### 3.14.11 Keywords

For completeness, the parameters that can be specified in the input block are summarized below:

```
%ice
        nel            8 # number of active electrons
        norb          23 # number of active orbitals
        nroots         1 # number of roots
        mult           1 # requested multiplicity
        irrep          0 # requested irrep (only supported for ICEType = CFGs)only␣
↪s
        tgen       1e-04 # generator threshold
        tvar       -1e-7 # negative -> 1e-7*tgen
        etol       1e-06 # convergence tolerance

        icetype     CFGs # The configuration based ICE-CI
                    CSFs # The CSF based ICE-CI
                    DETs # The CSF determinant based ICE-CI

        # algorithm details


        useMP2nat  false  # use MP2 natural orbitals
        useIVOs    false  # use improved virtual orbitals
        useQROs    false  # For UHF: use quasi-restricted MOs?
        integrals  exact  # exact or ri transformation
        CIMaxIter     64  # max. number of CI iterations in the Davidson procedure
        CINGuessMat  512 # size of the CI guess matrix in the Davidson procedure
        CIMaxDim      10  # max. size of expansion space in the Davidson procedure
        CIMode         3  # default=accelerated CI, other settings not recommended
        CSFCIwithRI    1  # 1=with RI, 0=without RI, use withour RI when norb >>␣
↪nel (e.g. (30e, 120o))
        CIPSIOrbSweepWindow 1 # Use in case of large size of the S+D list and␣
↪small available memory.
                              # MOs can be divided into chunks (e.g. MOblock = MO/
↪CIPSIOrbSweepWindow)

        # scratch space dimension (number of singles/double excitations per CSF)
        # If scratch is too small, the program may crash.
        CIBufferLength 10240 # ~82kB (factor 8/10^3 to get the memory in kB):

        # startup configurations(optional)
        refs { 2 2 2 2 0 }
             { 2 2 2 0 2 }
             end
        end

        # natural orbitals with the ICE ansatz
        NatOrbs 1 # generates the .cipsi.nat GWB file containing the natural␣
↪orbitals
```

### 3.14.12 A Technical Note: orcacclib

We should finally mention a technical aspect. The CI procedure in ICE-CI is based around the so-called one particle coupling coefficients

$$A_{pq}^{IJ} = \langle I|E_p^q|J\rangle$$

where $A_{pq}^{IJ}$ is a coupling coefficient, $I$ and $J$ are configuration state functions (CSFs) and $E_p^q$ is the spin-free excitation operator that promotes an electron from orbital $p$ to orbital $q$. The values of these coupling coefficients only depend on the logical relationship between the CSFs $I$ and $J$ but not on the absolute values of $I$, $J$, $p$, $q$. In fact, they only depend on the number of unpaired electrons in $I$ and the total spin $S$ that both CSFs refer to. Hence, prototype coefficients can be pre-tabulated. This is normally done in a CI run at the beginning of the run. However, in ICE-CI it may have to be repeated several dozen times and for large numbers of open shells (say 14), the process is time and memory consuming.

In order to ease the computational burden, we have provided a small utility program that tabulates the coupling coefficients for a given total spin $S$ (rather the multiplicity $M = 2S + 1$) and maximum number of open shells. This program is called `orcacclib`. It is called like:

```
orca_cclib Mult MaxNOpen
```

or - if you want to speed up the generation of the cclib:

```
mpirun -np 4 /full_path/orca_cclib_mpi Mult MaxNOpen cclib  # using 4 processes
```

It will produce a series of files orcacc.el.mult.nopen (electron density coupling coefficients) and orcacc.sp.mult.nopen (spin-density coupling coefficients) in the current directory. These files are binary files. They can be copied to an arbitrary directory. You instruct the program to read these coefficients (rather than to recalculate them all the time) by setting the path to this directory:

```
# My Job
! def2-SVP Auto-ICE
%cclib "/user/me/orca/cclib/orcacc"
```

The remaining part of the filename will be automatically added by the program. This option can save humongous amounts of time. The coupling coefficient library needs to be made for the desired multiplicities only once. The practical limit will be 14-16 open shells. If you are running the calculation on a cluster using some submit script, you have to ensure that the provided `cclib` path is accessible from the compute node.

## 3.15 Density Matrix Renormalization Group (DMRG)

*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. [445, 446, 447, 448, 460] The references given should be cited when using this part of the program. BLOCK is platform dependent and several crashes have been reported in the past. Please contact the BLOCK developers when you are affected.

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 $\pi$-active space of a $4\times25$ 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 $\pi$ and $\sigma$ 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!

### 3.15.1 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 [461].

### 3.15.2 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),
- 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 below by Chan and Sharma.

- G. K-.L. Chan and S. Sharma, *Ann. Rev. Phys. Chem.* **62**, 465 (2011),

### 3.15.3 Basic Usage

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 DMRGCI
   ...
   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 *Orbital optimization* on CASSCF). This is the responsibility of the user.

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 wavefunction 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 [462, 463, 464, 465]) 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 16000
%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 $\pi$-space (26 electrons in 24 orbitals), in a cc-pVDZ basis is given in Sec. *Example: Porphine \pi-active space calculation*.

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 `sweeptol` and `maxIter`.

### 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 `maxM=25` and the resulting sweep schedule), and to carry out a final single-point calculation using a larger value of `maxM`.

### 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 Keywords* 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
    switch_rst       1e-3
    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;
- switch_rst, defines the switching threshold of orbital gradient below which DMRG turns to onedot algorithm and restarts from previous operators and wavefunction. This is essential to avoid oscillation of energy values in the orbital optimization.

The default DMRG sweep schedule is selected automatically according to the choice of computational mode. By default two different sets of predefined schedules are supported for CASCI and CASSCF computations, respectively.

In CASCI mode, the default schedule corresponds to the following: starting from a given startM (where the default is 250 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 maxM has been reached.

In CASSCF mode, the orbital optimization requires much fewer renormalized states to converge the wavefunction with respect to orbital rotations. The default schedule therefore starts with startM (where the default is 25 and 2 sweeps), and increments by a factor of 2 every 2 sweeps util maxM is reached. The algorithm continues the sweep at maxM by decreasing the Davison tolerance sche_sweeptol and noise level sche_noise every 2 cycles by a factor of 10, until sche_sweeptol becomes smaller than sweeptol.

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.

### 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 present DMRG-SCF establishes the input order of active space orbitals according to their Hartree-Fock occupancy, even if these orbitals are ultimately canonical or split-localized canonical in nature. This is specified by hf_occ in which the Hartree-Fock occupancy is derived by default from the one-electron integrals. Other options for obtaining the occupancy are available (see *Keywords*).

Somet times the energy values produced from one SCF cycle to another may oscillate. Such a nonlinear numerical behaviour may occur typically by the last few iterations, most likely caused by the loss of a certain distribution of quantum numbers (eg, particle number, irrep symmetry and spin) in the blocking and decimation procedure due to incomplete many-body basis. On the other hand, the loss of quantum numbers is the main source of energy discontinuities on potential energy curves calculated by DMRG-SCF using a small number of renormalized states.

In the current release of DMRG-SCF implementation, the number of quantum states is locked to avoid these problems. The locking mechanism is turned on when the orbital gradient falls below a certain threshold defined by the keyword switch_rst (default: 0.001). The DMRG calculation then starts from previous operators and wavefunction in which a perturbative noise is not added. Locking quantum states and restaring DMRG wavefunction not only

ensures a smooth convergence towards the final energy but also minimizes the number of iterations. Note that the locking procedure introduces an arbitrariness to the final energy, when a very small $M$ is used, since the final digits of energy depend on where the locking begins. It is therefore not recommended to start locking too early in iterations which could trap the orbital solution in a local mimimum. Finally the quality of resulting orbitals can be checked by carrying out a DMRG calculation with sufficient renormalized states. Using the default value of `switch_rst` DMRG-SCF usually results in the orbitals that are good enough to reproduce the CASSCF energy.

### 3.15.4 Keywords

```
%casscf
 ...
 # enable DMRG as CI solver (mandatory)
 CIStep DMRGCI

 # refined optional settings for DMRG
 dmrg
 startM   25     # CASSCF mode: number of re-normalized states for a singlee root
          250    # CASCI mode: number of re-normalized states for a single root
 maxM     25     # CASSCF mode: number of re-normalized states for a singlee root
          250    # CASCI mode: number of re-normalized states for a single root
 DryRun   false  # just create an input for Block
 SweepTol 1e-9   # energy 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
 hf_occ   0 # user-defined initial Hartree-Fock occupancy manually
          1 # default: initial Hartree-Fock occupancy based on the values of
              the one-electron integrals
          2 # initial Hartree-Fock occupancy based on the energy ordering
              of canonical orbitals

 TwoDot_to_OneDot  1  # Switch from two-dot expressions to one-dot
 OneDot  0   # Only one-dot expressions. %In CASCI mode only.
 TwoDot  0   # Only two-dot expressions. %In CASCI mode only.
 switch_rst 1e-3   # Specify the threshold of orbital gradient below which DMRG
                     swithches to one-dot expression by reading previous␣
↪wavefunction.
 warmup  1   # wilson warm-up type
         2, 3 or 4   # n=3 is the default option.
                       The full configuration space of the n sites next to the␣
↪system
                       constitutes the environment states in the warm-up.
                       The remaining sites use the Hartree-Fock guess occupation
 nonspinadapted 0   # default: spin-adapted DMRG
                1   # non-spin-adatped DMRG in which the spin-density calculation
                      is available


 # Define a schedule for DMRG
 MaxIter   14   # Specify maximum number of iterations
 NSchedule -1   # default sweep schedule in CASSCF mode
            0   # default sweep schedule in CASCI mode
           >0   # Number of manual sweep 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␣
```

(continues on next page)

(continued from previous page)

```
↪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)

 # Define a separate maxM for DMRG-NEVPT2
 nevpt2_maxm 25   # set maximum number of renormalized states
                    for DMRG-NEVPT2 calculation (default: MaxM)

 end
end
```

### 3.15.5 Example: Porphine $\pi$-active space calculation

We provide a step-by-step basis on localizing the $\pi$-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
```

(continues on next page)

(continued from previous page)

```
H     -1.13385     -0.00000     0.00000
*
```

2. We then swap orbitals with $\pi$-character so they are adjacent to each other in the active space. ($\pi$ 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
    # Swap orbitals
    {70, 72}
    {65, 71}
    {61, 70}
    {59, 69}
    {56, 68}
    {88, 84}
    {92, 85}
    {93, 86}
    {96, 87}
    {99, 88}
    {102, 89}
    {103, 90}
    {104, 91}
    end
end
* xyz 0 1
[...]
*
```

3. After rotating the orbitals, we localize the 13 occupied $\pi$-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 $\pi$-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"
%MaxCore 16000

%casscf nel 26
norb 24
nroots 1
CIStep DMRGCI
end
* xyz 0 1
[...]
*
```

## 3.16 N-Electron Valence State Perturbation Theory (NEVPT2)

The N-electron valence state perturbation theory (NEVPT) belongs to the family of internally contracted methods and applies to a CASSCF/CAS-CI type reference wave functions. The NEVPT2 methodology developed by Angeli et al exists in two formulations namely the strongly-contracted NEVPT2 (SC-NEVPT2) and the partially contracted NEVPT2 (PC-NEVPT2). [466, 467, 468] Irrespective of the name "partially contracted" coined by Angeli et al, the latter approach employs a fully internally contracted wave function (FIC). Hence, we use the term "FIC-NEVPT2" in place of PC-NEVPT2. NEVPT2 has many desirable properties - among them:

- It is **intruder state free** due to the choice of the Dyall Hamiltonian [469] as the 0th order Hamiltonian.
- The **0th 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.
- "**strongly contracted**": Perturber functions only interact via their active part. Different subspaces are orthogonal and hence no time is wasted on orthogonalization issues.
- "**fully internally contracted**": Invariant to rotations of the inactive and virtual subspaces.

Both methods produces energies of similar quality as the CASPT2 approach.[470, 471] The strongly and fully internally contracted NEVPT2 methods are implemented in ORCA, along with a range of approximations that significantly enhance the methodology's scalability and make it highly attractive for large-scale applications. This includes the ability to handle *large basis sets*, *big molecules*, and *extended active spaces*.

In addition to corrections to the correlation energy, ORCA features a range of spectroscopic properties for NEVPT2, including UV, IR, CD, and MCD spectra, as well as EPR parameters. These properties are computed using the "quasi-degenerate perturbation theory" that is described in section *CASSCF Properties*. The NEVPT2 corrections enter as "improved diagonal energies" in this formalism. ORCA also offers the multi-state extension (*QD-NEVPT2* for the strongly contracted NEVPT2 variant.[472, 473] Here, the reference wave function is revised in the presence of dynamical correlation. For systems, where such reference relaxation is important, the computed spectroscopic properties will improve.

**NEVPT2 requires a single keyword on top of a working CASSCF input**.

```
!SC-NEVPT2    # for the strongly contracted NEVPT2
!FIC-NEVPT2   # for the fully internally contracted NEVPT2
!DLPNO-NEVPT2 # for the DLPNO variant of the FIC-NEVPT2
! ...
%casscf ...
```

Alternatively, the methods are called within the CASSCF block and detailed settings can be adjusted in the `PTSettings` subblock.

We will go through some of the *detailed setting* in the next few subsections. For historical reasons, a few features, such as the quasi-degenerate NEVPT2, are only available for the strongly contracted NEVPT2. As shown elsewhere, the strong contraction is not a good starting point for linear scaling approaches.[420] Thus newer additions such as the DLPNO and the F12 correction rely on the FIC variant. [474, 475, 476] Note that ORCA by default employs the frozen core approximation, which can be disabled with the simple keyword `!NoFrozenCore`. A complete description of the frozecore settings can be found in section *Frozen Core Options*.

### 3.16.1 A simple Example: $N_2$ Ground State

Let us consider the ground state of the nitrogen molecule as a simple example. After defining the computational details of our CASSCF calculation, we insert "`!SC-NEVPT2`" as simple input or specify "`PTMethod SC_NEVPT2`" in the `%casscf` block. Please note the difference in the two keywords' spelling: Simple input uses hyphen, block input uses underscore for technical reasons.

```
!def2-svp nofrozencore PAtom
%casscf nel  6
        norb 6
        mult 1
        PTMethod SC_NEVPT2  # SC_NEVPT2   for strongly contracted NEVPT2
                            # FIC_NEVPT2  for the fully internally contracted␣
↪NEVPT2
                            # DLPNO_NEVPT2 for the FIC-NEVPT2 with DLPNO
                            # DLPNO requires: trafostep RI and an aux basis
end

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

For better control over the program flow, it is recommended to split the calculation into two distinct parts. First, converge the CASSCF wave function and inspect the resulting orbitals. Then, in a second step, read the converged orbitals and run the actual NEVPT2 calculation.

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

...
PT2-SETTINGS:
A PT2 calculation will be performed on top of the CASSCF wave function (PT2 = SC-
↪NEVPT2)
...
------------------------------------------------------------
                          < 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
```

(continues on next page)

(continued from previous page)

```
  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 the SC-NEVPT2 approach lowers the energy by 150 mEh. ORCA also prints the contribution of each "excitation class V" to the first order wave function. We note that in the case of a single reference wave function corresponding to a CAS(0,0), the `V0_ijab` excitation class produces the exact MP2 correlation energy.

In section *Example: Breaking Chemical Bonds* the dissociation of the $N_2$ molecule has been investigated with the CASSCF method. Inserting `PTMethod SC_NEVPT2` into the `%casscf` block we obtain the NEVPT2 correction as additional information.

```
! def2-svp nofrozencore
%casscf nel  6
        norb 6
        mult 1
        PTMethod SC_NEVPT2
end

# scanning from the outside to the inside
%paras
        R = 2.5,0.7, 30
end

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

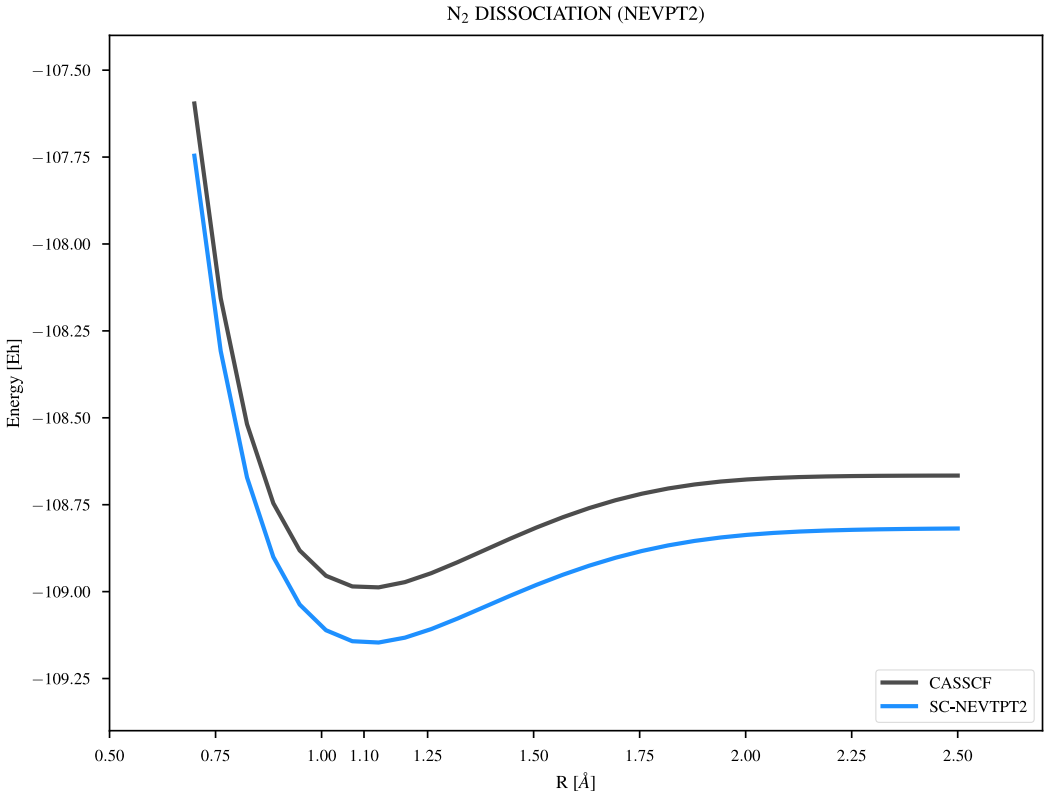


Fig. 3.37: Potential Energy Surface of the $N_2$ molecule from CASSCF(6,6) and NEVPT2 calculations (def2-SVP).

All of the options available in CASSCF can in principle be applied to NEVPT2. Since NEVPT2 is implemented as a submodule of CASSCF, it will inherit all settings from CASSCF (`!tightscf`, `!UseSym`, `!RIJCOSX`,...).

### 3.16.2 RI, RIJK and RIJCOSX Approximation

Setting the RI approximation on the *CASSCF* level, will set the RI options for NEVPT2 respectively. The three index integrals are computed and partially stored on disk. Three index integral with two internal labels are kept in main memory. The two-electron integrals are assembled on the fly. The auxiliary basis must be large enough to fit the integrals appearing in the CASSCF orbital gradient/Hessian and the NEVPT2 part. The auxiliary basis set of the type /J does not suffice here.

```
%casscf
  ...
  TrafoStep RI    # enable RI approximation in CASSCF and NEVPT2
  PTMethod  SC_NEVPT2 # or the NEVPT2 approach or your choice
end
```

Additional speedups can be achieved by approximating the Fock operator using the `!RIJCOSX` or `!RIJK` techniques. When using `RIJCOSX`, an additional auxiliary basis must be specified for the `AuxJ` auxiliary basis slot. For more information on basis set slots, see section *Orbital Basis Sets*.

```
#RIJCOSX one-liner
! def2-svp def2/J RIJCOSX def2-svp/C

# Commented out: alternative definition via %basis block
#%basis
```

(continues on next page)

(continued from previous page)

```
#auxJ "def2/J"
#auxC "def2-svp/C"
#end
```

Whereas the RIJK requires a single auxiliary basis set (AuxJK slot), that is large enough to fit integrals in the Fock-matrix construction, orbital gradient/Hessian and the correlation part. In contrast to COSX, the calculation can also be carried out in conv mode (storing the AO integrals on disk).

```
#RIJK one-liner: !conv is recommended.
! def2-svp def2/JK RIJK

# Commented out: Alternative definition via %basis block
#%basis
#auxJK "def2/JK"
#end
```

The described methodology allows the computation of systems with up to 2000 basis functions. Even larger molecules are accessible in the framework of DLPNO-NEVPT2 described in the next subsection. Several examples can be found in the CASSCF tutorial.

### 3.16.3 Beyond the RI approximation: DLPNO-NEVPT2

For systems with more than 80 atoms, we recommend the DLPNO-NEVPT2 method, which combines the DLPNO ansatz with the FIC-NEVPT2 approach.[474] This method is particularly effective, as it recovers 99.9% of the FIC-NEVPT2 correlation energies, even for large systems. The input structure for DLPNO-NEVPT2 is similar to that of its parent FIC-NEVPT2 method. Below, we provide an example input for the Fe(II)-complex depicted in Fig. 3.38, where the active space consists of the metal-3d orbitals. The example takes about 9 hour (including 3 hour for one CASSCF iteration) using 8 cores (2.60GHz Intel E5-2670 CPU) for the calculation to finish. A detailed description of the DLPNO-NEVPT2 methodology can be found in our article.[474]. A number of DLPNO specific parameters can be set in the PTSettings sub-block. The respective keywords are otherwise identical with the *DLPNO Coupled Cluster ansatz described elsewhere in the manual*. The most important parameters are TCutPNO, TCutMKNand TCutDO. However, it is recommended to use the simple keywords !TightPNO, !NormalPNO or !LoosePNO to adjust the accuracy. In contrast to the canonical FIC-NEVPT2 method, the DLPNO amplitude equations require the solution of a set of linear equations. The latter are solved iteratively via a DIIS solver, which comes with its own set of control parameters.

```
# DLPNO-NEVPT2 calculation for quintet state of FeC72N2H100
!PAL8 def2-TZVP def2/JK
!moread noiter
%moinp "FeC72N2H100.gbw-CASSCF"
%MaxCore 8000
%casscf
  nel 6
  norb 5
  mult 5

  # mandatory for DLPNO-NEVPT2:
  PTMethod DLPNO_NEVPT2  # will automatically set `TrafoStep RI`


  # optional settings reapting the defaults
  PTSettings
    # DLPNO specific settings (see DLPNO-NEVPT2 article):
    TCutPNO      1e-8 # cutoff for PNO occupation numbers
    TCutPNO_Core 1e-2 # cutoff for PNO occupation numbers for core orbitals
    TCutMKN      1e-3 # cutoff for Mulliken populations in the PNO integral␣
↪transformation
    TCutDO       1e-2 # cutoff for the sparse map construction of domain
```

(continues on next page)

(continued from previous page)

```
    TCutDOij     1e-5 # cutoff for the differential overlap matrix of ij pairs
                      # in the prescreening step

    # DLPNO DIIS solver settings:
    MaxIter    30    # maximum number of iterations
    MaxDIIS    7     # DIIS max. dimension
    DIISDamp1  0.5   # damping of the residual before DIIS is switched ON
    DIISDamp2  0     # damping of the residual after DIIS is switched ON
    LevelShift 0.2   # LevelShift during update
    LMP2TolE   1e-7  # energy convergence criteria
    LMP2TolR   5e-7  # residual convergence criteria
    LMP2Damp   1.0   # damping for the LMP2 amplitude update

  end
end
*xyz 0 5 FeC72N2H100.xyz
```

Just like RI-NEVPT2, the calculations requires an auxiliary basis. The aux-basis should be of /C or /JK type (more accurate). Aside from the paper of Guo et al,[474] a concise report of the accuracy can be found in the CASSCF tutorial, where we compute exchange coupling parameters. Note that in the snippet above, we have commented out the default setting in the `PTSettings` sub-block.

As mentioned earlier, the CASSCF step can be accelerated with the RIJK or RIJCOSX approximation. Both options are equally valid for the DLPNO-NEVPT2. The RIJK variant typically produces more accurate results than RIJCOSX. The input file is almost the same as before, except for the keyword line:

```
# The combination of RIJK with DLPNO-NEVPT2
!PAL8 def2-TZVP def2/JK conv RIJK
```

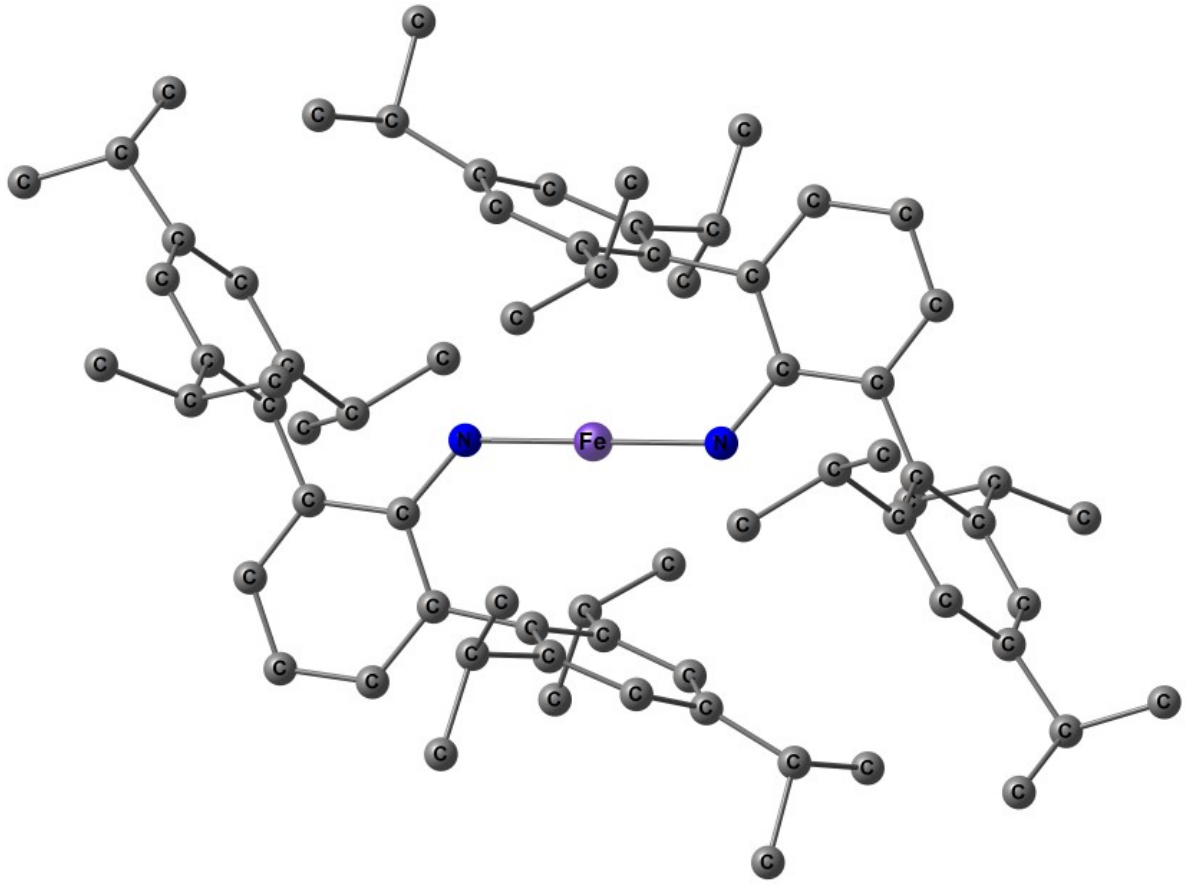


Fig. 3.38: Structure of the $FeC_{72}N_2H_{100}$

### 3.16.4 Explicit Correlation: NEVPT2-F12 and DLPNO-NEVPT2-F12

Like in the single-reference MP2 theory, the NEVPT2 correlation energy converges slowly with the basis set. Aside from basis set extrapolation, the R12/F12 method are popular methods to reach the basis set limit. For comparison of F12 and extrapolation techniques, we refer to the study of Liakos et al.[388] ORCA features an F12 correction for the FIC-NEVPT2 wave function using the RI approximation.[475] The RI approximation is mandatory as the involved integrals are expensive. In complete analogy to the single reference MP2-F12, the input requires an *F12 basis*, an *F12-cabs basis* and a sufficiently large RI basis (/JK or /C). It should be noted, that the CASSCF wave function itself is also affected by basis set incompleteness. When `F12` is invoked, a perturbative correction for CASSCF reference, denoted as CABS singles correction, is added.[477]

```
# aug-cc-pvdz/C used as RI basis
 ! cc-pvdz-F12 aug-cc-pvdz/C cc-pvdz-f12-cabs
 %casscf nel 8
         norb 6
         mult 3,1

         # mandatory:
         TrafoStep RI          # RI approximation must be on for F12
         PTMethod FIC_NEVPT2   # FIC-NEVPT2
         PTSettings
           F12 true            # request the F12 correction and CABS singles␣
↪correction.
         end
 end
 *xyz 0 3
  O      0.0 0.0 0.0
```

(continues on next page)

(continued from previous page)

```
 O      0.0 0.0 1.207
*
```

A linear scaling version of NEVPT2-F12, the DLPNO-NEVPT2-F12, allows to tackle systems with several thousand of basis functions.[475, 476] With the exception of the `DLPNO_NEVPT2` keyword, the input structure is otherwise identical to NEVPT2-F12 method.

```
# aug-cc-pvdz/C used as RI basis
! cc-pvdz-F12 aug-cc-pvdz/C cc-pvdz-f12-cabs
%casscf nel 8
        norb 6
        mult 3,1

        # mandatory:
        TrafoStep RI           # RI approximation must be on for F12
        PTMethod DLPNO_NEVPT2  # DLPNO variant
        PTSettings
          F12 true             # request the F12 correction and CABS singles␣
↪correction.
        end
end
*xyz 0 3
 O      0.0 0.0 0.0
 O      0.0 0.0 1.207
*
```

Note that the DLPNO-NEVPT2-F12 algorithm is unitary invariant with respect to subspace rotation of inactive and active orbitals. By tightening the DLPNO truncation thresholds (`TCutPNO`,`TCutDO`, `TCutCMO` and `TCutDOij`), the canonical NEVPT2-F12 can be reproduced, even with localized internal and active molecular orbitals.

### 3.16.5 Handling of Reduced Density Matrices

NEVPT2 calculations involve the computation of higher order reduced density matrices (RDMs), which can quickly become a major bottleneck for active spaces of CAS(10,10) and larger. Using Dyall's rank-reduction scheme,[478] the standard implementation involves at most the fourth order reduced density matrix (4-RDM).[468] In contrast, omitting the rank-reduction scheme, the full rank NEVPT2 (FR-NEVPT2) requires the 5-RDM and its contractions.[427] The latter should be used in the context of *approximate CAS-CI reference wave functions such as the ICE*.

ORCA uses an efficient reformulation of the NEVPT2 method, that avoids the explicit 4-RDM and 5-RDM construction.[479] The basic idea is similar to a recent development reported by Sokolov and coworkers.[480] The algorithm requires a memory-intense intermediates, that scales with the number of configuration state functions (CSFs) in the CAS-CI. With the option `D4Step fly`, the memory demands are reduced at the expense of computation time omitting the aforementioned intermediate.

```
%casscf
  ...
  PTMethod     FIC_NEVPT2  # or SC_NEVPT2

  # detailed settings (optional)
  PTSettings
  D4Step efficient # efficient: calling the default RDM handling
                   # fly      : slower but more economic code
                   #
                   # Cu4 : cumulant approximation of the 4-RDM
                   # Cu34: cumulant approximation of the 3-RDM and 4-RDM
  D4TPre 1e-12     # default PS approximation 4-RDM
  D3TPre 1e-14     # default PS approximation 3-RDM
```

(continues on next page)

(continued from previous page)

```
  imaginary 0.0    # imaginary shift (only for FIC-NEVPT2)
end
```

### PS Approximation (default)

To accelerate the evaluation of the higher-order RDMs, ORCA uses a prescreening approximation (PS) for the reference wave function.[422] Only configurations with a weight larger than a specified parameter `D4TPre` are taken into account. The same reduction is available for the third order density matrix using the keyword `D3TPre`. Both parameters can be adjusted within the `PTSettings` sub-block of the CASSCF module. The exact NEVPT2 energy is recovered with the parameters set to zero. The approximation is available for all variants of NEVPT2 (SC, FIC and DPLNO-FIC). The default thresholds, `D4TPre 1e-12` and `D3TPre=1e-14`, are chosen very conservative as lowering might lead to false intruder states.[422, 427]

These approximations naturally affect the "configuration RI" as well. In this context, it should be noted that a configuration corresponds to a set of CSFs 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
```

### Cumulant Approximation (optional)

Large computational savings can be achieved with the cumulant expansion, which have been recently re-evaluated.[422] The results should be treated with care as false intruder states can emerge.[481] In these cases, the imaginary level shift is the only mitigation tool.[482] Note that the imaginary shift is only implemented for FIC-NEVPT2.

```
PTSettings
  D4Step Cu4     # Cu4  : approximate the 4-RDM
                 # Cu34 : approximate the 3-RDM and 4-RDM
  imaginary 0.0 # imaginary shift (only for FIC-NEVPT2)
end
```

## 3.16.6 False Intruder States and Imaginary Shift

False intruder states are introduced by crude approximations of the reduced density matrices or the reference wave function.[422, 427] The occurance is highly system-specific. Indications are unreasonable correlation energy contributions from the 1-hole excitation class (denoted as "V_i" or "ITUV") or the 1-particle excitation class (denoted as "V_a" or "TUVA"), e.g., positive or large correlation energies compared to the 2-hole-2-particle excitation class (denoted as "V_ijab" or "IJAB"). For transparency, ORCA prints warnings when the Koopmans energies or the energy denominators are negative.

```
...
WARNING: Denominator is negative for DEpsilon ITUV i=2 mu=1 : -344.12813836782095
↪< 0                                                                          ␣
↪                                    |
WARNING: Denominator is negative for DEpsilon ITUV i=2 mu=2 : -228.47393616112132
↪< 0                                                                          ␣
↪                                    |
WARNING: Denominator is negative for DEpsilon ITUV i=2 mu=3 : -218.57612632153200
↪< 0                                                                          ␣
↪                                    |
WARNING: Denominator is negative for DEpsilon ITUV i=2 mu=4 : -127.85082085035380
```

(continues on next page)

(continued from previous page)

```
↪< 0                                                                                ␣
↪                                    |                                              ␣
Skipped =       126 of          432 LowestEV=-3.843455e+02 ... done in        0.1␣
↪sec                                                                                ␣
↪                                    |                                              ␣
 Reached max. number of intruder state warning (5). Thus, warnings were suppressed␣
↪for this class to limit the output!                                                ␣
↪                                    |                                              ␣
                                                                                    ␣
↪                                                                                   ␣
↪                                    |                                              ␣
                                                                                    ␣
↪                                                                                   ␣
↪                                    |                                              ␣
WARNING: Koopmans matrices have at least one negative eigenvalue (-1.519147e+06)   ␣
↪                                                                                   ␣
↪                                    |                                              ␣
         Please check your results carefully.
```

To avoid flooding the output, the number of warnings are limited by `MaxWarnings` (default=5). When faced with these warnings, it is advised to inspect the approximations of the reduced density matrices. For example, in the case of the PS approximation, tightening the thresholds (`D4TPre=1e-14`) should immediately alleviate the issue. The cumulant approximation in particular is prone to false intruder states and should thus be avoided.

As a last resort, an imaginary shift can be added to mitigate intruder states.[482] Note that imaginary shifts (default=0.0 ) are restricted to the canonical FIC-NEVPT2 implementation.

```
PTSettings
  imaginary 0.2 # imaginary shift (only for FIC-NEVPT2) . default = 0.0
  MaxWarnings 5 # Max number of denominator warnings before going silent.
end
```

### 3.16.7 Large Active Spaces: ICE-NEVPT2

Standard CAS-CI solver can perform routine calculation up to a CAS(14,14) active space. Larger active spaces are accessible with the *ICE ansatz*.[443] With a recent extension of the FIC-NEVPT2 methodology, that also allows an ICE reference wave functions, calculation with active spaces up to CAS(34,34) are possible.[459] As shown in the paper by Guo et al, for an approximate CAS-CI solution, the canonical implementation, that uses Dyall's rank reduction scheme, cannot be applied without introducing false intruder states. Hence, for an ICE reference wave function, ORCA defaults to the full-rank NEVPT2 formulation (FR-NEVPT2), where additional 5-RDM contributions are accounted for. The latter are enabled with `CASCI_Approx true`. While the default ICE parameter `TVar` is designed for the ICE-CASSCF-Iterations, it is less ideal for the NEVPT2 approach. Here, the computation of the higher-order RMDs is a major bottleneck and not feasible with the same threshold. As demonstrated by Guo et al, a parameter of `TVar 1e-7` (=TGen x $10^{-3}$) already leads to reasonable results that balance accuracy and efficiency. Following the reciple from the paper, it is recommended to split the calculation into two parts. First converge the ICE-CASSCF solution with the default parameters and inspect the resulting orbitals. Then, in a second step, run the FIC-NEVPT2 with a larger `TVar` reading the converged orbitals from the first step. For these type of calculation, we recommend to use the `D4Step fly` algorithm as it avoids memory-intense intermedidates processing the RDMs.

```
# Starting with converged CASSCF orbitals
!MORead NoIter
%moinp "converged.ice-casscf.gbw"
%casscf
  ...
  # mandatory keywords for ICE-NEVPT2
  CIStep ICE
  PTMethod FIC_NEVPT2
```

(continues on next page)

(continued from previous page)

```
  CI
    TVar 1e-7 # ICE parameter TVar for ICE-NEVPT2 runs.
              # TVar ~ TGen x 1e-3 - this is larger than the default.
  end


  # optional settings (repeating the defaults)
  PTSettings
    D4Step fly        # memory saving algorithm for larger active spaces
    CASCI_Approx true # setting FR-NEVPT2
    D4TPre 0.0        # recommendation: disabling PS approximation. No benefit␣
↪from screening here.
    D3TPre 1e-14      # recommendation: default PS approximation for 3-RDM.
  end
```

### 3.16.8 Large Active Spaces: DMRG-NEVPT2

An alternative for larger active spaces is offered by the DMRG ansatz. It is possible to run DMRG-NEVPT2 calculations for the FIC-NEVPT2 and SC-NEVPT2 ansatz using the methodology developed by the Chan group.[483] Here, the reduced density matrices are provided by the BLOCK program. It should be noted that the latter is platform dependent and several crashes have been reported in the past. For troubleshooting, please contact the BLOCK developers directly. Aside from the usual *DMRG input*, the program requires an additional parameter (`nevpt2_MaxM`) in the DMRG block.

```
%casscf
  cistep DMRGCI
  %dmrg
    ...
    nevpt2_MaxM 2000 # (see Note below)
  end
  PTMethod SC_NEVPT2 # or FIC_NEVPT2
end
```

### 3.16.9 Selecting or Specific States for NEVPT2

ORCA by default computes all states defined in the CASSCF block input with the NEVPT2 approach. There are cases, where this is not desired and the user wants to skip some of these states. The input mask of `SelectedRoots[block]` allows to select only few states for the computation, where "block" refers to numbering of the multiplicity blocks. The enumeration of blocks and roots starts with 0.

```
!NEVPT2 ...
%casscf
  ...
  MULT   3,1 # multiplicity block
  NRoots 2,3 # number of roots for the MULT blocks

  # detailed settings (optional) for the PT2 approaches
  PTSettings
    # option to skip the PT2 correction for all states except for the ones
    # specified for a selected multiplicity block and root
    # selectedRoots[block]  = root number counting from 0.
    selectedRoots[1]=0,1 # compute MULT=1 root=0,1 and skip all others.

  end
end
```