

These keywords represent the recommended way to control the accuracy of DLPNO calculations as follows. Manual changing of thresholds beyond these specifying these keywords is usually discouraged.

```
# Tight settings for increased accuracy, e.g. when investigating
# weak interactions or conformational equilibria
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD(T) TightPNO TightSCF

# OR: Default settings (no need to give NormalPNO explicitly)
# Useful for general thermochemistry
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD(T) NormalPNO TightSCF

# OR: Loose settings for rapid estimates
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD(T) LoosePNO TightSCF

%maxcore 2000

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

Since ORCA 4.0, the linear-scaling DLPNO implementation described in reference [401] is the default DLPNO algorithm. However, for comparison, the first DLPNO implementation from references [397] and [400] can still be called by using the DLPNO2013 prefix instead of the DLPNO- prefix.

```
# DLPNO-CCSD(T) calculation using the 2013 implementation
! cc-pVTZ cc-pVTZ/C DLPNO2013-CCSD(T)

# DLPNO-CCSD(T) calculation using the linear-scaling implementation
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD(T)

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

Until ORCA 4.0, the “semi-canonical” approximation is used in the perturbative triples correction for DLPNO-CCSD. It was found that the “semi-canonical” approximation is a very good approximation for most systems. However, the “semi-canonical” approximation can introduce large errors in rare cases (particularly when the HOMO-LUMO gap is small), whereas the DLPNO-CCSD is still very accurate. To improve the accuracy of perturbative triples correction, since 4.1, an improved perturbative triples correction for DLPNO-CCSD is available, DLPNO-CCSD(T1)[404]. In DLPNO-CCSD(T1), the triples amplitudes are computed iteratively, which can reproduce more accurately the canonical (T) energies.

It is necessary to clarify the nomenclature used in ORCA input files. The keyword to invoke “semi-canonical” perturbative triples correction approximation is DLPNO-CCSD(T). While, the keyword of improved iterative approximation is DLPNO-CCSD(T1). However, in our recent paper[404], the “semi-canonical” perturbative triples correction approximation is named DLPNO-CCSD(T0), whereas the improved iterative one is called DLPNO-CCSD(T). Thus, the names used in our paper are different from those in ORCA input files. An example input file to perform improved iterative perturbative triples correction for DLPNO-CCSD is given below,

```
# DLPNO-CCSD(T1) calculation using the iterative triples correction
! cc-pVTZ cc-pVTZ/C DLPNO-CCSD(T1)

%mdci
  TNOSCALES      10.0 #TNO truncation scale for strong triples,
  ↳TNOSCALES*TCutTNO.
                        Default setting is 10.0
  TNOSCALEW      100.0 #TNO truncation scale for weak triples, TNOSCALEW*TCutTNO
                        Default setting is 100.0
  TriTolE        1e-4 # (T) energy convergence tolerance
                        Default setting is 1e-4
%end
* xyz 0 1
```

(continues on next page)

(continued from previous page)

```
... (coordinates)
*
```

Since ORCA 4.2, the improved iterative perturbative triples correction for open-shell DLPNO-CCSD is available as well. The keyword of open-shell DLPNO-CCSD(T) is the same as that of the closed-shell case.

Since ORCA 4.0, the high-spin open-shell version of the DLPNO-CISD/QCISD/CCSD implementations have been made available on top of the same machinery as the 2016 version of the RHF-DLPNO-CCSD code. The present UHF-DLPNO-CCSD is designed to be an heir to the UHF-LPNO-CCSD and serves as a natural extension to the RHF-DLPNO-CCSD. A striking difference between UHF-LPNO and newly developed UHF-DLPNO methods is that the UHF-DLPNO approach gives identical results to that of the RHF variant when applied to closed-shell species while UHF-LPNO does not. Usage of this program is quite straightforward and shown below:

```
# (1) In case of ROHF reference
! ROHF DLPNO-CCSD def2-TZVPP def2-TZVPP/C TightSCF TightPNO

# (2) In case of UHF reference, the QROs are constructed first and used for
# the open-shell DLPNO-CCSD computations
! UHF DLPNO-CCSD def2-TZVPP def2-TZVPP/C TightSCF TightPNO

# (3) In case that UKS is specified, the QROs are constructed first and used as
# "unconverged" UHF orbitals for the open-shell DLPNO-CCSD computations.
# This approach is useful when the converged UHF wavefunction is qualitatively
# wrong but the UKS wavefunction is not
! UKS CAM-B3LYP DLPNO-CCSD def2-TZVPP def2-TZVPP/C TightSCF TightPNO
```

Note

DLPNO-CISD/QCISD/CCSD methods are dedicated to closed-shell and high-spin open-shell species, but not spin-polarized systems (e.g. open shell singlets or antiferromagnetically coupled transition metal clusters). Performing DLPNO-CISD/QCISD/CCSD calculations upon open shell singlet UHF/UKS wavefunctions will give results resembling the corresponding closed shell singlet calculations, because the DLPNO calculations will be done on the closed-shell determinant composed of the QRO orbitals. Similarly, calculations of spin-polarized systems other than open shell singlets may give qualitatively wrong results. Here, the UHF-LPNO-CCSD method or the DLPNO-NEVPT2 are better alternatives.

The same set of truncation parameters as closed-shell DLPNO-CCSD is used also in case of open-shell DLPNO. The open-shell DLPNO-CCSD produces more than 99.9 % of the canonical CCSD correlation energy as in case of the closed-shell variant. The computational timings of the UHF-DLPNO-CCSD and RIJCOSX-UHF for linear alkane chains in triplet state are shown in [Fig. 3.10](#).

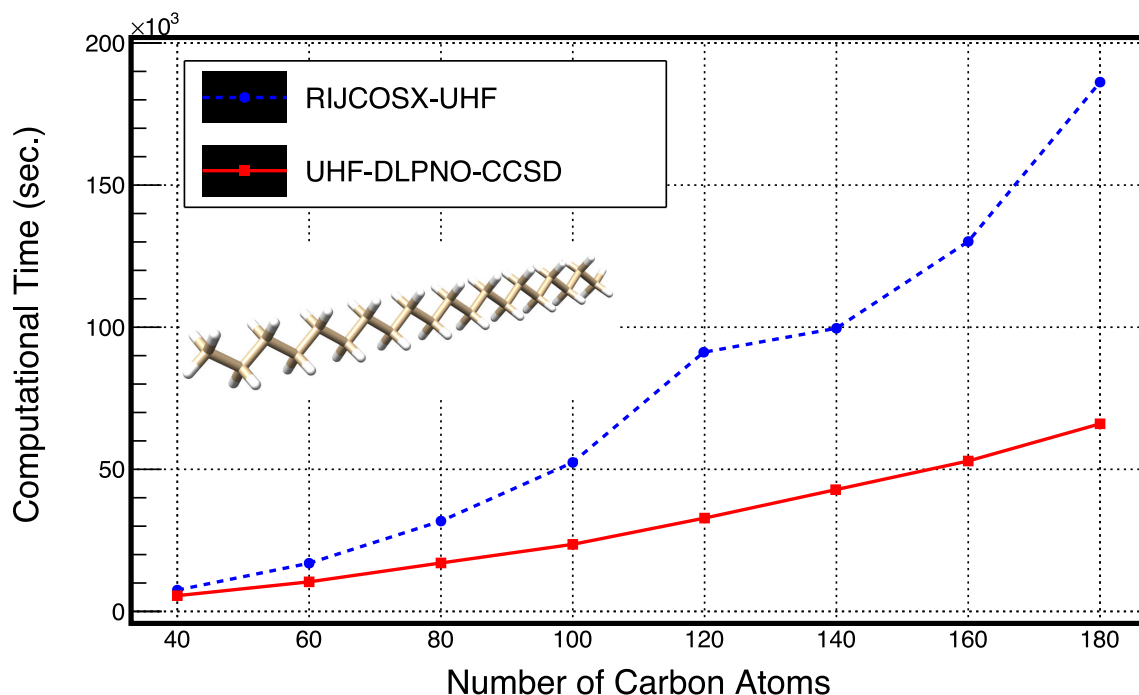


Fig. 3.10: Computational times of RIJCOSX-UHF and UHF-DLPNO-CCSD for the linear alkane chains (C_nH_{2n+2}) in triplet state with def2-TZVPP basis and default frozen core settings. 4 CPU cores and 128 GB of memory were used on a single cluster node.

Although those systems are somewhat idealized for the DLPNO method to best perform, it is clear that the preceding RIJCOSX-UHF is the rate-determining step in the total computational time for large examples. In the open-shell DLPNO implementations, SOMOs are included not only in the occupied space but also in the PNO space in the preceding integral transformation step. This means the presence of more SOMOs may lead to more demanding PNO integral transformation and DLPNO-CCSD iterations. The illustrative examples include active site model of the [NiFe] Hydrogenase in triplet state and the oxygen evolving complex (OEC) in the high-spin state, which are shown in Figures 7 and 8, respectively. With def2-TZVPP basis set and NormalPNO settings, a single point calculation on [NiFe] Hydrogenase (Fig. 3.11) took approximately 45 hours on a single cluster node by using 4 CPU cores of Xeon E5-2670. A single point calculation on the OEC compound (Fig. 3.12) with the same computational settings finished in 44 hours even though the number of AOs in this system is even fewer than the Hydrogenase: the Hydrogenase active site model and OEC involve 4007 and 2606 AO basis functions, respectively. Special care should be taken if the system possesses more than ten SOMOs, since inclusion of more SOMOs may drastically increase the prefactor of the calculations. In addition, if the SOMOs are distributed over the entire molecular skeleton, each pair domain may not be truncated at all; in this case speedup attributed to the domain truncation will not be achieved at all.

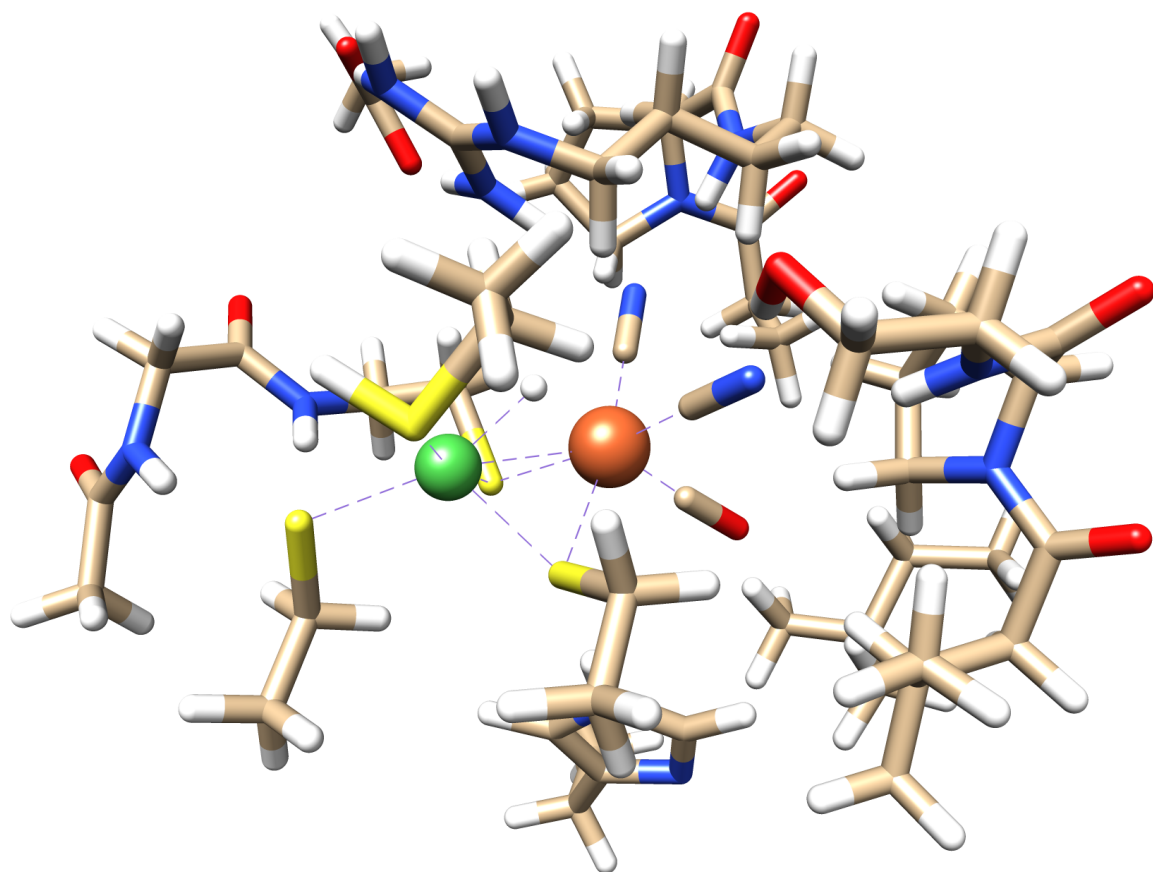


Fig. 3.11: Ni-Fe active center in the [NiFe] Hydrogenase in its second-coordination sphere. The whole model system is composed of 180 atoms.

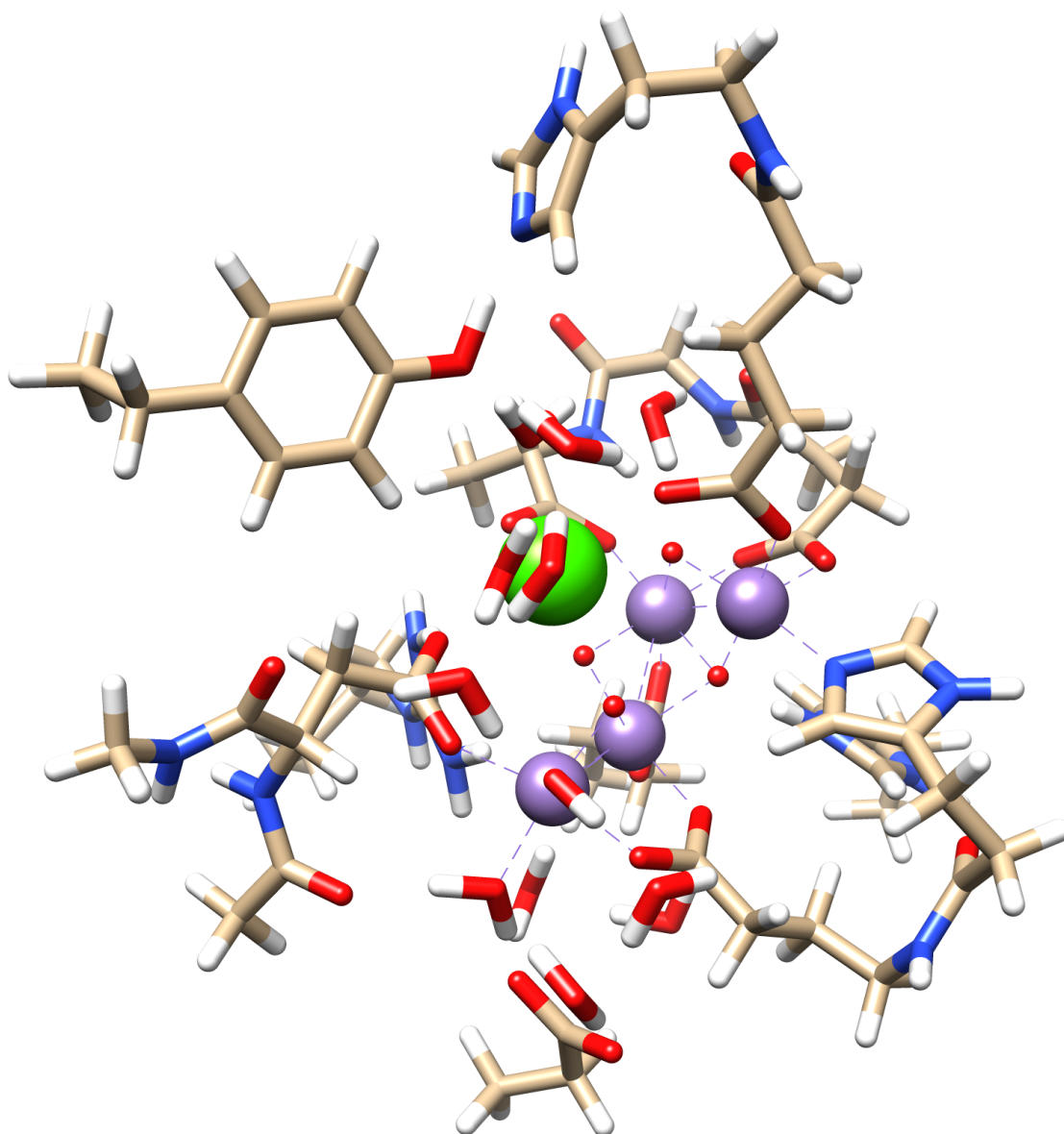


Fig. 3.12: A model compound for the OEC in the S_2 state of photosystem II which is composed of 238 atoms. In its high-spin state, the OEC possesses 13 SOMOs in total.

Calculation of the orbital-unrelaxed density has been implemented for closed-shell DLPNO-CCSD. This permits analytical computation of first-order properties, such as multipole moments or electric field gradients. In order to reproduce conventional unrelaxed CCSD properties to a high degree of accuracy, tighter thresholds may be needed than given by the default settings. Reading of the reference[402] is recommended. Calculation of the unrelaxed density is requested as usual:

```
%MDCI Density Unrelaxed End
```

There are a few things to be noticed about (D)LPNO methods:

- The DLPNO and LPNO methods obligatorily make use of the RI approximation. Hence, a correlation fit set must be provided.
- The DLPNO-CCSD(T) method is applicable to closed-shell or high-spin open-shell species. When performing DLPNO calculations on open-shell species, it is always better to have UCO option: If preceding SCF converges to broken-symmetry solutions, it is not guaranteed that the DLPNO-CCSD gives physically meaningful results.
- The open-shell version of the DLPNO approach uses a different strategy to the LPNO variant to define the

open-shell PNOs. This ensures that, unlike the open-shell LPNO, the PNO space converges to the closed-shell counterpart in the closed-shell limit. Therefore, in the closed-shell limit, the open-shell DLPNO gives identical correlation energy to the RHF variant up to at least the third decimal place. The perturbative triples correction referred to as, (T), is also available for the open-shell species.

- When performing a calculation on the open-shell species with either of canonical/DLPNO methods on top of the Slater determinant constructed from the QROs, special attention should be paid on the orbitals energies of those QROs. In some cases, the orbitals energy of the highest SOMO appear to be higher than that of the lowest VMO. Similarly to this, the orbital energy of the highest DOMO may appear to be higher than that of the lowest SOMOs. In such cases, the CEPA/QCISD/CCSD iteration may show difficulty in convergence. In the worst case, it just diverges. Most likely, in such cases, one has to suspect the charge and multiplicity might be wrong. If they are correct, you may need much prettier starting orbitals and a bit of good luck! Apart from a careful choice of starting orbitals (in particular, DFT orbitals can be used in place of the default HF orbitals if the latter have qualitative deficiencies, including but not limited to severe spin contamination), changing the maximum DIIS expansion space size (MaxDIIS) and the level shift (LShift) in the %mdci block may alleviate the convergence problems to some extent.
- DLPNO-CCSD(T)-F12 and DLPNO-CCSD(T1)-F12 (iterative triples) are available for both closed- and open-shell cases. These methods employ a perturbative F12 correction on top of the DLPNO-CCSD(T) correlation energy calculation. The F12 part of the code uses the RI approximation in the same spirit as the canonical RI-F12 methods (refer to section *Explicitly Correlated Methods: F12-MP2 and F12-CCSD(T)*). Hence, they should be compared with methods using the RI approximation for both CC and F12 parts. The F12 correction takes only a fraction (usually 10-30%) of the total time (excluding SCF) required to calculate the DLPNO-CCSD(T)-F12 correlation energy. Thus, the F12 correction scales the same (linear or near-linear) as the parent DLPNO method. Furthermore, no new truncation parameters are introduced for the F12 procedure, preserving the black-box nature of the DLPNO method. The F12D approximation is highly recommended as it is computationally cheaper than the F12 approach which involves a double RI summation. Keywords: DLPNO-CCSD(T)-F12D, DLPNO-CCSD(T)-F12, DLPNO-CCSD(T1)-F12D, DLPNO-CCSD(T1)-F12, DLPNO-CCSD-F12D, DLPNO-CCSD-F12.
- There are three thresholds that can be user controlled that can all be adjusted in the %mdci block: (a) T_{CutPNO} controls the number of PNOs per electron pair. This is the most critical parameter and has a default value of 3.33×10^{-7} . (b) T_{CutPairs} controls a perturbative selection of significant pairs and has a default value of 10^{-4} . (c) T_{CutMKN} is a technical parameter and controls the size of the fit set for each electron pair. It has a default value of 10^{-3} . All of these default values are conservative. Hence, no adjustment of these parameters is necessary. All DLPNO-CCSD truncations are bound to these three truncation parameters and should almost not be touched (Hence they are also not documented :)).
- The preferred way to adjust accuracy when needed is to use the “LoosePNO/NormalPNO/TightPNO” keywords. In addition, “TightPNO” triggers the full iterative (DLPNO-MP2) treatment in the MP2 guess, whereas the other options use a semicanonical MP2 calculation. Table 3.40 and Table 3.41 contain the thresholds used by the current (2016) and old (2013) implementations, respectively.
- Potential energy surfaces are virtually but not perfectly smooth (like any method that involves cut-offs). Numerical gradient calculations have been attempted and reported to have been successful.
- The DLPNO methods do work together with RIJCOSX, RI-JK and also with ANO basis sets and basis set extrapolation. They also work for conventional integral handling.
- The methods behave excellently with large basis sets. Thus, they stay efficient even when large basis sets are used that are necessary to obtain accurate results with wavefunction based *ab initio* methods. This is a prerequisite for efficient computational chemistry applications.
- For LPNO-CCSD, calculations with about 1000 basis functions are routine, calculations with about 1500 basis functions are possible and calculations with 2000-2500 basis functions are the limit on powerful computers. For DLPNO-CCSD much larger calculations are possible. There is virtually no crossover and DLPNO-CCSD is essentially always more efficient than LPNO-CCSD. Starting from about 50 atoms the differences become large. The largest DLPNO-CCSD calculation to date featured >1000 atoms and more than 20000 basis functions!
- Using large main memory is not mandatory but advantageous since it speeds up the initial integral transformation significantly (controlled by “MaxCore” in the %mdci block, see section *Local correlation (DLPNO)*).
- The open-shell versions are about twice as expensive as the corresponding closed-shell versions.

- Analytic gradients are not available.
- An unrelaxed density implementation is available for closed-shell DLPNO-CCSD, permitting calculation of first-order properties.

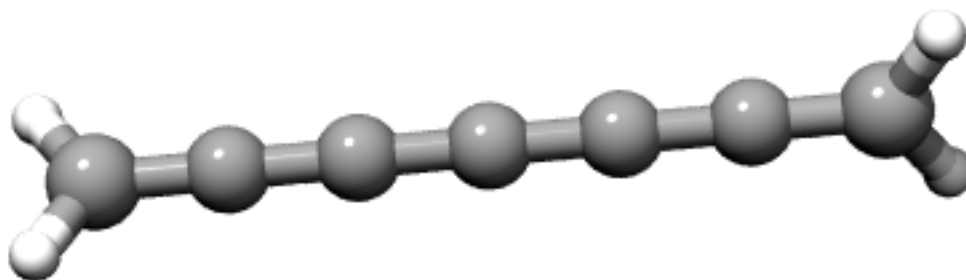
Table 3.40: Accuracy settings for DLPNO coupled cluster (current version).

Setting	T_{CutPairs}	T_{CutDO}	T_{CutPNO}	T_{CutMKN}	MP2 pair treatment
LoosePNO	10^{-3}	2×10^{-2}	1.00×10^{-6}	10^{-3}	semicanonical
NormalPNO	10^{-4}	1×10^{-2}	3.33×10^{-7}	10^{-3}	semicanonical
TightPNO	10^{-5}	5×10^{-3}	1.00×10^{-7}	10^{-3}	full iterative

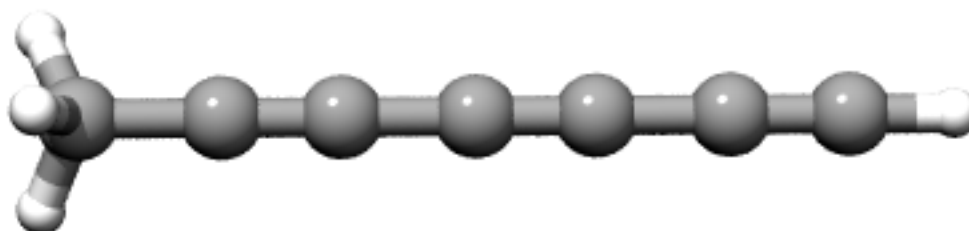
Table 3.41: Accuracy settings for DLPNO coupled cluster (deprecated 2013 version).

Setting	T_{CutPairs}	T_{CutPNO}	T_{CutMKN}	MP2 pair treatment
LoosePNO	10^{-3}	1.00×10^{-6}	10^{-3}	semicanonical
NormalPNO	10^{-4}	3.33×10^{-7}	10^{-3}	semicanonical
TightPNO	10^{-5}	1.00×10^{-7}	10^{-4}	full iterative

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



Isomerizes to:



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

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

The calculations are typical in the sense that: (a) the LPNO methods provide answers that are within 1 kcal/mol of the canonical results, (b) CEPA approximates CCSD(T) more closely than CCSD. The speedups of a factor of 2 – 5 are moderate in this case. However, this is also a fairly small calculation. For larger systems, speedups of the LPNO methods compared to their canonical counterparts are on the order of a factor >100–1000.

Keywords

Given these explanations the various cut-off parameters that can be controlled in LPNO and DLPNO calculations should be understandable and are listed below. We emphasize again that only the three thresholds T_{CutPairs} , T_{CutPNO} and T_{CutMKN} should be touched by the user, unless very specific questions are addressed. The recommended way to control the accuracy of calculations is to specify “TightPNO”, “NormalPNO” or “LoosePNO” keywords, rather than to change numeric values of cutoffs. Individual thresholds should normally not be changed, as the defaults are sensible and lead to good cost/performance ratios.

```
%mdci TcutPairs      1e-4 # cut-off for the pair truncation
      TcutPNO       3.33e-7 # cut-off for the PNO truncation
      TcutDO        1e-2 # cut-off for the DLPNO domain construction
      TcutMKN       1e-3 # cut-off for the local fit
                        # for DLPNO2013: also domain construction
                        # remaining options, tied to the three main cut-offs,
                        # EXPERTS ONLY!

      Localize      true # flag for using localized orbitals
      LocMet        AHFB # Localization method.
                        # Options: PM, FB, IAOIBO, IAOBOYS, NEWBOYS, AHFB
      LocTol        1e-6 # Absolute threshold for the localization procedure
                        # Automatically adjusted by default.
      LocTolRel      1e-8 # Relative threshold for the localization procedure
      LocMaxIter     128 # Maximum number of localization iterations
      LocRandom      1 # default, take random seed for any localization
                        # For internal orbitals: choose best of 32

↪localizations
                        # Switched off for AHFB
                        0 # take constant seed for any localization (for

↪testing)
      LocNAttempts   np # number of localization attempts
                        # default: number of processes, minimum 8, if
                        # randomize true
                        # 1, if randomize false
                        # any number larger or equal np, if randomize true

      PNONorm        MP2Norm # default, old IEPANorm can also be used
                        # (near identical results)
      NrMP2Pairs_Trip 1 # number of MP2 pairs to be included in the triples
                        # calculation
      PAOOverlapThresh 1e-8 # generation of non-redundant PAOs from redundant

↪ones
      UseFullLMP2Guess false # Use iterative full LMP2 (for DLPNO)
      SinglesFockUsePNOs true # compute the Singles Fock matrix (SFM) in PNOs.
                        # DLPNO2013: default for SinglesFockUsePNOs is false,
                        # by default RIJCOSX is used for the SFM, except

↪when
                        # RIJK is given. In that case the RIJK-SFM is

↪used.
      LMP2MaxIter     50 # max no of iterations in the MP2 equations
      LMP2TolE        1e-7 # LMP2 energy convergence tolerance
      LMP2TolR        5e-7 # LMP2 residual convergence tolerance
      LMP2ScaleTcutPNO # PNO cutoff for LMP2 is: TcutPNO*LMP2ScaleTcutPNO
                        # Default: TcutPNO (DLPNO-MP2) / TcutPNO (DLPNO-CCSD)

↪with
                        # respective TcutPNOs specific to Loose/Normal/

↪TightPNO
```

(continues on next page)

(continued from previous page)

LMP2FCut	1e-5	# LMP2 neglect cut-off for off-diagonal Fock matrix elements
TCutTNO	0	# Cut-off for triples natural orbitals (0=automatic)
TCutPNOSingles	-1	# -1= use 0.03*TCutPNO
TCutPreScr	-1	# -1= use 0.01*TCutPairs for multipole estimate based screening
TCutDeloc	0.1	# delocalization threshold for specification of extended domains. Necessary because PAOs are not strictly localized
TCutOSV	1e-6	# orbital specific virtuals used for pre-screening. No critical
TScaleMP2Pairs	0.1	
TScaleMKNStrong	10	
TScaleMKNWeak	100	

For larger systems and tighter thresholds the disk I/O of a DLPNO calculation may become challenging. In this case, it might be useful to keep some integrals in memory, if enough RAM is available. With the following flag

```
%mdci
  StorageType Shared
end
```

ORCA will try to store certain much-used integrals in shared memory. If the amount of memory is not sufficient, ORCA will fall back to on-disk storage. **NOTE:** This flag will only work if all processes work on the same node.

IMPORTANT NOTE REGARDING ORBITAL LOCALIZATION

- **Localized orbitals for DLPNO are obtained via the Foster-Boys method with an augmented Hessian converger (AHFB) by default.**
- The localization tolerance (LocTol) is coupled to the SCF gradient tolerance (TolG) with a constant factor by default. Selecting specific SCF convergence settings (such as TightSCF) therefore also ensures obtaining a set of appropriately well converged localized orbitals. This can be overridden by setting a different value for LocTol.
- An important feature of the augmented Hessian converger is that it systematically approaches a local maximum of the localization function (even though convergence to the global maximum cannot be guaranteed). As opposed to that, the conventional localization method (FB) may stop, for example, at a saddle point. In bad cases, this can lead to deviations of several kJ/mol in the DLPNO energy. Likewise, it can contribute towards lack of reproducibility of results.
- No similar procedure has been implemented for the other localization methods (such as Pipek-Mezey) yet. The same problems as with the FB converger can occur in these cases.
- No randomization is used for the AHFB converger.

The old default orbital localization settings of ORCA 4.0 can be reproduced with the following options:

```
%MDCI
  LocMet      FB
  LocTol      1.0e-6
  LocRandom 1
End
```

Regarding the methods that employ randomization (FB, PM, IAOIBO, IAOBOYS) only, the following notes apply:

- Generally, better DLPNO results are obtained when several runs of localization are undertaken using different initial guesses. The different initial guesses are obtained using randomization (LocRandom).
- However, randomization of the initial guess can lead to differently localized MOs in different calculations. This can yield non-identical correlation energies, varying in the sub-kJ/mol range, for different runs on the same machine.

- In order to yield identical correlation energy results, randomization can be switched off (LocRandom 0). However, switching off randomization only leads to identical results on the same machine, but can still lead to slightly different results (in the sub-kJ/mol range) on different machines.
- Reproducibility of the correlation energy is expected to increase further if LocNAttempts is set to higher values.

The input below shows how to perform a DLPNO calculation with settings that exactly reproduce the canonical RI-MP2 result. They are not recommended for production use, but merely to show that if the local approximations are pushed, then the result coincides with the canonical one. If one would set T_{CutPNO} to zero this would give canonical RI-CCSD. However, this is an absurdly inefficient calculation and hence not done.

```
#
! def2-SV(P) def2/JK def2-SVP/C RI-JK DLPNO-CCSD VeryTightSCF RI-MP2

# obtain a result that only contains errors from the PNO approximation
# but no others
%mdci TCutPairs 0
      TCutMKN 0
      UseFullLMP2Guess true
      LMP2FCut 1e-9
      LMP2MaxIter 25
      LMP2TolE 1e-10
      LMP2TolR 1e-11
      PAOOverlapThresh 1e-9
end

! Bohrs
* xyz 0 1
  C -1.505246952209632 1.048213673267046 -3.005665895986369
  C 1.289678561934891 0.246429688933291 -3.259735682020124
  C 2.834670835163566 1.157307360133605 -0.990383454919828
  C 1.924119415395082 -0.128330938291771 1.465070676514038
  C -0.931529472233802 -0.722841293992075 1.397639867298547
  C -2.347670084056626 1.213332291655600 -0.217984867773136
  H 2.084955694093313 0.973408301535989 -5.037750251258102
  H 1.426532559234904 -1.831017720289521 -3.371063003813707
  H -1.795307501459984 2.891278294563413 -3.927855043896308
  H -2.709613973668925 -0.308515546176734 -4.026627646697411
  H -4.404246093821399 0.941639912907262 -0.071175054238094
  H -1.962867323232915 3.122079490952855 0.528101313545138
  H -1.245096579039474 -2.621186110634707 0.594784162223769
  H -1.699155144887690 -0.782162821007662 3.328959985756973
  H 2.347109421287126 1.104305785540561 3.087624818244846
  H 2.990679065503112 -1.888017241218143 1.775287572161196
  H 4.862301668284708 0.796425411350593 -1.279131939569907
  H 2.634027658640572 3.226752635113244 -0.827936424652650
*
```

Including (semi)core orbitals in the correlation treatment

In some chemical applications some or all of the chemical (semi)core electrons must be included in the correlation treatment. In this case, it is necessary to tighten the TCutPNO thresholds for electron pairs in which chemical (semi)core electrons are involved. This is now the default in DLPNO calculations.

For instance, one can decide to switch off the frozen-core approximation and include all the electrons in the correlation treatment. In this case, the program will use tighter thresholds by default for all electron pairs and Singles that involve chemical core electrons. Note that, in this case, the use of properly optimized basis functions for correlating the inner electrons is highly recommended.

```
! DLPNO-CCSD(T) def2-SVP def2-SVP/C NoFrozenCore
%mdci TSCALEPNO_CORE 0.01 # scaling factor for TCutPNO for electron pairs and
```

(continues on next page)

(continued from previous page)

```

# Singles involving chemical core electrons
end
* xyz 0 1
  Ti  0.0001595288    0.0000775041    0.0000000000
  F   1.7595996122    0.0000634675   -0.0000000011
  F  -0.5865076471    1.6586935196    0.0000000018
  F  -0.5866248292   -0.8294172469   -1.4362516915
  F  -0.5866248311   -0.8294172443    1.4362516907
*
```

Another option is to choose the involved chemical core electrons by using an energy window. In this way all electron pairs and Singles that involve chemical core electrons, which are in the defined energy window, are affected by TScalePNO_CORE.

```

! DLPNO-CCSD(T) def2-SVP def2-SVP/C
%method
  FrozenCore FC_EWIN
end
%mdci
  EWIN -40, 10000
end
* xyz 0 1
  Ti  0.0001595288    0.0000775041    0.0000000000
  F   1.7595996122    0.0000634675   -0.0000000011
  F  -0.5865076471    1.6586935196    0.0000000018
  F  -0.5866248292   -0.8294172469   -1.4362516915
  F  -0.5866248311   -0.8294172443    1.4362516907
*
```

A summary with the number of electrons affected by TScalePNO_Core for the examples just discussed is shown in Table 3.42.

Table 3.42: Number of chemical core electrons included in the DLPNO calculation and affected by TScalePNO_Core for the TiF₄ examples

Keyword	Chemical Core Electrons		Valence Electrons
	Frozen	Included ^a	
FrozenCore (default)	18	0	40
NoFrozenCore	0	18	40
EWIN -40, 10000	16	2	40

^a using TScalePNO_Core.

By default, ORCA provides a chemical meaningful definition for the number of electrons which belong to the chemical core of each element. As already discussed, these default values define which pairs are affected by TScalePNO_Core. However, the user can modify the number of chemical core electrons for a specific element via the NewNCore keyword.

```

! DLPNO-CCSD(T) def2-SVP def2-SVP/C NoFrozenCore
%method
  NewNCore Ti 8 end
end
* xyz 0 1
  Ti  0.0001595288    0.0000775041    0.0000000000
  F   1.7595996122    0.0000634675   -0.0000000011

```

(continues on next page)

(continued from previous page)

F	-0.5865076471	1.6586935196	0.0000000018
F	-0.5866248292	-0.8294172469	-1.4362516915
F	-0.5866248311	-0.8294172443	1.4362516907
*			

In the previous example, the number of chemical core electrons for Ti has been fixed to 8.

Starting from ORCA 6.0, in DLPNO calculations, tightened TCutPNO thresholds are used by default for “semicore” electron pairs involving the 3s and 3p orbitals of first-row transition metals.[414] This improves not only the accuracy of the results noticeably but also the efficiency of the computations as the system size grows (ref.). To reproduce results obtained with earlier versions, the number of semicore orbitals on first-row transition metals needs to be set to zero (the old default) instead of eight (the current default), as done below for a Zn atom with the NewNSemiCore keyword in the method block:

```
%method NewNSemiCore Zn 0 end end
```

Note

- Of course, if electrons are replaced by ECPs, they are not included in the correlation treatment.
- If ECPs are used, the number for NewNCore has to include the electrons represented by the ECPs as well. E.g. if an element is supposed to have 60 electrons in the ECP and additional 8 electrons should be frozen in the correlation calculation, NewNCore should be 68.
- The different sets of orbitals (chemical core electrons included in the correlation treatment and valence electrons) are localized separately in order to avoid the mixing of core and valence orbitals.

3.10.11 Multi-Level Calculations

In many applications events are investigated that are located in a relatively small region of the system of interest. In these cases, combined quantum-mechanics/molecular-mechanics (QM/MM) approaches have been proved to be extremely useful, especially in the modeling of enzymatic reactions. The basic idea of any QM/MM method is to treat a small region of the system at the QM level and to use an MM treatment for the remaining part of the system. Alternatively, QM/QM methods, where different parts of a system are studied at various quantum mechanical levels, are also available. Quantum mechanical methods are more computationally demanding than the molecular mechanics treatment, and this limits the applicability of all-QM approaches. Nevertheless, QM/QM methods retain some strong advantages over QM/MM schemes. For instance, force field parameters for the molecular mechanics part of the calculation are not necessary, and thus there are no restrictions on the type of chemical systems that can be treated. Moreover, problems usually caused by boundaries between QM and MM parts do not occur. Finally, the accuracy of an all-QM calculation is expected to be higher compared to the accuracy of QM/MM approaches, that is limited by the MM treatment.

In ORCA, the different methods that can be used in a QM/QM calculations are:

- DLPNO-CCSD(T) with TightPNO thresholds
- DLPNO-CCSD(T) with NormalPNO thresholds
- DLPNO-CCSD(T) with LoosePNO thresholds
- DLPNO-CCSD
- DLPNO-MP2
- HF

The user can define an arbitrary number of *fragments* in the input, the level of theory to be used for each fragment and for the interaction between different fragments. Localized molecular orbitals are then assigned to a given fragment on the basis of their Mulliken populations.

The following example shows the calculation of a benzene dimer, for which the individual monomers are calculated on MP2 level, and the interaction between the two monomers is calculated on TightPNO DLPNO-CCSD(T) level. More realistic use cases are discussed in ref. [355].

```
! DLPNO-CCSD(T) Def2-SVP Def2-SVP/C
%mdci
  UseFullLmp2Guess True
  TightPNOFragInter {1 2}
  MP2FragInter {1 1} {2 2}
end

*xyz 0 1
C(1)      1.393      0.000      0.0
H(1)      2.474      0.000      0.0
C(1)      0.695      1.206      0.0
H(1)      1.238      2.143      0.0
C(1)     -0.695      1.206      0.0
H(1)     -1.238      2.143      0.0
C(1)     -1.393      0.000      0.0
H(1)     -2.474      0.000      0.0
C(1)     -0.695     -1.206      0.0
H(1)     -1.238     -2.143      0.0
C(1)      0.695     -1.206      0.0
H(1)      1.238     -2.143      0.0
C(2)      2.333      1.33      3.5
H(2)      3.414      1.33      3.5
C(2)      1.635      2.536      3.5
H(2)      2.178      3.473      3.5
C(2)      0.245      2.536      3.5
H(2)     -0.298      3.473      3.5
C(2)     -0.453      1.33      3.5
H(2)     -1.534      1.33      3.5
C(2)      0.245      0.124      3.5
H(2)     -0.298     -0.813      3.5
C(2)      1.635      0.124      3.5
H(2)      2.178     -0.813      3.5
*
```

- For the calculation of the interaction energy, the energy of the individual benzene monomer should be calculated on the accuracy level of the monomer in the dimer calculation, i.e. using MP2 with full LMP2 guess for the above example.

All possible settings for the multi-level calculation are listed below.

```
# The one-keyword line defines the default method for the multi-level calculation.
# Options here are DLPNO-CCSD(T) or DLPNO-CCSD with the addition of the
# LoosePNO, NormalPNO and TightPNO keyword
!DLPNO-CCSD(T)

# The below given keywords define the changes with respect to the
# above given default method. The user should take care that each intra- or
# interfragment combination is defined only once (unlike in the example given
# below)

%mdci
  LoosePNOFragInter {1 1} {2 2}      # use LoosePNO settings for the intrafragment
                                     # pair energies of fragments 1 and 2
  NormalPNOFragInter {1 2}           # use NormalPNO settings for the
# interfragment                      # pair energies between fragment 1 and 2
  TightPNOFragInter {1 3}            # use TightPNO settings for the interfragment
                                     # pair energies between fragment 1 and 3
  NormalPNOTightPairFragInter {1 2}  # use NormalPNO settings but with TCutPairs
```

(continues on next page)

(continued from previous page)

```

# 1.e-5 for the interfragment pair energies
# between fragment 1 and 2
LoosePNOtightPairFragInter {1 3}
↪e-5
# use LoosePNO settings but with TCutPairs 1.

# for the interfragment pair energies between
# fragment 1 and 3
NoTriplesFragments 1, 2
↪fragment
# if all MOs of a triple are located on

# 1 and / or 2, the triple is neglected
MP2FragInter {1 1} {2 2}
# compute the intrafragment pair energies of
# fragments 1 and 2 on MP2 level
HFFragInter {1 1} {2 2}
↪level
# compute the intrafragment energies on HF
UseFullLmp2Guess false
# default = false,
# if MP2FragInter is used: default = true

end

*xyz 0 1
C(1)      1.393      0.000      0.0
H(1)      2.474      0.000      0.0
...

```

3.10.12 Multi-Level Calculations for IP and EA-EOM-DLPNO-CCSD

The multi-layer method can be used to include the environmental effect in IP-and EA-EOM-DLPNO-CCSD method. A typical input file for the multi-layer IP-EOM-CCSD method will look like

```

! IP-EOM-DLPNO-CCSD TightSCF NORMALPNO def2-SVP def2-SVP/C def2/J RIJCOSX

%mdci
nroots 4
DTOL 1e-7
NORMALPNOFragInter { 1 1}
LOOSEPNOFragInter { 1 1}
HFFRAGMENTINTERACTION { 2 2}
end

*xyz 0 1
C(1)      2.782064      -1.456235      0.000000
C(1)      1.478695      -0.729491      0.000000
C(1)      0.274461      -1.343436      0.000000
N(1)     -0.914790      -0.659079      0.000000
C(1)     -0.988897       0.709718      0.000000
N(1)      0.241239       1.324758      0.000000
H(1)      0.224165       2.335424      0.000000
C(1)      1.507368       0.726274      0.000000
O(1)      2.518005       1.411594      0.000000
O(1)     -2.043648       1.337449      0.000000
H(1)     -1.808257      -1.143873      0.000000
H(1)      0.182931      -2.420317      0.000000
H(1)      2.626386      -2.532891      0.000000
H(1)      3.370859      -1.183830     -0.874506
H(1)      3.370859      -1.183830      0.874506
O(2)     -3.661424      -0.883408      0.000000
H(2)     -3.462053       0.068032      0.000000
H(2)     -4.615649      -0.964529      0.000000
*

```

Here the example is a mono-hydrated thymine molecule, where the thymine is treated at the main fragment and water is treated at the environment. It will result in the following output

```
-----
EOM-CCSD RESULTS (RHS)
-----
```

```

IROOT= 1: 0.322826 au      8.785 eV   70852.1 cm**-1
  Amplitude   Excitation
  -0.689911   37 -> x
Percentage singles character=      96.89

IROOT= 2: 0.364959 au      9.931 eV   80099.2 cm**-1
  Amplitude   Excitation
  -0.689956   35 -> x
Percentage singles character=      95.49

IROOT= 3: 0.378175 au     10.291 eV   82999.9 cm**-1
  Amplitude   Excitation
  -0.691827   36 -> x
Percentage singles character=      93.93

IROOT= 4: 0.403845 au     10.989 eV   88633.7 cm**-1
  Amplitude   Excitation
  -0.690254   34 -> x
Percentage singles character=      96.55

```

The result of a full a IP-EOM-DLPNO-CCSD calculation with NORMALPNO setting would have looked like

```

IROOT= 1: 0.322576 au      8.778 eV   70797.3 cm**-1
  Amplitude   Excitation
   0.689734   37 -> x
Percentage singles character=      96.88

IROOT= 2: 0.364691 au      9.924 eV   80040.3 cm**-1
  Amplitude   Excitation
  -0.689947   35 -> x
Percentage singles character=      95.50

IROOT= 3: 0.377966 au     10.285 eV   82954.0 cm**-1
  Amplitude   Excitation
  -0.691801   36 -> x
Percentage singles character=      93.94

IROOT= 4: 0.402497 au     10.953 eV   88337.9 cm**-1
  Amplitude   Excitation
  -0.690138   34 -> x
Percentage singles character=      96.50

```

The results in multi-layer IP-EOM-DLPNO-CCSD method has been found to be in excellent agreement with standard variant. The *MP2FragInter* treatment is not available for the EOM method. To get a reasonable accuracy one need to treat the fragment from where the ionization is happening (thymine in the above example) at the highest possible level. The interaction between the main fragment (thymine) and environment (water) should be treated at the intermediate level accuracy. The environment can safely be treated with *HFFragInter* for almost all the cases. One can decide the size of the main fragment by looking at HF occupied orbitals as the Koopmans' approximation is a very good zeroth order guess for the IP values. The electron attached states are much less localized as compared to the ionization problem. Consequently, the multi-layer EA-EOM-DLPNO-CCSD requires much more tighter thresholds than the IP variant. An typical input file for multi-layer EA-EOM-DLPNO-CCSD will look as follows.

```

! EA-EOM-DLPNO-CCSD NORMALPNO  ma-def2-SVP RIJCOSX aug-cc-pVDZ/C def2/J

%mdci
  NRoots          4
  FollowCIS       true

```

(continues on next page)

(continued from previous page)

```

TCutPNOSingles      1e-12
MaxIter             2000
DTol                1e-7
NDAV               10
NormalPNOFragInter { 1 1 }
LoosePNOFragInter  { 1 2 } { 2 2 }
end

* xyz 0 1
N(1)  -1.114  -0.934  -3.554
C(1)  -0.343  -0.202  -4.483
H(1)  -0.668  -0.311  -5.520
C(1)   0.635   1.107  -2.633
O(1)   1.241   2.018  -2.013
N(1)   0.022   0.050  -1.776
H(1)  -0.069   0.339  -0.800
C(1)  -0.975  -0.782  -2.233
O(1)  -1.697  -1.422  -1.333
C(1)   1.087   1.852  -4.986
H(1)   1.673   2.594  -4.418
H(1)   1.771   1.340  -5.697
H(1)   0.348   2.403  -5.609
H(1)  -1.823  -1.635  -3.888
N(2)  -4.904  -4.773  -4.958
H(2)  -4.634  -3.987  -5.572
C(2)  -4.875  -4.201  -3.599
C(2)  -3.704  -3.226  -3.310
H(2)  -5.818  -3.646  -3.423
H(2)  -4.859  -5.012  -2.850
O(2)  -3.026  -2.826  -4.301
H(2)  -4.078  -5.386  -5.029
O(2)  -3.559  -2.899  -2.077
H(2)  -2.494  -2.057  -1.754
C(2)   0.440   0.898  -4.018
end

```

It is a thymine-glycine complex where the thymine is treated as the main fragment and glycine as the environment. One needs to use a more tighter value of *TCutPNOSingles* for EA as in the case of standard EA-EOM-DLPNO-CCSD. The *TCutPNOSingles* for the respective fragments automatically gets adjusted based on their respective *TCutPNO* values. The output will be

```

-----
EOM-CCSD RESULTS (RHS)
-----

IROOT=  1: -0.039822 au      -1.084 eV   -8739.9 cm**-1
  Amplitude      Excitation
    0.689012      x ->  53
Percentage singles character=      93.33

IROOT=  2:  0.025156 au      0.685 eV    5521.0 cm**-1
  Amplitude      Excitation
   -0.614385      x ->  54
    0.139410      x ->  55
    0.297903      x ->  59
Percentage singles character=      96.86

IROOT=  3:  0.044569 au      1.213 eV    9781.7 cm**-1
  Amplitude      Excitation
    0.116867      x ->  54
    0.684240      x ->  55

```

(continues on next page)

(continued from previous page)

```

Percentage singles character=      98.16

IROOT=  4:  0.053677 au      1.461 eV   11780.7 cm**-1
  Amplitude   Excitation
    0.695211   x ->   56
Percentage singles character=      98.12

```

The results are in excellent agreement with the standard EA-EOM-DLPNO-CCSD method.

----- EOM-CCSD RESULTS (RHS) -----

```

IROOT=  1: -0.038862 au      -1.057 eV   -8529.3 cm**-1
  Amplitude   Excitation
    -0.689412   x ->   53
Percentage singles character=      93.47

IROOT=  2:  0.025448 au      0.692 eV    5585.2 cm**-1
  Amplitude   Excitation
    -0.654101   x ->   54
    0.131404   x ->   55
    0.207630   x ->   59
Percentage singles character=      97.47

IROOT=  3:  0.044651 au      1.215 eV    9799.8 cm**-1
  Amplitude   Excitation
    0.112183   x ->   54
    0.684562   x ->   55
Percentage singles character=      98.17

IROOT=  4:  0.053780 au      1.463 eV   11803.3 cm**-1
  Amplitude   Excitation
    0.695635   x ->   56
Percentage singles character=      98.16

```

To get the reasonable accuracy in multi-layer EA-EOM-CCSD one need to treat the environment and inter-fragment interaction atleast at *LoosePNOFragInter* level.

3.10.13 Excited States (EOM/STEOM/ADC)

The computation of excited states is described in section [Excited States Calculations](#).

3.10.14 Keywords

The keywords for the “simple” input lines are listed in [Table 3.43](#). The %mdci block options are listed below:

```

%mdci
  cotype CISD      # CI singles+doubles
    QCISD      # quadratic CI (singles+doubles)
    CCSD       # coupled-cluster singles+doubles
    CEPA_1     # coupled-electron pair approximation ''1''
    CEPA_2     #
    CEPA_3     #
    NCEPA_1    # our slightly modified versions of CEPA
    NCEPA_2    # and CPF
    NCEPA_3    #
    NCPF_1     #

```

(continues on next page)

(continued from previous page)

```

NCPF_2    #
NCPF_3    #
ACPF      # averaged coupled-pair functional
ACPF_2    # Gdanitz modification of it
NACPF     # our modification of it
AQCC      # Szalay + Bartlett
SEOI      # a strictly size extensive energy functional
           # maintaining unitary invariance (not yet
           # available for UHF)

ewin -3,1e3 # orbital energy window to determine which
           # MOs are included in the treatment
           # (respects settings in %method block)
Singles true # include single excitations in the
           # treatment (default true)
Triples 0    # (T) correction in CCSD(T)/QCISD(T)
           # default is no triples
           1    # Algorithm 1 (lots of memory, fast)
           2    # Algorithm 2 (less memory, about 2x slower,
           # not yet available for UHF)
Brueckner true # use Brueckner orbitals
           # (default false)
Denmat none    # no evaluation of density matrices
linearized    # density matrix obtained by retaining
           # only CEPA_0-like terms, i.e., those
           # linear in the excitation amplitudes
unrelaxed     # unrelaxed density matrices, i.e.,
           # density matrices without orbital
           # relaxation
orbopt        # perform orbital optimization yielding
           # fully relaxed density matrices (if
           # citype chosen as CCSD or QCISD this option
           # implies evaluation of the Z vector).
           # (default: linearized)
ZSimple true  # simplified evaluation of the Z vector
           # in case of orbital optimized CCD
           # (citype chosen as CCSD or QCISD and
           # Denmat as orbopt) by using an
           # analytical formula
false         # explicit solution of Z vector
           # equations
           # in case of orbital optimized CCD
           # (default: false)
UseQROs      # use of quasi-restricted orbitals
           # (default false)
Localize 0    # use localized MOs. Presently very little
           # use is made of locality. It may help
           # for interpretations. Localization is
           # incompatible with the (T) correction
           PM    # Use Pipek-Mezey localized MOs
           FB    # use Foster-Boys localized MOs
NatOrbIters 0 # Perform natural orbital iterations.
           # default is none. Not possible for CCSD
           # and QCISD
pCCSDAB      # the three parameters for parametrized
pCCSDCD      # coupled-cluster (default is 1.0 which
pCCSDEF      # corresponds to normal CCSD
# this defines how the rate limiting step is handled
# MO and AOX need lots of disk and I/O but if they
# can be done they are fast
KCOpt KC_MO   # Perform full 4-index transformation

```

(continues on next page)

(continued from previous page)

```

KC_AOBLAS# AO direct with BLAS (preferred)
          # (not yet available for UHF, switch to KC_AOX)
KC_AO     # AO direct handling of 3,4 externals
          # (not yet available for UHF, switch to KC_AOX)
KC_RI     # RI approximation of 3,4 externals
          # (not yet available for UHF)
KC_RI2    # Alternative RI (not recommended)
          # (not yet available for UHF)
KC_AOX    # Do it from stored AO exchange integrals
PrintLevel 2 # Control the amount of output. For 3 and
             # higher things like pair correlation
             # energies are printed.
MaxIter 35  # Max. number of iterations
# How the integral transformation is done.
# Note that it is fine to do AOX or AO or AOBLAS
# together with trafo_ri
TrafoType trafo_jk # Partial trafo to J+K operators
             trafo_ri # RI transformation of all
                     # integrals up to 2-externals
                     # (3-ext for (T))and rest on the
                     # fly
             trafo_full # Full four index transformation.
                     # Automatically chosen for
                     # KCOpt=KC_MO
MaxCore 350 # Memory in MB - used for integral
            # trafos and batching and for storage of
            # integrals and amplitudes
            # don't be too generous
STol 1e-5 # Max. element of the residual vector
          # for convergence check
LShift 0.3 # Level shift to be used in update of
           # coefficients
MaxDIIS 7 # Max number of DIIS vectors to be stored
# this lets you control how much and what is residing
# in central memory. May speed up things. Note that
# MaxCore is not respected here
InCore 0 # nothing in core
        1 # + sigma-vector and amplitudes (default)
        2 # + Jij(a,b) Kij(a,b) operators
        3 # + DIIS vectors
        4 # + 3-external integral Kia(b,c)
        5 # + 4-external integrals Kab(c,d)
          # this is identical to ALL
          # the default is AUTO which means that incore
          # is chosen based on MaxCore

#-----
# DLPNO specific settings:
# See local correlation section!
#-----

end

```

Table 3.43: Simple input keywords for the MDCI module.

Keyword	Description
Canonical method selection	
CCSD	Coupled-cluster singles and doubles
CCSD(T)	Same with perturbative triples correction
CCSD-F12	CCSD with F12 correction
CCSD(T)-F12	CCSD(T) with F12 correction
CCSD-F12/RI	CCSD with RI-F12 correction

continues on next p

Table 3.43 – continued from previous page

Keyword	Description
CCSD-F12D/RI	CCSD with RI-F12 correction employing the D approximation (less expensive)
CCSD (T) -F12/RI	CCSD(T) with RI-F12 correction
CCSD (T) -F12D/RI	CCSD(T) with RI-F12 correction employing the D approximation (less expensive)
QCISD	Quadratic Configuration interaction
QCISD (T)	Same with perturbative triples correction
QCISD-F12	QCISD with F12 correction
QCISD (T) -F12	QCISD(T) with F12 correction
QCISD-F12/RI	QCISD with RI-F12 correction
QCISD (T) -F12/RI	QCISD(T) with RI-F12 correction
NCPF/1	A “new” modified coupled-pair functional
CEPA/1	Coupled-electron-pair approximation
NCEPA/1	The CEPA analogue of NCPF/1
RI-CEPA/1-F12	RI-CEPA with F12 correction
MP3	MP3 energies
SCS-MP3	Grimme’s refined version of MP3
Integral handling for the canonical methods	
AOX-<Method>	Compute the 3- and 4-external integrals from pre-stored AO integrals
AO-<Method>	Compute the 3- and 4-external contributions on the fly
MO-<Method>	(Default) Perform a full four index integral transformation
RI-<Method>	Use RI-approximation for all integrals
RI34-<Method>	Use RI-approximation for 3- and 4-external integrals
Local correlation method selection	
DLPNO-CCSD	Domain based local pair natural orbital coupled-cluster method with single and double excitations
DLPNO-CCSD (T)	DLPNO-CCSD with perturbative triple excitations
DLPNO-CCSD (T1)	DLPNO-CCSD with iterative perturbative triple excitations
DLPNO-CCSD-F12	DLPNO-CCSD with F12 correction employing an efficient form of the C approximation
DLPNO-CCSD-F12/D	DLPNO-CCSD-F12 with approach D (less expensive than the C approximation)
DLPNO-CCSD (T) -F12	DLPNO-CCSD(T) with F12 correction employing an efficient form of the C approximation
DLPNO-CCSD (T) -F12/D	DLPNO-CCSD(T)-F12 with approach D (less expensive than the C approximation)
DLPNO-CCSD (T1) -F12	DLPNO-CCSD(T1) with F12 correction employing an efficient form of the C approximation
DLPNO-CCSD (T1) -F12/D	DLPNO-CCSD(T1)-F12 with approach D (less expensive than the C approximation)
Local correlation accuracy control	
LoosePNO	Selects loose DLPNO thresholds
NormalPNO	Selects default DLPNO thresholds
TightPNO	Selects tight DLPNO thresholds
DLPNO-HFC1	Tightened truncation setting for DLPNO-CCSD hyperfine coupling constants calculation
DLPNO-HFC2	Tighter truncation setting than for DLPNO-HFC1

3.11 Correlated Methods using Automatic Code Generation (AUTOCI)

The AUTOCI module hosts a number of methods, where their manual implementation would be tedious or practically impossible. The module works with all types of reference wave function available in ORCA, i.e., RHF, ROHF, UHF and *CASSCF* and offers CI and related methods.

All the methods are implemented in canonical orbital basis and storing all integrals on disk.

3.11.1 Introduction

All the theories are obtained by the means of automated programming within the ORCA-AGE (Automated Generator Environment for ORCA).[361, 362] The CI module reads in the SCF wavefunction and optimizes the coefficient of the CI expansion. Conceptually, the module is similar to `orca_mdci`, therefore the input and output do have a lot in common.