

(continued from previous page)

```

      refs cas(4,4) end
    end
    newblock 3 *
      nroots 7
      refs cas(4,4) end
    end
    newblock 5 *
      nroots 1
      refs cas(4,4) end
    end
  end
end

* xyz 0 5
O          0.00000000    0.00000000   -500.60500000
O          0.00000000   -0.00000000   -499.39500000
O         -0.60500000    0.00000000    500.00000000
O          0.60500000   -0.00000000    500.00000000
*
```

In Table Table 3.55, we have compiled the results of the size consistency test, taking the difference of the dimer state energies (at large separation) and the sum of the monomer state energies (in mE_h). It is evident that as more excitation classes are included in the similarity transformed Hamiltonian and the size of the final diagonalization manifold is decreased, the size-consistency errors decrease. Of particular note are the results for the MR-EOM-TIT[†]|SXD|U-h-v approach (only includes 1h and 1p configurations in the final diagonalization manifold), for which the largest deviation is $1.25 \times 10^{-2} mE_h$. The much larger deviations for the MR-EOM-TIT[†]|SXD-h-v approach clearly demonstrate the large effect that the 2h excitations have on the size-consistency errors.

Table 3.55: Test for size consistency in MR-EOM-CC Differences in energy (in mE_h) between the $_2\text{—O}_2$ dimer energies (at large separation) and the sum of the monomer energies for the ground state and various excited states. The results were obtained in an aug-cc-pVTZ basis using minimal active spaces.

	T T [†] -h-v	T T [†] SXD-h-v (with 1h1p)	T T [†] SXD-h-v	T T [†] SXD U-h-v
$\Delta E(^3\Sigma_g^- + ^3\Sigma_g^-)$	12.74	2.77	1.11	1.00×10^{-5}
$\Delta E(^3\Sigma_g^- + ^1\Delta_g)$	14.20	3.84	1.85	1.54×10^{-4}
$\Delta E(^3\Sigma_g^- + ^1\Sigma_g^+)$	17.21	5.52	2.83	4.13×10^{-4}
$\Delta E(^1\Delta_g + ^1\Delta_g)$	15.69	3.10	2.31	5.26×10^{-3}
$\Delta E(^1\Delta_g + ^1\Sigma_g^+)$	18.83	7.52	4.76	5.89×10^{-3}
$\Delta E(^1\Sigma_g^+ + ^1\Sigma_g^+)$	22.34	10.75	7.31	1.25×10^{-2}

To reduce the size-consistency errors, one can make use of the MR-CEPA(0) shift in the final diagonalization step. This MR-CEPA(0) shift can easily be activated by adding the line

```
citype mrcepa_0
```

to the beginning of the %mrci block. The results of the size-consistency test with the use of the MR-CEPA(0) shift are tabulated in Table Table 3.56. For each of the methods, we see a marked improvement over the results of Table *Test for size consistency in MR-EOM-CC Differences in energy (in mE_h) between the $_2\text{—O}_2$ dimer energies (at large separation) and the sum of the monomer energies for the ground state and various excited states. The results were obtained in an aug-cc-pVTZ basis using minimal active spaces.*, which do not make use of the MR-CEPA(0) shift. The greatest improvement occurs in the MR-EOM-TIT[†]|SXD-h-v and the MR-EOM-TIT[†]|SXD|U-h-v results. Namely, the errors in the former case are on the order of nano Hartrees, while the errors in the MR-EOM-TIT[†]|SXD|U-h-v results are not detectable (sub-nano Hartree), as the energy is only printed with nine decimal places. It is interesting to note that upon adding the 1h1p configurations to the diagonalization manifold in the MR-EOM-TIT[†]|SXD-h-v calculations (i.e. with 1h1p), the size-consistency errors increase greatly. Hence, it appears that the use of the MR-CEPA(0) shift is most effective at reducing the size-consistency errors resulting from the presence of the 1h, 1p and 2h configurations in the final diagonalization manifold. In any case, one can easily take advantage of this approach to obtain nearly size-consistent results with both the MR-EOM-TIT[†]|SXD-h-v and MR-EOM-TIT[†]|SXD|U-

h-v methods.

Table 3.56: Test for size consistency in MR-EOM-CC using the MR-CEPA(0) shift: Differences in energy (in mE_h) between the O₂—O₂ dimer energies (at large separation) and the sum of the monomer energies for the ground state and various excited states. The results were obtained in an aug-cc-pVTZ basis using minimal active spaces and the MR-CEPA(0) shift was applied in the final diagonalization in each case.

	T T [†] -h-v	T T [†] SXD-h-v (with 1h1p)	T T [†] SXD-h-v	T T [†] SXD U-h-v
$\Delta E(^3\Sigma_g^- + ^3\Sigma_g^-)$	2.75×10^{-3}	0.01	2.00×10^{-6}	0.00
$\Delta E(^3\Sigma_g^- + ^1\Delta_g)$	0.06	0.07	0.00	0.00
$\Delta E(^3\Sigma_g^- + ^1\Sigma_g^+)$	0.14	0.15	4.00×10^{-6}	0.00
$\Delta E(^1\Delta_g + ^1\Delta_g)$	0.21	0.22	1.00×10^{-6}	0.00
$\Delta E(^1\Delta_g + ^1\Sigma_g^+)$	0.42	0.44	5.00×10^{-6}	0.00
$\Delta E(^1\Sigma_g^+ + ^1\Sigma_g^+)$	0.82	0.87	9.00×10^{-6}	0.00

3.20.7 Perturbative MR-EOM-CCPT

The MR-EOM-T approach was developed for situations where the full accuracy of the iterative MR-EOM-CC method is not required. It performs on par with other multireference perturbation theories such as fic-NEVPT2 and does not have the convergence difficulties with the \hat{T} and \hat{S} , \hat{X} , \hat{D} amplitudes like its iterative parent method as these amplitudes are computed in a non-iterative fashion. The only iterative part of the MR-EOM-T method is the calculation of the \hat{U} amplitudes since they are quick to converge anyways [524].

The setup procedure for the MR-EOM-T method is the same as for the MR-EOM-CC method, and the foregoing also applies to the perturbative variant. Please note that the orbital selection scheme has not been tested with this variant and should be unnecessary anyways since calculations are much faster than with the iterative MR-EOM-CC method.

To invoke the new variant, set up the calculation as you would for an MR-EOM-CC calculation and then add the keyword `DomREOM_MRPT True` to the `%mdci` block.

The results are interpreted just like results for the iterative MR-EOM-CC method. After transforming the Hamiltonian with the perturbatively estimated amplitudes and the final MRCIS diagonalization step, the final state energies are printed along with their reference weights. For reliable results, we again recommend that the reference weight be >90%.

3.21 Dynamic Correlation Dressed CAS

Non-degenerate multireference perturbation theory (MRPT) methods, such as NEVPT2 or CASPT2, have the 0th order part of the wave function fixed by a preceding CASSCF calculation. The latter can be a problem if the CASSCF states are biased towards a wrong state composition in terms of electron configurations. In these instances, a quasi-degenerate or multi-state formulation is necessary, for example the QD-NEVPT2 described in Section *Quasi-Degenerate SC-NEVPT2*. A drawback of these approaches is that the results depend on the number of included states. The 2nd order Dynamic Correlation Dressed Complete Active Space method (DCD-CAS(2)) is a post-CASSCF MRPT method of the perturb-then-diagonalize kind, i.e. it can modify the CAS wavefunction compared to the previous CASSCF.[529] DCD-CAS(2) offers an alternative uncontracted approach, where a dressed CASCI matrix is constructed. Its diagonalization yields correlated energies and 0th order states that are remixed in the CASCI space under the effect of dynamic correlation.[529] In cases where CASSCF already provides a good 0th order wavefunction, DCD-CAS(2) energies are comparable to NEVPT2.

3.21.1 Theory of Nonrelativistic DCD-CAS(2)

The DCD-CAS(2) method is based on solving the eigenvalue problem of an effective Hamiltonian of the form

$$H_{IJ}^{\text{DCD},S} = \langle \Phi_I^{SS} | H | \Phi_J^{SS} \rangle - \sum_{K \in \text{FOIS}} \frac{\langle \Phi_I^{SS} | H | \tilde{\Phi}_K^{SS} \rangle \langle \tilde{\Phi}_K^{SS} | H | \Phi_J^{SS} \rangle}{E_K^S - E_0^S}$$

for each total spin S separately. The 0th order energies E_K^S of the perturbors $|\tilde{\Phi}_K^{SS}\rangle$ are obtained by diagonalizing the Dyll Hamiltonian in the first-order interacting space (FOIS, the space consisting of all single and double excitations with respect to the CASCI space). The effective Hamiltonian has the form of a CASCI Hamiltonian that is dressed with the effect of dynamic correlation (dynamic correlation dressed, DCD), hence the name for the method. E_0^S is chosen to be the ground state CASSCF energy for the respective total spin S . Since this choice is worse for excited states than for the ground state, excitation energies suffer from a “ground state bias”.

For the contribution coming from perturbors in which electrons are excited from two inactive (ij) to two virtual (ab) orbitals, we use (when writing the DCD Hamiltonian in a basis of CASCI states) the alternative expression

$$\begin{aligned} \langle \Psi_I^{SS} | H^{\text{DCD}}(ij \rightarrow ab) | \Psi_J^{SS} \rangle &= -\delta_{IJ} E_{\text{MP2}} \\ E_{\text{MP2}} &= \sum_{ijab} \frac{(ib|ja)^2 - (ib|ja)(ia|jb) + (ia|jb)^2}{\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j} \end{aligned}$$

Since in this version the $ij \rightarrow ab$ perturber class does not contribute at all to excitation energies (like it is assumed in the difference-dedicated configuration interaction method), we call this the difference-dedicated DCD-CAS(2) method. Since the $ij \rightarrow ab$ class contributes the largest part of the dynamic correlation energy, this also removes the largest part of the ground state bias. This option is used as default in DCD-CAS(2) calculations. In order to also remove the ground state bias from the other perturber classes, we furthermore apply a perturbative correction to the final energies. At first order (which is chosen as default), it takes the form

$$\begin{aligned} \Delta E_I &= -\Delta_I \sum_{K \in \text{FOIS}} \frac{\langle \tilde{\Psi}_I | H | \tilde{\Phi}_K \rangle \langle \tilde{\Phi}_K | H | \tilde{\Psi}_I \rangle}{(E_K - E_0)^2} \\ \Delta_I &= \langle \tilde{\Psi}_I | H | \tilde{\Psi}_I \rangle - E_0 \end{aligned}$$

for the correction ΔE_I to the total energy of the I th DCD-CAS(2) root $|\tilde{\Psi}_I\rangle$.

3.21.2 Treatment of spin-dependent effects

The theory so far is valid for a nonrelativistic or scalar-relativistic Hamiltonian H . If we modify it to a Hamiltonian $H + V$, where V contains effects that are possibly spin-dependent such as spin-orbit coupling (SOC) and spin-spin coupling (SSC), this leads us to a theory [432] which has a similar form as QDPT with all CAS roots included. The form of the spin-dependent DCD-CAS(2) effective Hamiltonian is

$$\begin{aligned} \langle \Phi_I^{SM} | H^{\text{DCD}} | \Phi_J^{S'M'} \rangle &= \delta_{SS'} \delta_{MM'} H_{IJ}^{\text{DCD},S,\text{corr}} + \langle \Phi_I^{SM} | V | \Phi_J^{S'M'} \rangle. \\ \mathbf{H}^{\text{DCD},S,\text{corr}} &= \mathbf{C}^{\text{DCD}} \mathbf{E} (\mathbf{C}^{\text{DCD}})^T. \end{aligned}$$

In order to construct it, we first need to solve the scalar-relativistic DCD-CAS(2) problem to construct the matrix $\mathbf{H}^{\text{DCD},S,\text{corr}}$ from the bias corrected energies \mathbf{E} and DCD-CAS(2) CI coefficients \mathbf{C}^{DCD} and then calculate the matrix elements of the operators contributing to V in the basis of CSFs $|\Phi_I^{SM}\rangle$.

Zero field splitting D tensors are extracted using the effective Hamiltonian technique, i.e. fitting the model Hamiltonian to a des Cloizeaux effective Hamiltonian that is constructed from the relativistic states and energies by projection onto the nonrelativistic multiplet (see section [Zero-Field Splitting](#) and the reference [530]). There are limitations to this approach if spin orbit coupling becomes so strong that the relativistic states cannot uniquely be assigned to a single nonrelativistic spin multiplet.

Hyperfine A-matrices and Zeeman g-matrices for individual Kramers doublets consisting of states $|\Phi\rangle, |\bar{\Phi}\rangle$ are extracted by comparing the spin Hamiltonians

$$H_{\text{Zeeman}} = \mu_B \vec{B} \cdot \mathbf{g} \cdot \vec{S}$$

$$H_{\text{HFC}} = \sum_A \vec{I}^A \cdot A^A \cdot \vec{S}$$

to the matrix representation of the many-electron Zeeman and HFC operators in the basis of the Kramers doublet. This yields [432]

$$\begin{aligned} g_{k1} &= 2\Re\langle\bar{\Phi}|L_k + g_e S_k|\Phi\rangle \\ g_{k2} &= 2\Im\langle\bar{\Phi}|L_k + g_e S_k|\Phi\rangle \\ g_{k3} &= 2\langle\Phi|L_k + g_e S_k|\Phi\rangle \\ A_{k1} &= -2\gamma_A \Re\langle\bar{\Phi}|B_k^{\text{HFC}}(\vec{R}_A)|\Phi\rangle \\ A_{k2} &= -2\gamma_A \Im\langle\bar{\Phi}|B_k^{\text{HFC}}(\vec{R}_A)|\Phi\rangle \\ A_{k3} &= -2\gamma_A \langle\Phi|B_k^{\text{HFC}}(\vec{R}_A)|\Phi\rangle \end{aligned}$$

where $B_k^{\text{HFC}}(\vec{R}_A)$ is the k th component of the magnetic hyperfine field vector at the position of nucleus A and γ_A is the gyromagnetic ratio.

3.21.3 Simple example

The basic usage is very simple: One just needs a `%casscf` block and the simple input keyword `!DCD-CAS(2)`. The following example is a calculation on the LiF molecule. It possesses two singlet states that can be qualitatively described as ionic (Li^+ and F^-) and covalent (neutral Li with electron in 2s orbital and neutral F with hole in $2p_z$ orbital). At distances close to the equilibrium geometry, the ground state is ionic, while in the dissociation limit the ground state is neutral. Somewhere in between, there is an avoided crossing of the adiabatic potential energy curves where the character of the two states quickly changes (see figure Fig. 3.39 for potential energy curves for this system at the (QD)NEVPT2 level). At the CASSCF level, the neutral state is described better than the ionic state, with the result that the latter is too high in energy and the avoided crossing occurs at a too small interatomic distance. In the region where the avoided crossing actually takes place, the CASSCF states are then purely neutral or purely ionic. DCD-CAS(2) allows for a remixing of the states in the CASCI space under the effect of dynamic correlation, which will lower the ionic state more in energy than the neutral one. The input file is as follows:

```
! def2-TZVP DCD-CAS(2)
!moread

%moinp "casorbs.gbwn" # guess with active orbitals in place

%casscf
  nel 2
  norb 2
  mult 1
  nroots 2
  actorbs locorbs
end

*xyz 0 1
Li 0.0 0.0 0.0
F 0.0 0.0 5.5
*
```

Since none of the standard guesses (`!PAtom`, `!PModel`, etc.) produces the correct active orbitals (Li 2s and F $2p_z$), we read them from the file `casorbs.gbwn`. We also use the `actorbs locorbs` option to preserve the atomic character of the active orbitals and interpret the states in terms of neutral and ionic components easier. The following is the state composition of LiF at an interatomic distance of 5.5 angstrom at the CASSCF and DCD-CAS(2) levels.

```
-----
CAS-SCF STATES FOR BLOCK 1 MULT= 1 NROOTS= 2
-----

ROOT 0: E= -106.8043590118 Eh
```

(continues on next page)

(continued from previous page)

```

0.99395 [ 1]: 11
0.00604 [ 2]: 02
ROOT 1: E= -106.7485794535 Eh 1.518 eV 12242.2 cm**-1
0.99396 [ 2]: 02
0.00604 [ 1]: 11

```

```

-----
DCD-CAS(2) STATES
-----

ROOT 0: E= -107.0917611937 Eh
0.60590 [ 2]: 02
0.39410 [ 1]: 11
ROOT 1: E= -107.0837717163 Eh 0.217 eV 1753.5 cm**-1
0.60590 [ 1]: 11
0.39410 [ 2]: 02

```

One can clearly see that while the CASSCF states are purely neutral (dominated by CFG 11) or purely ionic (dominated by CFG 02), the DCD-CAS(2) states are mixtures of neutral and ionic contributions. The calculation indicates that the interatomic distance of 5.5 is in the avoided crossing region. Note that the energies that are printed together with the DCD-CAS(2) state composition are the ones that are obtained from diagonalization of the DCD-CAS(2) dressed Hamiltonian. For excited states, these energies suffer from what we call *ground state bias* (see the original paper for a discussion [529]). A perturbative correction has been devised to overcome this problem. Our standard choice is first-order bias correction. The corrected energies are also printed in the output file and those energies should be used in production use of the DCD-CAS(2) method:

```

-----
BIAS-CORRECTED (ORDER 1) STATE AND TRANSITION ENERGIES
=====
ROOT      Energy/a.u.      DE/a.u.      DE/eV      DE/cm**-1
=====
0:  -107.093214435      0.000000      0.000      0.0
1:  -107.084988306      0.008226      0.224     1805.4

```

Warning

Note that the computational cost of a DCD-CAS(2) calculation scales as roughly the 3rd power of the size of the CASCI space. This makes calculations with active spaces containing more than a few hundred CSFs very expensive!

3.21.4 List of keywords

The following keywords can be used in conjunction with the DCD-CAS(2) method:

```

%casscf
dcdcas true           # Do a DCD-CAS(2) calculation
diffded true         # Use difference-dedicated DCD-CAS(2) for the
                      # ij->ab class
corrorder 1          # Maximum order for the perturbative bias correction
                      # (lower orders come for free)
dcd_ritrafo false     # Use RI approximation for electron repulsion
->integrals

dcd_soc false         # Relativistic DCD-CAS(2) with spin orbit coupling
dcd_ssc false         # Relativistic DCD-CAS(2) with direct electronic
                      # spin-spin coupling

```

(continues on next page)

(continued from previous page)

```

dcd_domagfield 0          # Number of user-specified finite magnetic fields
dcd_dtensor false        # Calculate an effective Hamiltonian D-tensor
dcd_nmultd 1              # The number of nonrelativistic multiplets for which
↪the                      # D-tensor is calculated
dcd_gmatrix false        # Calculate an effective Kramers pair Zeeman g-matrix
dcd_amatrix false        # Calculate an effective Kramers pair Hyperfine A-
↪matrix
dcd_kramerspairs 1        # The number of Kramers pairs for which g and/or A
                          # is calculated
dcd_magnetization false   # Calculate the magnetization of the molecule in an
                          # external magnetic field

dcd_cascimode false      # Run relativistic calculation in CASCI mode, i.e.
                          # without the dynamic correlation dressing
dcd_natorbs false        # Calculate natural orbitals for each state and write
                          # them to disk
dcd_populations false     # Perform population analysis on the DCD-CAS(2) states
end

```

Note that the calculation of SSC requires the definition of an auxiliary basis set, since it is only implemented in conjunction with RI integrals. If `dcd_magnetization` is requested, the values for magnetic flux density and temperature to be used can be specified via the keywords `MAGTemperatureMIN`, `MAGTemperatureMAX`, `MAGTemperatureNPoints`, `MAGFieldMIN`, `MAGFieldMAX`, `MAGNpoints` of the `rel` subblock of the `%casscf` block (see section [Magnetization and Magnetic Susceptibility](#)). If the keyword `dcd_domagfield` is set to a number different than 0, the magnetic fields can be entered as a matrix of xyz coordinates (in Gauss), e.g.

```

%casscf
  dcdcas true
  ...
  dcd_domagfield 2
  dcd_magneticfields[0] = 10000, 0, 0
  dcd_magneticfields[1] = 0, 10000, 0
end

```

Furthermore, there is the keyword `DCD_EDIAG` that when running the DCD-CAS(2) code in CASCI mode works analogously to the keyword `EDiag` in the `soc` subblock of the `%mrci` block (see section [Zero-Field Splitting](#)). The energies should be entered in atomic units, e.g.

```

%casscf
  ...
  dcdcas true
  dcd_cascimode true
  dcd_soc true
  DCD_EDIAG[0] -2220.920028
  DCD_EDIAG[1] -2220.834377
  DCD_EDIAG[2] -2220.835871
  DCD_EDIAG[3] -2220.810290
  DCD_EDIAG[4] -2220.812293
  DCD_EDIAG[5] -2220.756732
end

```

3.22 Full Configuration Interaction

ORCA provides several exact and approximate approaches to tackle the full configuration interaction (FCI) problem. These methods are accessible via the CASSCF module (see Section *Complete and Incomplete Active Space Self-Consistent Field (CASSCF and RAS/ORMAS)*) or the ICE module (described in Section *Approximate Full CI Calculations in Subspace: ICE-CI*).

In the following, we compute the FCI energy of the lithium hydride molecule using the CASSCF module, where a typical input requires the declaration of an active space. The latter defines the number of active electron and orbitals, which are evaluated with the FCI ansatz. In the special case that all electrons and orbitals are treated with the FCI ansatz, we can use the keyword `DoFCI` in the `%CASSCF` block and let the program set the active space accordingly. In this example, we focus on the singlet ground state. Note that excited states for arbitrary multiplicities can be computed with the keywords `Mult` and `NRoots`. The FCI approach is invariant to orbital rotations and thus orbital optimization is skipped in the CASSCF module. Nevertheless, it is important to employ a set of meaningful orbitals, e.g. from a converged Hartree-Fock calculation, to reduce the number of FCI iterations.

```
# Hartree-Fock orbitals
!def2-tzvp RHF

*xyz 0 1
  Li 0 0 0
  H  0 0 1.597
*
```

The output of the Hartree-Fock calculation also reports on the total number of electrons and orbitals in your system (see snippet below).

Number of Electrons	NEL	4
Basis Dimension	Dim	20

In the given example, there are 4 electrons in 20 orbitals, which is a “CAS(4,20)”. Reading the converged RHF orbitals, we can start the FCI calculation.

```
!def2-tzvp extremescf

!moread
%moinp "RHF.gbw"

%maxcore 2000

%casscf
  DoFCI true # sets NEL 4 and NORB 20 in this example.
end

*xyz 0 1
  Li 0 0 0
  H  0 0 1.597
*
```

The output reports on the detailed CI settings, the number of configuration state functions (CSFs) and the CI convergence thresholds.

```
CI-STEP:
  CI strategy                ... General CI
  Number of multiplicity blocks ... 1
  BLOCK 1 WEIGHT= 1.0000
  Multiplicity                ... 1
  # (Configurations)          ... 8455
  # (CSFs)                    ... 13300
  # (Roots)                   ... 1
  ROOT=0 WEIGHT= 1.000000
```

(continues on next page)

(continued from previous page)

```
PrintLevel          ...      1
N(GuessMat)         ...     512
MaxDim(CI)          ...      10
MaxIter(CI)         ...      64
Energy Tolerance CI ...  1.00e-13
Residual Tolerance CI ...  1.00e-13
Shift(CI)           ...  1.00e-04
...
```

The program then prints the actual CI iterations, the final energy, and the composition of the wave function in terms of configurations (CFGs).

CAS-SCF ITERATIONS

```
MACRO-ITERATION      1:  
--- Inactive Energy E0 = 0.99407115 Eh  
--- All densities will be recomputed  
CI-ITERATION        0:  
-8.012799617       0.526896429727   (    0.25)  
CI-ITERATION        1:  
-8.047996328       0.001601312242   (    0.25)  
CI-ITERATION        2:  
-8.048134967       0.000022625293   (    0.25)  
CI-ITERATION        3:  
-8.048137773       0.000000462227   (    0.25)  
CI-ITERATION        4:  
-8.048137841       0.000000035496   (    0.25)  
CI-ITERATION        5:  
-8.048137845       0.000000001357   (    0.25)  
CI-ITERATION        6:  
-8.048137845       0.000000000254   (    0.25)  
CI-ITERATION        7:  
-8.048137845       0.000000000006   (    0.25)  
CI-ITERATION        8:  
-8.048137845       0.000000000001   (    0.25)  
CI-ITERATION        9:  
-8.048137845       0.000000000000   (    0.25)  
CI-PROBLEM SOLVED  
DENSITIES MADE  
  
<<<<<<<<<<<<<<<INITIAL CI STATE CHECK>>>>>>>>>>>>>>>  
  
BLOCK   1 MULT= 1 NROOTS= 1  
ROOT     0:  E=          -8.0481378449 Eh  
0.97242 [         0]: 22000000000000000000  
0.00296 [        99]: 20000002000000000000  
0.00258 [       89]: 20000010001000000000  
0.00252 [      85]: 20000020000000000000
```

Aside from energies, the CASSCF module offers a number of properties (g-tensors, ZFS, ...), which are described in Section [CASSCF Properties](#).

The exact solution of the FCI problem has very steep scaling and is thus limited to smaller problems (at most active spaces of 16 electrons in 16 orbitals). Larger systems are accessible with approximate solutions, e.g. with the density matrix renormalization group approach (DMRG), described in Section *Density Matrix Renormalization Group (DMRG)*, or the iterative configuration expansion (ICE) reported in Section *Approximate Full CI Calculations in Subspace: ICE-CI*. For fun, we repeat the calculation with the ICE-CI ansatz, which offers a more traditional approach to get an approximate full CI result.


```

!def2-TZVP ExtremeSCF

!moread
%moinp "RHF.gbw"

%maxcore 2000

%ice
  Nel 4
  Norb 20
end

*xyz 0 1
  Li 0 0 0
  H 0 0 1.597
*
```

The single most important parameter to control the accuracy is TGen. It is printed with the more refined settings in the output. We note passing that the wave function expansion and its truncation can be carried out in the basis of CSFs, configurations, or determinants. The different strategies are discussed in detail by Chilkuri *et al.* [443, 444].

```

ICE-CI:
  General Strategy          ... CONFIGURATIONS (all CSFs to a given CFG,
↳spin adapted)
  Max. no of macroiterations ... 12
  Variational selection threshold ... -1.000e-07
  negative! => TVar will be set to 1.000e-07*Tgen=1.000e-11
  Generator selection threshold ... 1.000e-04
  Excitation level          ... 2
  Selection on initial CSF list ... YES
  Selection on later CSFs lists ... YES

  ...

*****
* ICECI MACROITERATION 3 *
*****

# of active configurations = 2808
Initializing the CI          ... (CI/Run=3,2 UseCC=0)done ( 0.0 sec)
Building coupling coefficients ... (CI/Run=3,2)Calling BuildCouplings_RI
↳UseCCLib=0 DoRISX=0
CI_BuildCouplings NCFG= 2808 NORB=20 NEL=4 UseCCLib=0 MaxCore=2000
PASS 1 completed. NCFG= 2808 NCFGK= 8416 MaxNSOMOI=4 MaxNSOMOK=4
PASS 2 completed.
PASS 3 completed.
Memory used for RI tree      = 2.99 MB (av. dim= 35)
Memory used for ONE tree     = 1.32 MB (av. dim= 46)
Memory used for coupling coefficients= 0.01 MB
done ( 0 sec)
Now calling CI solver (4095 CSFs)

****Iteration 0****
Maximum residual norm : 0.000293130557

****Iteration 1****
Maximum residual norm : 0.000000565920

****Iteration 2****
Maximum residual norm : 0.000001755176

****Iteration 3****
```

(continues on next page)

(continued from previous page)

```
Maximum residual norm :      0.000000435942
Rebuilding the expansion space

****Iteration      4****

*** CONVERGENCE OF ENERGIES REACHED ***
CI problem solved in    0.4 sec

CI SOLUTION :
STATE   0 MULT= 1: E=       -8.0481340246 Eh W=   1.0000 DE= 0.000 eV           0.0 cm**-1
0.97249 : 22000000000000000000000000
Selecting new configurations ... (CI/Run=3,2)done (   0.0 sec)
# of selected configurations ...    2747
# of generator configurations ...     43 (NEW=1 (CREF=43))
Performing single and double excitations relative to generators ... done (   0.0
↪sec)
# of configurations after S+D ...    7038
Selecting from the generated configurations ... done (   0.1 sec)
# of configurations after Selection ...    2808
Root   0:      -8.048134025          -0.000000023          -8.048134048
==>>> CI space seems to have converged. No new configurations
maximum energy change ... 1.727e-05 Eh

***** ICECI IS CONVERGED *****

Initializing the CI ... (CI/Run=3,3 UseCC=0)done (   0.0 sec)
Building coupling coefficients ... (CI/Run=3,3)Calling BuildCouplings_RI_
↪UseCCLib=0 DoRISX=
CI_BuildCouplings NCFG= 2808 NORB=20 NEL=4 UseCCLib=0 MaxCore=2000
PASS 1 completed. NCFG= 2808 NCFGK= 8416 MaxNSOMOI=4 MaxNSOMOK=4
PASS 2 completed.
PASS 3 completed.
Memory used for RI tree =         2.99 MB (av. dim=    35)
Memory used for ONE tree =        1.32 MB (av. dim=    46)
Memory used for coupling coefficients=    0.01 MB
done (     0 sec)
Now calling CI solver (4095 CSFs)

****Iteration      0****
Maximum residual norm :      0.000000471011

****Iteration      1****

*** CONVERGENCE OF ENERGIES REACHED ***
CI problem solved in    0.1 sec

CI SOLUTION :
STATE   0 MULT= 1: E=       -8.0481340245 Eh W=   1.0000 DE= 0.000 eV           0.0 cm**-1
↪1
0.97249 : 22000000000000000000000000
```

With Hartree-Fock orbitals and the default settings, the ICE converges in 3 macro iterations to an energy of $-8.048134047513 E_h$. The deviation from the exact solution is just $3.8 \times 10^{-6} E_h$ in this example.

3.23 Molecular Mechanics

Since version 4.2 ORCA features its own independent Molecular Mechanics (MM) implementation.

The minimum input necessary for an MM treatment looks as follows.

```
!MM
%mm
  ORCAFFFilename "UBQ.ORCAFF.prms"
end
```

In this section we discuss usage of MM as well as basic keywords and options, i.e.

- with which other features MM calculations can be combined
- the basic structure of the ORCA Force Field File,
- which force fields are supported,
- how to generate the ORCA Force Field File,
- how to manipulate the ORCA Force Field File,
- how to speed up MM calculations,
- further MM options and keywords.

Further options important for QM/MM calculations will be discussed in section *Overview on ORCA's Multiscale Implementation*.

3.23.1 Combining the MM module with other ORCA Features

Molecular mechanics was introduced as a prerequisite for multiscale simulations within ORCA. But MM calculations can also be used as a standalone, e.g. for cheap preoptimizations.

Possible ORCA features are:

- **Single Point Calculations:** Simple MM calculations as well as *QM/MM calculations*
- **Optimization:** Use all kinds of *geometry optimizations* with all kinds of constraints, TS optimization, relaxed surface scans, and the ScanTS feature. Use the L-Opt and L-OptH features including the combination of all kinds of fragment optimizations (fix fragments, relax fragments, relax only specific elements in fragments, treat a fragment as a rigid body).
- **Transition States and Minimum Energy Paths:** Bond breaking and bond forming are not possible using MM methods, but conformational changes or dispersion driven reactions can be studied on MM-level only. For such tasks ORCA *Nudged-Elastic Band method* can be used.
- **Conformational Searches:** Conformational Searches can be run with ORCA's *GOAT* module.
- **Frequency Calculations:** Use regular *frequency calculations*. If required, ORCA automatically switches on the *Partial Hessian Vibrational Analysis* (PHVA) [531] calculation. (not tested for IONIC-CRYSTAL-QMMM)
- **Molecular Dynamics:** Use the *Molecular Dynamics* (MD) module for Born-Oppenheimer MD (BOMD) with MM.

Important

Active regions can also be defined for MM calculations. They have to be *defined* via the %qmmm block.

3.23.2 Molecular Mechanics Force Field in ORCA

For the MM part of the QM/MM calculation force-field parameters are necessary. ORCA has its own parameter file format (ORCA force field file - ORCAFF.prms), which includes the atom specific parameters for nonbonded interactions:

- partial charge
- LJ coefficients

and parameters for bonded interactions:

- bonds
- angles
- Urey-Bradley terms
- dihedrals
- impropers
- CMAP terms (cross-terms for backbone, currently not used)

Individual parameters, like e.g. atomic charge, equilibrium bond length and force constant, ..., can be conveniently modified directly within the ORCA Force Field File.

Supported Force Field Types

For the MM part of the QM/MM calculation force field parameters are necessary. ORCA has its own force field file format (stored in files called `basename.ORCAFF.prms`), which is used internally. ORCA provides tools to convert other popular force fields into the ORCA force field file format. Currently, the following formats are supported to be converted to the ORCA force field file format:

- **CHARMM psf files:** protein structure file from the CHARMM force field. The psf files can be easily prepared with the popular molecular visualizer [VMD](#), together with its extensions ([psfgen](#), [QwikMD](#), [fftk](#), which works together with ORCA). The psf file contains information on the atom types and on the bonded interactions of all atoms. It does, however, not contain the parameters that belong to these interactions. These parameters are stored in specific files, often ending with `prm`, but also `par` or `str`. The CHARMM parameter files come with VMD installation, can be directly [downloaded](#).
- **AMBER prmtop files:** topology files from the AMBER force field. Tutorials on how to generate AMBER prmtop files (for standard and non-standard molecules) can be found [here](#).
- **Open Force Field:** xml files from the [openforcefield initiative](#). With the openff-toolkit xml topology files (compatible with the AMBER force field) can be easily generated for small and medium-sized non-standard molecules. For a tutorial see [here](#).
- **Simple force field** for small to medium-sized molecules: Alternatively, the `orca_mm` module can generate a simple approximate ORCAFF.prms file. For more options, see [Create simple force field: makeff](#).

This concept has the following advantages:

- **Modification of force field parameters:** Atom and bond specific parameters can be easily modified within the ORCA force field file, allowing the user maximum flexibility in modifying the force field, which might be particularly useful for chemical systems like transition metal complexes.
- **Standard and Non-Standard Ligands:** Ligands can be easily and flexibly exchanged or added to the system, see [How to generate and manipulate the ORCA Force Field File](#).

How to generate and manipulate the ORCA Force Field File

Once a ORCAFF.prms file is available, it can be manipulated, i.e. split up into several parts for individual molecules, new ORCAFF.prms files can be generated for non-standard molecules, and individual ORCAFF.prms files can be merged, as described in the following:

Conversion from other formats to ORCAFF.prms: convff

The `orca_mm` module can convert psf and prn files (CHARMM), prmtop files (AMBER) or xml files (open force field from the openff toolkit, compatible to AMBER) to the ORCAFF file with the `-convff` flag. Input options are:

```
orca_mm -convff <optional:-verbose> <FFInput> <PSFFILE> <PRMFILE(S)>
```

Keywords:

```
<FFInput> = -CHARMM
<FFInput> = -AMBER
<FFInput> = -OPENMM
```

For CHARMM topologies, when a psf file is available for a system with standard residues, prepared by e.g. QwikMD, psfgen or other vmd tools, the conversion needs the psf plus the prn files as input:

CHARMM example:

```
orca_mm -convff -CHARMM 1C1E.psf par_all136_prot.prm toppar_water_ions_namd.str
```

ORCA can also convert Amber topologies to the ORCAFF file. Here, only the prmtop file is required:

AMBER example:

```
orca_mm -convff -AMBER complex.prmtop
```

ORCA can also convert xml files from the openff toolkit (AMBER compatible) to the ORCAFF file. Here, only the xml file is required:

OPENFF example:

```
orca_mm -convff -OPENMM complex.xml
```

Divide a force field file: splitff

If a ORCAFF.prms file should be subdivided into several files, e.g. if the psf file stems from QWikMD with non-standard molecules included, e.g. a ligand. In that case first the parameters of the ligand are split from the remaining system, next the ligand needs to be protonated, then a simple ORCAFF.prms file is generated via `orca_mm`'s `makeff` option, and finally the ligand's new ORCAFF.prms file is added to the main systems file via the above described `mergeff` option. Note that the file can only be split into files for nonbonded fragments.

Input options:

```
orca_mm -splitff <optional:-verbose> <ORCAFFFILE> <A1> <optional:A2> ...
```

Keywords:

```
<ORCAFFFILE> = ORCA forcefield file.
<A1>         = Atom number of first atom of fragment that should belong
               to a new ORCA forcefield file
<A2>         = Atom number of first atom of fragment that should belong
               to a new ORCA forcefield file
...          = More split atoms possible
Note that atoms start counting at 1.
```

Example:

```
orca_mm -splitff 1C1E_substrate_noH.ORCAFF.prms 7208
```

Merge force field files: mergeff

If several ORCAFF.prms files are available and should be merged for an ORCA calculation, e.g. for a standard plus a non-standard molecule.

Input options:

```
orca_mm -mergeff <optional:-verbose> <ORCAFFFILE1> <ORCAFFFILE2> ...
```

Keywords:

```
<ORCAFFFILE1> = First ORCA forcefield file
<ORCAFFFILE2> = Second ORCA forcefield file
...           = More ORCA forcefield files possible
```

Example:

```
orca_mm -mergeff 1C1E.ORCAFF.prms substrate_withH.ORCAFF.prms
```

Repeat force field files: repeatff

In case the same ORCAFF.prms file needs to be repeated multiple times, the repeatff functionality is available.

Input options:

```
orca_mm -repeatff <optional:-verbose> <ORCAFFFILE> <A>
<ORCAFFFILE>      = ORCA forcefield file.
<A>               = Factor (integer) defining how often this forcefield_
→file should be repeated.
```

Example:

```
orca_mm -repeatff methanol.ORCAFF.prms 580
```

This feature can be useful e.g. in the case of solvating a molecule, i.e. adding multiple copies of a solvent to a solute. First the solvent can be repeated N times, and subsequently the solute's prms file can be merged together with the solvent prms file.

Divide a PDB file: splitpdb

When splitting a ORCAFF.prms file, also splitting of the pdb file is required. The file can be split into an arbitrary number of individual files.

This can be useful together with the splitff command.

Input options:

```
orca_mm -splitpdb <optional:-verbose> <PDBFILE> <A1> <optional:A2> ...
```

Keywords:

```
<PDBFILE> = PDB file.
<A1>      = Atom number of first atom of fragment that should belong
            to a new ORCA forcefield file
<A2>      = Atom number of first atom of fragment that should belong
            to a new ORCA forcefield file
...       = More split atoms possible
Note that atoms start counting at 1.
```

Example:

```
orca_mm -splitpdb 1C1E_substrate_noH.pdb 7208
```

Merge PDB files: mergepdb

If several PDB files are available and need to be merged for an ORCA calculation, e.g. a protein and a ligand or multiple ligands, or a ligand that was first removed from a complex, then modified, and finally should get back into the complex PDB file.

This can be useful together with the mergeff command.

Input options:

```
orca_mm -mergepdb <optional:-verbose> <1PDBFILE> <2PDBFILE> ...
```

Keywords:

```
<1PDBFILE>    = First PDB file
<2PDBFILE>    = Second PDB file
...           = More PDB files possible
```

Example:

```
orca_mm -mergepdb 1C1E.pdb substrate_withH.pdb
```

Create simple force field: makeff

The orca_mm tool can generate an approximate force field for a molecule, storing it in ORCAFF.prms format. Here, the LJ coefficients are based on UFF parameters and the partial charges are based on a simple PBE or XTB calculation. The resulting topology is certainly not as accurate as an original CHARMM topology, but can still be used for an approximate handling of the molecule. Herewith, the molecule can be part of the QM region (having at least the necessary LJ coefficients), or part of the MM region as a non-active spectator - being not too close to the region of interest. In the latter case it is important that the molecule is not active, since bonded parameters are not available. However, it can still be optimized as a rigid body, see sections *Geometry Optimizations using the L-BFGS optimizer* and the usage in QM/MM calculations in section *Optimization with the Cartesian L-BFGS Minimizer*, on MM level, optimizing its position and orientation with respect to the specific environment.

Input options:

```
orca_mm -makeff <optional:-verbose> <XYZ/PDBFILE> <optional:-C CHARGE>
               <optional:-M MULT> <optional:-nproc N> <optional:-CHARGE_OPTIONS>
```

Keywords:

```
<CHARGE>      = charge of system
<MULT>        = multiplicity of system
<-nproc N>    = number of processors (Default 1)
<-CHARGE_OPTIONS> = Structure      Charge calculation
  <-PBE>      input              PBE
  <-PBEOpt>    PBE opt.          PBE
  <-PBEOptH>   PBE H-opt.        PBE
  <-XTB>       input              XTB
  <-XTBOpt>    XTB opt.          XTB
  <-XTBOptH>   XTB H-opt.        XTB
  <-XTBOptPBE> XTB opt.          PBE
  <-noChargeCalc> distribute net charge evenly

PBE Opt and SP level: RI-PBE/def2-SVP CPCM(Water), CHELPG charges
XTB Opt and SP level: GFN2-XTB GBSA(Water), Mulliken partial charges
```

Example:

```
orca_mm -makeff substrate_withH.xyz -M 2 -XTBOptPBE
```

Note that ORCA generates bonds based on simple distance rules, which enables ORCA to detect where to add link atoms between QM and MM atoms, see also section *Boundary Region*. But the user is advised to treat a molecule,

for which the ORCAFF.prms file was generated with the makeff option, either fully in the QM, or fully in the MM region, unless the charge distribution has been properly taken care of (due to the need of integer charges in QM and MM system).

Get standard hydrogen bond lengths: getHDist

For the definition of the link atoms standard bond lengths between C, N and O and hydrogen are directly set by ORCA but can be modified by the user, see section *Boundary Region*. If other atom types are on the QM side of the QM-MM boundary, their distance to the link atom has to be defined. In this case a file can be provided to ORCA which defines the standard bond length to hydrogen for all possible atoms. Such a file can be generated via the following command:

Input options:

```
orca_mm -getHDist <optional:-verbose> <XYZ/PDBFILE>
```

Example:

```
orca_mm -getHDist 1C1E.xyz
```

This file can then be modified, the required values can be added, and the resulting file can be defined as input for the QMMM calculation.

Create ORCAFF.prms file for IONIC-CRYSTAL-QMMM

For IONIC-CRYSTAL-QMMM calculations, section *IONIC-CRYSTAL-QMMM*, an ORCAFF.prms file with initial charges and connectivities is required. If you are not using the `orca_crystalprep` tool for setting up such calculations, see section *orca_crystalprep*, you can directly prepare the ORCAFF.prms file with the command:

```
orca_mm -makeff <XYZ/PDBFILE> -CEL <ELEMENT1> <OXIDATION_STATE1>
                                     -CEL <ELEMENT2> <OXIDATION_STATE2>
```

Keywords:

<ELEMENT1>	= element name
<OXIDATION_STATE1>	= formal oxidation state of element 1
...	= More elements possible

Example:

```
orca_mm -makeff na4cl4.xyz -CEL Na 1.0 -CEL Cl -1.0
```

Here, na4cl4.xyz is the supercell structure file (it can contain tens of thousands of atoms).

Note

For supercells with more complex oxidation states', e.g. Co_3O_4 , the ORCAFF.prms file can be generated conveniently via the *orca_crystalprep* tool.

3.23.3 Speeding Up Nonbonded Interaction Calculation

For MM calculations of very large systems with hundreds of thousands of atoms, and for QMMM calculations with fast QM methods (e.g. XTB, AM1) and / or very small QM systems, the computation of the nonbonded interactions can become a bottleneck. Different schemes for speeding up the calculation of nonbonded interactions are available within the ORCA MM implementation. Two schemes truncate long-range interaction, another scheme can be used for calculations with active regions, i.e. calculations where only a part of the system is active or optimized. For more information on active regions see section *Active Atoms in Combination with Optimization, Frequency Calculation, MD*.

Force Switching for LJ Interaction

With force switching for the LJ interaction (described in reference [532]) a smooth switching function is used to truncate the LJ potential energy smoothly at the outer cutoff distance `LJCutOffOuter`. If switching is set to false, the LJ interaction is not truncated at `LJCutOffOuter`. The parameter `LJCutOffInner` specifies the distance at which the switching function starts taking an effect to bring the van der Waals potential to 0 smoothly at the `LJCutOffOuter` distance, ensuring that the force goes down to zero at `LJCutOffOuter`, without introducing discontinuities. Note that `LJCutOffInner` must always be smaller than `LJCutOffOuter`.

```
%mm
SwitchForceLJ true      # Use the switch force scheme for the LJ interaction.
                        # Default: true.
LJCutOffInner 10.        # Distance (in Ang). Default: 10 Ang.
LJCutOffOuter 12.        # Distance (in Ang). Default: 12 Ang.
end
```

Force Shifting for Electrostatic Interaction

With force shifting for the electrostatic interaction (described in reference [532]) the electrostatic potential is shifted to zero at the cutoff distance `CoulombCutOff`. If shifting is set to false, the electrostatic interaction is not truncated at `CoulombCutOff`.

```
%mm
ShiftForceCoulomb true  # Use the shift force scheme for the Coulomb interaction.
                        # Default: true.
CoulombCutOff 12.       # Distance (in Ang). Default: 12 Ang.
end
```

Neglecting Nonbonded Interactions Within Non-Active Region

When using active regions (see section *Active Atoms in Combination with Optimization, Frequency Calculation, MD*) for optimizations and MD runs, the nonbonded interactions at the MM level can be neglected for those atom pairs, which are both non-active, without loss of accuracy for the results. Even relative energies between two structures are correct, if the atom positions of the non-active atoms are identical. For all other cases, i.e. if the positions of atoms in the non-active region differ, the full nonbonded interaction should be computed in the final single-point energy calculation. By default this option is switched off.

```
%mm
Do_NB_For_Fixed_Fixed true # Compute MM-MM nonbonded interaction also for
                           # non-active atom pairs. Default true.
end
```

3.23.4 Rigid Waters

As TIP3P water might have to be treated as rigid bodies due to its parametrization - please check out the specifics of the applied force field parametrization - we offer a keyword for the automated rigid treatment of all active MM waters. The following keyword applies bond and angle constraints to active MM waters in optimization as well as MD runs:

```
%mm
  Rigid_MM_Water false      # Default: false.
end
```

3.23.5 Available Keywords for the MM module

Here we list all keywords that are accessible from within the mm block and that are relevant to MM calculations.

Table 3.57: %mm block input keywords.

Key-word	Op-tions	Description
ShiftF	true	Long range electrostatics: Use the Shift Force scheme. Default true. The Shift Force scheme for the Coulomb interaction shifts the Coulomb potential such that it becomes zero at the cutoff distance.
Coulom	<real>	Distance (in Ang). Default: 12 Ang.
Switch	true	Use the Switch Force scheme. Default true. With the Switch Force scheme for the LJ interaction is unchanged up to LJCutOffInner. Between LJCutOffInner and LJCutOffOuter a smooth switching function is applied onto the LJ potential so that the force goes down to zero at LJCutOffOuter, without introducing discontinuities.
LJCutC	<real>	Distance (in Ang). Default: 10 Ang.
LJCutC	<real>	Distance (in Ang). Default: 12 Ang.
Dielec	<real>	Dielectric constant used in calc. of electrostatic interaction. Default: 1.
Coulom	<real>	Scaling factor for electrostatic interaction between 1,4-bonded atoms. Default: 1.
PrintI	<int>	Printing options: Can be 0 to 4, 0=nothing, 1=normal, ...

Some of the MM keywords can also be accessed via the qmmm block, see section [Additional Keywords](#).

Table 3.58: %mm block input keywords which can also be accessed from the %qmmm block.

Keyword	Options	Description
ORCAFFilen	"<filename>	Filename of the ORCA Force Field prms file.
Do_NB_For_F	true/false	Compute MM-MM nonbonded interaction also within non-active region, i.e. for non-active atom pairs. Default true. For GOAT and Crystal-QMMM default is false.
RIGID_MM_WA	true/false	Optimization and MD of active MM waters. Default false.
ExtendActiv	distance	Scheme to extend the active region to activeRegionExt. See also here . distance: Use a distance criterion (VDW distance plus Dist_AtomsAroundOpt).
	cov_bond	Add only atoms bonded covalently to active atoms.
	no	Do not add any atoms to activeRegionExt.
Dist_AtomsA	<real>	Distance to be added to VDW distance for ExtendActiveRegion. In Angstrom. Default 0.
OptRegion_F	{ints}	List of atoms to define the extended active region. Default: empty list.
PrintOptReg	true/false	Additionally print xyz and trj for opt region. Default true.
PrintOptReg	true/false	Additionally print xyz and trj for extended opt region. Default false.
PrintPDB	true/false	Additionally print pdb file for entire system, is updated every iteration for optimization. Default false.

STRUCTURE AND REACTIVITY

4.1 Geometry Optimizations

ORCA is able to calculate equilibrium structures (minima and transition states) using the quasi Newton update procedure with the well known BFGS update [533, 534, 535, 536, 537, 538], 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 (Swart, 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. the energy and gradient calculations done by ORCA.

The usage of analytic gradients is necessary for efficient geometry optimization. In ORCA 5.0, the following methods provide analytic first derivatives

- Hartree-Fock (HF) and DFT (including the RI, RIJK and RIJCOSX approximations)
- MP2, RI-MP2 and DLPNO-MP2
- TD-DFT for excited states
- CAS-SCF

When the analytic gradients are not available, it is possible to evaluate the first derivatives numerically by finite displacements. This is available for all methods.

Some methods for locating transition states (TS) require second derivative matrices (Hessian), implemented analytically for HF, DFT and MP2. Additionally, several approaches to construct an initial approximate Hessian for TS optimization are available. A very useful feature for locating complicated TSs is the Nudged-Elastic Band method in combination with the TS finding algorithm (NEB-TS, ZOOM-NEB-TS). An essential feature for chemical processes involving excited states is the conical intersection optimizer. Another interesting feature are MECP (Minimum Energy Crossing Point) optimizations.

For very large systems ORCA provides a very efficient L-BFGS optimizer, which makes use of the `orca_md` module. It can also be invoked via simple keywords described at the end of this section.

4.1.1 Geometry Optimizations

The use of the geometry optimization module is relatively straightforward.¹ Here are some typical examples:

```
! BLYP SVP 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
*
```

ORCA employs the *RIJ* by default. If this is not wanted, RI can be deactivated generally by using the `!NORI` keyword.

An optimization of the first excited state of ethylene:

```
! BLYP SVP OPT

%tddft
  IRoot 1
end

* xyz 0 1
  C   0.000000   0.000000   0.666723
  C   0.000000   0.000000  -0.666723
  H   0.000000  -0.928802   1.141480
  H  -0.804366  -0.464401  -1.341480
  H   0.000000   0.928802   1.241480
  H   0.804366   0.464401  -1.241480
*
```

4.1.2 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, Swart 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`, `Swart` or `Schlegel` in redundant internal coordinates, further shown in the keywords list found within Section *Geometry Optimization Keywords*. Since version 2.5.30, 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 the default for minimizations. The `Lindh` and `Schlegel` Hessian yield a similar convergence behaviour. There is also the option for the `Swart` model Hessian, which is less parametrized and should improve for weakly interacting and/or unusual structures. 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.

¹ 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.