3.11.2 Basis Usage / List of Features

The following single-reference methods are currently implemented in the AUTOCI.

Reference	Correlation	Energy	Gradient
RHF	CID	✓	✓
RHF	CISD	✓	✓
RHF	CISDT	✓	
RHF	CCD	✓	✓
RHF	CCSD	✓	✓
RHF	CCSD[T]	✓	✓
RHF	CCSD(T)	✓	✓
RHF	CCSDT	✓	
RHF	CEPA(0)	✓	✓
RHF	CC2	✓	
RHF	QCISD	✓	
RHF	MP2	✓	✓
RHF	MP3	✓	✓
RHF	MP4	✓	✓
RHF	MP4(SDQ)	✓	✓
UHF	CID	✓	✓
UHF	CISD	✓	✓
UHF	CISDT	✓	✓
UHF	CCD	✓	✓
UHF	CCSD	✓	✓
UHF	CCSD[T]	✓	✓
UHF	CCSD(T)	✓	✓
UHF	CCSDT	✓	✓
UHF	CCSDTQ	✓	
UHF	CCSDT-1	✓	
UHF	CCSDT-2	✓	
UHF	CCSDT-3	✓	
UHF	CCSDT-4	✓	
UHF	CEPA(0)	✓	✓
UHF	CC2	✓	
UHF	CC3	✓	
UHF	QCISD	✓	
UHF	MP2	✓	✓
UHF	MP3	✓	✓
UHF	MP4	✓	✓
UHF	MP4(SDQ)	✓	✓
UHF	MP5	✓	✓
ROHF	CISD	✓	

Any AUTOCI single-reference methods can be called from the simple keyword by prepending `AUTOCI-` to the correlation method, for instance

```
! AUTOCI-CCSD
```

The prefix 'AUTOCI' is added to distinguish from the handwritten implementations in `orca_mdci` and `orca_mp2`.

> ⚠️ **Warning**
>
> Higher order CC calculations like CCSDT and CCSDTQ have extreme memory requirements in the CC Iterations. For UHF CCSDTQ particularly (computational efford scales as N^{10} !) only systems with few electrons and basis

> functions are feasible depending on your achitecture.

For a CASSCF reference wavefunction, the prefix is omitted. The following multi-reference methods and their un-relaxed densities are available.

Reference	Correlation	Energy	Unrelaxed Density
CASSCF	FIC-MRCI	✓	✓
CASSCF	FIC-MRCEPA(0)	✓	✓
CASSCF	FIC-MRACPF	✓	✓
CASSCF	FIC-MRAQCC	✓	✓
CASSCF	FIC-MRCC	✓	Linearized
CASSCF	FIC-DDCI3	✓	✓
CASSCF	FIC-NEVPT2	✓	✓
CASSCF	FIC-NEVPT3	✓	✓
CASSCF	FIC-NEVPT4(SD)	✓	✓

> ⓘ **Note**
>
> If a multi-reference method is specified as a simple keyword, all information about reference spaces, number of roots etc. is taken from the CASSCF module that is assumed to be run in advance. More refined settings require the `%autoci` block in the input.

3.11.3 Analytic Nuclear Gradients with AUTOCI

Obtaining accurate geometries is crucial to computing molecular properties accurately. In order to perform geometry optimisations, the nuclear gradient is necessary and while this can easily be obtained using numerical finite difference methods, it is also quite costly. More importantly, perhaps, is the fact that numeric derivatives tend to become unstable. Therefore, being able to evaluate analytic gradients is of vital importance. Using the AGE, a general framework has been built that supports arbitrary-order CI, CC and MPn nuclear gradients (and other derivatives).[362]

An example is shown below how to optimise a geometry using AUTOCI's gradients at the CCSD level of theory

```
! RHF cc-pVTZ AUTOCI-CCSD VerytightSCF Opt

%maxcore 10000

*xyz 0 1
...
*
```

The analytic gradients can even be used to perform semi-numerical frequency calculations

```
! AUTOCI-CCSD NumFreq
```

Besides nuclear gradients, all other first-order properties available in ORCA are available for the respective methods, such as dipole/quadrupole moments, hyperfine couplings or quadrupole splittings. As discussed above, (un)relaxed densities can be requested via

```
%autoci
  density relaxed
end
```

For geometry optimisations, both the unrelaxed and relaxed densities are computed automatically and do not need to be requested explicitly.

3.11.4 AUTOCI Response Properties via Analytic Derivatives

For single-reference methods (CCSD and CCSD(T)), some response properties could be calculated via the analytic derivative scheme.

Response Property	Reference	Correlation	Orbital Response
Static Polarizabilities*	RHF	CCSD	Unrelaxed / Relaxed
	RHF	CCSD(T)	Unrelaxed / Relaxed
	UHF	CCSD	Unrelaxed / Relaxed
	UHF	CCSD(T)	Unrelaxed / Relaxed
NMR Shielding (GIAO)	RHF	CCSD	Unrelaxed / Relaxed
	RHF	CCSD(T)	Unrelaxed / Relaxed
	UHF	CCSD	Unrelaxed / Relaxed
	UHF	CCSD(T)	Unrelaxed / Relaxed
G-Tensor (GIAO)	UHF	CCSD	Unrelaxed / Relaxed
	UHF	CCSD(T)	Unrelaxed / Relaxed

(* dipole-dipole, dipole-quadrupole, quadrupole-quadrupole) Note that for NMR shielding and EPR g-Tensor the fully analytic GIAO approach (see *Nuclear Magnetic Resonance (NMR) Parameters* and *Gauge origin treatment*) is the default and the common gauge approach (less accurate) is not available.

To calculate these properties, the following quantities need to be computed:

- Unperturbed CC amplitudes
- Unperturbed CC Λ (Lagrange Multiplier)
- Unperturbed 1-RDM
- Unperturbed 2-RDM (†)
- Unperturbed Z-Vector (†)
- Perturbed CC amplitudes
- Perturbed CC Λ (Lagrange Multiplier)
- Perturbed 1-RDM
- Perturbed 2-RDM (†)
- Perturbed Z-Vector (†)

Tasks labelled with (†) are optional - they are only required for the "relaxed" scheme of computing response properties. With the "relaxed" scheme, the orbitals are allowed to relax to external perturbations. (See: *orbital relaxation*

Unperturbed Parameters

From CC gradient theory and as stated in the previous *MDCI section*, the CC energy Lagrangian could be written as (denoting the similarity-transformed Hamiltonian as $\bar{H} = e^{-T}He^{T}$):

$$L_{\text{CC}} = \langle \Phi_0 | \bar{H} | \Phi_0 \rangle + \sum_{\eta} \lambda_{\eta} \langle \Phi_{\eta} | \bar{H} | \Phi_0 \rangle + \sum_{ai} f_{ai} z_{ai}$$

Equivalently, we could also express this Lagrangian in terms of density matrices:

$$L_{\text{CC}} = \sum_{pq} h_{pq} \frac{\partial L_{\text{CC}}}{\partial h_{pq}}$$

$$+ \sum_{pqrs} (pq|rs) \frac{\partial L_{\text{CC}}}{\partial (pq|rs)}$$

$$= \sum_{pq} h_{pq} D_{pq} + \sum_{pqrs} (pq|rs) d_{pqrs}$$

in which the one- and two-body reduced density matrices (RDMs) could be decomposed into the amplitude and orbital contributions:

$$D_{pq} = \gamma_{pq} + z_{pq} \delta_{pa} \delta_{qi}$$

$$d_{pqrs} = \Gamma_{qs}^{pr} + z_{pq} \sum_{k} \left(\delta_{pa} \delta_{qi} \delta_{rk} \delta_{sk} - \delta_{pa} \delta_{qk} \delta_{rk} \delta_{si} \right)$$

The above are known as the relaxed 1- and 2-RDMs, since the terms involving the z_{pq} arise due to *orbital relaxation*. The CC amplitude contributions (a.k.a. the unrelaxed 1- and 2-RDMs) are defined as:

$$\gamma_{pq} = \langle \Phi_0 | (1 + \Lambda) e^{-T} E_q^p e^{T} | \Phi_0 \rangle$$

$$\Gamma_{qs}^{pr} = \langle \Phi_0 | (1 + \Lambda) e^{-T} \left(E_q^p E_s^r - E_s^p \delta_{rq} \right) e^{T} | \Phi_0 \rangle$$

These quantities depend on the CC amplitudes t_{η} and Lagrange multipliers λ_{η}, for which we need to solve the amplitude- and Λ-equations:

$$\langle \Phi_{\eta} | \bar{H} | \Phi_0 \rangle = 0$$

$$\langle \Phi_0 | (1 + \Lambda)(\bar{H} - E_{\text{CC}}) | \Phi_{\eta} \rangle = 0$$

Perturbed Parameters

The well-known Wigner's $2n + 1$ rule of perturbation theory states: wavefunction parameters up to the n-th order are required for the energies at the order of $2n + 1$. For non-variational methods like CC, the Lagrangian approach is employed, resulting in a variational energy functional (albeit non-Rayleigh-Ritz, thus no guaranteed upper bound), which enables the analytic evaluation of derivatives. The $2n + 2$ rule of the Lagrange multiplier in such Lagrangian functional states: Lagrange multipliers up to the n-th order are required for the energies at the order of $2n + 2$.

In practice, and here in `ORCA-AUTOCI` implementation, these two rules (especially the $2n + 2$ rule) are not fully exploited. This is due to the consideration of computational efficiency for certain molecular propeties. For example, the general expression for the second derivative of the CC energy, subject to two external perturbations x and y, could be written as:

$$\frac{\text{d}^2 E}{\text{d}x\text{d}y} = \langle \Phi_0 | (1 + \Lambda) e^{-T} \frac{\partial^2 H}{\partial x \partial y} e^{T} | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | (1 + \Lambda) \left[e^{-T} \frac{\partial H}{\partial x} e^{T}, \frac{\text{d}T}{\text{d}y} \right] | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | (1 + \Lambda) \left[e^{-T} \frac{\partial H}{\partial y} e^{T}, \frac{\text{d}T}{\text{d}x} \right] | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | (1 + \Lambda) \left[\left[e^{-T} H e^{T}, \frac{\text{d}T}{\text{d}x} \right], \frac{\text{d}T}{\text{d}y} \right] | \Phi_0 \rangle$$

which is known as the "symmetric" expression, satisfying both the $2n + 1$ and $2n + 2$ rules. However, the following "asymmetric" expression might be preferred for mixed derivatives:

$$\frac{\mathrm{d}^2 E}{\mathrm{d}x \mathrm{d}y}$$

$$= \langle \Phi_0 | (1 + \Lambda) e^{-T} \frac{\partial^2 H}{\partial x \partial y} e^T | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | (1 + \Lambda) \left[e^{-T} \frac{\partial H}{\partial x} e^T, \frac{\mathrm{d}T}{\mathrm{d}y} \right] | \Phi_0 \rangle$$

$$+ \langle \Phi_0 | \frac{\mathrm{d}\Lambda}{\mathrm{d}y} e^{-T} \frac{\partial H}{\partial x} e^T | \Phi_0 \rangle$$

In this way, we solve the perturbed T and Λ equations subject to only a single perturbation. For NMR shielding this means we can solve these equations only for the magnetic-field perturbation and there are 3 equations for $\mathrm{d}T/\mathrm{d}B_i$ and 3 for $\mathrm{d}\Lambda/\mathrm{d}B_i$. If we were to use the first "symmetric" expression, although we could save some effort by not having to solve for any perturbed Λ, we still need to solve for perturbed T, subject to both the magnetic-field perturbation (3 equations) and the nuclear magnetic moment perturbation ($3N_{\text{atoms}}$ equations).

To obtain the equations for perturbed T and Λ, we differentiate the unperturbed equations, w.r.t. external perturbation χ

$$\frac{\partial}{\partial \chi} \langle \Phi_\eta | \bar{H} | \Phi_0 \rangle = 0$$

$$\frac{\partial}{\partial \chi} \langle \Phi_0 | (1 + \Lambda)(\bar{H} - E_{\text{CC}}) | \Phi_\eta \rangle = 0$$

Then we build the perturbed 1- and 2-RDMs, by taking derivative of the unperturbed expressions:

$$\frac{\partial \gamma_{pq}}{\partial \chi} = \frac{\partial}{\partial \chi} \langle \Phi_0 | (1 + \Lambda) e^{-T} E_q^p e^T | \Phi_0 \rangle$$

$$\frac{\partial \Gamma_{qs}^{pr}}{\partial \chi} = \frac{\partial}{\partial \chi} \langle \Phi_0 | (1 + \Lambda) e^{-T} \left(E_q^p E_s^r - E_s^p \delta_{rq} \right) e^T | \Phi_0 \rangle$$

The expressions and relevant modules are produced by the code generator. Note that for the currently available response properties - polarizabilities, NMR shielding, EPR g-tensors - which are technically one-particle properties, for which the 2-RDMs do not appear explicitly in the energy derivative expression, both the unperturbed 2-RDMs and the perturbed 2-RDMs are only needed when *orbital relaxation* needs to be computed.

Orbital Relaxation

By including the orbital relaxation effect into property calculations, we essentially allow the MO to respond to external perturbations. This leads to the following parameterization of MO coeffcients:

$$\mathbf{C}(\chi) = \mathbf{C}(0)\mathbf{U}(\chi)$$

then MO derivatives could be written as:

$$\mathbf{C}^\chi \equiv \frac{\partial \mathbf{C}(\chi)}{\partial \chi} = \mathbf{C}(0) \frac{\partial \mathbf{U}(\chi)}{\partial \chi} \equiv \mathbf{C}\mathbf{U}^\chi$$

where $\mathbf{U}^\chi$ are solutions to the *CPSCF* equations. For unrelaxed calculations the MO coefficients are always $\mathbf{C}(0)$. In relaxed calculations, the MO response enters relevant CC equations in two ways.

First, as described *above*, for second-order response properties (like polarizabilities, NMR shielding, etc.) we need to solve the first-order T and Λ equations. Within a relaxed scheme, any derivative integrals (Fock, ERI, etc.) entering

the CC equations contain CPSCF contributions. For example:

$$\frac{\partial(pq|rs)}{\partial\chi} =$$

$$\sum_{\mu\nu\kappa\tau} C_{\mu p} C_{\nu q} \frac{\partial(\mu\nu|\kappa\tau)}{\partial\chi} C_{\kappa r} C_{\tau s}$$

$$+ \sum_{\mu\nu\kappa\tau} C_{\mu p}^{\chi} C_{\nu q} (\mu\nu|\kappa\tau) C_{\kappa r} C_{\tau s}$$

$$+ \sum_{\mu\nu\kappa\tau} C_{\mu p} C_{\nu q}^{\chi} (\mu\nu|\kappa\tau) C_{\kappa r} C_{\tau s}$$

$$+ \sum_{\mu\nu\kappa\tau} C_{\mu p} C_{\nu q} (\mu\nu|\kappa\tau) C_{\kappa r}^{\chi} C_{\tau s}$$

$$+ \sum_{\mu\nu\kappa\tau} C_{\mu p} C_{\nu q} (\mu\nu|\kappa\tau) C_{\kappa r} C_{\tau s}^{\chi}$$

For unrelaxed calculations the last 4 terms are zero.

Second, to compute the relaxed 1- and 2-RDMs and their derivatives, the z-vector contribution needs to be included. This is a consequence of requiring the CC energy to be invariant to orbital rotation, in the presence of external perturbations. This nice trick by Schaefer and Handy allows gradient to be evaluated without having to solve $3N_{atoms}$ equations for the orbitals. Instead, only the so-called "z-vector" equation needs to be solved. It thus opens a door towards efficient analytic response property evaluations. The z-vector equations are of the same form as the *CPSCF* equations:

$$\mathbf{Az} = \mathbf{X}$$

where the $\mathbf{A}$ matrix is exactly the same as the CPSCF LHS, and the RHS matrix $\mathbf{X}$ is the orbital gradient of the CC energy functional. For second-order response properties, the derivatives (subject to each perturbation requested) of this equation also need to be solved.

The discussion of relaxed properties vs unrelaxed ones are quite common. Generally, we believe that for electric-field dependent properties like polarizabilities, inclusion of orbital relaxation does not necessarily lead to more accurate values compared with experiments. However it is generally regarded essential to include orbital relaxation for magnetic-field dependent properties like NMR and EPR parameters.

Finally, it is also worth noting that this orbital relaxtion scheme does not recover the MO response of using HF orbitals in a CC wavefunction setup, which relies on the orbital-optimized coupled-cluster (OOCC) methods, as described in the *previous section*.

Input Options

The input parameters for the wavefunctions are controlled by the `%autoci` block, while the property-related parameters are controlled by respective property blocks, like `elprop` and `eprnmr`. Some useful options are gathered below.

```
%autoci
    citype
        CCSD(T)
        CCSD
    STol      1e-06 # residue convergence tolerance
    MaxIter   50    # maximum number of iterations
    MaxDIIS   5     # depth of the DIIS memory
    density         # need at least unrelaxed density (see details below)
        unrelaxed
        relaxed
end

%elprop
```

```
    polar          true    # dipole-dipole polarizability
    polardipquad   true    # dipole-quadrupole polarizability
    polarquadquad  true    # quadrupole-quadrupole polarizability
end

%eprnmr
    gtensor        true    # calculate the g-tensor
    NMRShielding   1       # for chosen nuclei - specified with the Nuclei keyword
                   2       # for all nuclei, equivalent to the 'NMR' simple input
end
```

Output files: Calculation outputs generated from AutoCI-Response modules are re-directed into separate outputfiles, with the names `basename.autociresp_efield`, `basename.autociresp_bfield`, etc.

Input Examples

Highly accurate NMR shielding for HF:

```
! AUTOCI-CCSD(T) NMR pcsseg-4 cc-pwcv5z/c ExtremeSCF

%autoci
  STol 1e-10
  density relaxed
end

%eprnmr
  Tol 1e-12
end

%coords
  CTyp xyz
  Charge 0
  Mult 1
  Units bohrs
  coords
    H  0.00000000   0.00000000   1.64411926
    F  0.00000000   0.00000000  -0.08721704
  end
end
```

G-Tensor for NH radical, with the SOC operator being the effective nuclear charge (see *SOCoptions*):

```
! UHF cc-pVTZ G-Tensor ExtremeSCF AUTOCI-CCSD(T)
! NoFrozenCore Bohrs

%rel
  SOCType 1
  Zeff[1] = 1.0   # H
  Zeff[7] = 4.55  # N
end

%eprnmr
  Tol 1e-10
end

%autoci
  STol 1e-9
  Density relaxed
end
```

```
# NH radical
* xyz 0 3
N 0.0  0.0  0.0
H 0.0  0.0  1.957794
*
```

By changing the `%rel` block to the following we can do the same calculation, using a more accurate SOC description (SOMF):

```
%rel
  # these two are defaults
  SOCType 3
  SOCFlags 1,4,3,0
end
```

3.11.5 Fully Internally Contracted MRCI (FIC-MRCI)

Starting point for any multireference approach is a reference wavefunction that consists of multiple determinants or configurations state functions (CSFs). In many instances this is the complete active space SCF (CASSCF) wave-function. In the uncontracted MRCI approach, as implemented in the `orca_mrci` module, the wavefunction is expanded in terms of excited CSFs that are generated by considering excitations with respect to all reference CSFs. The methodology scales with the number of reference CSFs and hence is restricted to small reference spaces. More-over, the configuration driven algorithm used in `orca_mrci` keeps all integrals in memory, which further limits the overall size of the molecule.

Internal contraction as proposed by Meyer and Siegbahn avoids these bottlenecks [418, 419]. Here, excited CSFs are generated by applying the excitation operator to the reference wavefunction as whole. The fully internally contracted MRCI (FIC-MRCI) uses the same internal contraction scheme as the FIC-NEVPT2 (aka PC-NEVPT2). The entire methodology as well as a comparison with the conventional uncontracted MRCI is reported in our article [420]. The CEPA0, ACPF and AQCC variants are straight forward adoptions [421]. The residue of the FIC-MRCI ansatz

$$R_K = \langle \Phi_{qs}^{pr} | \hat{H} - E^{\mathrm{CAS}} - \lambda E_c | \Psi^{\mathrm{FIC}} \rangle \,, \qquad (3.130)$$

is modified by the factor

$$\lambda = \begin{cases} 1 & \mathrm{MRCI} \\ 0 & \mathrm{CEPA0} \\ \frac{2}{N_e} & \mathrm{ACPF} \\ \frac{1-(N_e-3)(N_e-2)}{N_e \cdot (N_e-1)} & \mathrm{AQCC} \end{cases}$$

Here, E_c is the correlation energy and Φ_{qs}^{pr} denote the internally contracted CSF that arise from the action of the spin-traced excitation operators on the CAS-CI reference wave function

$$\Phi_{qs}^{pr} = E_q^p E_s^r \, |\Psi^{\mathrm{CAS}}\rangle \,.$$

In case of ACPF and AQCC, the λ factor explicitly depends on the number of correlated electrons (N_e).

The general input structure is like that of the CASSCF module, e.g., the following example input reads an arbitrary set of orbitals and starts the FIC-MRCI calculation. The internal contracted formalism requires CAS-CI reduced densities up to fourth-order, which can be expensive to construct. By default, the density construction is sped up using the prescreening (PS) approximation reported in Section *N-Electron Valence State Perturbation Theory (NEVPT2)*.[422]

```
!def2-tzvp moread allowrhf noiter nofrozencore
%moinp "start.gbw"  # could be from CASSCF

%autoci
citype   FIC-MRCI      # Fully internally contracted MRCI (singles, doubles)
         FIC-MRCEPA(0) # CEPA0 version of FIC-MRCI
```

```
        FIC-MRACPF    # ACPF  version of FIC-MRCI
        FIC-MRAQCC    # AQCC  version of FIC-MRCI
        FIC-DDCI3     # FIC-MRCI without the IJAB excitation

# CAS-CI reference wavefunction
nel 2
norb 2
mult 1,3
nroots 3,1

nthresh 1e-6  # removal of linear dependencies in the IC-CSFs
D3TPre        1e-14 # default density truncation of the 3-RDM
D4TPre        1e-14 # default density truncation of the 4-RDM

# Davidson correction for the FIC-MRCI
DavidsonOpt 0  # none (default)
            1  # Davidson correction
end
```

Currently, the program is capable of computing total energies and vertical excitation energies. More features will be available with future releases.

Input Example

```
# potentially "!FIC-MRCEPA(0)" omitting the %autoci block.
!def2-SVP

# get CASSCF orbitals and define states
%casscf
  nel 8
  norb 6
  mult 3
  nroots 1
end

# run FIC-MRCEPA(0)
%autoci
  # CAS settings are automatically copied from the %casscf block, if
  # not defined in %autoci

  citype FIC-MRCEPA(0)
end

# O2 molecule
*xyz 0 3
O 0 0 0
O 0 0 1.28
*
```

3.11.6 Fully Internally Contracted MRCC (FIC-MRCC)

Several approaches have been taken towards extending CC theory to work with genuinely multiconfigurational reference wave functions [423], yet none of these approaches has found widespread adoption. As of 2011, the internally contracted MRCC theory has had a revival, with a rigorous theoretical investigation of several approximations that also proved its orbital invariance [424] and a first report of a polynomial-scaling code obtained through automatic equation generation [425].

Our implementation in ORCA is akin to the previously published formulations in Refs. [424, 425], although everything is formulated rigorously in terms of the spin-free excitation operators $\hat{E}_q^p = \hat{a}^{p\alpha} \hat{a}_{q\alpha} + \hat{a}^{p\beta} \hat{a}_{q\beta}$, using an improved version of the ORCA-AGE code generator.[362] To begin with, the *ansatz* for the wave function is

$$|\Psi\rangle = \mathrm{e}^{\hat{T}}|0\rangle \,, \tag{3.131}$$

where $|0\rangle$ denotes a zeroth order CASSCF reference wave function and the cluster operator can be written as (Einstein's summation convention implied)

$$\hat{T} = \frac{1}{2} t_{ab}^{ij} \hat{E}_i^a \hat{E}_j^b + t_{ab}^{it} \hat{E}_i^a \hat{E}_t^b + \frac{1}{2} t_{ab}^{tu} \hat{E}_t^a \hat{E}_u^b + t_{at}^{ij} \hat{E}_i^a \hat{E}_j^t + t_{au}^{it} \hat{E}_i^a \hat{E}_t^u$$

$$+ t_{ua}^{it} \hat{E}_i^u \hat{E}_t^a + t_{av}^{tu} \hat{E}_t^a \hat{E}_u^v + \frac{1}{2} t_{tu}^{ij} \hat{E}_i^t \hat{E}_j^u + t_{uv}^{it} \hat{E}_i^u \hat{E}_t^v \,.$$

Note that we do not use normal order in the cluster operator.

Inserting the *ansatz* from Eq. (3.131) into the Schrödinger equation and pre-multiplying with the inverse exponential, we obtain the similarity transformed Hamiltonian and the energy expression,

$$\bar{H}|0\rangle = \mathrm{e}^{-\hat{T}} \hat{H} \mathrm{e}^{\hat{T}}|0\rangle = E|0\rangle \,.$$

In our code, the similarity-transformed Hamiltonian is truncated after the quadratic terms since that approximation has been found to only have minor impact on the accuracy of the method [424],

$$\bar{H} \approx \hat{H} + [\hat{H}, \hat{T}] + \frac{1}{2}[[\hat{H}, \hat{T}], \hat{T}] \,.$$

The residual conditions are subsequently obtained by projecting with contravariant excited functions $\langle \tilde{\Phi}_P |$ onto the Schrödinger equation,

$$r_P = \langle \tilde{\Phi}_P | \bar{H} | 0 \rangle \,.$$

For a definition of the contravariant projection functions, we refer to Ref. [420] since this FIC-MRCC implementation uses the same contravariant functions as the published FIC-MRCI implementation. Despite using contravariant projection functions, this is not sufficient to remove all linear dependencies from the set of projection functions $\{\tilde{\Phi}_P\}$, i.e. the metric matrix

$$S_{PQ} = \langle \tilde{\Phi}_P | \tilde{\Phi}_Q \rangle \neq \delta_{PQ}$$

has off-diagonal elements *within* excitation classes and between classes with the same number of inactive and virtual indices (ITAU and ITUA). Hence, the set of projection functions needs to be orthonormalized, which is achieved with Löwdin's canonic orthogonalization in the AUTOCI module.[1]

Input Example

The FIC-MRCC module can be started by specifying the `CIType` keyword in the `%autoci` block or by adding `FIC-MRCC` to the simple input line of an ORCA input file. The following example computes the singlet ground state energy of four hydrogen atoms arranged as a square with a side length of $2a_0$, which is commonly known as the H4 model [426].

---

[1] This is similar to scheme A from Ref. [425].

```
! cc-pVTZ Bohrs  # it is possible to add the `FIC-MRCC' keyword here
                 # and omit the %autoci block below

%maxcore 10000

# get CASSCF orbitals and define states.
%casscf
  nel 2
  norb 2
  mult 1
  nroots 1
end

# run FIC-MRCC
%autoci
  # CAS settings are automatically copied from the %casscf block, if
  # not defined in %autoci
  citype fic-mrcc

  # optional parameters
  nthresh 1e-6        # default removal of linear dependencies in the IC-CSFs
  D3TPre       1e-14 # default density truncation of the 3-RDM
  D4TPre       1e-14 # default density truncation of the 4-RDM
  D5TPre       1e-14 # default density truncation of the 5-RDM
  Density  linearized  # compute the linearized density (not default).
end

* int 0 1
  H 0 0 0 0.0  0.0 0.0
  H 1 0 0 2.0  0.0 0.0
  H 2 1 0 2.0 90.0 0.0
  H 1 2 3 2.0 90.0 0.0
*
```

In this example, ORCA will first run a state-specific CASSCF calculation, and then immediately continue with the fic-MRCC calculation on top of the CASSCF solution from the first step. It is, however, not required to always run a CASSCF calculation before the `autoci` module. Any ORCA `gbw/mp2nat/…` file is accepted through `%moinp`, although that route requires the user to specify the active space in the `autoci` block. `autoci` will then compute a CASCI solution with the provided input orbitals and use that information to drive the correlated calculations.

Please be aware that FIC-MRCC is a very extensive theory, which leads to long run times. The computational effort depends mainly on the number of orbitals, the number of total electrons and the size of the active space. On modestly modern hardware, calculations of ~ 300 orbitals with a CAS(2,2) should be readily achievable. For larger active spaces, such as a CAS(6,6), calculations with a total of ~ 200 orbitals will also complete within a day.

3.11.7 Fully Internally Contracted NEVPT (FIC-NEVPT4(SD)/FIC-NEVPT3/FIC-NEVPT2)

Perturbation theory is a cost-effective alternative to the previously mentioned FIC-MRCI and FIC-MRCC methods. Most often, a reasonable description of electron correlation is already achieved at the second order of the expansion. The CASSCF module features an efficient implementation of the second-order methods, that are able to tackle large molecules and active spaces. For more details and references on the second-order variants, we refer to the dedicated section *N-Electron Valence State Perturbation Theory (NEVPT2)*.

In the following section, the focus is on higher orders of the n-electron valence state perturbation theory (NEVPT), that are available within the AUTOCI framework. Here, all integrals are computed and kept on disk, which limits the application to approximately 700 atomic orbitals (AOs). The appearance of higher-order reduced density matrices is a common feature of internally contracted methods. Like FIC-NEVPT2, the FIC-NEVPT3 and FIC-NEVPT4(SD) require at most the fourth-order reduced density matrix. Its computation can be accelerated with the prescreening method discussed in section *N-Electron Valence State Perturbation Theory (NEVPT2)*.[422, 427] In terms of computation time, the FIC-NEVPT4(SD) is as expensive as FIC-NEVPT3 or a single iteration of the FIC-MRCI method.

FIC-NEVPT3

The FIC-NEVPT3 methodlogy is desribed in detail by Angeli et al.[428] With the first order wave function $\Psi^{(1)}$, the third order energy is given by the expression

$$E^{(3)} = \langle \Psi^{(1)} | V | \Psi^{(1)} \rangle.$$

FIC-NEVPT4(SD)

The fourth-order perturbation theory requires the second wave function, which, in principle, involves triple and quadruple excitation defining the second-order interacting space. The inclusion of the latter is computationally demanding. In the following we report an approximate solution, coined FIC-NEVPT4(SD), which is restricted to the first-order interacting space - the single and double excitations with respect to the reference wave function. This method is reported and analyzed in more detail by Kempfer et al.[429] Denoting the second order wave function with $\Psi^{(2)}$, the fourth order energy is readily expressed as

$$E^{(4)} = \langle \Psi^{(1)} | V | \Psi^{(2)} \rangle - E^{(2)} \langle \Psi^{(1)} | \Psi^{(1)} \rangle.$$

The second contribution in the expression above, $E^{(2)} \langle \Psi^{(1)} | \Psi^{(1)} \rangle$, is a renormalization term. As discussed in the paper,[429] the renormalization term is dropped (default) as it destroys the size consistancy in the absence of the exact quadruple excitations.

Input Example

```
# potentially "!FIC-NEVPT4(SD)"/"!FIC-NEVPT3" instead of the %autoci block.
!def2-tzvp

# get CASSCF orbitals and define the states
%casscf
    nel 8
    norb 6
    mult 3
    nroots 1
end


# run NEVPT4(SD)
%autoci
# CAS settings are automatically copied from the %casscf block, if
# not defined in %autoci

 citype fic-nevpt4(sd)

# optional parameters
nthresh 1e-6       # default removal of linear dependencies in the IC-CSFs
D3TPre        1e-14 # default density truncation of the 3-RDM
D4TPre        1e-14 # default density truncation of the 4-RDM

Density  Unrelaxed  # compute the unrelaxed density (not default).
end

# O2 molecule
* xyz 0 3
O 0 0 0
O 0 0 1.28
*
```

3.11.8 FIC Ansatz: Unrelaxed Densities and Natural Orbitals

The FIC-NEVPT4(SD), the FIC-MRCC and the FIC-MRCI variants have a common set excitation parameters in their wavefunction. For these methods, (linearized) unrelaxed densities are available (`density unrelaxed`). With `!KeepDens` the densities are stored in the density container (**.densities** file on disk), which can be read with the *orca_plot program*. The name of the densities reflect its multiplicity, root count and the actual method. Below is an example snippet of the "orca_plot jobname.densities" call:

```
--------------------
List of density names
--------------------


Index:                                                    Name of Density
-----------------------------------------------------------------------
    0:                                                        jobname.scfp
    1:                                                        jobname.scfr
    2:                                          Tdens-CAS.mult.3.root.0.p
    3:                                          Tdens-CAS.mult.3.root.1.p
    4:                                          Tdens-CAS.mult.3.root.2.p
    5:                                          Tdens-CAS.mult.3.root.0.r
    6:                                          Tdens-CAS.mult.3.root.1.r
    7:                                          Tdens-CAS.mult.3.root.2.r
    8:                     jobname.mult.3.root.0.FIC-NEVPT4(SD).autoci.p
    9:                     jobname.mult.3.root.1.FIC-NEVPT4(SD).autoci.p
   10:                     jobname.mult.3.root.2.FIC-NEVPT4(SD).autoci.p
```

Here, the densities 8-10 are unrelaxed densities of the FIC-NEVPT4(SD) method for the first three triplet roots. With the unrelaxed densities, natural orbitals (`natorbs true`) can be generated. The resulting .gbw files ending with (.nat) can be used to carry out further analysis, e.g. using the MultiWFN program.

```
!FIC-NEVPT4(sd)
...
%autoci
  Density  Unrelaxed  # compute the unrelaxed density
  natorbs true        # generate natural orbitals
end
```

3.11.9 Keywords

Table 3.45: Simple input keywords for AUTOCI methods.

Keyword	Description
AUTOCI-<method>	Activate single-reference correlation module from AUTOCI. Disables MDCI.
FIC-MRCI	
FIC-MRCEPA(0)	
FIC-MRACPF	
FIC-MRAQCC	
FIC-DDCI3	
FIC-MRCC	
FIC-NEVPT2	
FIC-NEVPT3	
FIC-NEVPT4(SD)	

Table 3.46: `%autoci` block input keywords AUTOCI methods.

Keyword	Options	Description
CIType	CID	Correlation method
	CISD	
	CISDT	
	CCD	
	CCSD	
	CCSDT	
	CEPA(0)	
	QCISD	
	CC2	
	CC3	
	CCSD[T]	
	CCSD(T)	
	CCSDT-1	
	CCSDT-2	
	CCSDT-3	
	CCSDT-4	
	FIC-MRCI	
	FIC-MRCEPA(0)	
	FIC-MRACPF	
	FIC-MRAQCC	
	FIC-DDCI3	
	FIC-MRCC	
	FIC-NEVPT2	
	FIC-NEVPT3	
	FIC-NEVPT4(SD)	
	MP2	
	MP3	
	MP4(SDQ)	
	MP4	
	MP5	
STol	1e-6	Residue convergence tolerance
LevelShift	0.2	LevelShift in the DIIS update / denominators
MaxIter	50	Maximum number of iterations
MaxDiis	5	Depth of the DIIS memory
DIISStartIter	2	Apply DIIS starting at iteration 1 (counting starts with 0).
ExcludeHigherExcD]	false	Exclude triples and higher excitations from DIIS procedure
NEl	6	Number of active electrons (CAS only)
NOrb	7	Number of active orbitals (CAS only)
Mult	1	Requested multiplicity block (CAS only)
NRoots	1	Number of roots for mult block (CAS only)
Irrep	0	Requested irrep for mult block (CAS only)
NThresh	1e-6	Threshold for lin. dependencies in the IC-CSFs basis (CAS only)
D3TPre	1e-14	Density truncation in D3 (CAS only)
D4TPre	1e-14	Density truncation in D4 (CAS only)
PrintLevel	3	Amount of printing (1-7)
TrafoType	NoRI	Type of integral transformation
	RITrafo	
KeepInts	false	Keep the transformed integrals on disk
UseOldInts	false	Use the transformed integrals found on disk
RunROHFasUHF	false	Invokes AUTOCI UHF modules with orbitals from ROHF calculation
Density	none	Type of 1-body density requested
	linearized	
	unrelaxed	
	relaxed	
Density2	none	Type of 2-body density requested
	linearized	
	unrelaxed	
	relaxed	
NatOrbs	false	Generate natural orbitals (not available for all methods)

N.B. For a ROHF reference, only CISD calculations can be performed in the current version. However, it is possible to run UHF calculations with an ROHF reference by setting the `RunROHFasUHF` flag to true. Note that this only makes sense when the reference is indeed ROHF, e.g. (calculating the isotropic part of the HFC at CCSD level, running AutoCI UHF CCSD module, with orbitals obtained from the ROHF SCF calculation):

```
! ROHF def2-svp tightscf pmodel AUTOCI-CCSD
```

```
%autoci
    RunROHFasUHF true
end

* xyz 0 2
Cu  0.0  0.0  0.0
*

%eprnmr nuclei = all Cu {aiso} end
```

If one wishes to experiment with the module itself and the reference wavefunction stays constant, it is possible to store the transformed MO integrals on disk (`keepints`) and reuse them (`useoldints`). The program checks only whether the dimension of the integrals on disk match the problem actually solved, i.e. the user is responsible for valid data.

3.12 Arbitrary Order Coupled-Cluster (MRCC interface)

Various higher order theories (CC3, CISDT, CCSDTQ, MP4) are implemented using the *automatic code-generation*. In addition, ORCA features an interface to Kallay's powerful MRCC program. This program must be obtained separately. The interface is restricted to single point energies but can be used for rigid scan calculations or numerical frequencies.

The use of the interface is simple:

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

%mrcc method   "CCSDT"
      ETol    9
      end

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

The Method string can be any of:

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

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

3.13 Complete and Incomplete Active Space Self-Consistent Field (CASSCF and RAS/ORMAS)

3.13.1 Introduction

The complete active space self-consistent field (CASSCF) method is a special form of a multiconfigurational SCF method and can be thought of as an extension of the Hartree-Fock method. It is a very powerful method to study static correlation effects and a solid basis for MR-CI and MR-PT treatments. It can be applied to the ground state and excited states or averages thereof. The implementation in ORCA is fairly general and also allows incomplete active spaces omitting active-active rotations. Example for such *incomplete model spaces* include arbitrary references configurations, restricted active space (RAS) and occupation restricted multiple active spaces (ORMAS). In the following section, the primary focus is on the complete active spaces.

The subject is fairly complex and ultimately require a lot of insight from the user in order to be successful. Thus, in addition to the material in the manual, we have created a CASSCF tutorial, that covers many practical tips on the calculation design and usage of the program.

There are several situations where a CASSCF treatment is a good idea:

- Wave functions with significant multireference character arising from several nearly degenerate configurations (static correlation)
- Wave functions which require a multideterminantal treatment (for example multiplets of atoms, ions, transition metal complexes,)
- Situations in which bonds are broken or partially broken.
- Generation of orbitals which are a compromise between the requirements for several states.
- Starting point for multireference methods covering dynamic correlation (NEVPT2, MRCI, MREOM, …)
- Generation of genuine spin eigenfunctions for multideterminantal/multireference wave functions.

In all of these cases, the single-determinantal Hartree-Fock/DFT methods fail badly, whereas CASSCF remains a good choice. In the latter, the orbitals are divided into three-subspaces: (a) the internal (inactive) orbitals which are doubly occupied in all configuration state functions (CSFs) (b) partially occupied (active) orbitals (c) virtual (external) orbitals which are empty in all CSFs.

A fixed number of electrons is assigned to the internal subspace and the active subspace. If N electrons are "active" in M orbitals, one speaks of a CASSCF(N,M) wave functions. All spin-eigenfunctions for N electrons in M orbitals are included in the configuration interaction step (CAS-CI) and the energy is made stationary with respect to variations in the MO and the CI coefficients. Any number of roots of any number of different multiplicities can be calculated and the CASSCF energy may be optimized with respect to a user defined average of these states.

The CASSCF method has the nice advantage that it is fully variational which renders the calculation of analytical gradients relatively easy. Thus, the CASSCF method may be used for geometry optimizations and numerical frequency calculations. It should be noted that state-averaged CASSCF calculations are not fully variational in general. The orbitals must be optimized for the state of interest.

A number of properties are available in ORCA (g-tensor, ZFS splitting, CD, MCD, susceptibility, dipoles, …). The majority of CASSCF properties such as EPR parameters are computed in the framework of the *quasi-degenerate perturbation theory*. In addition, some properties such as ZFS splittings can also be computed via perturbation theory or rigorously extracted from an effective Hamiltonian. For a detailed description of the available properties and options see section Section 3.13.15. All the aforementioned properties are computed within the CASSCF module. An exception are *Mössbauer parameters*, which are computed with the usual keywords in the %EPRNMR block, when the density (in the density container) is available. In general, properties within the %EPRNMR block are derived using linear reponse theory. An extension of the latter for a CASSCF wave function is reported in section Section 5.26 for a few selected properties.

> ⓘ **Note**

> The CASSCF module hosts the *CASPT2* and *NEVPT2* methods and their variants. Higher order of multi-reference theories (e.g. *FIC-NEVPT4(sd)*) can be found in the *AutoCI module* and the *MRCI module*.

3.13.2 Theory

Denoting the state with I and its spin with S, the CASSCF wave function is written as

$$\left|\Psi_I^S\right\rangle = \sum_k C_{kI}\left|\Phi_k^S\right\rangle.$$

Here, the wave function is expanded in a set of configuration state functions (CSFs), that are a linear combination of determinants adapted to the total spin S.

To define the actual set of CSFs (Φ_I^S), the molecular orbital space is devided into three user defined subspaces:

- The "inactive orbitals" are the orbitals which are doubly occupied in all CSFs.

- The "active orbitals" are the orbitals with variable occupation numbers in the CSFs.

- The "external orbitals" are empty in all CSFs.

A complete model space, CAS(N,M), includes all CSFs, that arise from a full configuration interaction (FCI), where N active electron are distributed among the M active orbitals, while the remaining electrons of the system reside in the doubly occupied inactive orbitals. **The active orbitals are central to ansatz and must be chosen by the user with care.** The list of CSFs grows extremely quickly with the number of active electrons and orbitals (basically factorially). Depending on the actual system, the limit of feasibility is roughly around $\sim$14 active orbitals or about one million CSFs in the active space. Larger active spaces are tractable with approximate FCI solver such as the *Iterative-Configuration-ExpansionCI (ICE-CI)* or the *Density Matrix Renormalization Group (DMRG) approach.*[1] In certain situation, *incomplete model spaces such as the RAS or ORMAS* are viable alternatives.

The expansion coefficients C_{kI} represent the first set of variational parameters. Each CSF is constructed from a common set of orthonormal molecular orbitals $\psi_i(\mathbf{r})$ which are in turn expanded in basis functions $\psi_i(\mathbf{r}) = \sum_\mu c_{\mu i}\phi_\mu(\mathbf{r})$. The MO coefficients $c_{\mu i}$ form the second set of variational parameters.

The energy of the CASSCF wave function is given by the Rayleigh quotient

$$E(\mathbf{c},\mathbf{C}) = \frac{\left\langle\Psi_I^S\left|\hat{H}_{\mathrm{BO}}\right|\Psi_I^S\right\rangle}{\left\langle\Psi_I^S\left|\Psi_I^S\right\rangle\right.},$$

and represents an upper bound to the true total energy. However, CASSCF calculations are *not* designed to provide values for total energy which are close to the exact energy. The purpose of a CASSCF calculation is to provide a qualitatively correct wave function, which forms a good starting point for a treatment of dynamic electron correlation.

The CASSCF method is fully variational in the sense that the energy is made stationary with respect to variations in both sets of MO and CI coefficients. At convergence, the gradient of the energy with respect to the MO and CI coefficients vanishes

$$\frac{\partial E(\mathbf{c},\mathbf{C})}{\partial c_{\mu i}} = 0,$$

$$\frac{\partial E(\mathbf{c},\mathbf{C})}{\partial C_{kI}} = 0.$$

For a complete model space, the wave function and energy are invariant with respect to unitary transformations within the three subspaces. Hence, the orbitals within the subspaces are only defined up to a unitary transformation, and the program needs to make some canonicalization choice. In ORCA, the *final orbitals* by default are:

- natural orbitals in the active space,

- orbitals which diagonalize the CASSCF Fock matrix in the internal space and

- orbitals which diagonalize the CASSCF Fock matrix in the external space.

---

[1] For approximate full CI approaches, CASSCF is neither invariant to active-active rotations nor exactly size-consistent.

Optimization of CASSCF wave functions

In general, except for trivial cases, CASSCF wave functions are considerably more difficult to optimize than RHF (or UHF) wave functions. The underlying reason is that variations in **c** and **C** maybe strongly coupled and the energy functional may have many local minima in (**c**,**C**) space. Consequently, the choice of starting orbitals is of really high importance and the choice which orbitals and electrons are included in the active space has decisive influence on the success of a CASSCF study.

In general, after transformation to natural orbitals, one can classify the active space orbitals by their occupation numbers which vary between 0.0 and 2.0. In general, convergence problems are almost guaranteed if orbitals with occupation numbers close to zero or close to 2.0 are included in the active space. Occupation numbers between 0.02 and 1.98 are typically very reasonable and should not lead to large convergence problems. The reason for the occurrence of convergence problems is that the energy is only very weakly dependent on rotations between internal and active orbitals if the active orbital is almost doubly occupied and similarly for the rotations between external and weakly occupied active orbitals. However, in some cases (for example in the study of potential energy surfaces) it may not be avoidable to include weakly or almost inactive orbitals in the active space and in these cases the use of the most powerful convergence aids is necessary (*vide infra*).

With the exception, of the *TRAH-CASSCF* solver,[121] the ORCA implementation follows a *two-step procedure*, where in every cycle (Macro-Iteration) the CAS-CI problem is solved in the present set of molecular orbitals. The orbital coefficients are then updated with a unitary matrix U

$$\mathbf{c}^{\text{new}} = \mathbf{c}^{\text{old}}\mathbf{U}$$

and a new cylce begins. The process is repeated until convergence is reached, i.e., the energy does not change (default `ETol 1e-8`) or the orbital gradient is near zero (default `GTol 1e-3`). In contrast to that, a *one-step ansatz* updates the orbital coefficients and CI coefficients simultaneously.

In both cases, the unitary transformation is parametrized using an antisymmetric matrix ($X_{pq} = -X_{qp}$)

$$\mathbf{U} = \exp\left(\mathbf{X}\right),$$

where X_{pq} consists of the non-redundant orbital rotations (inter-subspace).

As in the case of single-determinant wave functions (RHF, UHF, RKS, UKS) there are first-order converging methods, such as the present default, the perturbative SuperCI (`SuperCI_PT`),[430] that only require the orbital gradients

$$\frac{\partial E\left(\mathbf{X}, \mathbf{C}\right)}{\partial X_{pq}}.$$

The necesarry transformed integrals $(tu|vx)$, with tuv active and x belonging to any subspace, are a very small and readily held in central storage even for larger calculations.

On the other hand, second-order CASSCF methods, compute the Hessian and require the integrals $(pq|xy)$ and $(px|qy)$ (p, q = internal, active; x, y = any orbital). This is a fairly large set of integrals and their generation is laborious in terms of CPU time and disk storage. Standard second-order methods such as the augmented Hessian (`NR`) are more limited in the size of the molecules which can be well treated. While the `NR` is highly efficient near the minimum, it is less robust when the initial orbital are too far away. The *restricted-step augmented Hessian (TRAH)* mitigates many of the disadvantages and offers a more robust second-order ansatz, which can also tackle larger systems.[431] Here, most of the integrals are processed using the *RI approximation* in an AO direct strategy. For more details, we refer to original publication.

State-averaging

In many situations, it is desirable to optimize the orbitals not for a single state but for the average of several states. In order to see what is done, the energy for state I is re-written as:

$$E_I\left(\mathbf{c}, \mathbf{C}\right) = \sum_{pq} \Gamma_q^{p(I)} h_{pq} + \sum_{pqrs} \Gamma_{qs}^{pr(I)}\left(pq|rs\right)$$

Here, $\Gamma_q^{p(I)}$ and $\Gamma_{qs}^{pr(I)}$ are the one-and two-particle reduced electron density matrices for this state (labels p, q, r, s span the inactive and active subspaces):

$$\Gamma_q^{p(I)} = \left\langle \Psi_I^S \left| E_q^p \right| \Psi_I^S \right\rangle$$

$$\Gamma_{qs}^{pr(I)} = \frac{1}{2} \left\langle \Psi_I^S \left| E_q^p E_s^r - \delta_{qr} E_s^p \right| \Psi_I^S \right\rangle$$

The average energy is simply obtained from averaging the density matrices using arbitrary weights w_I that are user defined but are constrained to sum to unity.

$$\Gamma_q^{p(av)} = \sum_I w_I \Gamma_q^{p(I)}$$

$$\Gamma_{qs}^{pr(av)} = \sum_I w_I \Gamma_{qs}^{pr(I)}$$

$$\sum_I w_I = 1$$

3.13.3 Basic Usage

The most elementary input information which is always required for CASSCF calculations is the specification of the number of active electrons and orbitals.

```
%casscf
        nel  4  # number of active space electrons
        norb 6  # number of active orbitals
end
```

The CASSCF program in ORCA can average states of several multiplicities. The multiplicities are given as a list. For each multiplicity the number of roots should be specified:

```
%casscf mult   1,3 # here: multiplicities singlet and triplet

        nroots 4,2 # four singlets, two triplets
        end
```

If the symmetry handling in ORCA is enabled (`! UseSym`) each multiplicity block must have an irreducible representation assigned. Numbers corresponding to the "irrep" within a given symmetry are printed in the output of ORCA.

```
%casscf mult   1,3 # here: multiplicities singlet and triplet
        irrep  0,1 # here: irrep for each mult. block (mandatory!)
        nroots 4,2 # four singlets, two triplets
```

Several roots and multiplicities usually imply a state average CASSCF (SA-CASSCF) calculation. Note that the program by default chooses equal weights for the multiplicity blocks. Roots within a given block have equal weight. Users can define a custom weighting scheme for the multiplicity blocks and roots:

```
%casscf mult 1,3   # here: multiplicities singlet and triplet
        nroots 4,2  # four singlets, two triplets
        bweight 2,1 # singlets and triplets weighted 2:1
        weights[0] = 0.5,0.2,0.2,0.2 # singlet weights
        weights[1] = 0.7,0.3         # triplet weights
        end
```

The program automatically normalizes these weights such that the sum over all weights is unity. If convergence on an excited state is desired then the `weights[0]` array may look like `0.0,0.0,1.0` (this would optimize the orbitals for the third excited state. If several states cross during the orbital optimization this will ultimately cause convergence problems.

We note passing that the converged orbitals of the state averaged procedure are a compromise for the set of states. ORCA by default only prints the SA-CASSCF gradient norm. State-specific gradients are summarized at the end of the calculation with the keyword `PrintGState`.

```
%casscf
  ...
  printgstate true # optional printing of the state-specific orbital gradients
end
```

Orbital Optimization (2-step approach)

In the following we discuss the available options for the two-step ansatz, where orbital update and the CI problem are solved in sequence. A more powerful and slightly more expensive 1-step ansatz is described in section Section 3.13.3. We remark that a number of convergence problems can already be resolved changing the guess orbitals.

> ⚠️ **Warning**
>
> The following keywords are optional and should only be used facing severe convergence difficulties. Ideally, the default or `!TRAH` should be enough.

Aside from the perturbative SuperCI (`SuperCI_PT` - default),[430] several orbital optimization methods are implemented.

```
# Keywords to be used as Orbstep/Switchstep
SuperCI_PT  # perturbative SuperCI (first-order) - default
TRAH        # restricted-step augmented Hessian (second-order 1-step)
#
SuperCI     # SuperCI (first-order)
DIIS        # DIIS (first-order)
KDIIS       # KDIIS (first-order)
SOSCF       # approx. Newton-Raphson (first-order)
NR          # augmented Hessian Newton-Raphson
            # unfolded two-step procedure
            # - still not true second-order
```

The different convergers have different strengths. First-order method are cheap but typically require more iterations compared to second-order methods. When the gradient is far off from convergence the program uses the converger defined as `orbstep` while close to convergence the `switchstep` is used. The actual criteria for switchstep are defined with the keywords `SwitchConv` and `SwitchIter`.

```
%casscf
  OrbStep    SuperCI # or any other from the list above
  SwitchStep DIIS    # or any other from the list above

  SwitchConv 0.03 # gradient at which to switch
  SwitchIter 15   # iteration at which the switch takes place
                  # irrespective of the gradient

  MaxIter    75   # Maximum number of macro-iterations
end
```

Picking a convergence strategy, the program has to balance speed and robustness. The default strategy uses the `SuperCI_PT` as converger for `orbstep` and `switchstep`.[430] It is robust with respect to orbitals that are exactly doubly occupied or empty. Rotations with orbital close to this critical occupations can further be eliminated with the keyword `DThresh` (default=1e-6). However, the method is quite aggressive in the orbital optimization. In some cases, such as basis set projection or `PATOM` guess (intrinsic basis set projection), the program might pick a step-size that is too big. Then restricting the step-size via the keyword `MaxRot` (default=0.2) might be useful. The keywords `DThresh` and `MaxRot` described below are specific to `SuperCI_PT`. For many users, `MaxRot`