of the Zero-Field Splittings in Halogenonickel(II) Scorpionate Complexes? A Theoretical Investigation Coupled to Ligand-Field Analysis, *J. Chem. Theory Comput.*, 8, 2344–2351.

130. Nesterov, V.; Ozbolat-Schon, A.; Schnakenburg, G.; Shi, L. L.; Cangonul, A.; van Gastel, M.; Neese, F.; Streubel, R. (**2012**) An Unusal Case of Facile Non-Degenerate P-C Bond Making and Breaking, *Chem. Asian J.*, 7, 1708–1712.

131. Bykov, D.; Neese, F. (**2012**) Reductive activation of the heme iron-nitrosyl intermediate in the reaction mechanism of cytochrome c nitrite reductase: a theoretical study, *J. Biol. Inorg. Chem.*, 17, 741–760.

132. Lancaster, K. M.; Zaballa, M.E.; Sproules, S.; Sundararajan, M.; DeBeer, S.; Richards, J. H.; Vila, A. J.; Neese, F.; Gray, H. B. (**2012**) Outer-Sphere Contributions to the Electronic Structure of Type Zero Copper Proteins, *J. Am. Chem. Soc.*, 134, 8241–8253.

133. Benkhauser-Schunk, C.; Wezisla, B.; Urbahn, K.; Kiehne, U.; Daniels, J.; Schnakenburg, G.; Neese, F.; Lutzen, A. (**2012**) Synthesis, Chiral Resolution, and Absolute Configuration of Functionalized Troger's Base Derivatives: Part II, *ChemPlusChem*, 77, 396–403.

134. Ye, S. F.; Riplinger, C.; Hansen, A.; Krebs, C.; Bollinger, J. M.; Neese, F. (**2012**) Electronic Structure Analysis of the Oxygen-Activation Mechanism by Fe$^{\text{II}}$- and α-Ketoglutarate (αKG)-Dependent Dioxygenases, *Chem. Eur. J.*, 18, 6555–6567.

135. Desrochers, P. J.; Sutton, C. A.; Abrams, M. L.; Ye, S. F.; Neese, F.; Telser, J.; Ozarowski, A.; Krzystek, J. (**2012**) Electronic Structure of Nickel(II) and Zinc(II) Borohydrides from Spectroscopic Measurements and Computational Modeling, *Inorg. Chem.*, 51, 2793–2805.

136. Torres-Alacan, J.; Krahe, O.; Filippou, A. C.; Neese, F.; Schwarzer, D.; Vöhringer, P. (**2012**) The Photochemistry of [Fe$^{\text{III}}$N$_3$(cyclam-ac)]PF$_6$ at 266 nm, *Chem. Eur. J.*, 18, 3043–3055.

137. Maekawa, M.; Römelt, M.; Daniliuc, C. G.; Jones, P. G.; White, P. S.; Neese, F.; Walter, M. D. (**2012**) Reactivity studies on [Cp$'$MnX(thf)]$_2$: manganese amide and polyhydride synthesis *Chem. Sci.*, 3, 2972–2979.

138. Pantazis, D. A.; Ames, W.; Cox, N.; Lubitz, W.; Neese, F. (**2012**) Two interconvertible structures that explain the spectroscopic properties of the oxygen-evolving complex of photosystem II in the S$_2$ state, *Angew. Chem. Int. Ed.*, 51, 9935–9940. (selected as cover article and VIP paper)

139. Vennekate, H.; Schwarzer, D.; Torres-Alacan, J.; Krahe, O.; Filippou, A. C.; Neese, F.; Vöhringer, P. (**2012**) Ultrafast primary processes of an iron-(III) azido complex in solution induced with 266 nm light, *Phys. Chem. Chem. Phys.*, <u>14</u>, 6165–6172.

140. Christian, G. J.; Ye, S.; Neese, F. (**2012**) Oxygen activation in extradiol catecholate dioxygenases – a density functional study, *Chem. Sci.*, <u>3</u>, 1600–1611.

141. Cowley, R. E.; Christian, G. J.; Brennessel, W. W.; Neese, F.; Holland, P. L. (**2012**) A Reduced (beta-Diketiminato)iron Complex with End-On and Side-On Nitriles: Strong Backbonding or Ligand Non-Innocence? *Eur. J. Inorg. Chem.*, 479–483.

142. Thiessen, A.; Wettach, H.; Meerholz, K.; Neese, F.; Hoger, S.; Hertel, D. (**2012**) Control of electronic properties of triphenylene by substitution, *Organic Electronics*, <u>13</u>, 71–83.

143. Lancaster, K. M.; Roemelt, M.; Ettenhuber, P.; Hu, Y. L.; Ribbe, M. W.; Neese, F.; Bergmann, U.; DeBeer, S. (**2011**) X-ray Emission Spectroscopy Evidences a Central Carbon in the Nitrogenase Iron-Molybdenum Cofactor, *Science*, <u>334</u>, 974–977.

144. Ames, W.; Pantazis, D. A.; Krewald, V.; Cox, N.; Messinger, J.; Lubitz, W.; Neese, F. (**2011**) Theoretical Evaluation of Structural Models of the S_2 State in the Oxygen Evolving Complex of Photosystem II: Protonation States and Magnetic Interactions, *J. Am. Chem. Soc.*, <u>133</u>, 19743–19757.

145. Antony, J.; Grimme, S.; Liakos, D. G.; Neese, F. (**2011**) Protein-Ligand Interaction Energies with Dispersion Corrected Density Functional Theory and High-Level Wave Function Based Methods, *J. Phys. Chem. A*, <u>115</u>, 11210–11220.

146. Radoul, M.; Bykov, D.; Rinaldo, S.; Cutruzzola, F.; Neese, F.; Goldfarb, D. (**2011**) Dynamic Hydrogen-Bonding Network in the Distal Pocket of the Nitrosyl Complex of Pseudomonas aeruginosa cd(1) Nitrite Reductase, *J. Am. Chem. Soc.*, <u>133</u>, 3043–3055.

147. Liakos, D. G.; Neese, F. (**2011**) Interplay of Correlation and Relativistic Effects in Correlated Calculations on Transition-Metal Complexes: The $Cu_2O_2^{2+}$ Core Revisited, *J. Chem. Theory Comput.*, <u>7</u>, 1511–1523.

148. Riplinger, C.; Neese, F. (**2011**) The Reaction Mechanism of Cytochrome P450 NO Reductase: A Detailed Quantum Mechanics/Molecular Mechanics Study, *ChemPhysChem*, <u>12</u>, 3192–3203.

149. Maganas, D.; Sottini, S.; Kyritsis, P.; Groenen, E. J. J.; Neese, F. (**2011**) Theoretical Analysis of the Spin Hamiltonian Parameters in $Co^{II}S_4$ Complexes, Using Density Functional Theory and Correlated ab initio Methods, *Inorg. Chem,* <u>50</u>, 8741–8754.

150. Surawatanawong, P.; Sproules, S.; Neese, F.; Wieghardt, K. (**2011**) Electronic Structures and Spectroscopy of the Electron Transfer Series $Fe(NO)L_2^z$ ($z = 1+, 0, 1-, 2-, 3-$; L = Dithiolene), *Inorg. Chem,* <u>50</u>, 12064–12074.

151. Cox, N.; Ames, W.; Epel, B.; Kulik, L. V.; Rapatskiy, L.; Neese, F.; Messinger, J.; Wieghardt, K.; Lubitz, W. (**2011**) Electronic Structure of a Weakly Antiferromagnetically Coupled Mn(II)Mn(III) Model Relevant to Manganese Proteins: A Combined EPR, [55]Mn-ENDOR, and DFT Study, *Inorg. Chem,* <u>50</u>, 8238–8251.

152. Rota, J. B.; Knecht, S.; Fleig, T.; Ganyushin, D.; Saue, T.; Neese, F.; Bolvin, H. (**2011**) Zero field splitting of the chalcogen diatomics using relativistic correlated wave-function methods, *J. Chem. Phys.*, <u>135</u>, 114106.

153. Atanasov, M.; Ganyushin, D.; Pantazis, D. A.; Sivalingam, K.; Neese, F. (**2011**) Detailed Ab Initio First-Principles Study of the Magnetic Anisotropy in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes, *Inorg. Chem,* <u>50</u>, 7460–7477.

154. Cox, N.; Rapatskiy, L.; Su, J. H.; Pantazis, D. A.; Sugiura, M.; Kulik, L.; Dorlet, P.; Rutherford, A. W.; Neese, F.; Boussac, A.; Lubitz, W.; Messinger, J. (**2011**) Effect of Ca^{2+}/Sr^{2+} Substitution on the Electronic Structure of the Oxygen-Evolving Complex of Photosystem II: A Combined Multifrequency EPR, ^{55}Mn-ENDOR, and DFT Study of the S_2 State, *J. Am. Chem. Soc.*, <u>133</u>, 3635–3648.

155. Maurice, R.; Sivalingam, K.; Ganyushin, D.; Guihery, N.; de Graaf, C.; Neese, F. (**2011**) Theoretical Determination of the Zero-Field Splitting in Copper Acetate Monohydrate, *Inorg. Chem,* <u>50</u>, 6229–6236.

156. Su, J. H.; Cox, N.; Ames, W.; Pantazis, D. A.; Rapatskiy, L.; Lohmiller, T.; Kulik, L. V.; Dorlet, P.; Rutherford, A. W.; Neese, F.; Boussac, A.; Lubitz, W.; Messinger, J. (**2011**) The electronic structures of the S_2 states of the oxygen-evolving complexes of photosystem II in plants and cyanobacteria in the presence and absence of methanol, *Biochim. Biophys. Acta-Bioenergetics*, <u>1807</u>, 829–840.

157. Bykov, D.; Neese, F. (**2011**) Substrate binding and activation in the active site of cytochrome c nitrite reductase: a density functional study, *J. Biol. Inorg. Chem,* <u>16</u>, 417–430.

158. Gennari, M.; Orio, M.; Pecaut, J.; Bothe, E.; Neese, F.; Collomb, M. N.; Duboc, C. (**2011**) Influence of Mixed Thiolate/Thioether versus Dithiolate Coordination on the Accessibility of the Uncommon +I and +III Oxidation States for the Nickel Ion: An Experimental and Computational Study, *Inorg. Chem,* <u>50</u>, 3707–3716.

159. Ye, S. F.; Neese, F. (**2011**) Nonheme oxo-iron(IV) intermediates form an oxyl radical upon approaching the C-H bond activation transition state, *Proc. Natl. Acad. Sci. USA*, <u>108</u>, 1228–1233.

160. Gennari, M.; Pecaut, J.; DeBeer, S.; Neese, F.; Collomb, M. N.; Duboc, C. (**2011**) A Fully Delocalized Mixed-Valence Bis-μ-(Thiolato) Dicopper Complex: A Structural and Functional Model of the Biological Cu(A) Center, *Angew. Chem., Int. Ed.*, <u>50</u>, 5661–5665.

161. Gennari, M.; Retegan, M.; DeBeer, S.; Pecaut, J.; Neese, F.; Collomb, M. N.; Duboc, C. (**2011**) Experimental and Computational Investigation of Thiolate Alkylation in Ni(II) and Zn(II) Complexes: Role of the Metal on the Sulfur Nucleophilicity, *Inorg. Chem,* <u>50</u>, 10047–10055.

162. Atanasov, M.; Delley, B.; Neese, F.; Tregenna-Piggott, P. L.; Sigrist, M. (**2011**) Theoretical Insights into the Magnetostructural Correlations in Mn(3)-Based Single-Molecule Magnets, *Inorg. Chem,* <u>50</u>, 2112–2124.

163. Neese, F.; Pantazis, D. A. (**2011**) What is not required to make a single molecule magnet, *Faraday Discussions*, <u>148</u>, 229–238.

164. Lassalle-Kaiser, B.; Hureau, C.; Pantazis, D. A.; Pushkar, Y.; Guillot, R.; Yachandra, V. K.; Yano, J.; Neese, F.; Anxolabéhère-Mallart, E. (**2010**) Activation of a water molecule using a mononuclear Mn complex: from Mn-aquo, to Mn-hydroxo, to Mn-oxyl via charge compensation, *Energy Environ. Sci.*, <u>3</u>, 924–938.

165. Pantazis, D. A.; Krewald, V.; Orio, M.; Neese, F. (**2010**) Theoretical magnetochemistry of dinuclear manganese complexes: broken symmetry density functional theory investigation on the influence of bridging motifs on structure and magnetism, *Dalton Trans.*, <u>39</u>, 4959–4967.

166. Woertink, J. S.; Tian, L.; Maiti, D.; Lucas, H. R.; Himes, R. A.; Karlin, K. D.; Neese, F.; Wurtele, C.; Holthausen, M. C.; Bill, E.; Sundermeyer, J.; Schindler, S.; Solomon, E. I. (**2010**) Spectroscopic and Computational Studies of an End-on Bound Superoxo-Cu(II) Complex: Geometric and Electronic Factors That Determine the Ground State, *Inorg. Chem,* <u>49</u>, 9450–9459.

167. McNaughton, R. L.; Roemelt, M.; Chin, J. M.; Schrock, R. R.; Neese, F.; Hoffman, B. M. (**2010**) Experimental and Theoretical EPR Study of Jahn–Teller-Active HIPTN(3)N MoL Complexes (L = N_2, CO, NH_3), *J. Am. Chem. Soc.*, <u>132</u>, 8645–8656.

168. Geng, C. Y.; Ye, S. F.; Neese, F. (**2010**) Analysis of Reaction Channels for Alkane Hydroxylation by Nonheme Iron(IV)-Oxo Complexes, *Angew. Chem., Int. Ed.*, <u>49</u>, 5717–5720.

169. Orio, M.; Jarjayes, O.; Kanso, H.; Philouze, C.; Neese, F.; Thomas, F. (**2010**) X-Ray Structures of Copper(II) and Nickel(II) Radical Salen Complexes: The Preference of Galactose Oxidase for Copper(II), *Angew. Chem., Int. Ed.*, <u>49</u>, 4989–4992.

170. Gennari, M.; Orio, M.; Pecaut, J.; Neese, F.; Collomb, M. N.; Duboc, C. (**2010**) Reversible Apical Coordination of Imidazole between the Ni(III) and Ni(II) Oxidation States of a Dithiolate Complex: A Process Related to the Ni Superoxide Dismutase, *Inorg. Chem,* <u>49</u>, 6399–6401.

171. Ye, S. F.; Price, J. C.; Barr, E. W.; Green, M. T.; Bollinger, J. M.; Krebs, C.; Neese, F. (**2010**) Cryoreduction of the NO-Adduct of Taurine:alpha-Ketoglutarate Dioxygenase (TauD) Yields an Elusive {FeNO}8 Species, *J. Am. Chem. Soc.*, <u>132</u>, 4739–4751.

172. Maganas, D.; Grigoropoulos, A.; Staniland, S. S.; Chatziefthimiou, S. D.; Harrison, A.; Robertson, N.; Kyritsis, P.; Neese, F. (**2010**) Tetrahedral and Square Planar Ni(SPR$_2$)$_2$N$_2$ complexes, R = Ph & *i*Pr Revisited: Experimental and Theoretical Analysis of Interconversion Pathways, Structural Preferences, and Spin Delocalization, *Inorg. Chem,* <u>49</u>, 5079–5093.

173. Anoop, A.; Thiel, W.; Neese, F. (**2010**) A Local Pair Natural Orbital Coupled Cluster Study of Rh Catalyzed Asymmetric Olefin Hydrogenation, *J. Chem. Theory Comput.*, <u>6</u>, 3137–3144.

174. Duboc, C.; Collomb, M. N.; Neese, F. (**2010**) Understanding the Zero-Field Splitting of Mononuclear Manganese(II) Complexes from Combined EPR Spectroscopy and Quantum Chemistry, *Appl. Magn. Res.*, <u>37</u>, 229–245.

175. Ye, S. F.; Neese, F. (**2010**) The Unusual Electronic Structure of Dinitrosyl Iron Complexes, *J. Am. Chem. Soc.*, <u>132</u>, 3646–3647.

176. Kochem, A.; Orio, M.; Jarjayes, O.; Neese, F.; Thomas, F. (**2010**) Unsymmetrical one-electron oxidized Ni(II)-bis(salicylidene) complexes: a protonation-induced shift of the oxidation site, *Chem. Commun.*, <u>46</u>, 6765–6767.

177. Ozbolat-Schon, A.; Bode, M.; Schnakenburg, G.; Anoop, A.; van Gastel, M.; Neese, F.; Streubel, R. (**2010**) Insights into the Chemistry of Transient P-Chlorophosphanyl Complexes, *Angew. Chem., Int. Ed.*, <u>49</u>, 6894–6898.

178. Vancoillie, S.; Chalupsky, J.; Ryde, U.; Solomon, E. I.; Pierloot, K.; Neese, F.; Rulisek, L. (**2010**) Multireference Ab Initio Calculations of g tensors for Trinuclear Copper Clusters in Multicopper Oxidases, *J. Phys. Chem. B*, <u>114</u>, 7692–7702.

179. Grote, D.; Finke, C.; Kossmann, S.; Neese, F.; Sander, W. (**2010**) 3,4,5,6-Tetrafluorophenylnitren-2-yl: A Ground-State Quartet Triradical, *Chem. Eur. J.*, <u>16</u>, 4496–4506.

180. Ye, S. F.; Neese, F.; Ozarowski, A.; Smirnov, D.; Krzystek, J.; Telser, J.; Liao, J. H.; Hung, C. H.; Chu, W. C.; Tsai, Y. F.; Wang, R. C.; Chen, K. Y.; Hsu, H. F. (**2010**) Family of V(III)-Tristhiolato Complexes Relevant to Functional Models of Vanadium Nitrogenase: Synthesis and Electronic Structure Investigations by Means of High-Frequency and -Field Electron Paramagnetic Resonance Coupled to Quantum Chemical Computations, *Inorg. Chem,* <u>49</u>, 977–988.

181. Hegele, P.; Santhamma, B.; Schnakenburg, G.; Frohlich, R.; Kataeva, O.; Nieger, M.; Kotsis, K.; Neese, F.; Dotz, K. H. (**2010**) Hydroquinoid Chromium Complexes Bearing an Acyclic Conjugated Bridge: Chromium-Templated Synthesis, Molecular Structure, and Haptotropic Metal Migration, *Organometallics*, <u>29</u>, 6172–6185.

182. Ye, S. F.; Neese, F. (**2010**) Accurate Modeling of Spin-State Energetics in Spin-Crossover Systems with Modern Density Functional Theory, *Inorg. Chem,* 49, 772–774.

183. Orio, M.; Philouze, C.; Jarjayes, O.; Neese, F.; Thomas, F. (**2010**) Spin Interaction in Octahedral Zinc Complexes of Mono- and Diradical Schiff and Mannich Bases, *Inorg. Chem,* 49, 646–658.

184. Pantazis, D. A.; Orio, M.; Petrenko, T.; Zein, S.; Lubitz, W.; Messinger, J.; Neese, F. (**2009**) Structure of the Oxygen-Evolving Complex of Photosystem II: Information on the S_2 state through Quantum Chemical Calculation of its Magnetic Properties. *Phys. Chem. Chem. Phys.*, 11, 6788–6798.

185. Baffert, C.; Orio, M.; Pantazis, D. A.; Duboc, C.; Blackman, A.G.; Blondin, G.; Neese, F.; Deronzier,A.; Collomb, M-N. (**2009**) A trinuclear terpyridine frustrated spin system with a $Mn_3^{IV}O_4$ core: synthesis, physical characterization and quantum chemical modeling of its magnetic properties. *Inorg. Chem.*, 48, 10281–10288.

186. Liakos, D.; Neese, F. (**2009**) A multiconfigurational *ab initio* study of the zero-field splitting in the di- and trivalent hexaquo-chromium complexes. *Inorg. Chem.*, 48, 10572–10580.

187. Astashkin, A.V.; Klein, E.C.; Ganyushin, D.; Johnson.Winters, K.; Neese, F.; Kappler, U.; Enemark, J.H. (**2009**) Exchangeable oxygens in the vicinity of the molybdenum center of the high-pH form of sulfite oxidase and sulfite dehydrogenase. *Phys. Chem. Chem. Phys.*, 11, 6733–6742.

188. Orio, M.; Pantazis, D. A.; Petrenko, T.; Neese, F. (**2009**) Magnetic and spectroscopic properties of mixed valence manganese(III,IV) dimers: a systematic study using broken symmetry density functional theory, *Inorg. Chem.*, 48, 7251–7260.

189. Klein, E.L.; Astashkin, A.V.; Ganyushin, D.; Johnson-Winters, K.; Wilson, H.L.; Rajagopalan, K. V.; Neese, F.; Enemark, J.H. (**2009**) Direct Detection and Characterization of Chloride in the Active Site of the Low-pH Form of Sulfite Oxidase Using ESEEM Spectroscopy, Isotopic Labeling, and DFT Calculations, *Inorg. Chem.*, 48(11), 4743–4752.

190. Vancoillie, S.; Rulisek, L.; Neese, F.; Pierloot, K. (**2009**) Theoretical description of the structure and magnetic properties of nitroxide-Cu(II)-nitroxide spin triads, *J. Phys. Chem.*, 113, 6149–6157.

191. Cowley, R.E.; Bill, E.; Neese, F.; Brennessel,W.W.; Holland, P.L. (**2009**) Iron(II) Complexes With Redox-Active Tetrazene (RNNNNR) Ligands, *Inorg. Chem.*, 48, 4828–4836.

192. Gansäuer, A.; Fleckhaus, A.; Lafon, A.; Okkel, M.; Anakuthil, A.; Kotsis, K.; Neese, F. (**2009**) Catalysis via Homolytic Substitutions with C-O and Ti-O Bonds: Oxidative Additions and Reductive Eliminations in Single Electron Steps. *J. Am. Chem. Soc.*, 131, 16989–16999.

193. Ye, S.; Neese, F. (**2009**) Quantum Chemical Studies of C-H Activation Reactions by High-Valent Nonheme Iron Centers *Curr. Op. Chem. Biol.*, 13(1), 89–98.

194. Krahe, O.; Neese, F.; Streubel, R. (**2009**) The quest for ring-opening of oxaphosphirane complexes: a coupled cluster and density functional study of CH_3PO isomers and their $Cr(CO)_5$ complexes *Chem. Eur. J.*, 15, 2594–2601.

195. Romain, S.; Duboc, C.; Neese, F.; Riviere, E.; Hanton, L. R.; Blackman, A. G.; Philouze, C.; Lepretre, J. C.; Deronzier, A.; Collomb, M. N. (**2009**) An Unusual Stable Mononuclear Mn(III) Bis-terpyridine Complex Exhibiting Jahn-Teller Compression: Electrochemical Synthesis, Physical Characterisation and Theoretical Study, *Chem. Eur. J.*, 15, 980–988

196. Zein, S.; Neese, F. (**2008**) *Ab initio* and Coupled Perturbed DFT Calculation of Zero-Field Splittings in Mn(II) Transition Metal complexes. *J. Phys. Chem. A*, 112, 7976–7983.

197. Ye, S.; Tuttle, T.; Bill, E.; Gross, Z.; Thiel, W.; Neese, F. (**2008**) The Noninnocence of Iron Corroles: A combined Experimental and Quantum Chemical Study. *Chem. Eur. J.* (selected as very important paper), <u>34</u>, 10839–10851.

198. Duboc, C.; Collomb, M.-N.; Pecaut, J.; Deronzier, A.; Neese, F. (**2008**) Definition of Magneto-Structural Correlations for the Mn(II) Ion. *Chem. Eur. J.*, <u>21</u>, 6498–6509.

199. Berry, J.F.; DeBeer-George, S.; Neese, F. (**2008**) Electronic Structure and Spectroscopy of "Superoxidized" Iron Centers in Model Systems: Theoretical and Experimental Trends. *Phys. Chem. Chem. Phys.*, <u>10</u>, 4361–4374.

200. Sander, W.; Grote, D.; Kossmann, S.; Neese, F. (**2008**) 2.3.5.6-Tetrafluorophenylnitren-4-yl: EPR Spectroscopic Characterization of a Quartet Ground State Nitreno Radical, *J. Am. Chem. Soc.*, <u>130</u>, 4396–4403.

201. Scheifele, Q.; Riplinger, C.; Neese, F.; Weihe, H.; Barra, A.L.; Jurany, F.; Podlesnyak, A.; Tregenna-Piggot, P.W.L. (**2008**) Spectroscopic and Theoretical Study of a Mononuclear Mn(III) Bioinorganic Complex Exhibiting a Compressed Jahn-Teller Octahedron, *Inorg. Chem.*, <u>47</u>, 439–447.

202. Zein, S.; Kulik, L.V.; Yano, J.; Kern, J.; Zouni, A.; Yachandra, V.K.; Lubitz, W.; Neese, F.; Messinger, J. (**2008**) Focussing the View on Nature's Water Splitting Catalyst *Phil. Trans. Roy. Soc. London B*, <u>363</u>, 1167–1177.

203. Zein, S.; Duboc, C.; Lubitz, W.; Neese, F. (**2008**) Theoretical Characterization of zero-Field Splittings in Mn(II) Complexes. *Inorg. Chem.*, <u>47</u>, 134–142.

204. Parker, D.J.; Hammond, D.; Davies, E.S.; Garner, C.D.; Benisvy, L.; McMaster, J.; Wilson, C.; Neese, F.; Bothe, E.; Bittl, R.; Teutloff, C. (**2007**) A stable H-bonded *ortho*-Thioether Phenoxyl-Radical: A Chemical and Spectroscopic Analogue of •Tyr$_{272}$ in *apo*-Galactose Oxidase, *J. Biol. Inorg. Chem.* (Ed Stiefel memorial issue), <u>101</u>, 1859–1864.

205. Chlopek, K.; Muresan, N.; Neese, F.; Wieghardt, K. (**2007**) Electronic Structures of Five-Coordinate Complexes of Iron Containing Zero, One, or Two π Radical Ligands: A Broken Symmetry Density Functional Theoretical Study, *Chem. Eur. J.*, <u>13</u>, 8391–8403.

206. Muresan, N.; Chlopek, K.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2007**) Bis(α-diimine)nickel Complexes: Molecular and Electronic Structure of Three Members of the Electron-Transfer Series [Ni(L)$_2$]^z ($z = 0, 1+, 2+$) (L = 2-Phenyl-1,4-bis(isopropyl)-1,4-diazabutadiene). A Combined Experimental and Theoretical Study, *Inorg. Chem.*, <u>46</u>, 4905–4916.

207. Ray, K.; Petrenko, T.; Wieghardt, K.; Neese, F. (**2007**) Joint Spectroscopic and Theoretical Investigations of Transition Metal Complexes Involving Non-Innocent Ligands. *Dalton Trans.*, 1552 (selected for cover picture).

208. Sinnecker, S.; Svensen, N.; Barr, E.W.; Ye, S.; Bollinger, J.M.; Neese, F.; Krebs, C. (**2007**) Spectroscopic and Theoretical Evaluation of the Structure of the High-Spin Fe(IV)-Oxo Intermediates in Taurine:α-Ketoglutarate Dioxygenase from *Escherichia coli* and its His99Ala Ligand Variant *J. Am. Chem. Soc.*, <u>129</u>, 6168–6179.

209. Duboc, C.; Phoeung, T; Zein, S.; Pécaut, J.; Collomb, M.-N.; Neese. F. (**2007**) Origin of the zero field splitting in mononuclear dihalide Mn(II) complexes: an investigation by multifrequency high-field EPR and density functional theory (DFT), *Inorg. Chem.*, <u>46</u>, 4905–4916.

210. DeBeer-George, S.; Petrenko, T.; Aliaga-Alcade, N.; Bill, E.; Mienert, B.; Sturhan, W.; Ming, Y.; Wieghardt, K.; Neese, F. (**2007**) Characterization of a Genuine Iron(V)Nitrido Species by Nuclear Resonant Vibrational Spectroscopy Coupled to Density Functional Calculations, *J. Am. Chem. Soc.*, <u>129</u>, 11053–11060.

211. Lehnert, N.M; Cornelissen, U.; Neese, F.; Ono, T.; Noguchi, Y.; Okamoto, K.-I.; Fujisawa, K. (**2007**) Synthesis and Spectroscopic Characterization of Cu(II)-Nitrite Complexes with Hydrotris(pyrazolyl)borate and Related Ligands. *Inorg. Chem.*, 46, 3916–3933.

212. Carmieli, R.; Larsen, T.; Reed, G.H.; Zein, S.; Neese, F.; Goldfarb, D. (**2007**) The Catalytic Mn^{2+} Sites in the Enolase-Inhibitor Complex - Crystallography, Single Crystal EPR and DFT calculations. *J. Am. Chem. Soc.*, 129, 4240–4252.

213. Kokatam, S.; Ray, K.; Pap, J.; Bill, E.; Geiger, W.E.; LeSuer, R.J.; Rieger, P.H.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2007**) Molecular and Electronic Structure of Square Planar Gold Complexes Containing Two 1,2-di(4-*tert*-butylphenyl)ethylene-1,2-dithiolato Ligands: $[Au(L)_2]^{1+/0/1-/2-}$. A Combined Experimental and Computational Study, *Inorg. Chem.*, 46, 1100–1111.

214. Ray, K.; DeBeer-George, S.; Solomon, E.I.; Wieghardt, K.; Neese, F. (**2007**) Description of the Ground State Covalencies of the Bis(dithiolato)Transition Metal Complexes Using X-ray Absorption Spectral and Time-Dependent-Density-Functional Studies. *Chem. Eur. Journal*, 13(10), 2753 (selected for cover picture).

215. Chalupský, J.; Neese, F.; Solomon, E.I.; Ryde, U.; Rulíšek, L. (**2006**) Identification of intermediates in the reaction cycle of multicopper oxidases by quantum chemical calculations of spectroscopic parameters, *Inorg. Chem.*, 45, 11051–11059.

216. Bart, S.C.; Chłopek, K.; Bill, E., Bouwkamp, B.W.; Lobkovsky, E.; Neese, F.; Wieghardt, K.; Chirik, P.J. (**2006**) Electronic Structure of Bis(imino)pyridine Iron Dichloride, Monochloride and Neutral Ligand Complexes: A Combined Structural, Spectroscopic and Computational Study, *J. Am. Chem. Soc.*, 128, 13901–13912.

217. Patra, A.K.; Bill, E.; Bothe, E.; Chlopek, K.; Neese, F.; Weyhermüller, T.; Stobie, K.; Ward, M.D.; McCleverty, J.A.; Wieghardt, K. (**2006**) The Electronic Structure of Mononuclear Bis(1,2-diaryl-1,2-ethylenedithiolate)iron Complexes Containing a Fifth Cyanide or Phosphite Ligand: A Combined Experimental and Computational Study, *Inorg. Chem.*, 45, 7877–7890.

218. Berry, J.F.; Bill, E.; Bothe, E.; DeBeer-George, S.; Mienert, B.; Neese, F.; Wieghardt, K. (**2006**) An Octahedral Coordination Complex of Iron(VI) – One Step Ahead of Nature?, *Science*, 312, 1937–1941.

219. Petrenko, T.; Ray, K.; Wieghardt, K.; Neese, F. (**2006**) Vibrational Markers for the Open-Shell Character of Metal bis-Dithiolenes: An Infrared, resonance Raman and Quantum Chemical Study. *J. Am. Chem. Soc.*, 128, 4422–4436.

220. Chłopek, K.; Bothe, E.; Neese, F.; Weyhermüller, T.; Wieghardt, K. (**2006**) The Molecular and Electronic Structures of Tetrahedral Complexes of Nickel and Cobalt Containing N, N'-Disubstituted, Bulky o-Diiminobenzosemiquinonate(1-) π-Radical Ligands, *Inorg. Chem.*, 45, 6298–6307.

221. Kababya, S.; Nelson. J.; Calle, C.; Neese, F.; Goldfarb, D. (**2006**) The electronic structure of bi-nuclear mixed valent copper azacryptates derived from integrated advanced EPR and DFT calculations. *J. Am. Chem. Soc.*, 128, 2017–2029.

222. Berry, J.F.; Bill, E.; Neese, F.; Garcia-Serres, R.; Weyhermüller, T.; Wieghardt, K. (**2006**) Effect of N-Methylation of Macrocyclic Amine Ligands on the Spin State of Fe(III): A Tale of Two Fluoro Complexes. *Inorg. Chem.*, 45, 2027–2037.

223. Kapre, R.; Ray, K.; Sylvestre, I.; Weyhermüller, T.; DeBeer-George, S.; Neese, F.; Wieghardt, K. (**2006**) The Molecular and Electronic Structure of Oxo-bis(benzene-1,2-dithiolato)chromate(V) Monoanions. A Combined Experimental and Density Functional Study. *Inorg. Chem.*, 45, 3499–3509.

224. Zhu, W.; Marr, A.C.; Wang, Q.; Neese, F.; Spencer, J.E.; Blake, A.J.; Cooke, P.A.; Wilson, C.; Schröder, M. (**2005**) Modulation of the Electronic Structure and the Ni-Fe Distance in Heterobimetallic Models for the Active Site in [NiFe]Hydrogenase: Is there a Ni-Fe Bond? *Proc. Natl. Acad. Sci. (USA)*, <u>102</u>, 18280–18285.

225. Astashkin, A.V.; Neese, F.; Raitsimaring, A.M.; Cooney, J.J.A.; Bultman, E.; Enemark, J.H. (**2005**) Pulsed EPR investigation of systems modelling molybdenum enzymes: hyperfine and quadrupole parameters of oxo-^{17}O in [Mo^{17}O(SPh)$_4$]$^-$, *J. Am. Chem. Soc.*, <u>127</u>, 16713–16722.

226. Benisvy, L.; Bittl, R.; Bothe, E.; Garner, C.D.; McMaster, J.; Ross, S.; Teutloff, C.; Neese, F. (**2005**) Phenoxyl Radicals Hydrogen-Bonded to Imidazolium – Analogues of Tyrosyl D$^\bullet$ of Photosystem II: High-Field EPR and DFT Studies. *Angew. Chem. Int. Ed.*, <u>44</u>, 5314–5317.

227. Praneeth, V.K.K.; Neese, F.; Lehnert, N. (**2005**) Spin Density Distribution in Five- and Six-Coordinate Iron(II)-Porphyrin NO Complexes Evidenced by Magnetic Circular Dichroism Spectroscopy. *Inorg. Chem.*, <u>44</u>, 2570–2572.

228. Sinnecker, S.; Neese, F.; Lubitz, W. (**2005**) Dimanganese Catalase – Spectroscopic Parameters from Broken Symmetry Density Functional Theory of the Superoxidized Mn^{III}/Mn^{IV} state, *J. Biol. Inorg. Chem.*, <u>10</u>, 231–238.

229. Blanchard, S.; Neese, F.; Bothe, E.; Bill, E.; Weyhermüller, T.; Wieghardt, K. (**2005**) Square Planar vs. Tetrahedral Coordination in Diamagnetic Complexes of Nickel(II) Containing Two Bidentate π Radical Monoanions, *Inorg. Chem.*, <u>44</u>, 3636–3656.

230. Mader-Cosper, M.; Neese, F.; Astashkin, A.V.; Carducci, M.A.; Raitsimring, A.M.; Enemark, J.H. (**2005**) Determination of the Magnitude and Orientation of the g-Tensors for *cis,trans*-(L-N$_2$S$_2$)Mo^{V}OX (X=Cl, SCH$_2$Ph) by Single Crystal EPR and Molecular Orbital Calculations, *Inorg. Chem.*, <u>44</u>, 1290–1301.

231. Fouqeau, A.; Casida, M.E.; Lawson, L.M.; Hauser, A.; Neese, F. (**2005**) Comparison of Density Functionals for Energy and Structural Differences Between the High-[^{5}T$_{2g}$: (t$_{2g}^4$)(e$_g^2$)] and Low-[^{1}A$_{1g}$: (t$_{2g}^6$)(e$_g^0$)] Spin States of Iron(II) Coordination Compounds: II. Comparison of Results for More than Ten Modern Functionals with Ligand Field Theory and *Ab Initio* Results for Hexaquoferrous Dication, [Fe(H$_2$O)$_6$]$^{2+}$ and Hexaminoferrous Dication [Fe(NH$_3$)$_6$]$^{2+}$, *J. Chem. Phys.*, <u>122</u>, 044110.

232. Aliaga-Alcade, N.; DeBeer George, S.; Bill, E.; Wieghardt, K.; Neese, F. (**2005**) The Geometric and Electronic Structure of [(Cyclam-acetato)Fe(N)]+: a Genuine Iron(V) Species with Ground State Spin $S = 1/2$. *Angew. Chem. Int. Ed.*, <u>44</u>, 2908–2912.

233. Bill, E.; Bothe, E.; Chaudhuri, P.; Chlopek, K.; Herebian, D.; Kokatam, S.; Ray, K. Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2004**) Molecular and Electronic Structure of Four- and Five-Coordinate Cobalt Complexes Containing Two *o*-Phenylendiamine- or Two *o*-Aminophenol-Type Ligands at Various Oxidation Levels: An Experimental, Density Functional and Correlated *ab initio* Study. *Chem. Eur. J.*, <u>11</u>, 204–224.

234. Paine, T.; Bothe, W.; Bill, E.; Weyhermüller, T.; Slep, L.; Neese, F.; Chaudhuri, P. (**2004**) Nonoxo Vanadium(IV) and Vanadyl(V) Complexes with Mixed O,X,O-Donor Ligand (X = S, Se, P, PO), *Inorg. Chem.*, <u>43</u>, 7324–7338.

235. Baute, D.; Arieli, D.; Zimmermann, H.; Neese, F.; Weckhuysen, B.; Goldfarb, D. (**2004**) The Structure of Copper Histidine Complexes in Solution and in Zeolite Y: A Combined X- and W-Band Pulsed EPR/ENDOR and DFT Study, *J. Am. Chem. Soc.*, <u>126</u>, 11733–11745.

236. Garcia Serres R.; Grapperhaus, C.A.; Bothe, E.; Bill, E.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2004**) Structural, Spectroscopic and Computational Study of an Octahedral, Non-heme {FeNO}^{6,7,8} Series: [Fe(NO)(cyclam-ac)]$^{2+/1+/0}$, *J. Am. Chem. Soc.*, <u>126</u>, 5138–5153.

237. Sinnecker, S.; Noodleman, L.; Neese, F.; Lubitz, W. (**2004**) Calculation of the EPR Parameters of a Mixed Valence Mn(III)/Mn(IV) Model Complex with Broken Symmetry Density Functional Theory. *J. Am. Chem. Soc.*, <u>126</u>, 2613–2622.

238. Sinnecker, S.; Neese, F.; Lubitz, W. (**2004**) Benzosemichinone Solvent Interactions. A Density Functional Study of Electric and Magnetic Properties for Probing Hydrogen Bond Strengths and Geometries. *J. Am. Chem. Soc.*, <u>126</u>, 3280–3290.

239. van Gastel, M.; Fichtner, C.; Neese, F.; Lubitz, W. (**2005**) EPR Experiments to Elucidate the Structure of the Ready and Unready States of the [NiFe] Hydrogenase of *Desulfovibrio vulgaris* Miyazaki F. *Biochem. Soc. Trans.*, <u>33</u>, 7–11.

240. van Gastel, M.; Lassman, G.; Lubitz, W.; Neese, F. (**2004**) The unusual EPR parameters of the cysteine radical: a DFT and correlated *ab initio* study *J. Am. Chem. Soc.*, <u>126</u>, 2237–2246.

241. Fouqueau, A.; Mer, S.; Casida, M.E.; Daku, L.M.L.; Hauser, A.; Mieva, T.; Neese, F. (**2004**) Comparison of Density Functionals for Energy and Structural Differences between the High $[^5T_{2g}: t_{2g}^4 e_g^2]$ and Low $[^1A_{1g}: t_{2g}^6]$ Spin States of the Hexaquo-Ferrous Ion, $[Fe(H_2O)_6]^{2+}$, *J. Chem. Phys.*, <u>120</u>, 9473–9486.

242. Slep, L.D.; Mijovilovich, A.; Meyer-Klaucke, W.; Weyhermüller, T.; Bill, E.; Bothe, E.; Neese, F.; Wieghardt, K. (**2003**) The Mixed-valent $\{Fe^{IV}(\mu\text{-}O)(\mu\text{-carboxylato})_2Fe^{III}\}^{3+}$ Core. *J. Am. Chem. Soc.*, <u>125</u>, 15554–15570.

243. Herebian, D.; Wieghardt, K.; Neese, F. (**2003**) Analysis and Interpretation of Metal-Radical Coupling in a Series of Square Planar Nickel Complexes. Correlated *Ab Initio* and Density Functional Investigation of $[Ni(L^{ISQ})_2]$ (L^{ISQ} =3,5-di-tert-butyl-*o*diiminobenzosemquinone). *J. Am. Chem. Soc.*, <u>125</u>, 10997–11005.

244. Herebian, D.; Bothe, E.; Neese, F.; Weyhermüller, T.; Wieghardt, K. (**2003**) The Molecular and Electronic Structures of Bis(*o*-diiminobenzosemiquinonato)metal(II) Complexes (Ni, Pd, Pt), their Monocations and Anions, and their Dimeric Dications Containing Weak Metal-Metal Bonds. *J. Am. Chem. Soc.*, <u>125</u>, 9116–9128.

245. Ghosh, P.; Bill, E.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2003**) The non-Innocence of the Ligand Glyoxal-bis (2-mercaptoanil). The Electronic Structures of $[Fe(gma)]_2$, $[Fe(gma)(py)]^{\bullet}py$, $[Fe(gma)(CN)]^{1-/0}$, $[Fe(gma)I]$, $[Fe(gma)(PR_3)_n]$ ($n = 1, 2$). Experimental and Theoretical Evidence for 'Excited State' Coordination. *J. Am. Chem. Soc.*, <u>125</u>, 1293–1308.

246. Einsle, O.; Messerschmidt, A.; Huber, R.; Kroneck, P.M.H.; Neese, F. (**2002**) Mechanism of the Six Electron Reduction of Nitrite to Ammonia by Cytochrome *c* Nitrite Reductase (CCNIR). *J. Am. Chem. Soc.*, <u>124</u>, 11737–11745.

247. Sun, X.; Chun, H.; Hildenbrand, K.; Bothe, E.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2002**) *o*-Iminobenzosemiquinonato(1-) and *o*-Amidophenolato(2-) Complexes of Palladium(II) and Plantinum(II): A Combined Experimental and Density Functional Theoretical Study, *Inorg. Chem.*, <u>41</u>, 4295–4303.

248. Li, M.; Bonnet, D.; Bill, E.; Neese, F.; Weyhermüller, T.; Blum, N.; Sellmann, D.; Wieghardt, K. (**2002**) Tuning the Electronic Structure of Octahedral Iron Complexes [FeL(X)] (L = 1-alkyl-4,7-bis(4-tert-butyl-2-mercaptobenzyl)-1,4,7-triazacyclo-nonane, X = Cl, CH₃O, CN, CO). The $S = 1/2 \Leftrightarrow S = 3/2$ Spin-Equilibrium of $[FeL^{Pr}(NO)]$. *Inorg. Chem.*, <u>41</u>, 3444–3456.

249. Lehnert, N.; Neese, F.; Ho, R.Y.N.; Que Jr., L.; Solomon, E.I. (**2002**) Electronic Structure and Reactivity of Low-Spin Fe(III)-Hydroperoxo Complexes: Comparison to Activated Bleomycin. *J. Am. Chem. Soc.*, <u>124</u>, 10810–10822.

250. Grapperhaus, C.A.; Bill, E.; Weyhermüller, T.; Neese, F.; Wieghardt, K. (**2001**) Electronic and Geometric Structure and Spectroscopy of a High Valent Manganese(V) Nitrido Complex. An Experimental and DFT Study. *Inorg. Chem.*, <u>41</u>, 4191–4198.

251. Neese, F., Solomon, E.I. (**1998**) Detailed Spectroscopic and Theoretical Studies on [Fe(EDTA)(O$_2$)]$^{3-}$: the Electronic Structure of the Side-On Ferric Peroxide Bond and its Relevance to Reactivity. *J. Am. Chem. Soc.*, <u>120</u>, 12829–12848.

Some Reviews of interest:

252. Neese, F.; Liakos, D. G.; Ye, S. F. (**2011**) Correlated Wavefunction Methods in Bioinorganic Chemistry, *J. Biol. Inorg. Chem,* <u>16</u>, 821–829.

253. Neese, F.; Ames, W.; Christian, G.; Kampa, M.; Liakos, D. G.; Pantazis, D. A.; Roemelt, M.; Surawatanawong, P.; Ye, S. F. (**2010**) Dealing with Complexity in Open-Shell Transition Metal Chemistry from a Theoretical Perspective: Reaction Pathways, Bonding, Spectroscopy, and Magnetic Properties, *Adv. Inorg. Chem.*, <u>62</u>, 301–349.

254. Orio, M.; Pantazis, D. A.; Neese, F. (**2009**) Density Functional Theory, *Photosynth. Res.*, <u>102</u>, 443–453.

255. Neese, F. (**2009**), Density Functional Theory and EPR Spectroscopy: a guided tour. *EPR Newsletter*, <u>18</u>(4), Pro & Contra section.

256. Neese, F. (**2009**) Prediction of Molecular Spectra and Molecular Properties with Density Functional Theory: from Fundamental Theory to Exchange Coupling. *Coord. Chem. Rev.*, <u>253</u>, 526–563.

257. Neese, F. (**2009**) Spin Hamiltonian Parameters from First Principle Calculations: Theory and Application. In: Hanseon, G.; Berliner, L. (Eds.) *Biological Magnetic Resonance. Vol 28*, pp 175–232.

258. Ray, K.; Petrenko, T.; Wieghardt, K.; Neese, F. (**2007**) Joint Spectroscopic and Theoretical Investigations of Transition Metal Complexes Involving Non-Innocent Ligands. *Dalton Trans.*, 1552. (selected for cover picture)

259. Kirchner, B.; Wennmohs, F.; Ye, S.; Neese, F. (**2007**) Theoretical Bioinorganic Chemistry: Electronic Structure Makes a Difference, *Curr. Op. Chem. Biol.*, <u>11</u>, 131–141.

260. Neese, F.; Petrenko, T.; Ganyushin, D.; Olbrich, G. (**2007**) Advanced Aspects of *ab initio* Theoretical Spectroscopy of Open-Shell Transition Metal Ions. *Coord. Chem. Rev.*, <u>205</u>, 288–327.

261. Ye, S.; Neese, F. (**2006**) Combined Quantum Chemical and Spectroscopic Studies on Transition Metal Complexes with Coordinating Radicals. *Chemtracts* (Special Volume on Computational Inorganic Chemistry), <u>19</u>, 77–86.

262. Sinnecker, S.; Neese, F. (**2006**) Theoretical Bioinorganic Spectroscopy, Invited Chapter in the Series *Current Topics in Chemistry*, Editor M. Reiher, Springer, Heidelberg.

263. Neese, F. (**2006**) Quantum Chemical Approaches to Spin-Hamiltonian Parameters. *Specialist Periodical Reports on EPR Spectroscopy Vol. 20*, (Ed. B. Gilbert) Royal Scoiety Press.

264. Neese, F. (**2006**) A Critical Evaluation of DFT, including Time-Dependent DFT, Applied to Bioinorganic Chemistry. *J. Biol. Inorg. Chem.*, (commentary on invitation), <u>11</u>, 702–711.

265. Neese, F.; Munzarova, M.L. (**2004**) Historical Aspects of EPR Parameter Calculations. In: Kaupp, M.; Bühl, M.; Malkin, V. (Eds) *Calculation of NMR and EPR Parameters. Theory and Applications.* Wiley-VCH, pp 21–32.

266. Neese, F. (**2004**) Zero-Field Splitting. In: Kaupp, M.; Bühl, M.; Malkin, V. (Eds) *Calculation of NMR and EPR Parameters. Theory and Applications.* Wiley-VCH, pp 541–564.

267. Neese, F. (**2004**) Application of EPR Parameter Calculations in Bioinorganic Chemistry. In: Kaupp, M.; Bühl, M.; Malkin, V. (Eds) *Calculation of NMR and EPR Parameters. Theory and Applications.* Wiley-VCH, pp 581–591.

268. Neese, F. (**2003**) Quantum Chemical Calculations of Spectroscopic Properties of Metalloproteins and Model Compounds: EPR and Mössbauer Properties. *Curr. Op. Chem. Biol.*, <u>7</u>, 125–135.

269. Neese, F.; Solomon, E.I. (**2003**) Calculation and Interpretation of Spin-Hamiltonian Parameters in Transition Metal Complexes. Invited review, (Wiley series: Magnetoscience – From Molecules to Materials edited by J.S. Miller and M. Drillon), Volume IV, p 345–466.

Bibliography

[1] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006.

[2] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. *J. Chem. Phys.*, 132:154104, 2010.

[3] Jingjing Zheng, Xuefei Xu, and Donald G Truhlar. *Theor. Chem. Acc.*, 128:295–305, 2010.

[4] Dmitrij Rappoport and Filipp Furche. *J. Chem. Phys.*, 133:134105, 2010.

[5] D. A. Pantazis, X.-Y. Chen, C. R. Landis, and F. Neese. *J. Chem. Theory Comput.*, 4:908–919, 2008.

[6] M. Buehl, C. Reimann, D. A. Pantazis, T. Bredow, and F. Neese. *J. Chem. Theory Comput.*, 4:1449–1459, 2008.

[7] D. A. Pantazis and F. Neese. *J. Chem. Theory Comput.*, 5:2229–2238, 2009.

[8] D. A. Pantazis and F. Neese. *J. Chem. Theory Comput.*, 7:677–684, 2011.

[9] D. A. Pantazis and F. Neese. *Theor. Chem. Acc.*, 131:1292, 2012.

[10] P. J. Hay and W. R. Wadt. *J. Chem. Phys.*, 82:270 & 284 & 299, 1985.

[11] P. Fuentealba, H. Preuss, H. Stoll, and L. von Szentpaly. *Chem. Phys. Lett.*, 89:418–422, 1982.

[12] L. von Szentpaly, P. Fuentealba, H. Preuss, and H. Stoll. *Chem. Phys. Lett.*, 93:555–559, 1982.

[13] P. Fuentealba, H. Stoll, L. von Szentpaly, P. Schwerdtfeger, and H. Preuss. *J. Phys. B*, 16:L323, 1983.

[14] H. Stoll, P. Schwerdtfeger P. Fuentealba, J. Flad, L. von Szentpaly, and H. Preuss. *J. Chem. Phys.*, 81:2732–2736, 1984.

[15] P. Fuentealba, L. von Szentpaly, H. Preuss, and H. Stoll. *J. Phys. B*, 18:1287, 1985.

[16] M. Dolg, U. Wedig, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 86:866–872, 1987.

[17] G. Igel-Mann, H. Stoll, and H. Preuss. *Mol. Phys.*, 65:1321–1328, 1988.

[18] M. Dolg, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 90:1730–1734, 1989.

[19] P. Schwerdtfeger, M. Dolg, W. H. E. Schwarz, G. A. Bowmaker, and P. D. W. Boyd. *J. Chem. Phys.*, 91:1762–1774, 1989.

[20] M. Dolg, H. Stoll, A. Savin, and H. Preuss. *Theor. Chim. Acta*, 75:173–194, 1989.

[21] D. Andrae, U. Haeussermann, M. Dolg, H. Stoll, and H. Preuss. *Theor. Chim. Acta*, 77:123–141, 1990.

[22] M. Kaupp, P. v. R. Schleyer, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 94:1360–1366, 1991.

[23] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. *Mol. Phys.*, 74:1245–1263, 1991.

[24] M. Dolg, P. Fulde, W. Küchle, C.-S. Neumann, and H. Stoll. *J. Chem. Phys.*, 94:3011–3017, 1991.

[25] M. Dolg, H. Stoll, H.-J. Flad, and H. Preuss. *J. Chem. Phys.*, 97:1162–1173, 1992.

[26] A. Bergner, M. Dolg, W. Küchle, H. Stoll, and H. Preuss. *Mol. Phys.*, 80:1431–1441, 1993.

[27] M. Dolg, H. Stoll, H. Preuss, and R. M. Pitzer. *J. Phys. Chem.*, 97:5852–5859, 1993.

[28] M. Dolg, H. Stoll, and H. Preuss. *Theor. Chim. Acta*, 85:441–450, 1993.

[29] U. Haeussermann, M. Dolg, H. Stoll, and H. Preuss. *Mol. Phys.*, 78:1211–1224, 1993.

[30] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 100:7535–7542, 1994.

[31] A. Nicklass, M. Dolg, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 102:8942–8952, 1995.

[32] T. Leininger, A. Nicklass, H. Stoll, M. Dolg, and P. Schwerdtfeger. *J. Chem. Phys.*, 105:1052–1059, 1996.

[33] T. Leininger, A. Nicklass, W. Küchle, H. Stoll, M. Dolg, and A. Bergner. *Chem. Phys. Lett.*, 255:274, 1996.

[34] T. Leininger, A. Berning, A. Nicklass, H. Stoll, H.-J. Werner, and H.-J. Flad. *Chem. Phys.*, 217:19, 1997.

[35] F. Schautz, H.-J. Flad, and M. Dolg. *Theor. Chem. Acc.*, 99:231, 1998.

[36] Y. Wang and M. Dolg. *Theor. Chem. Acc.*, 100:124, 1998.

[37] B. Metz, H. Stoll, and M. Dolg. *J. Chem. Phys.*, 113:2563, 2000.

[38] B. Metz, M. Schweizer, H. Stoll, M. Dolg, and W. Liu. *Theor. Chem. Acc.*, 104:22, 2000.

[39] J. M. L. Martin and A. Sundermann. *J. Chem. Phys.*, 114:3408, 2001.

[40] X. Cao and M. Dolg. *J. Chem. Phys.*, 115:7348, 2001.

[41] H. Stoll, B. Metz, and M. Dolg. *J. Comput. Chem.*, 23:767, 2002.

[42] X. Cao and M. Dolg. *J. Mol. Struct. (Theochem)*, 581:139, 2002.

[43] K. A. Peterson. *J. Chem. Phys.*, 119:11099, 2003.

[44] K. A. Peterson, D. Figgen, E. Goll, H. Stoll, and M. Dolg. *J. Chem. Phys.*, 119:11113, 2003.

[45] D. Figgen, G. Rauhut, M. Dolg, and H. Stoll. *Chem. Phys.*, 311:227, 2005.

[46] I. S. Lim, P. Schwerdtfeger, B. Metz, and H. Stoll. *J. Chem. Phys.*, 122:104103, 2005.

[47] K. A. Peterson and C. Puzzarini. *Theor. Chem. Acc.*, 114:283, 2005.

[48] J. Yang and M. Dolg. *Theor. Chem. Acc.*, 113:212, 2005.

[49] I. S. Lim, H. Stoll, and P. Schwerdtfeger. *J. Chem. Phys.*, 124:034107, 2006.

[50] K. A. Peterson, B. C. Shepler, D. Figgen, and H. Stoll. *J. Phys. Chem. A*, 110:13877, 2006.

[51] K. A. Peterson, D. Figgen, M. Dolg, and H. Stoll. *J. Chem. Phys.*, 126:124101, 2007.

[52] A. Moritz, X. Cao, and M. Dolg. *Theor. Chem. Acc.*, 117 & 118:473 & 845, 2007.

[53] A. Moritz and M. Dolg. *Theor. Chem. Acc.*, 121:297, 2008.

[54] M. Huelsen, A. Weigand, and M. Dolg. *Theor. Chem. Acc.*, 122:23, 2009.

[55] D. Figgen, K. A. Peterson, M. Dolg, and H. Stoll. *J. Chem. Phys.*, 130:164108, 2009.

[56] A. Weigand, X. Cao, J. Yang, and M. Dolg. *Theor. Chem. Acc.*, 126:117–127, 2010.

[57] R. Flores-Moreno, R. J. Alvares-Mendez, A. Vela, and A. M. Köster. *J. Comput. Chem.*, 27:1009, 2006.

[58] L. E. Roy, J. Hay, and R. L. Martin. *J. Chem. Theor. Comput.*, 4:1029, 2008.

[59] S. Grimme. *J. Chem. Phys.*, 118:9095–9102, 2003.

[60] T. J. Lee and P. R. Taylor. *Int. J. Quant. Chem. Symp*, 23:199, 1989.

[61] F. Wennmohs and F. Neese. *Chem. Phys.*, 343:217–230, 2008.

[62] D. P. Chong and S. R. Langhoff. *J. Chem. Phys.*, 84:5606, 1986.

[63] R. Ahlrichs, P. Scharf, and C. Ehrhardt. *J. Chem. Phys.*, 82:890, 1985.

[64] R. Ahlrichs and P. Scharf. *Ab Initio Methods in Quantum Chemistry Part I.* Advances in Chemical Physics. Wiley, 1987.

[65] D. G. Liakos and F. Neese. *J. Phys. Chem. A*, 116:4801–4816, 2012.

[66] P. Pulay, S. Saebo, and W. Meyer. *J. Chem. Phys.*, 81:1901, 1984.

[67] M. Schütz and H. J. Werner. *Chem. Phys. Lett.*, 318:370, 2000.

[68] M. Schütz and H. J. Werner. *J. Chem. Phys.*, 114:661, 2001.

[69] F. Neese, F. Wennmohs, A. Hansen, and U. Becker. *Chem. Phys.*, 356:98–109, 2009.

[70] S. Kossmann and F. Neese. *Chem. Phys. Lett.*, 481:240–243, 2009.

[71] S. Grimme. *J. Chem. Phys.*, 124:034108, 2006.

[72] T. Schwabe and S. Grimme. *Phys. Chem. Chem. Phys.*, 8:4398, 2006.

[73] F. Neese, T. Schwabe, and S. Grimme. *J. Chem. Phys.*, 126:124115, 2007.

[74] S. Grimme, S. Ehrlich, and L. Goerigk. *J. Comput. Chem.*, 32:1456, 2011.

[75] S. Grimme. *J. Comput. Chem.*, 25:1463, 2004.

[76] H. Iikura, T. Tsuneda, T. Yanai, and K. Hirao. *J. Chem. Phys.*, 115:3540–3544, 2001.

[77] T. Yanai, D. P. Tew, and N. C. Handy. *Chem. Phys. Lett.*, 393:51–57, 2004.

[78] Y. Tawada, T. Tsuneda, S. Yanagisawa, T. Yanai, and K. Hirao. *J. Chem. Phys.*, 120:8425–8433, 2004.

[79] J.-D. Chai and M. Head-Gordon. *J. Chem. Phys.*, 128:084106, 2008.

[80] Y.-S. Lin, G.-D. Li, S.-P. Mao, and J.-D. Chai. *J. Chem. Theory Comput.*, 9:263–272, 2013.

[81] E. van Lenthe, E. J. Baerends, and J. G. Snijders. *The Journal of Chemical Physics*, 99(6):4597–4610, 1993.

[82] C. van Wüllen. *J. Chem. Phys.*, 109:392–399, 1998.

[83] H. B. Schlegel in: Lawley (Ed.). *Ab initio methods in quantum chemistry-I.* John Wiley and Sons, 1987.

[84] H. B. Schlegel in: D. Yarkony (Ed). *Modern Electronic Structure Theory.* World Scientific, Singapore, 1995.

[85] H. B. Schlegel in: P. v. R. Schleyer (Ed.). *Encyclopedia of Computational Chemistry.* John Wiley and Sons, 1998.

[86] F. Eckert, P. Pulay, and H. J. Werner. *J. Comput. Chem.*, 12:1473, 1997.

[87] H. Horn, H. Wei, M. Häser, M. Ehrig, and R. Ahlrichs. *J. Comput. Chem.*, 12:1058, 1991.

[88] J. Baker. *J. Comput. Chem.*, 7:385, 1986.

[89] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl. *J. Chem. Theory Comput.*, 4:435, 2008.

[90] J. N. Harvey, M. Aschi, H. Schwarz, and W. Koch. *Theor. Chem. Acc.*, 99:95, 1998.

[91] H. Li and J. H. Jensen. *Theor. Chem. Acc.*, 107:211, 2002.

[92] F. Neese. *Inorg. Chim. Acta*, 337C:181–192, 2002.

[93] H. C. Andersen William C. Swope, P. H. Berens, and K. R. Wilson. *J. Chem. Phys.*, 76:648–649, 1982.

[94] L. Verlet. *Phys. Rev.*, 159:98–103, 1967.

[95] R. Hockney. *Methods Comp. Phys.*, 9:136–211, 1970.

[96] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory.* Dover Publications, 1989.

[97] W. Koch and M. C. Holthausen. *A Chemist's Guide to Density Functional Theory.* Wiley-VCH, 2000.

[98] R. G. Parr. *Density Functional Theory of Atoms and Molecules.* International Series of Monographs on Chemistry. Oxford University Press, 1994.

[99] A. D. Becke. *J. Chem. Phys.*, 98:1372, 1993.

[100] A. D. Becke. *J. Chem. Phys.*, 98:5648, 1993.

[101] S. H. Vosko, L. Wilk, and M. Nusair. *Can. J. Phys*, 58:1200, 1980.

[102] J. P. Perdew and Y. Wang. *Phys. Rev. B*, 45:13245, 1992.

[103] C. Lee, W. Yang, and R. G. Parr. *Phys. Rev. B.*, 37:785, 1988.

[104] R. Ahlrichs, M. Bär, H. P. Baron, R. Bauernschmitt, S. Böcker, M. Ehrig, K. Eichkorn, S. Elliott, F. Furche, F. Haase, M. Häser, H. Horn, C. Huber, U. Huniar, M. Kattanek, C. Kölmel, M. Kollwitz, K. May, C. Ochsenfeld, H. Öhm, A. Schäfer, U. Schneider, O. Treutler, M. von Arnim, F. Weigend, P. Weis, and H. Weiss. *TurboMole - Program System for ab initio Electronic Structure Calculations, Version 5.2.* Universität Karlsruhe, Karlsruhe, Germany, 2000.

[105] R. Ahlrichs in: P. v. R. Schleyer (Ed.). *Encyclopedia of Computational Chemistry.* John Wiley and Sons, 1998.

[106] R. Ahlrichs, M. Bär, M. Häser, H. Horn, and C. Kölmel. *Chem. Phys. Lett.*, 162:165, 1989.

[107] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, Jr., R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, A. G. Baboul, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, J. L. Andres, C. Gonzalez, M. Head-Gordon, E. S. Replogle, and J. A. Pople. *Gaussian 98, Revision A.8.* Gaussian, Inc., Pittsburgh PA, 1998.

[108] R. H. Hertwig and W. Koch. *Chem. Phys. Lett.*, 268:345, 1997.

[109] M. Ernzerhof in: D. P. Joubert (Ed.). *Density Functionals: Theory and Applications. Vol. 500 of Lecture Notes in Physics.* Springer Verlag, 1998.

[110] C. Adamo, A. di Matteo, and V. Barone. *Adv. Quant. Chem.*, 36:45, 2000.

[111] C. Adamo and V. Barone. *J. Chem. Phys.*, 110:6158, 1999.

[112] A. D. Becke. *Chem. Phys.*, 88:2547, 1988.

[113] O. Treutler and R. J. Ahlrichs. *Chem. Phys.*, 102:346, 1994.

[114] P. M. Boerrigter, G. Te Velde, and E. J. Baerends. *Int. J. Quant. Chem.*, 33:87, 1988.

[115] E. J. Baerends, D. E. Ellis, and P. Ros. *Chem. Phys.*, 2:41, 1973.

[116] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin. *J. Chem. Phys.*, 71:3396, 1979.

[117] C. Van Alsenoy. *J. Comput. Chem.*, 9:620, 1988.

[118] R.A. Kendall and H.A. Früchtl. *Theor. Chem. Acc.*, 97:158, 1997.

[119] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs. *R. Chem. Phys. Lett.*, 240:283, 1995.

[120] K. Eichkorn, F. Weigend, O. Treutler, and R. Ahlrichs. *Theor. Chem. Acc.*, 97:119, 1997.

[121] J. L. Whitten. *J. Chem. Phys.*, 58:4496, 1973.

[122] G. Te Velde and E. J. Baerends. *J. Comp. Phys.*, 99:84, 1992.

[123] C. W. Murray, N. C. Handy, and G. J. Laming. *Mol. Phys.*, 78:997, 1993.

[124] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch. *Chem. Phys. Lett.*, 257:213, 1996.

[125] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 220:377, 1994.

[126] P. M. W. Gill, B. G. Johnson, and J. A. Pople. *Chem. Phys. Lett.*, 209:506, 1993.

[127] Y. C. Zheng and J. Almlöf. *Chem. Phys. Lett.*, 214:397, 1993.

[128] Y. C. Zheng and J. Almlöf. *J. Mol. Struct. (Theochem)*, 388:277, 1996.

[129] G. Berghold, J. Hutter, and M. Parinello. *Theor. Chem. Acc.*, 99:344, 1998.

[130] M. Krack and A. M. Köster. *J. Chem. Phys.*, 108:3226, 1998.

[131] V. I. Lebedev. *Zh. Vychisl. Mat. Fiz.*, 15:48, 1975.

[132] V. I. Lebedev. *Zh. Vychisl. Mat. Fiz.*, 16:293, 1976.

[133] V. I. Lebedev and A. L. Skorokhodov. *Sov. Phys.-Dokl.*, 45:587, 1992.

[134] Lebedev V. I. and D. N. Laikov. *Doklady Mathematics*, 59:477, 1999.

[135] M. Eden and M. H. Levitt. *J. Magn. Res.*, 132:220, 1998.

[136] A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall, Englewood Cliffs, 1971.

[137] D. Laikov and C. Van Wüllen. Software. http://www.ccl.net/cca/software/SOURCES/FORTRAN/Lebedev-Laikov-Grids/. These routines are based on C code provided by Dmitri Laikov (laikov@chem.msu.ru). Christoph van Wüllen (Christoph.van.Wuellen@Ruhr-Uni-Bochum.de) has made the effort to obtain the code and convert it to Fortran77, and he obtained Dmitri's kind permission to make the routines publically available.

[138] E. Clementi and D. Raimondi. *IBM Res. Note*, NJ-27, 1963.

[139] O. Vahtras, J. Almlöf, and M. W. Feyereisen. *Chem. Phys. Lett.*, 213:514, 1993.

[140] F. Weigend and M. Häser. *Theor. Chem. Acc.*, 97:331, 1997.

[141] R. Izsak and F. Neese. *J. Chem. Phys.*, 135:144105, 2011.

[142] A. D. Becke and E. R. Johnson. *J. Chem. Phys.*, 122:154101, 2005.

[143] E. R. Johnson and A. D. Becke. *J. Chem. Phys.*, 123:024101, 2005.

[144] E. R. Johnson and A. D. Becke. *J. Chem. Phys.*, 124:174104, 2006.

[145] S. Grimme. *Chem. Eur. J.*, 18:9955–9964, 2012.

[146] Stefan Grimme. AK Grimme homepage. http://www.thch.uni-bonn.de/tc/grimme.

[147] K. Lee, E. D. Murray, L. Kong, B. I. Lundqvist, and D. C. Langreth. *Phys. Rev. B*, 82:081101, 2010.

[148] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 133:244103, 2010.

[149] W. Hujo and S. Grimme. *J. Chem. Theory Comput.*, 2011. dx.doi.org/10.1021/ct200644w.

[150] L. Goerigk and S. Grimme. *J. Chem. Theory Comput.*, 6:107, 2010.

[151] L. Goerigk and S. Grimme. *J. Chem. Theory Comput.*, 7:291–309, 2011.

[152] L. Goerigk and S. Grimme. *Phys. Chem. Chem. Phys.*, 13:6670, 2011.

[153] J. Řezáč, K.E. Riley, and P. Hobza. *J. Chem. Theory Comput.*, 7:2427, 2011.

[154] W. Hujo and S. Grimme. *Phys. Chem. Chem. Phys.*, 13:13942, 2011.

[155] P. Jurecka, J. Sponer, J. Cerny, and P. Hobza. *Phys. Chem. Chem. Phys.*, 8:1985, 2006.

[156] H. Kruse and S. Grimme. *J. Chem. Phys.*, 16:136, 2012.

[157] H. Kruse, L. Goerigk, and S. Grimme. *J. Org. Chem.*, 23:10824, 2012.

[158] R. Sure and S. Grimme. *J. Comput. Chem*, 2013, DOI: 10.1002/jcc.23317.

[159] R. McWeeny. *Methods of Molecular Quantum Mechanics. 2nd Edition.* Academic Press, 1992.

[160] D. Cremer in: P. v. R. Schleyer (Ed.). *Encyclopedia of Computational Chemistry.* John Wiley and sons Ltd: Chichester, England, 1998.

[161] S. Saebo and J. Almlöf. *Chem. Phys. Lett.*, 154:83, 1989.

[162] M. Head-Gordon and J. A. Pople. *Chem. Phys. Lett.*, 153:503, 1988.

[163] W. J. Lauderdale, J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett. *Chem. Phys. Lett.*, 187:21, 1991.

[164] P. J. Knowles, J. S. Andrews, R. D. Amos, N. C. Handy, and J. A. Pople. *Chem. Phys. Lett.*, 186:130, 1991.

[165] J. A. Pople, J. S. Binkley, and R. Seeger. *Int. J. Quant. Chem. Symp.*, 10:1, 1976.

[166] R. Krishnan, M. J. Frisch, and J. A. Pople. *J. Chem. Phys.*, 72:4244, 1980.

[167] N. C. Handy, P. J. Knowles, and K. Somasundram. *Theor. Chem. Acc.*, 68:87, 1985.

[168] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs. *Chem. Phys. Lett.*, 294:143, 1998.

[169] M. Feyereisen, G. Fitzerald, and A. Komornicki. *Chem. Phys. Lett.*, 208:359, 1993.

[170] D. E. Bernholdt and R. J. Harrison. *Chem. Phys. Lett.*, 250:477, 1996.

[171] J. A. Pople and D. L. Beveridge. *Approximate Molecular Orbital Theory.* McGraw Hill Inc, 1970.

[172] J. Sedlej and I. L. Cooper. *Semi-Emipirical Methods of Quantum Chemistry.* 1985, John Wiley and Sons.

[173] M. J. S. Dewar and W. Thiel. *Theor. Chim. Acta*, 46:89, 1977.

[174] W. Thiel and A. A. Voityuk. *Theor. Chim. Acta*, 81:391, 1992.

[175] M. J. S. Dewar and W. Thiel. *J. Am. Chem. Soc.*, 99:4899, 1977.

[176] M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, and J. P. Stewart. *J. Am. Chem. Soc.*, 107:3902, 1985.

[177] J. P. Stewart. *J. Comput. Chem.*, 10:209 & 221, 1989.

[178] J. A. Pople and G. A. Segal. *J. Chem. Phys.*, 43:136, 1965.

[179] J. A. Pople and G. A. Segal. *J. Chem. Phys.*, 44:3289, 1966.

[180] D. P. Santry. *J. Am. Chem. Soc.*, 90:3309, 1968.

[181] D. P. Santry and G. A. Segal. *J. Chem. Phys.*, 47:158, 1967.

[182] J. A. Pople, D. L. Beveridge, and P. A. Dobosh. *J. Chem. Phys.*, 47:2026, 1967.

[183] D. W. Clack, N. S. Hush, and J. R. Yandle. *J. Chem. Phys.*, 57:3503, 1972.

[184] D. W. Clack. *Mol. Phys.*, 27:1513, 1974.

[185] D. W. Clack and W. Smith. *Theor. Chim. Acta*, 36:87, 1974.

[186] M. C. Böhm and R. Gleiter. *Theor. Chim. Acta*, 59:127 & 153, 1981.

[187] J. Ridley and M. C. Zerner. *Theor. Chim. Acta*, 32:111, 1973.

[188] A. D. Bacon and M. C. Zerner. *Theor. Chim. Acta*, 53:21, 1979.

[189] M. C. Zerner, G. H. Loew, R. F. Kirchner, and U. T. Mueller-Westerhoff. *J. Am. Chem. Soc*, 102:589, 1980.

[190] W. P. Anderson, W. D. Edwards, and M. C. Zerner. *Inorg. Chem.*, 25:2728, 1986.

[191] W. P. Anderson, T. R. Cundari, R. S. Drago, and M. C. Zerner. *Inorg. Chem.*, 29:3, 1990.

[192] W. P. Anderson, T. R. Cundari, and M. C. Zerner. *Int. J. Quant. Chem.*, XXXIX:31, 1991.

[193] M. C. Zerner in: K. B. Lipkowitz and D. B. Boyd (Eds.). *Reviews in Computational Chemistry Vol. 2.* VCH, Heidelberg, 1990.

[194] M. C. Zerner in: D. R. Salahub and N. Russo (Eds.). *Metal-Ligand Interactions: from Atoms to Clusters to Surfaces.* Kluwer Academic Publishers, the Netherlands, 1992.

[195] M. C. Zerner in: D. R. Salahub and N. Russo (Eds.). *Metal-Ligand Interactions.* Kluwer Academic Publishers, the Netherlands, 1992.

[196] M. G. Cory and M. C. Zerner. *Chem. Rev.*, 91:813, 1991.

[197] M. Kotzian, N. Rösch, and M. C. Zerner. *Theor. Chim. Acta*, 81:201, 1992.

[198] C. Nieke and J. Reinhold. *Theor. Chim. Acta*, 65:99, 1984.

[199] H. J. Köhler and F. Birnstock. *Z. Chem.*, 5:196, 1972.

[200] Molecular Science Computing Facility. Extensible computational chemistry environment basis set database. http://www.emsl.pnl.gov:2080/forms/basisform.html, 2000. Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version Mon Apr 17 10:05:30 PDT 2000, as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multi-program laboratory operated by Battelle Memorial Institue for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. Contact David Feller or Karen Schuchardt for further information.

[201] J. Andzelm, M. Klobukowski, E. Radzio-Andzelm, Y. Sakai, and H.Tatewaki in: S. Huzinaga (Ed). *Gaussian basis sets for molecular calculations.* Elsevier, Amsterdam, 1984.

[202] W. J. Hehre, R. F. Stewart, and J. A. Pople. *J. Chem. Phys.*, 51:2657, 1969.

[203] R. F. Stewart. *J. Chem. Phys.*, 50:2485, 1969.

[204] W. J. Hehre, R. Ditchfield, R. F. Stewart, and J.A. Pople. *J. Chem. Phys.*, 52:2769, 1970.

[205] W. J. Pietro, B. A. Levi, W. J. Hehre, and R. F. Stewart. *Inorg. Chem.*, 19:2225, 1980.

[206] W. J. Pietro, E. S. Blurock, R. F. Hout, Hehrem W. J., D. J. DeFrees, and R. F. Stewart. *Inorg. Chem.*, 20:3650, 1981.

[207] W. J. Pietro and W. J. Hehre. *J. Comput. Chem.*, 4:241, 1983.

[208] J. S. Binkley, J. A. Pople, and W. J. Hehre. *J. Am. Chem. Soc.*, 102:939, 1980.

[209] M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre. *J. Am. Chem. Soc.*, 104:2797, 1983.

[210] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 7:359, 1986.

[211] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 8:861 & 880, 1987.

[212] W. J. Hehre, R. Ditchfield, and J. A. Pople. *J. Chem. Phys.*, 56:2257, 1972.

[213] J. D. Dill and J. A. Pople. *J. Chem. Phys.*, 62:2921, 1975.

[214] M. M. Francl, W. J. Petro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees, and J. A. Pople. *J. Chem. Phys.*, 77:3654, 1982.

[215] V. Rassolov, J. A. Pople, M. Ratner, and T. L. Windus. *J. Chem. Phys.*, accepted 1998.

[216] T. Clark, J. Chandrasekhar, and P. v. R. Schleyer. *J. Comput. Chem.*, 4:294, 1983.

[217] Hariharan P. C. and J. A. Pople. *Theor. Chim. Acta*, 28:213, 1973.

[218] R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople. *J. Chem. Phys.*, 72:650, 1980.

[219] A. D. McLean and G. S. Chandler. *J. Chem. Phys.*, 72:5639, 1980.

[220] J.-P. Blaudeau, M. P. McGrath, L. A. Curtiss, and L. Radom. *J. Chem. Phys.*, 107:5016, 1997.

[221] L. A. Curtiss, M. P. McGrath, J.-P. Blandeau, N. E. Davis, R. C. Binning Jr., and L. Radom. *J. Chem. Phys.*, 103:6104, 1995.

[222] M. J. Frisch, J. A. Pople, and J. S. Binkley. *J. Chem. Phys.*, 80:3265, 1984.

[223] A. V. Mitin, G. Hirsch, and R. Buenker. *Chem. Phys. Lett.*, 259:151, 1996.

[224] A. V. Mitin, G. Hirsch, and R. Buenker. *J. Comput. Chem.*, 18:1200, 1997.

[225] A. Schäfer, H. Horn, and R. J. Ahlrichs. *Chem. Phys.*, 97:2571, 1992.

[226] N. Godbout, D. R. Salahub, J. Andzelm, and E. Wimmer. *Can. J. Chem.*, 70:560, 1992.

[227] T. H. Dunning Jr. *J. Chem. Phys.*, 90:1007, 1989.

[228] D. E. Woon and T. H. Dunning Jr. *J. Chem. Phys.*, 100:2975, 1994.

[229] D. E. Woon and T. H. Dunning Jr. *J. Chem. Phys.*, 98:1358, 1993.

[230] R. A. Kendall, T. H. Dunning Jr, and R. J. Harrison. *J. Chem. Phys.*, 96:6769, 1992.

[231] V. Barone in: D. P. Chong (Ed.). *Recent Advances in Density Functional Methods, Part I.* World Scientific, 1996.

[232] W. Kutzelnigg, U. Fleischer, and M. Schindler. *The IGLO-Method: Ab Initio Calculation and Interpretation of NMR Chemical Shifts and Magnetic Susceptibilities*, volume 23. Springer Verlag, 1990.

[233] A. J. H. Wachters. *J. Chem. Phys.*, 52:1033, 1970.

[234] C. W. Jr. Bauschlicher, S. R. Langhoff, and L. A. Barnes. *J. Chem. Phys.*, 91:2399, 1989.

[235] H. Partridge. *J. Chem. Phys.*, 90:1043, 1989.

[236] H. Partridge. *J. Chem. Phys.*, 87:6643, 1987.

[237] Ahlrichs R. and coworkers. unpublished.

[238] Gamess program. Gamess user manual. Polarization functions obtained from the Gamess programm.

[239] L. Giordano, G. Pacchioni, T. Bredow, and J. F. Sanz. *Surf. Sci.*, 471:21, 2001.

[240] A. T. Amos and G. G. Hall. *Proc. R. Soc. Ser. A.*, 263:483, 1961.

[241] H. F. King, R. E. Stanton, H. Kim, R. E. Wyatt, and R. G. Parr. *J. Chem. Phys.*, 47:1936, 1967.

[242] M. C. Zerner and M. Hehenberger. *Chem. Phys. Lett.*, 62:550, 1979.

[243] M. F. Guest and V. R. Saunders. *Mol. Phys.*, 28:819, 1974.

[244] V. R. Saunders and I. H. Hillier. *Int. J. Quant. Chem.*, VII:699, 1973.

[245] P. Pulay. *Chem. Phys. Lett.*, 73:393, 1980.

[246] P. Pulay. *J. Comput. Chem.*, 3:556, 1992.

[247] T. H. Fischer and J. Almlöf. *J. Phys. Chem.*, 96:9768, 1992.

[248] F. Neese. *Chem. Phys. Lett.*, 325:93, 2000.

[249] M. J. S. Dewar, J. A. Hashmall, and C. G. Venier. *J. Am. Chem. Soc.*, 90:1953, 1968.

[250] R. McWeeny. *Mol. Phys.*, 28:1273, 1974.

[251] F. W. Brobowicz and H. F. (Ed.) Goddard, W. A. in: Schaefer III. *Methods of Electronic Structure Theory*. Plenum Press, 1977.

[252] R. Carbo and J. M. Riera. *A General SCF theory. Lecture notes in Chemistry*. Springer Verlag, 1978.

[253] J. S. Binkley, J. A. Pople, and P. A. Dobosh. *Mol. Phys.*, 28:1423, 1974.

[254] W. D. Edwards and M. C. Zerner. *Theor. Chim. Acta*, 72:347, 1987.

[255] R. P. Muller, J. M. Langlois, M. N. Ringnalda, R. A. Friesner, and W. A. Goddard. *J. Chem. Phys.*, 100:1226, 1994.

[256] C. Kollmar. *J. Chem. Phys.*, 105:8204, 1996.

[257] C. Kollmar. *Int. J. Quant. Chem.*, 62:617, 1997.

[258] J. M. Bofill, H. Bono, and J. Rubio. *J. Comput. Chem.*, 19:368, 1998.

[259] K. K. Stavrev and M. C. Zerner. *Int. J. Quant. Chem.*, 65:877, 1997.

[260] M. C. Zerner. *Int. J. Quant. Chem.*, XXXV:567, 1989.

[261] K. Almlöf, J. Faegri and K. Korsell. *J. Comput. Chem.*, 3:385, 1982.

[262] J. Almlöf and P. R. Taylor in: Dykstra (Ed.). *Advanced Theories and Computational Approaches to the Electronic Structure of Molecules*. D. Reidel, 1984.

[263] J. Almlöf in: D. R. Yarkony (Ed.). *Modern Electronic Structure Theory*. World Scientific, 1995.

[264] M. Häser and R. Ahlrichs. *J. Comput. Chem.*, 10:104, 1989.

[265] F. Jensen. *Introduction to Computational Chemistry*. Wiley, 1999.

[266] F. Neese. *J. Am. Chem. Soc.*, 128:10213, 2006.

[267] R. Ahlrichs. *Comp. Phys. Comm.*, 17:31, 1979.

[268] R. J. Gdanitz. *Int. J. Quant. Chem.*, 85:281, 2001.

[269] R. J. Gdanitz and R. Ahlrichs. *Chem. Phys. Lett.*, 1988:413, 143.

[270] P. G. Szalay and R. J. Bartlett. *Chem. Phys. Lett.*, 214:481, 1993.

[271] C. Kollmar and F. Neese. *Mol. Phys.*, 108:2449, 2010.

[272] C. Kollmar and F. Neese. *J. Chem. Phys.*, 135:064103, 2011.

[273] G. E. Scuseria and H. F. Schaefer III. *Chem. Phys. Lett.*, 142:354, 1987.

[274] N. C. Handy, J. A. Pople, M. Head-Gordon, K. Raghavachari, and G. W. Trucks. *Chem. Phys. Lett.*, 164:185, 1989.

[275] C. Kollmar and A. Hesselmann. *Theor. Chem. Acc.*, 127:311, 2010.

[276] E. A. Salter, G. W. Trucks, and R. J. Bartlett. *J. Chem. Phys.*, 90:1752, 1989.

[277] C. Kollmar and F. Neese. *J. Chem. Phys.*, 135:084102, 2011.

[278] L. M. J. Huntington and M. Nooijen. *J. Chem. Phys.*, 133:184109, 2010.

[279] L. M. J. Huntington, A. Hansen, F. Neese, and M. Nooijen. *J. Chem. Phys.*, 136:064101, 2012.

[280] C. Hampel, K. A. Peterson, and H. J. Werner. *Chem. Phys. Lett.*, 190:1, 1992.

[281] G. E. Scuseria, C. L. Janssen, and H. F. Schaefer. *J. Chem. Phys*, 89:7382, 1988.

[282] J. L. Heully and J.-P. Malrieu. *J. Mol. Struct. Theochem.*, 768:53, 2006.

[283] F. Neese, A. Hansen, and D. G. Liakos. *J. Chem. Phys.*, 131:064103, 2009.

[284] A. Hansen, D. G. Liakos, and F. Neese. *J. Chem. Phys.*, 135:214102, 2011.

[285] F. Neese, A. Hansen, F. Wennmohs, and S. Grimme. *Acc. Chem. Res.*, 42:641, 2009.

[286] F. Neese, D. G. Liakos, and S. F. Ye. *J. Biol. Inorg. Chem.*, 16:821, 2011.

[287] G. K.-L. Chan and M. Head-Gordon. *J. Chem. Phys.*, 116:4462, 2002.

[288] G. K.-L. Chan. *J. Chem. Phys.*, 120:3172, 2004.

[289] D. Gosh, J. Hachmann, T. Yanai, and G. K.-L. Chan. *J. Chem. Phys.*, 128:144117, 2008.

[290] S. Sharma and G. K.-L. Chan. *J. Chem. Phys.*, 136:124121, 2012.

[291] D. Ganyushin and F. Neese. *J. Chem. Phys*, 138:104113, 2013.

[292] C. Angeli, R. Cimiraglia, S. Evangelisti, T. Leininger, and J.-P. Malrieu. *J. Chem. Phys.*, 114:10252–10264, 2001.

[293] C. Angeli, R. Cimiraglia, and J.-P. Malrieu. *Chem. Phys. Letters*, 350:297–305, 2001.

[294] C. Angeli, R. Cimiraglia, and J.-P. Malrieu. *J. Chem. Phys.*, 117:9138–9153, 2002.

[295] M. Cestari C. Angeli, S. Borini and R. Cimigraglia. *J. Chem. Phys.*, 121:4043, 2004.

[296] K. Andersson, P. A. Malmqvist, B. O. Roos, A. J. Sadlej, and K. Wolinski. *J. Phys. Chem.*, 94:5483–5488, 1990.

[297] B.O. Roos J. Finley, P.-A. Malmqvist and L. Serrano-Andres. *Chem. Phys. Lett.*, 288:299, 1998.

[298] K. G. Dyall. *J. Chem. Phys.*, 102:4909–4918, 1995.

[299] A. Domingo, M.-A. Carvajal, C. de Graaf, K. Sivalingam, F. Neese, and C. Angeli. *Theor. Chem. Acc.*, 131(9):1264, 2012.

[300] J. des Cloizeaux. *Nucl. Phys.*, 20:321, 1960.

[301] G. K.-L. Chan. *Ann. Rev. Phys. Chem.*, 65:465, 2011.

[302] G. K.-L. Chan. Dmrg homepage. http://www.princeton.edu/chemistry/chan/software/dmrg/.

[303] Miroslav Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23:298, 1973.

[304] Miroslav Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. J.*, 25:619, 1975.

[305] JE Atkins, EG Boman, and Bruce Hendrickson. A spectral algorithm for seriation and the consecutive ones problem. *SIAM J. Comput.*, 28(1):297–310, 1998.

[306] G. Barcza, Ö. Legeza, K. H. Marti, and M. Reiher. Quantum-information analysis of electronic states of different molecular structures. *Phys. Rev. A*, 83:012508, 2011.

[307] E. van Lenthe, E. J. Baerends, and J. G. Snijders. *J. Chem. Phys.*, 101:9783–9792, 1994.

[308] B. Sandhoefer and F. Neese. *J. Chem. Phys.*, 137:094102, 2012.

[309] L. Visscher and K. G. Dyall. *Atom. Data Nucl. Data Tabl.*, 67:207, 1997.

[310] F. Neese. *J. Biol. Inorg. Inorg. Chem.*, 11:702, 2006.

[311] F. Neese. *Coordination Chemistry Reviews*, 253:526, 2009.

[312] T. Petrenko, S. Kossmann, and F. Neese. *J. Chem. Phys.*, 134:054116, 2011.

[313] M. Head-Gordon, R. A. Rico, M. Oumi, and T. J. Lee. *Chem. Phys. Letters*, 219:21–29, 1994.

[314] M. Grüning, O. V. Gritsenko, S. J. A. Gisbergen, and E. J. Baerends. *J. Chem. Phys.*, 114:652, 2001.

[315] M. Roemelt and F. Neese. *J. Phys. Chem. A*, 117:3069–3082, 2013.

[316] F. Neese and E. I. Solomon. *Inorg. Chem.*, 37:6568–6582, 1998.

[317] D. Ganyushin and F. Neese. *J. Chem. Phys.*, 128:114117, 2008.

[318] S. Koseki, M. W. Schmidt, and M. S. Gordon. *J. Phys. Chem.*, 96:10768–10772, 1992.

[319] S. Koseki, M. S. Gordon, M. W. Schmidt, and N. Matsunaga. *J. Phys. Chem.*, 99:12764–12772, 1995.

[320] S. Koseki, M. W. Schmidt, and M. S. Gordon. *J. Phys. Chem. A*, 102:10430–10435, 1998.

[321] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. J. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery. *J. Comput. Chem.*, 14:1347, 1993.

[322] G. Chaban, M. W. Schmidt, and M. S. Gordon. *Theor. Chem. Acc.*, 97:88, 1997.

[323] J. Fernandez-Rico, R. Lopez, A. Aguado, I. Ema, and G. Ramirez. *J. Comput. Chem.*, 11:1284, 1998. and references therein.

[324] P. A. M. Dirac. *Proc. Camb. Phil. Soc.*, 26:376, 1930.

[325] J. C. Slater. *The quantum theory of atoms molecules and solids, Vol. 4.* McGraw Hill, New York, 1974.

[326] A. D. Becke. *Phys. Rev. A*, 38:3098, 1988.

[327] P. M. W. Gill. *Mol. Phys.*, 89:433, 1996.

[328] J. P. Perdew, J.A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. *Phys. Rev. A.*, 46:6671, 1992.

[329] J. P. Perdew, K. Burke, and M. Ernzerhof. *Phys. Rev. Lett.*, 77:3865, 1996.

[330] B. Hammer, L. B. Hansen, and J. K. Nørskov. *Phys. Rev. B*, 59:7413, 1999.

[331] R. S. Mulliken. *J. Chem. Phys.*, 23:1833, 1955.

[332] K. B. Wiberg. *Tetrahedron*, 24:1083, 1968.

[333] I. Mayer. *Chem. Phys. Lett.*, 97:270, 1983.

[334] I. Mayer. *Int. J. Quant. Chem.*, 26:151, 1984.

[335] I. Mayer. *Theor. Chim. Acta*, 67:315, 1985.

[336] I. Mayer in: Z. B. Maksic (Ed.). *Modelling of Structure and Properties of Molecules.* John Wiley and Sons, 1987.

[337] T. Fueno K. Takatsuka and K. Yamaguchi. *Theor. Chem. Acta*, 48:175, 1978.

[338] M. Head-Gordon. *Chem. Phys. Lett.*, 372:508, 2003.

[339] I. Mayer E. Ramos-Cordoba, E. Matito and P. Salvador. *J.C.T.C.*, 8:1270, 2012.

[340] A.E. Clark E.R. Davidson. *Phys. Chem. Chem. Phys.*, 9:1881, 2007.

[341] A. Torre D.R. Alcoba, L. Lain and R.C. Bochicchio. *Chem. Phys. Lett.*, 470:136, 2009.

[342] C. M. Breneman and K. B. Wiberg. *J. Comput. Chem.*, 11:361–373, 1990.