Fig. 5.28: Lowest transitions in  $\text{Ni}(\text{H}_2\text{O})_6^{2+}$ .

### 5.8.4 X-Ray Absorption Spectroscopy

It is also possible to compute X-ray absorption spectra (XAS) with the MCRPA module by making use of the core-valence separation (CVS) approximation.[659] For such calculations, the same active space as for ground state CASSCF or valence excitation energy MCRPA calculations should be chosen. To access MCRPA core excitations, two restrictions are introduced by the CVS approximation:

- The configuration part of the eigenvectors is omitted as it is responsible for valence excitations within the valence active orbitals.
- The orbital part of the eigenvectors is restricted to core-active and core-virtual pairs with a user given list of core orbitals

For computing the lowest MCRPA oxygen K-edge transitions of ozone, one can employ the following input for an CVS-MCRPA calculation:

```
%casscf
nel      12 # number of active electrons
norb     9 # number of active orbitals
nroots   1 # only the ground state
mult     1 # singlet multiplicity
end

%mcrpa
NRroots 15          # number of roots / excited states
```

(continues on next page)

(continued from previous page)

```

DoCVS  true      # turn on core valence separation
OrbWin  0, 2, -1, -1 # select on the lowest 3 core orbitals
DoNTO   true      # generate NTOs for converged roots
end

```

Here the first 3 inactive MOs are selected by OrbWin that are primarily expanded in the 3 1s atomic orbitals of the three O atoms. There is no restriction on the virtual orbitals ( -1 is chosen for first and last virtual MO index).

Note that in contrast to other *multi-reference protocols*, the core orbitals must not get added to the active space for CVS-MCRPA calculations.

The most dominant NTO pairs for the four lowest core excitations are given below:

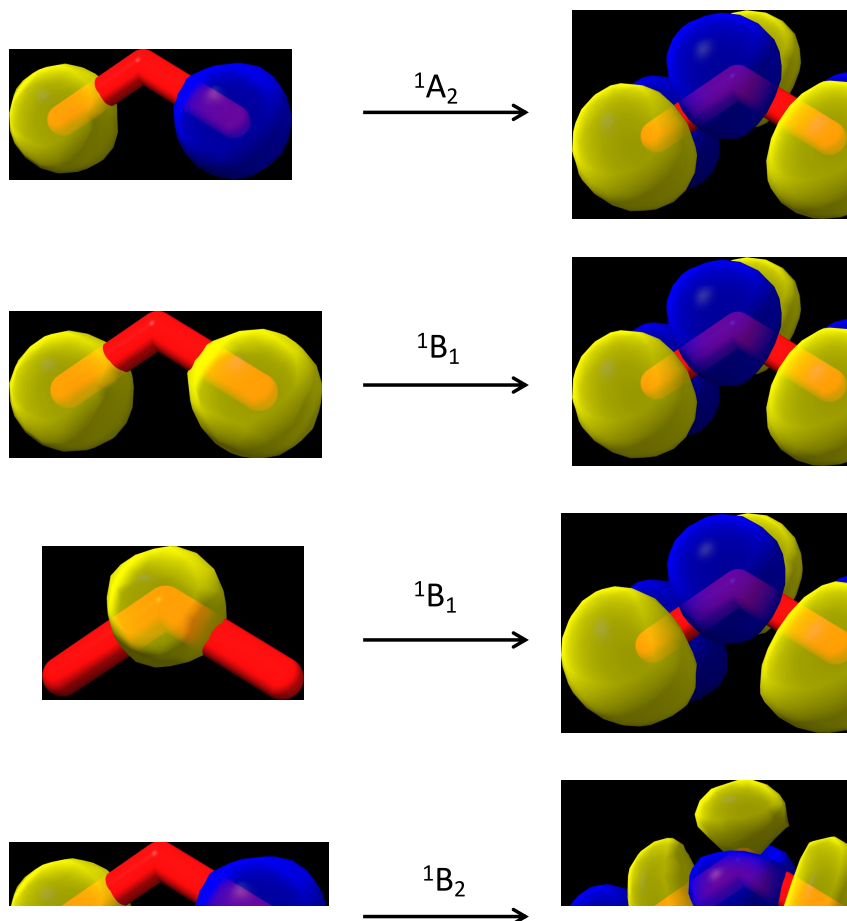


Fig. 5.29: Lowest CVS-MCRPA oxygen K-edge transitions in ozone.

## 5.8.5 Computational Aspects

### Large Molecules

The code is intended to be used for medium-sized and larger open-shell molecules. It has the same scaling as ORCA's first-order CASSCF energy implementation though a larger pre-factor as the computational cost grow "in principle" linearly with the number of roots.

The implementation is AO-driven meaning that the computational bottleneck is the Fock matrix construction for the several state-specific pseudo AO densities. Note that there are up to 6 pseudo AO densities for each state. The computational costs can be reduced significantly if the *RIJCOSX* approximation is employed, which is highly recommended.

The second most expensive part of the MCRPA computation are the two-electron integrals with 3 active indices  $g_{ptuv}$ . As we aim for running calculations on larger systems, there is only an implementation of the integral transformation that uses the resolution-of-the-identity (RI) approximation.

The restrictions on the auxiliary basis sets are the same as for the CASSCF code (Sec. *CASSCF Linear Response*). That is

- If the Fock matrices are constructed in `Direct` or `Conventional` mode, the /C bases are used for the RI approximation of the  $g_{ptuv}$  integrals.
- If the *RIJCOSX* approximation for the Fock matrices is employed, the /JK bases are used for both the Fock matrices and the  $g_{ptuv}$  integrals.

Note that the *RIJK* approximation is not yet available.

### Convergence

Before starting running MCRPA, it is recommended to converge the state-specific CASSCF energy calculation very tightly. The *TRAH-CASSCF* is ideally suited for that purpose. Note that property calculations in general assume vanishing electronic gradients otherwise numerical issues in the eigenvalue / response equations may occur.

Eigenvalues and eigenvectors are determined by a dedicated variant of the Davidson algorithm for solving large-scale TDA or RPA eigenvalue equations.[594, 660] The accuracy of the eigenvalues and eigenvectors is solely determined by `TolR` (default 1.e-5) that is compared to the Frobenius of the residual vector for each root.

Similar to converging CASSCF equations, the MCTDA and MCRPA convergence rates are also affected by the choice of molecules orbitals. This has to be setup in the `%casscf` input block. The MCRPA program will not change the converged orbitals and CI coefficients. For most calculations, the default settings are most appropriate which are *canonical orbitals* for inactive and virtual orbitals and *natural orbitals* for active orbitals. as explained in subsection *Final Orbitals: Canonicalization Choices*. For multi-center systems as bimetallic transition metal complexes, *localized orbitals* are a better choice. Purified d or f orbitals (`ActOrbs dorbs` or `ddorbs` and `forbs`) for single-metal complexes are often helpful for analyzing excitations, though, their choice usually result in slower convergence of the eigenvalue equations.

To facilitate quick convergence of the eigenvalue equations, we start the iterations from eigenvectors that were obtained from direct diagonalization of full Hessian and metric matrices that were compute for a subset of elements. This is referred to as P-space or `full` method which is the default for obtaining initial eigenvectors but also the preconditioner of the TDA/RPA Davidson algorithm (`PreCond full`). The P-space method contains the orbital-orbital and configuration-configuration blocks that are diagonalized individually. Orbital-configuration coupling is omitted here. The size of the P space is controlled by the keyword `PreConMaxRed` which is set to 400 by default. Faster convergence can be expected with larger P-space dimensions, though the cost of the full diagonalization increase cubically. Alternatively, the approximate diagonal Hessian[661] and metric can be used for preconditioning (`PreCond diag`), but not for the initial eigenvectors. It is recommended to employ `PreCond full`.

## 5.8.6 Keyword List

```
%mcrpa

NRoots      0      # The number of desired roots

TolR         1e-5   # Convergence threshold for residual norm
PrintWF      CFG    # print CI part in (CFG (default), CSF, or DET basis)
TolPrint     1e-2   # Threshold for printing elements of excitation vector

MaxRed       200    # maximum size of reduces space for ALL VECTORS IN TOTAL
MaxIter      100    # maximum number of (Davidson) iterations

TDA          false  # Switch off full TD-CASSCF (Tamm-Dancoff approximation)

DoNTO        false  # Generate Natural Transition Orbitals
NTOTresh     1e-3   # Threshold for printing occupation numbers

DoCVS        false  # toggle core valence separation (CVS) for XAS
OrbWin       i0, i1, a0, a1 # first and last inactive and virtual MO,
↪respectively

PreCond      full   # (default) P-space preconditioner uses parts of exact
↪Hessian matrix
              diag   #          diagonal preconditioner approximate diagonal
↪Hessian elements

PreConMaxRed 400    # dimension of exact Hessian for preconditioning

DoOrbResp    true   # turn orbital response off -> CAS-CI method

DoCD         true   # circular dichroism calculation
DoDipoleLength true # electric moments in a length formulation
DoDipoleVelocity true # electric moments in a velocity formulation
DoFullSemiclassical false # request the calculation of complete semiclassical
↪multipolar moments
DecomposeFosc false # request the decomposition of the oscillator
↪strengths in a multipolar expansion.
```

## 5.9 Excited States via EOM-CCSD

Many of the previously described methods are based on the single excitation framework. For a more accurate treatment, double excitations should also be considered. The equation of motion (EOM) CCSD method (and the closely related family of linear response CC methods) provides an accurate way of describing excited, ionized, and electron-attached states based on singles and doubles excitations within the coupled-cluster framework.

### 5.9.1 Theory

The EOM wave function is parametrized in the following manner

$$\mathcal{R}|\Psi_{CC}\rangle,$$

i.e. via the action of a linear excitation operator  $\mathcal{R}$  on the coupled-cluster ground state wave function  $\Psi_{CC}$ . Here,  $\mathcal{R}$  is a particle conserving operator, in the case of the excitation energy problem. However, this is not true for the ionization or electron attachment problems, where  $\mathcal{R}$  is a net annihilation or net creation operator, respectively. The ground state coupled-cluster T-amplitudes are obtained from a CCSD calculation, and our task is to obtain  $\mathcal{R}$ . Note that since the CC Hamiltonian is nonsymmetric, a left hand solution ( $\mathcal{L}$ ) would also be needed to evaluate properties. For the calculation of excitation, ionization or electron attachment energies, however, it is enough to obtain the right

hand solutions ( $\mathcal{R}$ ). In principle, this is done by building the Hamiltonian and diagonalizing it in order to obtain energy expectation values.

In practice, the size of the CCSD Hamiltonian matrix is prohibitively large and thus, various methods have been devised to obtain its lowest few eigenvalues and eigenstates. One of the most popular of these approaches is the Davidson method, which can be summarized as follows:

- Construct an initial guess of orthogonal trial vectors,  $C$ .
- Evaluate the sigma vectors  $\sigma = HC$ .
- Build model Hamiltonian  $\mathcal{H} = C^T \sigma$ .
- Diagonalize  $\mathcal{H}$ :  $\mathcal{E} = U^T \mathcal{H} U$ .
- Compute Ritz vectors  $X = CU$ .
- Compute residuals  $R = X\mathcal{E} - \sigma U$ , check convergence: if yes, pass  $X, \mathcal{E}$  as solutions.
- Preconditioning:  $T = MR$  (many possible choices for the preconditioner  $M$ ).
- Check if adding new trial vectors would exceed the maximum number of trial vectors:
  - if no, add  $T$  to  $C$ , and orthonormalize the united set
  - if yes, then set  $X$  as  $C$  (orthonormalize if  $H$  is nonsymmetric); then add  $T$  and orthonormalize

The advantage of the above method is that, instead of the full Hamiltonian, only the sigma vectors have to be explicitly evaluated and stored.

It is also possible to use a lower scaling version of the EOM-CCSD methods, which relies on the perturbative truncation of the coupled-cluster similarity transformed Hamiltonian. Presently, only the second order truncated version (CCSD(2) approximation) is available for closed-shell molecules (RHF). However, it is better to use the PNO based implementation, as it has the cost of EOM-CCSD(2), but its accuracy is comparable to canonical EOM-CCSD.

## Initial Guess

The present initial guess in the RHF EOM implementation consists of constructing a CIS Hamiltonian of a certain dimension, and diagonalizing it. The roots are preselected based on the energetic ordering of the diagonal elements of the Hamiltonian. In the UHF case, the guess is constructed from the solutions of a UHF CIS calculation. The number of roots in the initial guess is determined as 20 times the number of roots desired in EOM (NRoots) if NDAV is 20 or smaller, otherwise it is set to NDAV times the number of EOM roots. If the parameter NInits is larger than zero, then the number of initial guess roots will be set to this parameter times NRoots. The maximum possible number of roots is the full CIS dimension, ( $N_O N_V$  (RHF) or  $N_{O_\alpha} N_{V_\alpha} + N_{O_\beta} N_{V_\beta}$  (UHF)). One should keep in mind, while increasing the number of initial guess vectors, that this corresponds to diagonalizing a matrix of increasing dimension. If, for example NRoots is 10, then by default 200 roots are considered in the initial guess (unless it exceeds the size of the CIS space), or if NInits is set to 100, then there will be 1000 roots in the guess. In some cases, the roots calculated using EOM may not be the lowest ones, but a few of these may be replaced by some higher roots which are “easier” to find. In such cases, it may help to increase NRoots or NInits to converge to the proper roots. The program can be made to follow the initial CIS guess by setting FollowCIS to true and is necessary if we wish to ionize or excite from inner-valence or core orbitals. In the RHF implementation, the core orbital, from which the ionization or excitation originates, can be specified using the keyword CoreHole, in addition to setting DoCore and FollowCIS to true. The CoreHole keyword is quite general and in principle, ionization or excitation processes from any occupied orbital can be specified using this keyword.

## Hamiltonian Construction

The Hamiltonian construction begins by calling the sigma routines. In the case of the closed-shell code, the logical variables `UseEOMOptS` and `UseEOMOptD` choose the routines to be used in the evaluation of the singles and doubles sigma vectors, respectively. If true, the optimized sigma routine, using dressed integrals, will be used. This should not be changed, the option is there mainly for debugging purposes. If set to false, an automatically generated, and much slower serial code will be used instead. In the case of the open-shell UHF implementation, optimized sigma routines have been generated using the ORCA Automated Generator Environment (AGE) [361]. In each early iteration,  $N_R$  sigma vectors will be determined, except in the case of a restart, where the number of sigma vectors is  $2N_R$ . For further details on convergence, see *Convergence, Restart, Preconditioning and Subspace Expansion* below.

The most time consuming part of the sigma vector construction is the formation of the external exchange contribution, which can be influenced via the CC keyword `KCOpt`. Currently, there are three options that are compatible with the RHF EOM implementation: `KC_MO`, `KC_AOX` and `KC_AOBLAS` and `KC_AOX` is the only option available in the UHF EOM code. The external exchange term can be treated most efficiently using `COSX`, which in the closed-shell case, leads to average speed ups of 10x for the external exchange term and an overall speedup of 3x for the EOM calculation. This is accompanied by a drastic reduction of the storage cost[662]. The error introduced is below 1 meV, which is 200-fold less than the error bar of the method[662] itself. It is the default for `KCOpt KC_AOX` and `KC_AOBLAS` and can be controlled by the keyword `DOCOSXEOM`. The default grid settings for EOM are `GridX 1` and `IntAccX 2.68`.

Once the sigma vectors are available, they are multiplied with the trial vectors to yield the reduced space Hamiltonian. The Hamiltonian is built in a way that, in each iteration, only the new vector products are added to the “edge” of the old Hamiltonian, so that a full build is avoided. It should be clear that the parameter `NDav` plays an important role here, since it determines the maximum size of the Hamiltonian ( $N_D N_R$ ), and also controls how much memory is needed for the trial and sigma vectors, as seen above. Since the choice of this parameter influences convergence properties, it will be discussed further in *Convergence, Restart, Preconditioning and Subspace Expansion*.

## Solution of the (Nonsymmetric) Eigenproblem

Following the construction of the Hamiltonian, a nonsymmetric eigensolver is called. In this case, it is possible to have complex eigenvalues. In practice, this is rarely the case, and indicates a problem of some kind. A warning will be given if this happens, however, one may get away with this if it only happens in an isolated iteration step.

Once the eigenvectors are available, they are compared with those of the previous iteration, if root homing is turned on, i.e. if the `RootHoming` keyword is set to true. This means evaluating the overlap of the old and new eigenvectors, in order to keep track of the possible movement of the eigenvectors if root flipping occurs. If converged roots are removed from further iterations (see next section), it is important to keep track of changes in ordering, especially if a converged and a non-converged root is swapped. After diagonalization, the Ritz vectors and residuals can be evaluated.

## Convergence, Restart, Preconditioning and Subspace Expansion

Convergence is signaled once a residual square norm based criteria is fulfilled. This criteria is determined by the `CheckEachRoot` keyword. If it is true (default), the convergence of the residual square norm of each root is checked separately. This is due to the fact that different roots converge at a different rate. Once a root is converged, no new trial vectors will be generated, belonging to that vector. This means that the EOM iterations will progressively become faster (until restart). Turning off the rootwise convergence check is possible, but not recommended. In this case, the maximum of all residual square norms is checked for convergence, and all iterations will take roughly the same amount of time since no vectors are removed in any iteration. However, this procedure can be numerically unstable, since the residuals of some roots might become very close to zero, and trying to generate new vectors, which are orthogonal to these, may lead to numerical disaster. In short, the recommended default is having both `CheckEachRoot` and `RootHoming` set to true. If `CheckEachRoot` is false, then `RootHoming` should also be set to false, as it may cause problems if `NDav` is too small. The convergence threshold of the residual in Davidson’s method can be larger than that for the ground state CC residual threshold in order to obtain converged results. Namely, a value of `DTol` of  $1e-5$  is almost always enough to get well converged energies.

At this point it is worth discussing the role of the keyword `NDav`. This keyword determines at what point the Davidson algorithm should be restarted. If it is chosen too small, it may cause slow convergence. If this value is too large, this

may result in overwhelming demands on memory/disk space requirements. The default value (20) is chosen with the hope that no, or maybe one restart will be required. It should only be changed if computational resources demand it. However, the treatment of core ionization or core excitation processes often requires a large value of `NDav`. At restart, the Ritz vectors are copied as new trial vectors for all roots, which will then be orthonormalized, while new vectors will only be generated for the non-converged roots. This means that the step after the rebuilding of the expansion space will be 1-2 times as expensive as one of the initial steps.

New directions (trial vectors) are generated from the preconditioned residual vectors. If no preconditioning is applied (the preconditioner is taken to be a unit matrix), one falls back to the Lanczos algorithm, which is inferior to the Davidson algorithm. This happens if the keyword `DoLanczos` is true. This is not recommended, as the Lanczos algorithm converges several times slower than Davidson's, and is there for debugging mainly. The original Davidson preconditioner is the inverse of a diagonal matrix which contains the difference of the diagonal elements of the Hamiltonian and the current approximation to the eigenvalue belonging to the given root. Let us consider the closed-shell RHF implementation for simplicity. If  $R_{ia}$  and  $R_{ijab}$  are elements of the singles and doubles amplitudes, respectively, then the updated vectors ( $T_{ia}$ ,  $T_{ijab}$ ) have the form

$$T_{ia} = \frac{R_{ia}}{D_{ia} + \mathcal{E}_R}$$

for singles, and

$$T_{ijab} = \frac{R_{ijab}}{D_{ijab} + \mathcal{E}_R}$$

for doubles. Here,  $D_{ia}$  and  $D_{ijab}$  are related to, and possibly approximations of, the respective diagonal Hamiltonian elements. The simplest approximation is just to construct these from diagonal Fock matrix elements (i.e. orbital energies) as  $D_{ia} = \varepsilon_a - \varepsilon_i$  and  $D_{ijab} = \varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j$ . A slightly better preconditioning can be obtained as follows. For singles, take the exact CIS diagonal elements,  $D_{ia} = \varepsilon_a - \varepsilon_i + \bar{g}_{iaaa}$ , where the last term is the respective antisymmetrized integral; and construct the doubles as  $D_{ijab} = D_{ia} + D_{jb}$ . This is the default, and can be changed back to the simple Fock matrix guess by setting `UseCISUpdate` to false.

Following the preconditioning step, the resulting vectors are orthogonalized to the previous set of trial vectors, and orthonormalized among themselves. Since, the trial vectors do not change once they are generated (unless a restart occurs), only the new elements of the overlap matrix need to be generated for the orthonormalization. The numerical threshold for the inversion (and other division steps) is controlled by the parameter `OTol`. Finally, the amount of printed information can be controlled via the `PrintLevel` keyword. If not given or equal to 2, only basic iteration information will be printed. If set to 3, detailed iteration information will be printed (recommended if timing results for individual steps are required), while 4 or higher triggers additional (and very verbose) information from other subroutines as well.

The default solver is a multi-root Davidson procedure. The single-root solver can be initiated by setting `DoRootwise` and `FollowCIS` to true. The latter is more stable when a large number of roots are requested.

## Properties in the RHF EOM implementation

The only property that can be calculated with the current RHF EOM implementation is the transition moment. It is calculated as a CI-like expectation value, as proposed by Stanton and Bartlett. The right and left transition density are defined as

$$\rho_{pq}^{Gr \rightarrow Ex} = \langle \phi_0 | (1 + \Lambda) [e^{-T} \{p^+ q^-\} e^T, R] | \phi_0 \rangle$$

$$\rho_{pq}^{Ex \rightarrow Gr} = \langle \phi_0 | L e^{-T} \{p^+ q^-\} e^T | \phi_0 \rangle$$

In the above equation,  $\Lambda$  corresponds to the ground state left vector, which needs to be solved once and  $L$  is the left vector, which needs to be solved separately for each root. Once the right and left vectors have been obtained, the left and right transition densities are constructed and the oscillator strength is calculated using the following formula

$$f = \frac{2}{3} \varepsilon |\mu_{pq} \rho_{pq}^{Ex \rightarrow Gr}| |\mu_{pq} \rho_{pq}^{Gr \rightarrow Ex}|$$

The oscillator strength, calculated by default, employs a linear approximation for  $\Lambda$ . The  $L$  vectors are, on the other hand, calculated as a general inverse of the corresponding  $R$  vectors. This approximation requires no additional effort



over the energy calculation and gives similar accuracy as that of the exact oscillator strength calculation, which is at least twice the cost of the energy calculation. Exact EOM-CC transition moments can, however, be calculated by setting `DoLeft` and `DoTDM` to true. Please note that transition moments have not yet been implemented for the UHF EOM-CCSD approach.

## 5.9.2 General Use

The EOM-CCSD routine is part of the `orca_mdci` module of the ORCA program package. It is called after a successful coupled-cluster calculation, if the appropriate flags and the number of roots have been set. In the following chapter, the general program flow and the input parameters of the EOM routine will be described. For an RHF or UHF reference, the EE-, IP- and EA-EOM-CCSD approaches are available for the computation of excitation energies, ionization potentials and electron affinities, respectively. Currently, the following simple input keywords are available:

```
!EOM-CCSD      # same as !EE-EOM-CCSD
!EE-EOM-CCSD   # EOM for electronically excited states
!IP-EOM-CCSD   # IP version
!EA-EOM-CCSD   # EA version
```

The simplest way to perform an EOM calculation is via the usage of the `EOM-CCSD` keyword, together with the specification of the desired number of roots:

```
! RHF EOM-CCSD cc-pVDZ TightSCF

%mdci
  nroots 9
end

*xyz 0 1
  C      0.016227   -0.000000    0.000000
  O      1.236847    0.000000   -0.000000
  H     -0.576537    0.951580   -0.000000
  H     -0.576537   -0.951580   -0.000000
*
```

The above input will call the EOM routine with default settings. The main output is a list of excitation energies, augmented with some further state-specific data. For the above input, the following output is obtained:

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

IROOT=  1:  0.147823 au      4.022 eV   32443.5 cm**-1
  Amplitude   Excitation
    0.107945    4 ->    8
    0.665496    7 ->    8
    0.104633    7 ->    8    6 ->    8
  Ground state amplitude:  0.000000
Percentage singles character=      92.32

IROOT=  2:  0.314133 au      8.548 eV   68944.3 cm**-1
  Amplitude   Excitation
    0.671246    7 ->    9
  Ground state amplitude: -0.000000
Percentage singles character=      90.42

IROOT=  3:  0.343833 au      9.356 eV   75462.6 cm**-1
  Amplitude   Excitation
   -0.670633    5 ->    8
   -0.112538    6 ->    8    5 ->    8
```

(continues on next page)



(continued from previous page)

```

Ground state amplitude: 0.000000
Percentage singles character= 92.00

IROOT= 4: 0.364199 au 9.910 eV 79932.5 cm**-1
Amplitude Excitation
0.102777 4 -> 10
-0.484661 6 -> 8
0.438311 7 -> 10
-0.167512 6 -> 8 6 -> 8
Ground state amplitude: -0.021060
Percentage singles character= 87.22

IROOT= 5: 0.389398 au 10.596 eV 85463.0 cm**-1
Amplitude Excitation
0.646812 4 -> 8
-0.122387 7 -> 8
0.171366 7 -> 8 6 -> 8
Ground state amplitude: 0.000000
Percentage singles character= 87.47

IROOT= 6: 0.414587 au 11.281 eV 90991.4 cm**-1
Amplitude Excitation
-0.378418 6 -> 8
-0.537292 7 -> 10
-0.124246 6 -> 8 6 -> 8
Ground state amplitude: -0.061047
Percentage singles character= 89.13

IROOT= 7: 0.423861 au 11.534 eV 93026.7 cm**-1
Amplitude Excitation
0.673806 7 -> 11
Ground state amplitude: 0.000000
Percentage singles character= 93.14

IROOT= 8: 0.444201 au 12.087 eV 97490.8 cm**-1
Amplitude Excitation
0.664877 6 -> 9
0.130475 6 -> 9 6 -> 8
Ground state amplitude: -0.000000
Percentage singles character= 87.17

IROOT= 9: 0.510514 au 13.892 eV 112044.8 cm**-1
Amplitude Excitation
-0.665791 6 -> 10
0.114259 6 -> 15
-0.124374 6 -> 10 6 -> 8
Ground state amplitude: -0.000000

```

The IP and EA versions can be called using the keywords IP-EOM-CCSD and EA-EOM-CCSD, respectively. For open-shell systems (UHF reference wavefunction), IP/EA-EOM-CCSD calculations require an additional keyword. Namely, an IP/EA calculation involving the removal/attachment of an  $\alpha$  electron is requested by setting the DoAlpha keyword to true in the %mdci block, while setting the DoBeta keyword to true selects an IP/EA calculation for the removal/attachment of a  $\beta$  electron. Note that DoAlpha and DoBeta cannot simultaneously be true and that the calculation defaults to one in which DoAlpha is true if no keyword is specified on input. A simple example of the input for a UHF IP-EOM-CCSD calculation for the removal of an  $\alpha$  electron is given below.

```

! IP-EOM-CCSD cc-pVDZ
%mdci
  DoAlpha true
  NRoots 7
end

```

(continues on next page)

(continued from previous page)

```
*xyz 0 3
  O      0.0 0.0 0.0
  O      0.0 0.0 1.207
*
```

## Capabilities

At present, the EOM routine is able to perform excited, ionized and electron attached state calculations, for both closed- or open-shell systems, using RHF or UHF reference wave functions, respectively. It can be used for serial and parallel calculations. The method is available in the back-transformed PNO and DLPNO framework enabling the calculation of large molecules - see Section *Excited States via PNO-based coupled cluster* and Section *Excited States with DLPNO based coupled cluster methods*. In the closed-shell case (RHF), a lower-scaling version can be invoked by setting the CCSD2 keyword to true in the %mdci section. The latter is a second-order approximation to the conventional EOM-CCSD. For the time being, the most useful information provided is the list of the excitation energies, the ionization potentials or the electron affinities. The ground to excited state transition moments are also available for the closed-shell implementation of EE-EOM-CCSD.

## 5.9.3 Memory Management

The most important data coming from the coupled-cluster routine are the ground state energy and wave function, and the molecular integrals. The integrals are then used to create “dressed” integral containers, which allows for an efficient factorization of the EOM equations, since these dressed quantities do not change during the calculation. Most of these are written on disk, with the possible exception of the integral container which has three external labels. This, and the solver files may remain in core if enough memory is available. The program sequentially tries to allocate memory for the files in the order of their importance, and what cannot be kept in core, goes on disk. The order of allocation is as follows: 1. residual vectors, 2. Ritz vectors, 3. three external integrals, 4. sigma vectors and 5. state (trial) vectors, as seen in the example below:

```
-----
AUTOMATIC CHOICE OF INCORE LEVEL
-----

Memory available                ...    6512.00 MB
Memory needed for Residual-vectors ...    71.27 MB
Memory needed for Ritz-vectors   ...    71.27 MB
Memory needed for 3-ext integrals ...    92.05 MB
Memory needed for Sigma-vectors  ...   1425.31 MB
Memory needed for State-vectors  ...   1425.31 MB
-> Final InCoreLevel           ...    5
```

Half of the memory specified with the keyword MaxCore is distributed among the five candidates. In the above case, everything fits in memory. Note that these are only the largest contributors to memory consumption, and there should ideally be a safety margin when allocating memory.

In order to estimate the amount of necessary memory, it should be kept in mind that, in the closed shell case, the memory requirements of the residual and Ritz vectors are proportional to  $N_R N_P N_V^2$ , the three external integrals to  $N_R N_O N_V^3$  and the sigma and trial vectors to  $N_D N_R N_P N_V^2$ , where  $N_O$  and  $N_V$  are the number of occupied and virtual orbitals,  $N_P = \frac{N_O(N_O+1)}{2}$  is the number of occupied pairs,  $N_R$  is the number of roots, and  $N_D$  is the maximum size of the reduced space. The keyword NRoots sets  $N_R$ , while NDAV determines  $N_D$ . Luckily, the contributions that, in our experience, are the most important to keep in memory, are also the ones that require the smallest amount of it. It is advisable to use KCOpt AOBLAS, as it has the lowest memory requirements.

Note that in the UHF EE-EOM-CCSD implementation, the memory requirements of the residual and Ritz vectors are proportional to  $N_R(N_{P_\alpha} N_{V_\alpha}^2 + N_{P_\beta} N_{V_\beta}^2 + N_{O_\alpha} N_{O_\beta} N_{V_\alpha} N_{V_\beta})$ , the three external integrals to  $N_R(N_{O_\alpha} N_{V_\alpha}^2 + N_{O_\beta} N_{V_\beta}^2 + N_{O_\alpha} N_{V_\alpha} N_{V_\beta}^2 + N_{O_\beta} N_{V_\beta} N_{V_\alpha}^2)$  and the sigma and trial vectors memory requirements are proportional to  $N_D N_R(N_{P_\alpha} N_{V_\alpha}^2 + N_{P_\beta} N_{V_\beta}^2 + N_{O_\alpha} N_{O_\beta} N_{V_\alpha} N_{V_\beta})$ , where  $N_{O_\alpha}$ ,  $N_{O_\beta}$ ,  $N_{V_\alpha}$  and  $N_{V_\beta}$  are respectively, the

number of occupied alpha, occupied beta, virtual alpha and virtual beta orbitals and  $N_{P_\alpha} = \frac{N_{O_\alpha}(N_{O_\alpha}-1)}{2}$  and  $N_{P_\beta} = \frac{N_{O_\beta}(N_{O_\beta}-1)}{2}$  are the number of alpha and beta occupied pairs, respectively.

## 5.9.4 Some tips and tricks for EOM-CC calculation

- The COSX approximation gives significant savings in terms of memory use, disk space use and computational timings with almost no loss of accuracy[662]. Therefore, the preferred setting for large-scale calculations should include `DoCOSXEOM true`, `DoAOX3e true` and `KCOpt KC_AOBLAS` (Note that `KC_AOX` is the only option available for `KCOpt` in the UHF implementation).
- The EOM-CC code in ORCA has three versions of the Davidson solver. The default one is the multi-root solver which does optimization of all the roots together. It gives the fastest convergence and is more suitable when one is interested only in a few roots of a big molecule. However, the multi-root solver can run into numerical issues, if more than 10 roots are desired. In that case, one can invoke the root-wise solver by setting `DoRootwise true`. The single root solver is very stable and should be used when a large number of roots are desired. However, the convergence of the single root solver is slower than the multi-root one. In the RHF implementation, there is also a batchwise solver, where a subset of the total number of roots is optimized together. This can be invoked by setting `NRootsPerBatch true` and is intermediate between the multi-root and single-root solvers in terms of stability and convergence.
- If the EOM iterations do not converge within 50 cycles, one can try to increase the number of iterations by setting `MaxIter` in the `%mdci` block to a larger value. One can also try to increase the dimension of the Krylov space in the Davidson solver by increasing the `NDav` value and this generally helps to accelerate convergence. However, setting `NDav` to a value larger than 200 can make the calculation prohibitively costly especially in terms of memory.
- Convergence thresholds of `DTol 1e-5` (Davidson convergence) and `STol 1e-7` (ground state CCSD convergence) generally yield sufficiently converged energies, and are suitable for most purposes.
- The normal Davidson solver generally leads to the lowest-energy solutions. This procedure can also yield roots dominated by double excitations (the so-called satellite states) for the IP and EA variants of EOM-CC, when one asks for a large number of roots. If one is interested in the low-lying Koopman's type of IP and EA states, they can be obtained by setting `FollowCIS` to true. This will follow the initial guess provided by the Fock operators.

## 5.9.5 Full keyword list

The simple input keywords related to EOM-CC are collected in Table 5.10. The input keywords of the `%mdci` block related to UHF EOM-CC are collected in Table 5.12. The input keywords of the `%mdci` block related to RHF EOM-CC are collected in Table 5.11.

Table 5.10: Simple input keywords for EOM-CC.

Keyword	Description
EOM-CCSD	Same as EE-EOM-CCSD
EE-EOM-CCSD	EOM-CC for electronically excited states
IP-EOM-CCSD	IP version
EA-EOM-CCSD	EA version

Table 5.11: `%mdci` block input keywords for RHF EOM-CC with defaults in parentheses.

Keyword	Options	Description
DoEOM (false)	true/false	Perform EOM?
UseEOMOptS (true)	true/false	Use optimized sigma routines for singles?
UseEOMOptD (true)	true/false	Use optimized sigma routines for doubles?
NDav (20)	<int>	Max. size of reduced space (i.e. 20*NRoots)

continues on next page

Table 5.11 – continued from previous page

Keyword	Options	Description
CCSD2 (false)	true/false	Use the lower scaling CCSD(2) approximation?
CheckEachRoot (true)	true/false	Check convergence for each root separately?
RootHoming (true)	true/false	Apply root homing?
DoLanczos (false)	true/false	Use the Lanczos procedure rather than Davidson?
UseCISUpdate (true)	true/false	Use diagonal CIS for updating?
NInits (0)	<int>	Number of roots in the initial guess, if 0, use preset value
DRESS3ES (true)	true/false	Construct the external dressing to singles or calculate on the fly?
DRESS3ED (false)	true/false	Construct the external dressing to doubles or calculate on the fly?
DOCOSXEOM (false)	true/false	Use COSX approximation for external exchange term in EOM?
DOAOX3E (false)	true/false	Use COSX approximation for 4 external terms contribution to 3 external intermediates
DoRootwise (false)	true/false	Solve for each root separately? More stable for large number of roots
DoTDM (false)	true/false	Option for calculation of default transition moment?
Doleft (false)	true/false	Calculation of exact left vector?
NRootsPerBatch (1)	<int>	Number of roots calculated together
FOLLOWCIS (false)	true/false	Follows the initial singles guess?
DoCore (true)	true/false	Initiates ionization or excitation from core orbital?
CoreHole (0)	<int>	Core orbital from which ionization or excitation is needed
CVSEP (false)	true/false	Separate core orbitals from valence?
DTol (1e-5)	<real>	Default for EOM residual threshold
NRoots (9)	<int>	Number of roots
OTol (1e-14)	<real>	Orthogonalization threshold
KCOpt		Method for external exchange formation
	KC_MO	
	KC_AOX	When asked for exact TDM calculation
	KC_AOBLAS	Most efficient
PrintLevel (3)	<int>	The amount of information to be printed
MaxCore (500)	<int>	Total amount of memory

Table 5.12: %mdci block input keywords for UHF EOM-CC with defaults in parentheses.

Keyword	Options	Description
DoEOM (false)	true/ false	Perform EOM?
DoAlpha (false)	true/ false	Removal/attachment of an alpha electron (IP/EA calculations)?
DoBeta (false)	true/ false	Removal/attachment of a beta electron (IP/EA calculations)?
NDav (20)	<int>	Max. size of reduced space (i.e. 20*NRoots)
UseQROs (false)	true/ false	Use QROs?
CheckEachRoot (true)	true/ false	Check convergence for each root separately?
RootHoming (true)	true/ false	Apply root homing?
DoLanczos (false)	true/ false	Use the Lanczos procedure rather than Davidson?
DoOlsen (false)	true/ false	Use the Olsen procedure rather than Davidson?
UseCISUpdate (true)	true/ false	Use diagonal CIS for updating?
NInits (0)	<int>	Number of roots in the initial guess, if 0, use preset value
DOCOSXEOM (false)	true/ false	Use COSX approximation for external exchange term in EOM?
DOAOX3E (false)	true/ false	Use COSX approximation for 4 external terms contribution to 3 external intermediate?
DoRootwise (false)	true/ false	Solve for each root separately? More stable for large number of roots
NRootsPerBatch (1)	<int>	Number of roots calculated together
FOLLOWCIS (true)	true/ false	Follow the initial singles guess?
DTol (1e-5)	<real>	Default for EOM residual threshold
NRoots (9)	<int>	Number of roots
OTol (1e-14)	<real>	Orthogonalization threshold
KCOpt	KC_AOX	AO exchange for the four external contributions
PrintLevel (3)	<int>	The amount of information to be printed
MaxCore (500)	<int>	Total amount of memory

## 5.10 Excited States via STEOM-CCSD

The EOM-CCSD approach for excitation energies becomes prohibitively costly for large systems because of its  $O(N^6)$  scaling. Therefore, one needs a more compact form of the wave-function ansatz. A second similarity transformation can compress the final matrix diagonalization step to the CIS space only. The resulting STEOM-CCSD method of Marcel Nooijen and co-workers [663] is an efficient way for accurate calculations of excitation energies.

### 5.10.1 General Description

In the standard EOM-CC method, the transformed Hamiltonian is diagonalized over a singles and doubles space to obtain ionized, attached, or excited states of the reference state. In STEOM-CC, one performs a second similarity transformation

$$\hat{G} = \{e^{\hat{S}}\}^{-1} \hat{H} \{e^{\hat{S}}\}$$

The transformation operator  $\hat{S}$ , including singles and doubles, is defined as

$$\begin{aligned}\hat{S} &= \hat{S}^{IP} + \hat{S}^{EA}, \\ \hat{S}^{IP} &= S_{i'}^m \hat{E}_{i'}^m + \frac{1}{2} S_{ij}^{mb} \hat{E}_{ij}^{mb}, \\ \hat{S}^{EA} &= S_e^{a'} \hat{E}_e^{a'} + \frac{1}{2} S_{ej}^{ab} \hat{E}_{ej}^{ab}.\end{aligned}$$

In the above equations,  $m$  and  $e$  denote active indices of the hole and particle type respectively, while a prime denotes a restriction to orbitals that are not active. The amplitudes of the operator  $\hat{S}$  are defined in such a way that matrix elements of the transformed Hamiltonian, in second quantized notation, become equal to zero.

$$g_{i'}^m = g_{ij}^{mb} = g_e^{a'} = g_{ej}^{ab} = 0$$

In addition, the zeros which pre-existed in  $\bar{H}$ , after solving the CCSD equations, remain preserved. The above equations are linear in  $\hat{S}$  and are equivalent to the Fock space multireference coupled cluster equations for the one valence problem. However, to ensure numerical stability, the equations are re-casted as matrix diagonalization problem and solved as IP-EOM-CCSD and EA-EOM-CCSD problems. The  $\hat{S}^{IP}$  and  $\hat{S}^{EA}$  are then extracted from the converged previous calculations, respectively, by invoking intermediate normalization on the suitably chosen eigenvectors corresponding to active holes and active particles. The total process can be described as following

- Solution of the ground state coupled cluster equations
- Construct the first similarity transformed Hamiltonian as  $\hat{\bar{H}} = e^{-\hat{T}} \hat{H} e^{\hat{T}}$
- Solution of the IP-EOM and EA-EOM equations
- Extraction of the  $\hat{S}$  amplitudes
- Construct the second similarity transformed Hamiltonian as  $\hat{\bar{G}} = e^{-\hat{S}} \hat{\bar{H}} e^{\hat{S}}$
- Diagonalization of  $\hat{\bar{G}}$  in CIS space

The advantage of the above method is that, instead of one iterative  $O(N^6)$  scaling diagonalization step, it requires two iterative  $O(N^5)$  scaling steps, one non-iterative  $O(N^5)$  scaling step and one iterative  $O(N^4)$  scaling matrix diagonalization step. The presence of so-called ‘implicit triples excitation’ term ensures the charge transfer separability of the excited states, which is absent in EOM-CCSD. In addition, since the final diagonalization step is performed in a CIS space, the spin adaption is trivial and excited states of triplet multiplicity can be obtained without going through the complications of a spin orbital based implementation.

The STEOMCC approach has also recently been extended for applications to open-shell systems within the UHF formalism [664] [665]. In this case, the expressions for the operators  $\hat{S}^{IP}$  and  $\hat{S}^{EA}$  take the form,

$$\begin{aligned}\hat{S}^{IP} &= \frac{1}{2} \sum_{i,e,a,b} s_{ie}^{ab} \{\hat{a}^\dagger \hat{b}^\dagger \hat{e} \hat{i}\} + \sum_{\bar{i},e,\bar{a},b} s_{ie}^{\bar{a}b} \{\hat{\bar{a}}^\dagger \hat{b}^\dagger \hat{e} \hat{i}\} \\ &+ \frac{1}{2} \sum_{\bar{i},\bar{e},\bar{a},\bar{b}} s_{ie}^{\bar{a}\bar{b}} \{\hat{\bar{a}}^\dagger \hat{\bar{b}}^\dagger \hat{e} \hat{i}\} + \sum_{i,e,a,\bar{b}} s_{ie}^{a\bar{b}} \{\hat{a}^\dagger \hat{b}^\dagger \hat{e} \hat{i}\}, \\ \hat{S}_- &= \frac{1}{2} \sum_{i,j,a,m} s_{ij}^{am} \{\hat{a}^\dagger \hat{m}^\dagger \hat{j} \hat{i}\} + \sum_{\bar{i},j,\bar{a},m} s_{ij}^{\bar{a}m} \{\hat{\bar{a}}^\dagger \hat{m}^\dagger \hat{j} \hat{i}\} \\ &+ \frac{1}{2} \sum_{\bar{i},\bar{j},\bar{a},\bar{m}} s_{ij}^{\bar{a}\bar{m}} \{\hat{\bar{a}}^\dagger \hat{\bar{m}}^\dagger \hat{j} \hat{i}\} + \sum_{i,\bar{j},a,\bar{m}} s_{ij}^{a\bar{m}} \{\hat{a}^\dagger \hat{\bar{m}}^\dagger \hat{j} \hat{i}\}.\end{aligned}$$

where we use overbars to distinguish the  $\beta$  orbitals from the  $\alpha$  orbitals. The amplitudes  $\{s_{ie}^{ab}, \bar{s}_{ie}^{ab}\}$  are determined by solving the UHF EA-EOM-CCSD equations for the attachment of an  $\alpha$  electron, while the  $\{s_{ie}^{\bar{a}\bar{b}}, \bar{s}_{ie}^{\bar{a}\bar{b}}\}$  amplitudes are extracted from a UHF EA-EOM-CCSD calculation for the attachment of a  $\beta$  electron. Similarly, the sets of amplitudes  $\{s_{ij}^{am}, \bar{s}_{ij}^{am}\}$  and  $\{s_{ij}^{\bar{a}\bar{b}}, \bar{s}_{ij}^{\bar{a}\bar{b}}\}$  are determined by solving the decoupled UHF IP-EOM-CCSD problems for the ionization of an  $\alpha$  electron and the ionization of a  $\beta$  electron, respectively. Hence, an UHF STEOMCC calculation involves two separate IP calculations ( $O(N^5)$  scaling) and two separate EA calculations ( $O(N^5)$  scaling steps).

All the speed up options, including CCSD(2) (only available in RHF implementation) and COSX, which are available for EOM-CCSD are also available for STEOMCC. The most important steps in a STEOMCC calculation are the IP-EOM and EA-EOM calculations. These steps are performed using the EOM-CCSD module and the relevant keywords are the same as that described in [Excited States via EOM-CCSD](#). The keywords which are exclusive to the RHF STEOM module are:

```
%mdci
#RHF STEOM parameters - defaults displayed
DoCISNat true           # automatic selection of active space
NActIP 3                 # number of states defined as active in the IP calculation
NActEA 2                 # number of states defined as active in the EA calculation
DoTriplet false         # target state of triplet multiplicity
DoDbFilter true          # filters out states with doubles excitation character
DoNewActSch true         # new active space selection scheme for STEOM-CCSD
DoSOLV                  # perturbative correction for solvation effects (experimental)
#Default values for automatic active space selection scheme
OTresh 0.001            # Cut off occupation of CIS natural orbitals in IP calculation
VThresh 0.001           # Cut off occupation of CIS natural orbitals in EA calculation
IPSThrs 80              # The percentage singles threshold for the IP calculation
EASThrs 80              # The percentage singles threshold for the EA calculation
end
```

The keywords pertaining to the UHF STEOM module are:

```
%mdci
#UHF STEOM parameters - defaults displayed
DoCISNat true           # automatic selection of active space
NActIP_a 3              # number of states defined as active in the IP calculation
                        # for the removal of an  $\alpha$  electron
NActIP_b 3              # number of states defined as active in the IP calculation
                        # for the removal of a  $\beta$  electron
NActEA_a 2              # number of states defined as active in the EA calculation
                        # for the attachment of an  $\alpha$  electron
NActEA_b 2              # number of states defined as active in the EA calculation
                        # for the attachment of a  $\beta$  electron
DoDbFilter true          # filters out states with doubles excitation character
UseQROs false           # use QROs or not
DoNewActSch true         # new active space selection scheme for STEOM-CCSD
#Default values for automatic active space selection scheme
OTresh 0.001            # Cut off occupation of CIS natural orbitals in tIPcalc
↪calculations
VThresh 0.001           # Cut off occupation of CIS natural orbitals in EAcalc
↪calculations
IPSThrs 80              # The percentage singles threshold for the IP calculations
EASThrs 80              # The percentage singles threshold for the EA calculations
end
```



## 5.10.2 Selection of Active space

The results of a STEOM-CC calculation depend upon the number of roots selected as active in the EOMIP and EOMEA calculations. In ORCA, they are chosen automatically, by using state-averaged CIS natural transition orbitals (NTO). By default, the number of roots included in this initial CIS computation is equal to the number of roots requested in STEOM (NRoots). However, this can be modified setting NRootsCISNAT to higher values. The orbitals up to a predefined occupation are then chosen to be active in the EOMIP and EOMEA calculations, and this is controlled by the keywords OThresh and VThresh respectively. Now, there are two possible ways to choose active space. One is to use the criteria of percentage occupation of NTO's as described in ref [666]. However, a newer and more robust approach is to use the criteria of absolute occupation, which is default in the current implementation. One can switch on the old percentage occupation based active space selection by setting DoNewActSch to false (not recommended).

One can also select the active spaces manually by turning the DoCISNat to false and setting the NActIP and NActEA (RHF STEOM calculation) or the NActIP\_a, NActIP\_b, NActEA\_a and NActEA\_b (UHF STEOM calculation) to desired values. However, this is not recommended for general uses. The following snippet shows the output of the active orbital selection procedure on a closed-shell molecule:

```
-----
STATE AVERAGED NATURAL ORBITALS FOR ACTIVE SPACE SELECTION
-----

Solving eigenvalue problem for the occupied space ... Occupied block occupation :
  0  0.000478
  1  0.002266
  2  0.169928
  3  0.171663
  4  0.310125
  5  0.345541
Orbital taken as active for IP roots:
  0  0.345541
  1  0.310125
  2  0.171663
  3  0.169928
done
Solving eigenvalue problem for the virtual space ... Virtual block occupation :
  6  0.640886
  7  0.332262
  8  0.017272
  9  0.005326
 10  0.001752
 11  0.000667
 12  0.000574
 13  0.000540
 14  0.000160
 15  0.000150
 16  0.000139
 17  0.000086
 18  0.000082
 19  0.000037
 20  0.000023
 21  0.000016
 22  0.000013
 23  0.000008
 24  0.000003
 25  0.000002
 26  0.000001
 27  0.000000
 28  0.000000
 29  0.000000
 30  0.000000
 31  0.000000
```

(continues on next page)

(continued from previous page)

```

32  0.000000
33  0.000000
34  0.000000
35 -0.000000
Orbital taken as active for EA roots :
  0  0.640886
  1  0.332262
  2  0.017272
done
No of roots active in IP calculation: 4
No of roots active in EA calculation: 3

```

### 5.10.3 Active space selection using TD-DFT densities

Instead of using a CIS calculation for selected the Active Space roots, a TD-DFT based one can also be considered. Be aware that using DFT Kohn-Sham orbitals for computing the CCSD GS energy can lead to some instabilities and give incorrect results.

The main interest of this approach is to start the STEOM-CCSD calculation with TD-DFT electronic densities which are in general better than the CIS one, especially for some specific compounds (metallic complexes for example). The computed TD-DFT densities are also often more stable than the CIS one. It will however slow down the calculation.

The input has to be written like this:

```

!RHF/UHF BHANDHLYP STEOM-CCSD TZVP

%mdci
nroots 10
tddftguess true
end

%tddft
nroots 10
end

*xyz 0 1

```

Any DFT functional can be used but we recommend one with a decent amount of HF exchange. On top of this, the keyword `TDDFTGuess` has to be set to true in `mdci` block and the `tddft` has to be added together with the `NRoots` keyword. In both input blocks (`%mdci` and `%tddft`) the same number of roots has to be given. Starting from ORCA 6, using TD-DFT guess with an UHF reference is also possible.

### 5.10.4 The reliability of the calculated excitation energy

The excitation energy for any states calculated in STEOM-CC are only reliable when the dominant excitation for that states are confined within the active space. This can be verified from the percentage active character of the calculated states, an *a posteriori* diagnostic which is defined as

$$\%activecharacter = \frac{\sum_{m,e} C(m,e) * C(m,e)}{\sum_{i,a} C(i,a) * C(i,a)} * 100$$

for closed-shell systems and takes the form,

$$\%activecharacter = \frac{\sum_{m,e} C(m,e) * C(m,e) + \sum_{\bar{m},\bar{e}} C(\bar{m},\bar{e}) * C(\bar{m},\bar{e})}{\sum_{i,a} C(i,a) * C(i,a) + \sum_{\bar{i},\bar{a}} C(\bar{i},\bar{a}) * C(\bar{i},\bar{a})} * 100.$$

within the UHF formalism. The roots which have *%activecharacter* higher than 98.0 are considered to be converged with respect to the active space.

-----  
STEOM-CCSD RESULTS  
-----

IROOT= 1: 0.145412 au 3.957 eV 31914.3 cm\*\*<sup>-1</sup>  
 Amplitude Excitation  
 -0.169361 4 -> 8  
 -0.984822 7 -> 8

Percentage Active Character 99.86

Amplitude Excitation in Canonical Basis  
 -0.166580 4 -> 8  
 -0.975432 7 -> 8  
 -0.124356 7 -> 13

IROOT= 2: 0.309409 au 8.419 eV 67907.5 cm\*\*<sup>-1</sup>  
 Amplitude Excitation  
 0.994141 7 -> 9

Percentage Active Character 99.78

Amplitude Excitation in Canonical Basis  
 -0.990029 7 -> 9

IROOT= 3: 0.336993 au 9.170 eV 73961.4 cm\*\*<sup>-1</sup>  
 Amplitude Excitation  
 -0.994078 5 -> 8

Percentage Active Character 99.10

Amplitude Excitation in Canonical Basis  
 -0.984116 5 -> 8  
 -0.136769 5 -> 13

IROOT= 4: 0.357473 au 9.727 eV 78456.2 cm\*\*<sup>-1</sup>  
 Amplitude Excitation  
 0.181761 4 -> 10  
 0.728209 6 -> 8  
 0.611668 7 -> 10  
 -0.191540 7 -> 12

Percentage Active Character 94.10

Warning:: the state may have not converged with respect to active space  
 ----- Handle with Care -----

Amplitude Excitation in Canonical Basis  
 -0.184144 4 -> 10  
 -0.725183 6 -> 8  
 -0.633718 7 -> 10

IROOT= 5: 0.386654 au 10.521 eV 84860.8 cm\*\*<sup>-1</sup>  
 Amplitude Excitation  
 0.980406 4 -> 8  
 -0.178551 7 -> 8

Percentage Active Character 99.79

(continues on next page)

(continued from previous page)

Amplitude	Excitation in Canonical Basis	
0.971678	4 ->	8
0.122877	4 ->	13
-0.179242	7 ->	8
IROOT= 6: 0.444881 au 12.106 eV 97640.1 cm** <sup>-1</sup>		
Amplitude	Excitation	
-0.995150	6 ->	9
Percentage Active Character 99.69		
Amplitude	Excitation in Canonical Basis	
-0.989966	6 ->	9

If the *%activecharacter* for any calculated state is less than 98, that state may have not converged with respect to active space and the excitation energy for that particular state is less reliable. The user should request a larger number of roots under those conditions.

### 5.10.5 Removal of IP and EA states with double excitation character

To obtain accurate results with STEOM-CCSD, only the  $\hat{S}$  amplitudes corresponding to the states dominated by single excitations should be included in the second similarity transformation. This is ensured in ORCA in two ways. First, the root following (FollowCIS) is activated by default so that it converges to the states dominated by singly excited guess vectors. This avoids the calculation of so called 'satellite states', which are of double excitation character with respect to the ground state. Secondly, among the converged IP and EA roots, the states which have *%singles* character below a certain predefined threshold (i.e. controlled by the keywords *IPThresh* and *EAThresh*) are automatically excluded from the second similarity transformation.

EOM-CCSD RESULTS				
-----				
IROOT=	1:	0.105316 au	2.866 eV	23114.2 cm** <sup>-1</sup>
Amplitude		Excitation		
0.697547		x ->	8	
IROOT=	2:	0.217925 au	5.930 eV	47829.1 cm** <sup>-1</sup>
Amplitude		Excitation		
-0.701454		x ->	9	
IROOT=	3:	0.304098 au	8.275 eV	66741.8 cm** <sup>-1</sup>
Amplitude		Excitation		
-0.700458		x ->	10	
IROOT=	4:	0.350387 au	9.535 eV	76901.1 cm** <sup>-1</sup>
Amplitude		Excitation		
0.702705		x ->	11	
IROOT=	5:	0.651462 au	17.727 eV	142979.4 cm** <sup>-1</sup>
Amplitude		Excitation		
0.637352		x ->	12	
0.121747		x ->	8	4 -> 10
0.177039		x ->	8	5 -> 9
0.109987		x ->	9	5 -> 8
-0.206789		x ->	8	7 -> 10
-0.109870		x ->	10	7 -> 8
EA STATE=	0:	percentage singles	95.282	
EA STATE=	1:	percentage singles	96.981	
EA STATE=	2:	percentage singles	96.540	
EA STATE=	3:	percentage singles	97.844	
EA STATE=	4:	percentage singles	68.884	

(continues on next page)

(continued from previous page)

Warning: high double excitation character, excluding from the STEOM transformation  
 Final no active EA roots: 4

Note that the use of CIS natural transition orbitals can lead to convergence issues for the IP and EA states which are dominated by double excitation character. This can be remedied by setting `DoDbFilter` to true.

### 5.10.6 Transition and difference densities

At the end of a STEOM computation, it is possible to store the final eigenvectors in a file “*job.cis*”, in analogy with what is done for CIS and TD-DFT computations. This file can be obtained by setting `DoStoreSTEOM` true in the input. This file can then be processed by `orca_plot` to obtain the difference and / or the transition densities.

An Natural Transition Orbitals analysis can also be performed within the STEOM-CCSD scheme, as described in *Natural Transition Orbitals*. It can be performed by setting the keyword `DoSTEOMNatTransOrb` to true.

### 5.10.7 Properties

The dipolar and transition moments (as well as the oscillator strength) can be computed within the STEOM module using different kinds of approximations. Please cite our paper on these corrected STEOM transition densities [667]! Starting from ORCA 5, new defaults (`DoSimpleDens` false) are used that are much better than the previous CIS-like approximation, and the full option is of CC3-like quality.

```
%mdci
DosimpleDens false      # Default, using the STEOM-CCSD density + some doubles_
↪effect.
AddL2term true

DosimpleDens false      # using the STEOM-CCSD density + some doubles effect.
AddL2term true          # + neglected GS double
UpdateL1 true

DosimpleDens false      # using the STEOM-CCSD density + some doubles effect.
AddL2term true          # + neglected GS double + doubles from EOM-CCSD
UpdateL1 true           # (expensive, but of CC3 quality - see reference)
AddDDTerm true

end
```

By default, the STEOM-CCSD densities with `AddL2term` true should be used for all calculation as discussed in ref. [667].

### 5.10.8 Solvation (Experimental)

In STEOM-CCSD, the excitation energies and densities can be corrected using the CPCM solvation scheme in ORCA.

To use it, the keyword `DoSolv` has to be set to true in the `%mdci` block and the simple keyword `CPCM` (or `SMD`) + name of the solvent has to be given.

```
!CPCM(ethanol) STEOM-CCSD TightSCF def2-TZVP def2-TZVP/C def2/J

%mdci
Nroots 5
DoSolv true
end
```