is less palpable than level shifting. Therefore, the present version allows level shifts as well. In contrast to other convergers, level shifts are not needed and highly discouraged. With the exception of GradScaling (vide infra), other damping techniques described further below do not apply to the SuperCI_PT.

```
# SuperCI_PT specific settings
MaxRot 0.05  # cap stepsize
DThresh 1e-6  # thresh for critical occupation
```

In case of convergence problems with the default settings, it is recommended to try the !TRAH option. Alternatively, the combination of orbstep SuperCI and switchstep DIIS in conjuction with a large level shift (2 Eh) often lead to immediate success. The proposed scheme typically requires more iterations. Moreoever, in contrast to the SuperCI(PT), the SuperCI, DIIS and KDIIS can struggle with active orbitals, that have an occupation of exactly 2.0 or 0.0! The KDIIS [6, 7] — based on perturbation theory — is an approximation to the regular DIIS procedure avoiding redundant rotations. Both DIIS schemes avoid linear dependencies in the expansion space.

```
MaxDIIS 15 # max. no of DIIS vectors to keep
DIISThresh 1e-7 # overlap criteria for linear dependency
```

The combination of SuperCI and DIIS (switchstep) is particularly suited to protect the active space composition. Adjusting the level shift will do the job. **Here, level shift is the single most important lever to control convergence**.

```
# default = dynamic level-shifting based on Ext-Act, Int-Act
ShiftUp    2.0  # static up-shift the virtual orbitals
ShiftDn    2.0  # static down-shift the internal orbitals
MinShift    0.6  # minimum separation subspaces
```

Level-shift is particularly important if the active, inactive and virtual orbitals overlap in their orbital energies. The energy separation of the subspaces is printed in the output. Ideally, the entries <code>Ext-Act</code> and <code>Act-Int</code> should be positive and larger than 0.2 Eh. This will help the program to preserve your active space composition throughout the iterations. If no shift is specified in the input, ORCA will choose a level-shift to guarantee an energy separation between the subspaces (MinShift).

```
E(CAS) = -230.590325053 Eh DE = -0.000798832

--- Energy gap subspaces: Ext-Act = -0.244 Act-Int = -0.002

--- current l-shift: Up(Ext-Act) = 0.54 Dn(Act-Int) = 0.30
```

In all of the implemented orbital optimization schemes the step-size correlates with the gradient-norm. A constant damping factor can be set with the keyword GradScaling. Note, damping and level shifting techniques are not recommended for the default converger (SuperCI_PT).

```
GradScaling 0.5 # constant damping in all steps
```

There are situations when the active space has been chosen carefully, but the initial gradient is still far off. To keep the "good" active space, we can suppress all rotation but the inactive-external ones until the gradient-norm is small enough to continue safely. The threshold can be set with <code>FreezelE</code> keyword. Once the components of the gradient in the inactive-external direction have a weight of less than <code>FreezelE</code>, all constraints are lifted. ORCA by default freezes active rotations if the total gradient norm is larger than 1.0 and the active rotations have a weight of less than 5%. The feature can be turned off setting the threshold to zero.

Similarly, rotations of the almost doubly occupied orbitals with the inactive orbitals can be damped using the threshold FreezeActive. Rotations of this type are damped as long as all their weight is smaller than FreezeActive. In contrast to the ShiftDn, it damps just the "troublesome" parts of internal-active rotations. This option applies to all of the orbital optimization schemes but the SuperCI_PT.

```
FreezeIE 0.4  # keep active space until int-ext rotation have # a contribution of less than 40% to the ||g|| FreezeActive 0.03  # keep almost doubly occupied orbitals as long as # their contribution is less than 3% to the ||g||
```

If the calculation starts from a converged Hartree-Fock orbitals, the core orbitals should not change dramatically by the CASSCF optimization. Often trailing convergence is associated to rotations with low lying orbitals. Their

contribution to the total energy is fairly small. With the keyword FreezeGrad these rotations can be omitted from the orbital optimization procedure.

```
FreezeGrad 0.2 # omit hitting a gradient norm ||g|| <0.2
```

The affected orbitals are printed at the startup of CASSCF.

```
FreezeGrad ... enabled if ||g|| is below 0.02

Note Convergence can be signaled if the reduced gradient reaches GTol

Last frozen orbital ... 9

First deleted orbital ... 320

Once rotations with core and deleted orbitals are stabilized they will be damped.
```

By default rotations with frozencore (or deleted virtuals) are not omitted. If the option FreezeGrad is active, the ratio with respect to the total gradient is printed.

```
||g|| = 0.001240414 Max(G) = -0.000431747 Rot=319,1

--- Option=FreezeGrad: ||g|| = 0.001081707

= 87.21%

Omitting frozencore elements
```

The DIIS based algorithms may sometimes converge slowly or "trail" towards the end. This is typically the region, where the augmented Hessian method (NR) and the cheaper quasi-Newton (SOSCF) perform better. Keep in mind that the Newton-Raphson is designed for optimization on a **convex surface** (Hessian is semidefinite). If the NR is switched on too early, there is a good chance that this condition is not fulfilled. In this case the program will complain about negative eigenvalues or diagonal elements of the Hessian as can be seen in the snippet below. The next optimization step is large and unpredictable. It is a wildcard that can get you closer to convergence or immediate divergence of the CASSCF procedure.

```
||g|| = 0.771376945 Max(G) = 0.216712933 Rot=140,53

--- Orbital Update [ NR]

Warning: NEGATIVE diagonal element D(81,53) = -4.733590

Warning: NEGATIVE diagonal element D(82,53) = -4.737955

...
```

For larger system, the augmented Hessian equations are solved iteratively (NR iterations). The augmented Hessian is considered solved when the residual norm, < r | r >, is small enough. Aside from the overall CASSCF convergence, negative eigenvalues affect these NR iterations.

```
--- Orbital Update [
 AugHess Tolerance (auto): Tol= 1.00e-07
 AUGHESS-ITER 0: E= -0.174480747 < r | r>=
                                                   0.558679452
 AUGHESS-ITER 1: E=
                         -0.308672359 < r | r > =
                                                   0.468254671
 AUGHESS-ITER 2: E=
                         -0.434272813 < r | r > =
                                                   0.286305469
 AUGHESS-ITER
                         -0.439149451 < r | r > =
                3: E=
                                                   0.286514628
 AUGHESS-ITER
                         -0.605787445 < r|r>=
                4: E=
                                                   0.191691955
 AUGHESS-ITER
                5: E=
                         -0.607766529 < r|r>=
                                                   0.310450670
 AUGHESS-ITER
                6: E=
                         -0.611674930 < r | r > =
                                                   0.141402593
               7: E=
                         -0.623145299 < r | r > =
 AUGHESS-ITER
                                                   0.394505306
 AUGHESS-ITER 8: E=
                         -0.658410333 < r | r > =
                                                   0.166915094
                         -0.790571374 < r | r > =
 AUGHESS-ITER 9: E=
                                                   4.722929453
 AUGHESS-ITER 10: E=
                         -0.790590554 < r | r > =
                                                   4.716012014
 AugHess: No convergence in the Davidson procedure
```

There are a number of refined NR settings that influence the convergence behavior on a non-convex energy surface. We mention the keywords for completeness and disencourage from changing the default settings. If overall convergence cannot be changed due to negative eigenvalues, it is recommended to delay the NR switchstep (switchconv, switchiter). This will require some trial and error, since the curvature of the surface is *a priori* not know.

```
%casscf
  # NR specific settings:
  aughess
   Solver 0
                           # Davidson (default)
            1
                          # Pople (pure NR steps)
                           # DIIS
                       # max. no. of CI iters.
# Davidson expansion space
   MaxIter 35
   MaxDim 35
   MaxDIIS 12
                           # max. number of DIIS vectors
    UseSubMatrixGuess true # diag a submatrix of the Hessian
                           # as an initial guess
   NGuessMat 512
                           # size of initial guess matrix (part of
                            # the Hessian exactly diagonalized)
   ExactDiagSwitch 512
                           # up to this dimension the Hessian
                           # is exactly diagonalized (small problems)
   PrintLevel 1
Tol 1e-6
Compress true
DiagShift 0.0
                           # amount of output during AH iterations
                           # convergence tolerance
                           # use compressed storage
                           # shift of the diagonal elements of the
                           # Hessian
   UseDiagPrec true
SecShift 1e-4
                          # use the diagonal in updating
                           # shift the higher roots in the Davidson
                           # secular equations
    UpdateShift 0.5
                           # shift of the denominator in the
                            # update of the AH coefficients
  end
end
```

1 Note

- Let us stress again: it is strongly recommended to first LOOK at your orbitals and make sure that the ones that will enter the active space are really the ones that you want to be in the active space! Many problems can be solved by thinking about the desired physical contents of the reference space before starting a CASSCF. A poor choice of orbitals results in poor convergence or poor accuracy of the results! Choosing the active orbitals always requires chemical and physical insight into the molecules that you are studying!
- Please try the program with default settings before playing with the more advanced options. If you encounter convergence problems, have a look into your output, read the warning and see how the gradient and energy evolves. Try !TRAH. Increasing MaxIter will not help in many cases.
- Be careful with keywords such as !tightscf, !verytightscf and so on. These keywords set higher integral thresholds, which is a good idea, but also tighten the CASSCF convergence thresholds. If you do not need a tighter energy convergence, reset the criteria in the casscf block using ETol. For many applications an energy convergence beyond 10^{-8} is unnecessary.

Incremental Fock

In general, convergence is strongly influenced by numerical noise, especially in the final iterations. One source of numerical noise is the incremental Fock build. Thus, it can help to enforce more frequent full Fock matrix formation.

```
ResetFreq 1 # reset frequency for direct SCF
```

If the orbital change in the active space is small, the active Fock matrix in ORCA is approximated using the density matrix from the previous cycle saving a second Fock matrix build. However, this approximation might also be a source of numerical instability. The threshold "SwitchDens" can be set to zero to enable the exact build. The program default starts with a rather large value (1e-2) and will reduce this parameter automatically when necessary.

```
switchdens 0.0001 # ~gtol * 0.1
```

Monitoring the active space

During the iterations, the active orbitals are chosen to maximize the overlap with active orbitals from the previous iterations. Maximizing the overlap does not make any restrictions on the nature of the orbitals. Thus initially localized orbitals will stay localized and ordered, which is sometimes a desired feature, e.g., in the density matrix renormalization group approach (DMRG). This feature is set with the keyword ActConstraints and is enabled by default (after the first 3 macro-iterations). For some orbital optimization procedures, such as the SuperCI, natural orbitals are more advantageous. Therefore, the ActConstraints can be turned off in favor of natural orbital construction (see below). If the keyword is not set by the user, ORCA picks the best choice for the given orbital optimization step.

```
ActConstraints 0 # no checks and no changes
1 # maximize overlap of active orbitals and check sanity (default...)

for DIIS)
2 # make natural orbitals in every iteration (default SuperCI)
3 # make canonical orbitals in every iteration
4 # localize orbitals
```

In addition to maximizing the overlap, "ActConstraints 1" checks if the composition of the active space has changed, i.e., an orbital has been rotated out of the active space. In this case, ORCA aborts and stores the last valid set of orbitals. Below is an example error message.

```
--- Orbital Update [ DIIS]
--- Failed to constrain active orbitals due to rotations:
Rot(37, 35) with OVL=0.960986
Rot(38, 34) with OVL=0.842114
Rot(43,104) with OVL=0.031938
```

In the snippet above, the active space ranges from 37-43. The program reports that orbitals 37,38 and 43 have changed their character. The overlap of orbital 43 (active) with the previous set of active orbitals is just 3% and the program aborts. There are a number of reasons, why this happens in the calculation. If this error occurs constantly with the same orbitals, it is worthwhile to inspect the rotating partner orbitals (visualize). It might be sign that the active space is not balanced and should be extended. In many instances changing the level-shift or lowering switchconv is sufficient to protect the active space. In some cases, turning off the sanity check ("ActConstraints 0") and re-rotating orbitals will bring CASSCF closer to convergence. Some problems can be avoided by a better design of the calculation. The CASSCF tutorial elaborates on the subject in more detail.

There are situations such as parameter scans, where "actconstraints" is counter-productive and should be disabled. In other words, we want to allow changes in the active space composition. As an example, consider the rotation of ethylene around its double-bond represented by a CAS(2,2). Although the active space consists of the bonding and anti-bonding orbitals π -orbitals, their composition in terms of atomic orbitals changes from the eclipsed to the staggered conformation. Depending on the actual input settings (orbstep and number of scan points), this might trigger an abort.

Orbital Optimization (1-step approach): Robust Convergence with TRAH-CASSCF

The restricted-step second-order converger TRAH is now also available for both state-specific and state-averaged CASSCF calculations.[121] To activate TRAH for your CASSCF calculation, you just need to add! TRAH in one of the simple input lines and add an auxiliary basis.

```
# Auxiliary basis of type /C needed for !TRAH
! TRAH Def2-SVP Def2-SVP/C TightSCF
%casscf
  # define CAS(6,6)
 nel 6
 norb 6
  # two lowest singlets states
 nroots 2
end
*xyz 0 1
      0.0
            0.0
                  0.0
 Ν
      0.0
            0.0
                  1.1
 Ν
end
```

In most cases, there is no need to play with any input parameters. The only exception is the choice of active molecular orbital representations that can have a significant impact on the convergence rate for spin-coupled systems. As can be seen from Fig. Fig. 3.13, for such calculations localized active orbitals perform best. In any other case, the natural orbitals (default) should be employed.

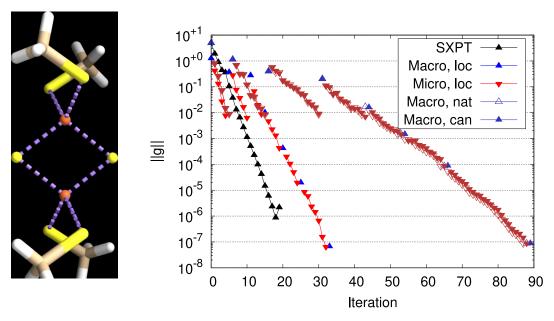


Fig. 3.13: SXPT and TRAH error convergence using different choices for the active-orbital basis.

Possible input options for the active-orbital basis are

```
%casscf
ActConstraints Natural # default
end
```

Active Orbitals	Meaning
NatOrbs	Keeps the one-electron density matrix (1-RDM) diagonal. Default
LocOrbs	A Foster-Boys localization of active MOs is performed in every macro iteration.
	This is recommended for spin-coupled systems .
Unchanged	The active MO basis is not changed. Primarily debug option.
CanonOrbs	Keeps the total active-MO Fock matrix diagonal. Experimental option.

Note that, in contrast to the SCF program, there is **no** AutoTRAH feature for CASSCF yet. The TRAH feature has to be requested explicitly in the input.

Using the RI Approximation

A significant speedup of CASSCF calculations on larger molecules can be achieved with the RI, RI-JK and RIJ-COSX approximations. [430] There are two independent integral generation and transformation steps in a CASSCF procedure. In addition to the usual Fock matrix construction, that is central to HF and DFT approaches, more integrals appear in the construction of the orbital gradient and Hessian. The latter are approximated using the keyword trafostep RI, where an auxiliary basis (/C or the more accurate /JK auxiliary basis) is required. Note that auxiliary basis sets of the type /J are not sufficient to fit these integrals. If no suitable auxiliary basis set is available, the AutoAux feature might be useful.[59] We note passing, that there are in principle three distinguished auxiliary basis slots, that can be individually assigned in the %basis block (section Section 2.7). As an example, we recompute the benzene ground state example from section Section 3.13.6 with a CAS(6,6).

```
! SV(P) def2-svp/C
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"
# Commented out: Detailed settings of the auxiliary basis in the %basis block,
                where the AuxC slot is relevant for the option TrafoStep RI.
# %basis
# auxC "def2-svp/C"
# end
%casscf
 # define CAS(6,6)
 nel
       6
 norb
 # singlet ground state
 mult.
 nroots 1
 trafostep ri # enables the RI approximation
```

The energy of this calculation is -230.590328~ Eh compared to the previous result -230.590271~ Eh. Thus, the RI error is only 0.06 mEh which is certainly negligible for all intents and purposes. With the larger /JK auxiliary basis the error is typically much smaller (0.02 mEh in this example). Even if more accurate results are necessary, it is a good idea to pre-converge the CASSCF with RI. The resulting orbitals should be a much better guess for the subsequent calculation without RI, and thus save computation time.

The TrafoStep RI only affects the integral transformation in CASSCF calculations while the Fock operators are still calculated in the standard way using four index integrals. In order to fully avoid any four-index integral evaluation, you can significantly speed up the time needed in each iteration by specifying <code>!RIJCOSX</code>. The keyword implies <code>TrafoStep RI</code>. The COSX approximation is used for the construction of the Fock matrices. In this case, an additional auxiliary basis (/J auxiliary basis) is mandatory.

```
! SV(P) def2-svp/C RIJCOSX def2/J
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"
# Commented out: Detailed settings of the auxiliary basis in the %basis block,
                 where the AuxJ and AuxC slot are mandatory.
# %basis
# auxJ "def2/J"
# auxC "def2-svp/C"
# end
%casscf
 # define CAS(6,6)
 nel 6
 norb
       6
  # singlet ground state
 mult.
       1
 nroots 1
end
```

The speedup and accuracy is similar to what is observed in RHF and UHF calculations. In this example the RIJCOSX leads to an error of 1 mEh. The methodology performs better for the computation of energy differences, where it profits from error cancellation. The RIJCOSX is ideally suited to converge large-scale systems. Note that for large calculations the integral cut-offs and numerical grids should be tightened. See section Section 2.8.4 for details. With a floppy numerical grid setting the accuracy as well as the convergence behavior of CASSCF deteriorate. The RIJK approximation offers an alternative ansatz. The latter is set with <code>!RIJK</code> and can also be run in conventional mode (conv) for additional speed-up. With conv, a **single auxiliary basis must be provided** that is sufficiently larger to approximate the Fock matrices as well the gradient/Hessian integrals. In direct mode an additional auxiliary basis set can be set for the AuxC slot.

```
! SV(P) RIJK def2/JK

# Commented out: Detailed settings of the auxiliary basis in the %basis block,

# where only the auxJK slot must be set.

# %basis

# auxJK "def2/JK" # or "AutoAux"

# end
```

The RIJK methodology is more accurate and robust for CASSCF, e.g., here the error is just 0.5 mEH.

Final Orbitals: Canonicalization Choices

Once the calculation has converged, ORCA will do a final macro-iteration, where the orbital are "finalized". For complete active spaces (CAS), these transformations do not alter the final energy and wave function. Note, that solutions from approximate CAS-CI solvers such as the ICE approach or the DMRG ansatz are affected by the final orbital transformation. The magnitude depends on the truncation level (e.g. TGen, TVar and MaxM) of the approximated wave function. The default final orbitals are canonical in the internal and external space with respect to state-averaged Fock operator. The active orbitals are chosen as natural orbitals. Other orbital choices are equally valid and can be selected for the individual subspaces.

```
#internal space
IntOrbs CanonOrbs # canonical
   LocOrbs # localized
   unchanged # no changes

# partner orbitals for the active space based
   # on various concepts
   PMOS # based on orthogonalization tails.
```

```
OSZ
                  # based on oszillator orbital
        DOI
                  # based on differential overlap
#external space
ExtOrbs CanonOrbs # canonical
       LocOrbs # localized
        unchanged # no changes
        # partner orbitals for the active space based
        # on various concepts
              # based on orthogonalization tails.
        OSZ
                  # based on oszillator orbital
                  # based on differential overlap
        DoubleShell # based on the shell and angular momentum
                    # of the highest active orbitals, the first few
                    # virtual orbitals correspond to the doubled-shell.
                    # All other virt. orbitals are canonicalized.
                    # For 3d-metal complexes, these are the 4d orbitals!
                    # For 4d-metal complexes, these are the 5d orbitals!
                    # And so on...
#active space
ActOrbs NatOrbs # natural
       CanonOrbs # canonical
       LocOrbs # localized
        unchanged # no changes
        dOrbs # purify metal d-orbital and call the AILFT
        f0rbs
                  \ensuremath{\text{\#}}\xspace purify metal f-orbital and call the AILFT
        SDO
                  # Single Determinant Orbitals: this is only possible if the
                  # active space has a single hole or a single electron.
                  # SDOs are then the unique choice of orbitals that simultaneously
                  # turns each CASSCF root into a single determinant.
```

SDOs are specific for the active orbital space.[432] The set of options (PMOS, OSZ, DOI, DoubelShell) are specific for the inactive and external space. They aim to assist the extension of the current active space. All four options, re-design the first NOrb (number of active orbitals) next to the active space, while the remaining orbitals of the same subspace are canonical. The re-designed orbital are based on different concepts.

- PMOS generates the bonding / anti-bonding partner orbitals for the chosen active space. It is based on the orthogonalization tail of the active orbitals.
- OSZ generates a single orbital for each active orbital, that maximizes the dipole-dipole interaction.
- DOI follows the same principle as OSZ, but the differential overlap is maximized instead.
- DoubleShell is specific to the external space. The highest active MO or DoubleShellMO is analyzed. A set of orbitals with the same angular momentum but larger radial distribution is generated.

Optionally, the four options above can be supplemented with a reference MO using the keyword RefMO/DoubleShellMO. The presence of RefMO/DoubleShellMO changes the default behavior. In case of PMOS, OSZ and DOI, all orbitals of the given subspace are chosen to maximize a single objective function with respect to the reference MO (must be active). This contrasts the default settings, where for each active MO an objective function is maximized and a single "best" orbital is picked. In other words, in the default setting, each active orbital has a corresponding "best" orbital in the selected subspace neighboring the active space.

```
RefMO 17  # MO with number 17 (default =-1)
DoubleShellMO 17 # MO with number 17 (default =-1)
```

The aforementioned options are aids and the resulting orbitals should be inspected prior extension of the active space. In particular the PMOS option is useful in the context of transition metal complexes to find suitable Ligand based orbitals. There are more options (dorbs, forbs, DoubleShell), that are specifically designed for coordination chemistry. A more detailed description is found in the CASSCF tutorial that supplements the manual.

If the active space consists of a single set of metal d-orbitals, natural orbitals may be a mixture of the d-orbitals.

The active orbitals are remixed to obtain "pure" d-orbitals (ligand field orbitals) if the actorbs is set to dorbs. The same holds for f-orbitals and the option forbs. Furthermore, the keyword dorbs automatically triggers the *ab initio* ligand field analysis (AILFT).[433, 434]The approach has been reported in a number of applications.[435, 436, 437, 438, 439, 440, 441, 442] Note that the actual representation depends on the chosen axis frame. It is thus recommended to align the molecule properly. For more details on the AILFT approach, we refer to the AILFT section Section 3.13.16, the original paper and the CASSCF tutorial, where examples are shown.

CI Solvers (CSFCI, ACCCI, ..., ICE)

The CI-step default setting is CSF based and is done in the present program by generating a partial "formula tape" which is read in each CI iteration. The tape may become quite large beyond several hundred thousand CSFs which limits the applicability of the CASSCF module. The accelerated CI (ACCCI) has the same limitations, but uses a slightly different algorithm that handles multi-root calculations much more efficiently.

Larger active spaces are tractable with the *DMRG* approach or *the iterative configuration expansion (ICE)* developed in our own group.[443, 444] DMRG and ICE return approximate full CI results. The maximum size of the active space depends on the system and the required accuracy. Active spaces of 10–40 orbitals should be feasible with both approaches. The CASSCF tutorial covers examples with ACCCI and ICE as CI solvers.

```
%casscf
CIStep CSFCI # CSF based CI (default)
ACCCI # CSF based CI solver with faster algorithm for multi-root.
→calculations
ICE # CSF based approximate CI -> ICE/CIPSI algorithm
DMRGCI # density matrix renormalization group approach instead of the CI end
```

In the ICE approach, the computation of the coupling coefficients is time-consuming and by default repeated in every macro-iteration. To avoid the reconstruction, it is recommended to once generate a coupling coefficient library (cclib) and to use it for all of your ICE calculations. The details of the methodology and the cclib are described in the ICE section Section 3.14.

Detailed settings for the conventional CI solvers (CSFCI, ACCCI, ICE) can be controlled in a sub-block.

```
%casscf
 # CI specific settings
 Сi
   MaxIter 64
                # max. no. of CI iters.
   MaxDim 10 # Davidson expansion space = MaxDim * NRoots
   NGuessMat 512 # Initial guess matrix: 512x512
   PrintLevel 3 # amount of output during CI iterations
   ETol 1e-10 # default 0.1*ETol in CASSCF
   RTol 1e-10 # default 0.1*ETol in CASSCF
   TGen 1e-4 # ICE generator thresh
   TVar 1e-11 # ICE selection thresh, default = TGen*1e-7
   TPrint 1e-3 # Thresh to print the CI Coefficients / CI weights
   PrintWF 0  # (default) prints only the CFGs
                # Printing of the wave function in the basis of CSFs
          det # Printing of the wave function in the basis of Determinants
   # ICE/CIPSI specific settings
   # See ICE section of the manual:
   CIPSIType 0 # 0: CFG based (default)
                  # 1: CSF based
           -1e.7 # variational threshold. negative means 1e-7*TGen
           1e-4 # generator threshold
   TGen
```

```
end
end
```

The CI-step DMRGCI interfaces to the **BLOCK** program developed in the group of G. K.-L. Chan [445, 446, 447, 448]. A detailed description of the **BLOCK** program, its input parameters, general information and examples on the density matrix renormalization group (DMRG) approach, are available in the section *Density Matrix Renormalization Group (DMRG)* of the manual.

The implementation of DMRG in **BLOCK** is fully spin-adapted. However, spin-densities and related properties are not available in the current version of the **BLOCK** code. To start a DMRG calculation add the keyword "CIStep DMRGCI" into a regular CASSCF input. ORCA will set default parameters and generate and input for the **BLOCK** program. In general, DMRG is not invariant to rotation in the active space. The program by **default** will run an automatic ordering procedure (Fiedler). More and refined options can be set in the dmrg sub-block of CASSCF—see section Section 3.15 for a complete list of keywords.

```
%casscf
nel 8
norb 6
mult 3
CIStep DMRGCI

# Detailed settings
dmrg
# more/refined options
...
end
end
```

It is highly recommended to start the calculation with split-localized orbitals. Any set of starting orbitals (gbw file) can be localized using the <code>orca_loc</code> program. Typing <code>orca_loc</code> in the shell will return a small help-file with details on how to setup an input for the localization. Examples for DMRG including the localization are in the corresponding section of the manual *Density Matrix Renormalization Group (DMRG)*. The utility program <code>orca_loc</code> is documented in section Section 9.2.5. Split-localization refers to an independent localization of the internal and virtual part of the desired active orbitals.

Model Spaces: CAS, RAS and ORMAS

In the MCSCF approach, the CI- and orbital coefficients are optimized. The complete active space (CAS) is a specific models space, where the CI consists of all possible CSFs, that have N active electron in M active orbitals (i.e. CAS(N,M)). This is naturally the default for the <code>%casscf</code> module. However, the module is general enough to define arbitrary CSFs using their configurations (CFGs). The latter refer to the orbital occupation of the CSFs.

Warning

The orbital optimization is carried out omitting the active-active rotation, which might be non-negligible. Hence, the energy is sensitive to canonicalization choices (see actorbs,actconstraints).

Arbitrary References

With the refs sub-block, arbitrary CFGs can be selected to construct the CI space. All CSFs within a given CFG are accounted for. This is for example useful in the *context of X-ray spectroscopy*.

As simple example, lets take the benzene molecule from section Section 3.13.6 and restrict the CI space to the RHF determinant

```
! SV(P)
! moread
%moinp "Test-CASSCF-Benzene-2.mrci.nat"
 \# define arbitrary CI space using the CFG input mask
 refs
    \{2\ 2\ 2\ 0\ 0\ 0\}\ \#\ CFG\ with the lowest orbitals being doubly occupied.
                   # Arbitrary CFGs can be added in this block.
                   # The number of electrons and orbitals
                   # must match the active space defined below.
  end
  # define active space
 nel
         6
 norb
         6
  # define singlet ground state
 mult 1
 nroots 1
end
```

Restricted Active Space (RAS)

In the RAS model space, the active space is further partitioned into RAS1, RAS2 and RAS3. The input RAS (Nel: R1 H/R2/R3 P) requires 6 parameters: 'Nel' refers to the total number of electrons in the system. 'R1/R2/R3' define the number orbitals in RAS1/RAS2/RAS3. 'H' defines the number of holes in the RAS1 CFGs, whereas 'P' the number of particles in the RAS3. With the restrictions above, all possible CFGs are constructed. RAS is particularly useful to define *core-excited states*.

```
%casscf
# define the active space (mandatory)
nel 11
norb 8
# define the actual RAS partitioning
refs
   RAS(11:3 1/5/0 0) # (Nel: NRAS1 MaxHoles / NRAS2 / NRAS3 MaxParticles)
end
...
end
```

Occupation Restricted Multiple Active Spaces (ORMAS)

A more general partitioning is possible with the ORMAS model space, which consists of multiple active spaces (up to 25). Each active space is defined with the number of orbitals('m') and the number of electrons ranging from 'min' to 'max'. With the input mask ORMAS (nel: m1 min1 max1, m2 min2 max2,...), the active spaces are defined such that a total of 'nel' electrons are distributed among all the sub-space. The first sub-space (m1, min1, max1) consists of 'm1' orbitals with electrons from 'min1' to 'max1'. The subsequent parameters define the next sub-space (m2,min2,max2) and so on. Note that the standard CASSCF parameters nel and norb are overwritten with the parameters from the ORMAS input mask.

The RASCI and ORMAS Module

ORCA features a rather flexible module that is designed to handle rather flexible general CI expansions. This is not meant to be in competition with the individually selecting and uncontracted MRCI or the internally contracted AUTOCI program or the iteratively refined ICE program. Rather, this is an attempt to create a simple and flexible module that allows expert users to play with various general CI expansions using moderate orbital spaces. This is not a module that can be used on large molecules or systems with hundreds let alone thousands of orbitals. One can use it for larger molecules only in a setting where there is a limited orbital window.

Importantly, the module can accept individual configurations or CSFs as references, RAS reference and also ORMAS references as detailed below. In addition to these flexible reference space, the program can also perform excitations of various degrees on these references. The module is connect with the QDPT driver for the calculation of magnetic properties and the OPA driver for the calculation of optical spectra.

```
! RHF cc-PVDZ VeryTightSCF PModel
# AVAS guess for sigma/sigma* active space
%scf
 avas
 svstem
   shell 2, 2
   1
          1, 1
          0, 0
   m 1
   center 0, 1
  end
 end
end
# casscf single bond correlation
%casscf nel
             2
             2
        norb
        irrep 0
        mult
               1
        nroots 1
        cistep 0
        trafostep exact
```

```
end
%rasci refs RAS (14: 6 2 / 2 / 18 2)
    # this is MRCISD
    # MRCISDT would be RAS(14: 6 3 / 2 / 18 3)
    # no of corr. el.: RAS1 norb nholes/ RAS2 nel/RAS3 norb nel)
    #ORMAS(14: 6 10 12/ 2 0 2/ 18 0 2)
    #would be identical. ORMAS can be used with up to 25
    # subspaces all given as norb nelmin nelmax
    # Indiviual CFGS { 2 2 2 2 2 2 1 1 0 } would be possible
  # irrep lists can be used as in CASSCF
          0
  # irrep
  mult 1,3
                 # multiplicity lists as in CASSCF
                # number of root lists as in CASSCF
  nroots 5,5
  cistep accci # accelerated CI as in CASSCF
          #csfci : CI in CSFs as in CASSCF
          #detci : CI in determinants as in ICE
          #treecsf: CI in CSFs as in ICE
  ExcLevel 0
                           # excitation level on top of refs
  RejectInvalidRefs false # reject references of invalid spin and
                           # space symmetry before excitations
                           # as in MRCI
                           # optical spectra (UV,CD,MCD)
  douv true
  rel dosoc true end
                          # Magnetic properties
  end
* xyz 0 1
F 0 0 0
F 0 0 1.457513043738
```

Relevant Papers:

Ivanic, Joseph. Direct configuration interaction and multiconfigurational self-consistent-field method for multiple active spaces with variable occupations. I. Method. *The Journal of Chemical Physics*, 2003, 119 (18), 9364–9376. arXiv:https://pubs.aip.org/aip/jcp/article-pdf/119/18/9364/19131699/9364_1_online.pdf, DOI: 10.1063/1.1615954.

3.13.4 Keywords

Simple input keywords for CASSCF are given in Table 3.47. All of them require the CASSCF block as minimal input, the options for which are given below.

	Tuble 3.17. Simple input key words for Crisber
Keyword	Description
DMRG	Sets DMRG as "CIStep" in CASSCF
NEVPT2	SC NEVPT2
SC-NEVPT2	SC-NEVPT2 same as NEVPT2
RI-NEVPT2	SC-NEVPT2 with the RI approximation
FIC-NEVPT2	FIC-NEVPT2 aka PC-NEVPT2
DLPNO-NEVPT2	FIC-NEVPT2 in the framework of DLPNO
CASPT2	FIC-CASPT2
RI-CASPT2	FIC-CASPT2 with the RI approximation
DCD-CAS(2)	2nd order Dynamic Correlation Dressed CAS
RI-DCD-CAS(2)	2nd order Dynamic Correlation Dressed CAS with RI approximation

Table 3.47: Simple input keywords for CASSCF

▲ Warning

Please always try the default convergence settings before tweaking the convergence manually!

```
%casscf
 PrintLevel 3 # amount of printing
 # definition of active space
 # (mandatory)
 nel 6 # number of active space electrons
 norb 6 # number of active orbitals
 # incomplete model spaces
 # (optional)
 # define arbitrary CI space using the CFG input mask
   # arbitrary CFGs.
   {2 2 2 0 0 0}
   # RAS(Nel: NRAS1 MaxHoles / NRAS2 / NRAS3 MaxParticle)
   RAS(6:2 1/2/2 1)
   #ORMAS partitioning (overwritting nel/norb)
   ORMAS(10: 3 4 6 / 2 0 4/ 10 0 2) \# 10 electron distributed to:
                                    \# first space = 3 orbitals with 4-6 electrons
                                    \# second space = 2 orbitals with 0-4 electrons
                                    # third space = 10 orbitals with 0-2_
⇔electrons
 # definition of states
 # (mandatory)
 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
 # State-Averaging
 # (optional)
 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
 printgstate true # optional printing of the state-specific orbital gradients
 # final orbital option (canonicalization)
 # (optional)
 #internal space
 IntOrbs CanonOrbs # canonical - default
         LocOrbs # localized
         unchanged # no changes
         # partner orbitals for the active space based
         # on various concepts
               # based on orthogonalization tails.
```

```
# based on oszillator orbital
       OSZ
       DOI
               # based on differential overlap
#external space
ExtOrbs CanonOrbs # canonical - default
       LocOrbs # localized
       unchanged # no changes
       # partner orbitals for the active space based
       # on various concepts
             # based on orthogonalization tails.
       PMOS
       OSZ
                # based on oszillator orbital
       DOI
                # based on differential overlap
       DoubleShell # based on the shell and angular momentum
                   # of the highest active orbitals, the first few
                   # virtual orbitals correspond to the doubled-shell.
                   # All other virt. orbitals are canonicalized.
                   # For 3d-metal complexes, these are the 4d orbitals!
                   # For 4d-metal complexes, these are the 5d orbitals!
                   # And so on...
#active space
ActOrbs NatOrbs # natural - default
       CanonOrbs # canonical
       LocOrbs # localized
       unchanged # no changes
       # purify metal f-orbital and call the AILFT
       f0rbs
       LMORBS # calls extended AI-LFT
                # Single Determinant Orbitals: this is only possible if the
                 # active space has a single hole or a single electron.
                 # SDOs are then the unique choice of orbitals that
                 # simultaneously turns each CASSCF root into a
                 # single determinant.
# PMO reference orbitals
RefMO 17 # MO
               # MO with number 17 (default =-1)
# DoubleShell reference orbital
DoubleShellMO 17 \# MO with number 17 (default =-1)
# NTO / NDO printing
        #Generate Natural Transition Orbitals
NTOStates 1,2,3 #States to consider for NTO analysis. If empty, all are done
NTOThresh 1e-4 #Threshold for printing occupation numbers
DoNDO #Generate Natural Difference Density Orbitals
NDOStates 1,2,3 #States to consider for NDO analysis. If empty, all are done
# RI approximation of the gradient/ Hessian integrals
# (optional)
TrafoStep RI # enabled RI approximation with /C aux basis
           _____
# FOCK options
ResetFreq 15  # reset frequency for direct SCF
switchdens 0.0001 # approximate active Fock when density is considered unchanged
# CI settings
```

```
_____
CIStep CSFCI # CSF based CI (default)
      ACCCI # CSF based CI solver, faster algorithm for multi-root calculations
      ICE  # CSF based approximate CI -> ICE/CIPSI algorithm
      DMRGCI # density matrix renormalization group approach instead of the CI
# CI specific settings
             # max. no. of CI iters.
 MaxIter 64
 MaxDim 10
                # Davidson expansion space = MaxDim * NRoots
  NGuessMat 512 # Initial guess matrix: 512x512
               # amount of output during CI iterations
# default 0.1*ETol in CASSCF
  PrintLevel 3
  ETol 1e-10
  RTol 1e-10 # default 0.1*ETol in CASSCF
  TGen 1e-4
               # ICE generator thresh
  TVar 1e-11 # ICE selection thresh, default = TGen*1e-7
 TPrint 1e-3 # Thresh to print the CI Coefficients / CI weights
 PrintWF 0 # (default) prints only the CFGs
               # Printing of the wave function in the basis of CSFs
               # Printing of the wave function in the basis of Determinants
  # ICE/CIPSI specific settings
  # See ICE section of the manual:
  ICEType CFGs # CFG based (default)
          CSFs # CSF based
          DETs # determinant based
         -1e.7 # variational threshold. negative means 1e-7*TGen
  TVar
 TGen
          1e-4 # generator threshold
# Numerical gradient state-selection
# -----
imult 0 # multiplicity block counting from zero
iroot 0 # root for given imult counting from zero
# orbital optimization
# (optional)
GTol 1e-3 # convergence criteria for ||g||
ETol 1e-8 # convergence criteria for energy
# The following options allowed for 'orbstep'/'switchstep'
# SuperCI_PT - perturbative SuperCI (first-order) - default 2-step
# 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)
            - augmented Hessian Newton-Raphson
            - unfolded two-step procedure (not true second order)
                    # default
orbstep SuperCI_PT
switchstep SuperCI_PT # default after ||g|| < 'SwitchConv'</pre>
SwitchConv 0.03 # gradient norm ||g|| at which to switch
SwitchIter 15  # iteration at which the switch takes place
               # irrespective of the gradient
MaxIter
         75 # Maximum number of macro-iterations
```

```
# level shifts / damping etc
 ShiftUp 2.0 # static up-shift the virtual orbitals
            2.0 # static down-shift the internal orbitals
 MinShift
            0.6 # minimum separation subspaces
 # monitoring the active space
 ActConstraints 0 # no checks and no changes
        1 # maximize overlap of active orbitals and check sanity (default for_
→DIIS)
         2 # make natural orbitals in every iteration (default SuperCI)
         3 # make canonical orbitals in every iteration
         4 # localize orbitals
 # 'SuperCI_PT' specific settings
 MaxRot 0.05 # cap stepsize
 DThresh 1e-6 # thresh for critical occupation
 # 'TRAH' specific settings
 # see 'TRAH' section of manual.
 # DIIS/KDIIS/SuperCI_PT specific
         15 # max. no of DIIS vectors to keep
 DIISThresh 1e-7 # overlap criteria for linear dependency
 # 'NR' specific settings:
 aughess
                     # Davidson (default)
   Solver
                         # Pople (pure NR steps)
                         # DIIS
                        # max. no. of CI iters.
   MaxIter 35
   MaxDim 35
                         # Davidson expansion space
   MaxDIIS 12
                        # max. number of DIIS vectors
   UseSubMatrixGuess\ true\ \#\ diag\ a\ submatrix\ of\ the\ Hessian
                         # as an initial guess
   NGuessMat 512
                         # size of initial guess matrix (part of
                         # the Hessian exactly diagonalized)
   ExactDiagSwitch 512
                        # up to this dimension the Hessian
                         # is exactly diagonalized (small problems)
  PrintLevel 1
                        # amount of output during AH iterations
                        # convergence tolerance
                         # use compressed storage
   Compress true
   DiagShift 0.0
                         # shift of the diagonal elements of the
                         # Hessian
   UseDiagPrec true
                         # use the diagonal in updating
   SecShift 1e-4
                         # shift the higher roots in the Davidson
                         # secular equations
                         # shift of the denominator in the
  UpdateShift 0.5
                         # update of the AH coefficients
 end
 # more dampng options restricted to a few selected 'orbstep'/'switchstep'
 FreezeIE 0.4  # keep active space until int-ext rotation have
                    # a contribution of less than 40% to the ||g||
```

3.13.5 A simple Example: Be Ground State

One standard example of a multireference system is the Be atom. Let us run two calculations, a standard closed-shell calculation $(1s^22s^2)$ and a CASSCF(2,4) calculation which also includes the $(1s^22s^12p^1)$ and $(1s^22s^02p^2)$ configurations.

```
! TZVPP TightSCF
* xyz 0 1
Be 0 0 0
*
```

This standard closed-shell calculation yields the energy -14.56213241 Eh. The CASSCF calculation

```
! TZVPP TightSCF
%casscf
    # defining CAS(2,4)
    nel 2
    norb 4

    # ground state singlet:
    mult 1 # define the multiplicity.
    nroots 1 # define the number of roots for the mult=1
end

* xyz 0 1
Be 0 0 0
*
```

yields the energy -14.605381525 Eh. Thus, the inclusion of the 2p shell results in an energy lowering of 43 mEh which is considerable. The CASSCF program also prints the composition of the wave function:

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

ROOT 0: E= -14.6053815294 Eh
0.90060 [ 0]: 2000
0.03313 [ 4]: 0200
0.03313 [ 9]: 0002
0.03313 [ 7]: 0020
```

This information is to be read as follows: The lowest state is composed of 90% of the configuration which has the active space occupation pattern 2000 which means that the first active orbital is doubly occupied in this configuration while the other three are empty. The MO vector composition tells us what these orbitals are (ORCA uses natural orbitals to canonicalize the active space).

	0	1	2	3	4	5
	-4.70502	-0.27270	0.11579	0.11579	0.11579	0.16796
	2.00000	1.80121	0.06626	0.06626	0.06626	0.00000
0 Be s	100.0	100.0	0.0	0.0	0.0	100.0
0 Be pz	0.0	0.0	13.6	6.1	80.4	0.0
0 Be px	0.0	0.0	1.5	93.8	4.6	0.0
0 Be py	0.0	0.0	84.9	0.1	15.0	0.0

Thus, the first active space orbital has occupation number 1.80121 and is the Be-2s orbital. The other three orbitals are 2p in character and all have the same occupation number 0.06626. Since they are degenerate in occupation number space, they are arbitrary mixtures of the three 2p orbitals. It is then clear that the other components of the wave function (each with 3.31%) are those in which one of the 2p orbitals is doubly occupied.

How did we know how to put the 2s and 2p orbitals in the active space? The answer is – WE DID NOT KNOW! In this case it was "good luck" that the initial guess produced the orbitals in such an order that we had the 2s and 2p orbitals active. IN GENERAL IT IS YOUR RESPONSIBILITY THAT THE ORBITALS ARE ORDERED SUCH THAT THE ORBITALS THAT YOU WANT IN THE ACTIVE SPACE COME IN THE DESIRED ORDER. In many cases this will require re-ordering and CAREFUL INSPECTION of the starting orbitals.

A Attention

If you include orbitals in the active space that are nearly empty or nearly doubly occupied, convergence problems are likely. The SuperCI(PT) [430] and Newton-Raphson method are less prone to these problems.

3.13.6 Guess: Natural Orbitals Example

The initial guess and selection of the active orbitals is crucial for the success of any CASSCF calculation. Often, the default <code>!PModel</code> guess does not provide good starting orbitals or the correct active space. Which orbitals enter as active, depends on the system and requires the users insight. The ideal choice might vary with the molecules and calculation design. For instance, in the case of lanthanides/actinides, the fragment guess, is particularly successful and leads to a rapid convergence. A number of typical examples, including the fragment guess, can be found in the CASSCF tutorial. If the nature of the active space is a priori known, the atomic valence space selection (AVAS) offers a user-friendly guess alternative.

Irrespective of the origin of the initial starting orbitals, it is always recommended to inspect the orbitals prior the orbital optimization.



In many cases, natural orbitals of a simple correlated calculation of some kind provide an idea of which orbitals to include in a CASSCF calculation.

Let us illustrate this principle with a calculation on the Benzene molecule where we want to include all six π -orbitals in the active space. After doing a RHF calculation:

```
! RHF SV(P)

* int 0 1
C 0 0 0 0.000000 0.000
C 1 0 0 1.389437 0.000 0.000
C 2 1 0 1.389437 120.000 0.000
```

```
C 3 2 1 1.389437 120.000 0.000

C 4 3 2 1.389437 120.000 0.000

C 5 4 3 1.389437 120.000 180.000

H 1 2 3 1.082921 120.000 180.000

H 2 1 3 1.082921 120.000 180.000

H 3 2 1 1.082921 120.000 180.000

H 4 3 2 1.082921 120.000 180.000

H 5 4 3 1.082921 120.000 180.000

H 6 5 4 1.082921 120.000 180.000

*
*Output

Print[P_ReducedOrbPopMO_L] 1

End
```

We can look at the orbitals around the HOMO/LUMO gap:

		2.00000	2.00000	2.00000	15 -0.59153 2.00000	2.00000	2.00000
0 C	S	2.9	0.0		0.1	0.0	
0 C	pz	0.0	0.0	0.0			
0 C	r –	1.4	12.4	5.9	0.3	0.0	11.2
0 C	ру	4.2		10.1	5.9	0.0	0.1
0 C	dyz	0.0	0.0	0.0		0.1	0.0
0 C	dx2y2	0.1	0.1	0.2	0.2	0.0	0.5
0 C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
1 C	s	2.9	0.0	0.3		0.0	0.0
1 C	pz	0.0	0.0		0.0		0.0
1 C	рх	1.4	12.4	5.9	0.3	0.0	
1 C	ру	4.2	4.1	10.1	5.9	0.0	0.1
1 C	dyz	0.0	0.0	0.0		0.1	0.0
1 C	dx2y2	0.1	0.1	0.2	0.2	0.0	
1 C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
2 C	S	2.9	0.0	0.0	0.4	0.0	0.1
2 C	pz	0.0	0.0	0.0	0.0	16.5	0.0
2 C	px	5.7	0.0	0.0		0.0	10.1
2 C	ру	0.0	16.5	1.3	0.0	0.0	0.0
2 C	dxz	0.0	0.0	0.0	0.0	0.1	0.0
2 C	dx2y2	0.6	0.0	0.0		0.0	1.2
2 C	dxy	0.0	0.1	0.5	0.0	0.0	0.0
3 C	S	2.9	0.0	0.3	0.1	0.0	0.0
3 C	pz	0.0	0.0	0.0	0.0	16.5	0.0
3 C	px	1.4	12.4	5.9	0.3	0.0	11.2
3 C	ру	4.2	4.1	10.1		0.0	0.1
3 C	dyz	0.0	0.0	0.0	0.0	0.1	0.0
3 C	dx2y2	0.1	0.1	0.2	0.2	0.0	0.5
3 C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
4 C	S	2.9	0.0	0.3	0.1	0.0	0.0
l C	pz	0.0	0.0	0.0	0.0	16.5	0.0
4 C	px	1.4	12.4	5.9	0.3	0.0	11.2
1 C	ру	4.2	4.1	10.1	5.9	0.0	0.1
l C	dyz	0.0	0.0	0.0	0.0	0.1	0.0
1 C	dx2y2	0.1	0.1	0.2	0.2	0.0	0.5
l C	dxy	0.4	0.0	0.0	0.2	0.0	0.0
5 C	S	2.9	0.0	0.0	0.4	0.0	0.1
5 C	pz	0.0	0.0	0.0	0.0	16.5	0.0
5 C	px	5.7	0.0	0.0	20.9	0.0	10.1
5 C	ру	0.0	16.5	1.3	0.0	0.0	0.0
5 C	dxz	0.0	0.0	0.0	0.0	0.1	0.0
5 C	dx2y2	0.6	0.0	0.0	0.2	0.0	1.2
C	anzyz	0.0	0.0	0.0	0.2	0.0	(continues