

Note

- The FrozenCore options are applied to all post Hartree-Fock methods.
- If including all electrons is desired, the `!NoFrozenCore` keyword can be simply inserted. For MP2: Frozen virtual orbitals are not allowed in gradient runs or geometry optimization!
- If ECPs are used, the number for NewNCore has to include the electrons represented by the ECPs as well. E.g. if an element is supposed to have 60 electrons in the ECP and additional 8 electrons should be frozen in the correlation calculation, NewNCore should be 68.
- In ORCA we use rather conservative frozencore settings, i. e. a large number of electrons are included in the correlation treatment. Therefore, we recommend to use properly optimized correlating basis functions in all cases, such as the cc-pwCVXZ basis sets.
- For DLPNO calculations the virtual space for core-core and core-valence correlation is adjusted by default, which is described in detail in section [Including \(semi\)core orbitals in the correlation treatment](#).
- In general, NewNCore only has an effect in calculations with FC_ELECTRONS. In calculations using the DLPNO approximation (except DLPNO-NEVPT2), NewNCore has also an effect in the other cases, as is described in section [Including \(semi\)core orbitals in the correlation treatment](#).
- Double-hybrid density functional (section [Double-Hybrid DFT](#)) calculations by default use the FrozenCore option for the perturbative part, as is the case for MP2.

2.22 CP-SCF Options

The coupled perturbed self-consistent field (CP-SCF) equations have to be solved in many cases, such as when second derivative properties (e.g. vibrational frequencies, polarizability, NMR shielding, indirect spin-spin coupling, hyperfine coupling, g-tensor) or the MP2 relaxed density (in this case they are referred to as Z-vector equations) are calculated. They are a set of linear equations generally expressed as

$$\mathbf{A}\mathbf{U}^x = \mathbf{B}^x,$$

where \mathbf{U}^x is the vector of solutions for perturbation x , the right-hand side (RHS) matrix \mathbf{B}^x is perturbation-specific and the left-hand side (LHS) matrix \mathbf{A} is perturbation-independent and contains, among other terms, the two-electron repulsion integrals $(ij|ab)$ and $(ia|jb)$. The equations are solved iteratively and the LHS is reassembled at every step, while the RHS does not change. The generation and transformation of the two-electron integrals are therefore the most time-consuming parts of the CP-SCF solution.

The ORCA module which solves these equations accepts several options given below with their default values:

```
%method
  Z_Solver      Pople # (default) Use the Pople algorithm to solve the equations
                  DIIS # Use the DIIS algorithm
                  CG   # Use the conjugate gradient algorithm
  Z_Tol         1e-3 # Convergence tolerance for the residual norm.
                  # Default is 1e-5 for VeryTightOpt
                  # and varies from 3e-3 to 3e-6 from LooseSCF to ExtremeSCF
  Z_MaxIter     128  # Maximum number of iterations
  Z_MaxDIIS     12   # Maximum number of DIIS vectors
  Z_Shift       0.3  # Level shift for DIIS
  Z_GridXC      1    # XC angular grid used for the LHS
  Z_IntAccXC    3.467 # XC radial grid accuracy used for the LHS
  Z_GridX       1    # COSX angular grid used for the LHS
  Z_IntAccX     3.067 # COSX radial grid accuracy used for the LHS
  Z_GridX_RHS   2    # COSX grid used for the RHS of MP2 Z-vector eqs (see below)
  Z_COSX_Algo   0    # (default) choose the best COSX algorithm automatically
                  1    # better prescreening, more efficient for few densities
```

(continues on next page)

(continued from previous page)

```

2      # uses more memory, more efficient for many densities
end

```

Since ORCA 6, the same settings are used for all electric response property calculations as well as for CIS/TD-DFT gradients and relaxed densities. For convenience, the keywords in the `%elprop` input block are still available but they modify the same internal variables as those in `%method`. For magnetic response properties, the solver and convergence tolerance are set separately in `%eprnmr`, because the convergence behavior of the magnetic response CP-SCF equations is sometimes different.

```

%elprop
  Solver      # Alias, see: %method Z_Solver
  Tol         # Alias, see: %method Z_Tol
  MaxIter     # Alias, see: %method Z_MaxIter
  MaxDIIS     # Alias, see: %method Z_MaxDIIS
  LevelShift  # Alias, see: %method Z_Shift
end

%eprnmr
  Solver      # Solver for magnetic response, see options at: %method Z_Solver
  Tol         # Convergence tolerance for magnetic response
  MaxIter     # Alias, see: %method Z_MaxIter
  MaxDIIS     # Alias, see: %method Z_MaxDIIS
  LevelShift  # Alias, see: %method Z_Shift
end

```

The keywords `Z_GridX` and `Z_IntAccX` are applicable if the RIJCOSX approximation is chosen for the treatment of two-electron integrals. They determine the angular and radial COSX integration grids, as discussed in section *Changing TD-DFT, CP-SCF and Hessian Grids*. Analogously, the keywords `Z_Grid` and `Z_IntAcc` determine the integration grid for DFT XC functionals.

Integrals on the RHS are evaluated differently for different perturbations - refer to sections *RIJCOSX*, *EPRNMR - keywords for magnetic properties*, *RIJCOSX-RI-MP2 Gradients*, *MP2 and RI-MP2 Second Derivatives* and *RI-MP2 and Double-Hybrid DFT Response Properties* for SCF-level gradients, EPR/NMR calculations with GIAOs, MP2 gradients, and MP2 second derivatives, respectively. For MP2 Z-vector equations, the RIJCOSX Fock-response terms in the RHS are evaluated with the COSX grid specified by `Z_GridX_RHS`. Note that it is used differently to `Z_GridX`: instead, it selects one of the three grids used in the SCF (see Sections *RIJCOSX*, *COSX Numerical Grids*, *Numerical Integration* for details) and it is not recommended to change the default value of 2.

If the RIJONX or RIJK approximation is used in the SCF, the same is also employed in the CP-SCF. Note, however, that the RI-K approximation is not efficient for these terms.

Note

When running RPA or the CP-SCF with COSX, the algorithm from ORCA-6.0.0 was hard-coded to a memory-hungry one, which was not the original intention. In ORCA-6.0.1 that was fixed, and the automatic one is now the default. This might cause very small differences to these results, specially if the number of densities is small. As a result, calculations give slightly different results with ORCA-6.0.0. This does not affect any “normal” energy, gradient or etc. calculations. Only those particular cases of RPA and CP-SCF.

2.23 Numerical Gradients

If the analytic gradient is not available, the numerical gradient can simply be requested via:

```
! NumGrad
```

as in the following example:

```
!CCSD(T) TZVPP
!Opt NumGrad
* int 0 1
C 0 0 0 0 0 0 0
O 1 0 0 1.2 0 0
H 1 2 0 1.1 120 0
H 1 2 3 1.1 120 180
*
```

NOTE

- Be aware that the numerical gradient is quite expensive. The time for one gradient calculation is equal to $6 \times (\text{number of atoms}) \times (\text{time for one single point calculation})$.
- The numerical gradient can be calculated in a multi-process run, using a maximum of three times the number of atoms (see section [Parallel and Multi-Process Modules](#)).

More details on various options, geometry convergence criteria and the like are found in section [Geometry Optimizations](#).

2.23.1 Some Notes and Tricks

Note

- TightSCF in the SCF part is set as default to avoid the buildup of too much numerical noise in the gradients.
- Even if the optimization does not converge, the ORCA output may still end with “****ORCA TERMINATED NORMALLY****”. Therefore do not rely on the presence of this line as an indicator of whether the geometry optimization is converged! Rather, one should instead rely on the fact that, an optimization job that terminates because the maximum number of iterations has been reached, will generate the following output message:

Warning

The optimization did not converge but reached the maximum number of optimization cycles. Please check your results very carefully.

While a successfully converged job will generate the following message instead:

```
*****HURRAY*****
***      THE OPTIMIZATION HAS CONVERGED      ***
*****
```

Tip

- In rare cases the redundant internal coordinate optimization fails. In this case, you may try to use `COPT` (optimization in Cartesian coordinates). This will likely take many more steps to converge but should be stable.
- For optimizations in Cartesian coordinates the initial guess Hessian is constructed in internal coordinates and thus these optimizations should converge only slightly slower than those in internal coordinates. Nevertheless, if you observe a slow convergence behaviour, it may be a good idea to compute a Hessian initially (perhaps at a lower level of theory) and use `InHess read` in order to improve convergence.
- At the beginning of a TS optimization more information on the curvature of the PES is needed than a model Hessian can give. The best choice is analytic Hessian, available for HF, DFT and MP2. In other cases (e.g. CAS-SCF), the numerical evaluation is necessary. Nevertheless you do not need to calculate the full Hessian when starting such a calculation. With ORCA we have good experience with approximations to the exact Hessian. Here it is recommended to either directly combine the TS optimization with the results of a relaxed surface scan or to use the Hybrid Hessian as the initial Hessian, depending on the nature of the TS mode. Note that these approximate Hessians do never replace the exact Hessian at the end of the optimization, which is always needed to verify the minimum or first order saddle point nature of the obtained structure.

2.23.2 Numerical Gradients

If you want to use numerical instead of analytic gradients you have to use

```
! NumGrad
```

in your input file. Additionally the settings for the numerical differentiation can be changed:

```
%numgrad
  CentralDiff true # You should use two-sided numerical differentiation, but it
                  # is possible to switch to one-sided numerical differentiation.
  DX 0.005        # Increment in Bohr for the differentiation.
  TransInvar true # Take advantage of translation invariance
end
```

MODEL CHEMISTRIES

3.1 Wavefunction Types: RHF/RKS, UHF/UKS, ROKS and more

The basic variable that controls the type of wavefunction to be computed is the variable `HFTyp` in the `%scf` block. If nothing is specified for `HFTyp`, the program will check the multiplicity given in the input: for closed-shell molecules with multiplicity 1, RHF/RKS is assumed; for open shell molecules with multiplicity larger than 1, UHF/UKS is invoked. RHF will lead to a spin restricted closed-shell type computation [180] whereas UHF does a spin unrestricted calculation.

For DFT calculations, RKS, UKS and ROKS can be used as synonyms for RHF, UHF and ROHF. The restricted open-shell DFT method (ROKS) is only operative for high-spin states that have n unpaired electrons and $S = n/2$. UKS wavefunctions will not be spin-purified.

```
%scf
  HFTyp  RHF      # closed-shell (RKS for DFT)
         UHF      # unrestricted open-shell (UKS for DFT)
         ROHF     # restricted open-shell (ROKS for DFT)
         CASSCF   # complete active space SCF
end
```

In certain cases you may want to run open-shell systems with RHF/RKS to get a “half-electron” type wavefunction [181]. The total energy is not corrected! Sometimes these half-electron computations lead to acceptable convergence, and the resulting orbitals may be used as input for ROHF, UHF or MRCI calculations. Special attention should be taken for transition metal complexes where the RHF/RKS orbitals are quite different from ROHF or UHF orbitals, so that it is not recommended to over-interpret the wavefunctions from such calculations. The calculation is set up in the following way:

```
%method AllowRHF true end
# or simply: ! AllowRHF
```

3.1.1 ROHF Options

For ROHF calculations[6, 7, 182, 183, 184, 185, 186, 187, 188] the program will try to figure out what type of open-shell situation is present on the basis of the initial guess orbitals and their energies. Most “simple” cases are well recognized, but sometimes a little help from the user is needed.

The simplest ROHF case is the `HIGHSPIN` case, where all unpaired electrons in the open-shell are ferromagnetically coupled (parallel to each other), resulting in the highest multiplicity possible. The user can request this case as follow:

```
%scf
  HFTyp      ROHF
  ROHF_case  HIGHSPIN
  ROHF_NEl[1] 4      # number of electrons in the open-shell
end
```

The ROHF code also has a very powerful feature that goes back to insights of Mike Zerner [189, 190]. It can average over either *all* states of a given configuration (CAHF) or *all states of a given spin* for a given configuration (SAHF).

Especially, the SAHF feature gives you easy access to most degenerate high symmetry situations and the orbitals resulting from such calculations will be very convenient as input for CI calculations.

```
%scf
  HFTyp      ROHF
  ROHF_case   CAHF      # configuration averaged HF
                      # spin averaged HF
  ROHF_NumOp   3         # number of operators (3, 2 or 1)
  ROHF_NOrb[1] 2,1       # number of orbitals in each open-shell
  ROHF_NEl[1]  1,1       # number of electrons in each open-shell
end
```

The hypothetical example below could represent an excited state of an octahedral d^3 transition metal complex. In this case there are five open-shell orbitals. The first three open-shell orbitals contain two electrons and the last two one electron. The input for a SAHF calculation is identical, just replace CAHF with SAHF.

```
%scf
  HFTyp      ROHF
  ROHF_case   CAHF      # configuration averaged HF
  ROHF_NumOp   3         # 3 operators in this case: closed, open1, open2
  ROHF_NOrb[1] 3,2       # 3 orbitals in first open shell, 2 in the second
  ROHF_NEl[1]  2,1       # 2 electrons in first open shell, 1 in the second
end
```

Another feature of the ROHF code is the ability to converge the SCF to a given Configuration State Function (CSF-ROHF) [191]. In this way one can approach results from MCSCF calculations. This can be requested in two ways.

The user can give a specific coupling situation.

```
%scf
  HFTyp      ROHF
  ROHF_CASE   USER_CSF      # User defined CSF
  ROHF_REF    {1 1 -1 -1} end # CSF to be converged to a 2 open shell
↪antiferromagnetically coupled
end
```

Or the user can give how many orbitals per shell. Where each open-shell will couple with antiparallel spin with the previous one.

```
%scf
  HFTyp      ROHF
  ROHF_CASE   AF_CSF        # User defined CSF
  ROHF_AFORBS 2,2          # Coupling Situation
end
```

As an example, one can think of a Fe(III) dimer, where each center is locally high spin, but they couple antiferromagnetically to each other. In order to get the ROHF solution for this system, first one needs a set of initial guess orbitals. The guess orbitals can be obtained either from the QROs of an UHF calculation, or a high spin ROHF calculation, or even a SAHF or CAHF. Independently of the method used, the orbitals need to be localized and ordered in a way that the set of 3d orbitals of each iron are grouped together in sequence. From this, one can run a CSF-ROHF calculation for the antiferromagnetic CSF as shown below:

```
%scf
  HFTyp      ROHF
  ROHF_CASE   USER_CSF      # User defined CSF
  ROHF_REF    {1 1 1 1 1 -1 -1 -1 -1 -1} end # 2 open shells coupling antiparallel
end

or

%scf
```

(continues on next page)

(continued from previous page)

```

HFTyp      ROHF
ROHF_CASE  AF_CSF                      # User defined CSF
ROHF_AFORBS 5,5                      # 2 open shells coupling antiparallel
end

```

The CSF-ROHF procedure can recognize doubly occupied and virtual orbitals in the definition of the CSF when the USER_CSF case is invoked. When detected, these orbitals will be rotated out of the open-shell defined in the ROHF method and the calculation will run normally:

```

%scf
  HFTyp      ROHF
  ROHF_CASE  USER_CSF                  # User defined CSF
  ROHF_REF   {1 1 2 1 -1 -1 0 -1} end # the DOMO will be rotated to the closed-
↪shell and
                                           # the VMO will be rotated to the virtual_
↪space.
end

```

The user can also directly input the ROHF variables by means of the ROHFOP Case User keyword. For example, for the high spin case with three electrons in three orbitals, it gives two operators with vector coupling coefficients $a = 1$ and $b = 2$ (Zerner convention).

```

%scf
  HFTyp      ROHF
  ROHFOP Case User # manual input of ROHF variables
    Nop      2      # number of operators
    Norb[1]  3      # number of open-shell orbitals
    Nel[1]   3      # number of open-shell electrons
    A[1,1]   1      # Coulomb vector in the open shell
    B[1,1]   2      # Exchange vector in the open shell
end
end

```

One awkward feature of the ROHF theory is that the Fock operator is somewhat arbitrarily defined. Different choices lead to the same wavefunction, but have different convergence properties that may vary from system to system. ORCA thus lets the user choose the desired variant. Playing around with these choices may turn a divergent or slowly converging ROHF calculation into a successful calculation!

The ROHF_Restrict feature is another feature that may be useful. If you suspect that the ROHF calculation does not converge because an open-shell and a closed-shell orbital are flipping back and forth, you can try to avoid this behavior by choosing ROHF_Restrict true. Of course there is no guarantee that it will work, and no guarantee that the system stays in the desired state. However, it decreases the chances of large, uncontrolled steps.

```

%scf
  ROHF_Mode 0 # construct F according to Pulay (default)
             1 # construct F as in the Gamess program
             2 # construct F according to Kollmar

  ROHF_Restrict false # restrict orbital interchanges and off-diagonal elements
                      # (default=false)

# a complete list of ROHF variables
ROHFOP
  Case      User # manual input of ROHF variables
  Nop       2    # number of operators
  Norb[1]   3    # number of open-shell orbitals
  Nel[1]    3    # number of open-shell electrons
  A[1,1]    1    # Coulomb vector in the open shell
  B[1,1]    2    # Exchange vector in the open shell
  Mode      2    # use the Kollmar operator
  Restrict  false # do not restrict

```

(continues on next page)

(continued from previous page)

```
end
end
```

3.1.2 UHF Natural Orbitals

The program can produce the UHF natural orbitals (UNOs). With these, the open-shell wavefunction can be pictured conveniently. The syntax is simple:

```
%scf
  UHFNO true
end
# or simply: ! UNO
```

There are various printing options for UNOs described in the output section *Population Analysis*. The UNOs can also be plotted as described in the plots section *Orbital and Density Plots*. In general the program stores a file `BaseName.uno`, where `BaseName` is by default the name of your input file with `.inp` stripped off. Accordingly, the `gbw` file is named `BaseName.gbw`. The `.uno` file is a normal `gbw` file that contains the geometry, basis set and the UNO orbitals. It could be used, for example, to start a ROHF calculation.

3.1.3 UNO, UCO and QROs input

For calculations on open-shell systems we recommend to use the keywords `!UNO !UCO` in the input line. This will generate quasi-restricted molecular orbitals QRO, unrestricted natural spin-orbitals UNSO, unrestricted natural orbitals UNO and unrestricted corresponding orbitals UCO. Moreover, it will print the UCO overlaps in the output, which can provide very clear information about the spin-coupling in the system. Below an example of the input and section of the output is provided.

```
!B3LYP def2-SVP UNO UCO TightSCF
```

The UCO overlap section in the output will look like:

```
***UHF Corresponding Orbitals were saved in MyJob.uco***

-----
Orbital      Overlap(*)
-----
.
.
.
96:          0.99968
97:          0.99955
98:          0.99947
99:          0.99910
100:         0.99873
101:         0.99563
102:         0.74329
103:         0.00000
```

The overlap that corresponds to a value usually less than 0.85 denotes a spin-coupled pair. Whereas, values close to 1.00 and 0.00 refers to the doubly occupied and singly occupied orbitals respectively.

3.2 Hartree Fock Theory

The Hartree-Fock (HF) method[192, 193, 194] is a basic *ab-initio* quantum mechanical method that is used to approximate many-body electronic wavefunctions within mean field theory. Many post-Hartree-Fock methods like Coupled-Cluster typically make use of HF reference wavefunctions.

A basic Hartree-Fock calculation can be invoked via the simple input keyword `!HF`

```
! HF
```

or via the `%method` block.

```
%method
  method HF      # equivalent to method HFGTO
end
```

By default, ORCA will make use of the `def2-SVP` *basis set* if not specified else. Hartree-Fock calculations can be accelerated drastically by using different *RI approximations*.

➡ See also

ORCA provides several choices of wavefunction types like restricted (RHF), unrestricted (UHF), or unrestricted open-shell HF (ROKS). These options are documented in [Section 3.1](#). HF is also an integral part of *hybrid*- and *double-hybrid* DFT.

💡 Tip

Grimme and co-workers developed a low-cost method based on HF called *HF-3c* that can be employed for very large systems. Some HF-based *semi-empirical methods* like AM1 or PM3 are also available.

3.3 Density Functional Theory (DFT)

Due to its good cost-accuracy ratio, Kohn-Sham density functional theory (DFT) has become the workhorse of quantum chemistry. Over the last few decades, a whole zoo of density functionals has been developed that differ in terms of their construction and design philosophy. The so-called Jacob's Ladder picture ([Fig. 3.1](#)) coined by Perdew [195] categorizes density functionals according to their basic physical ingredients into rungs. With higher rungs, the cost of the given functionals increased but also higher accuracy is conceptionally expected. However, the accuracy of density functionals strongly varies and careful study of benchmark studies is generally recommended.

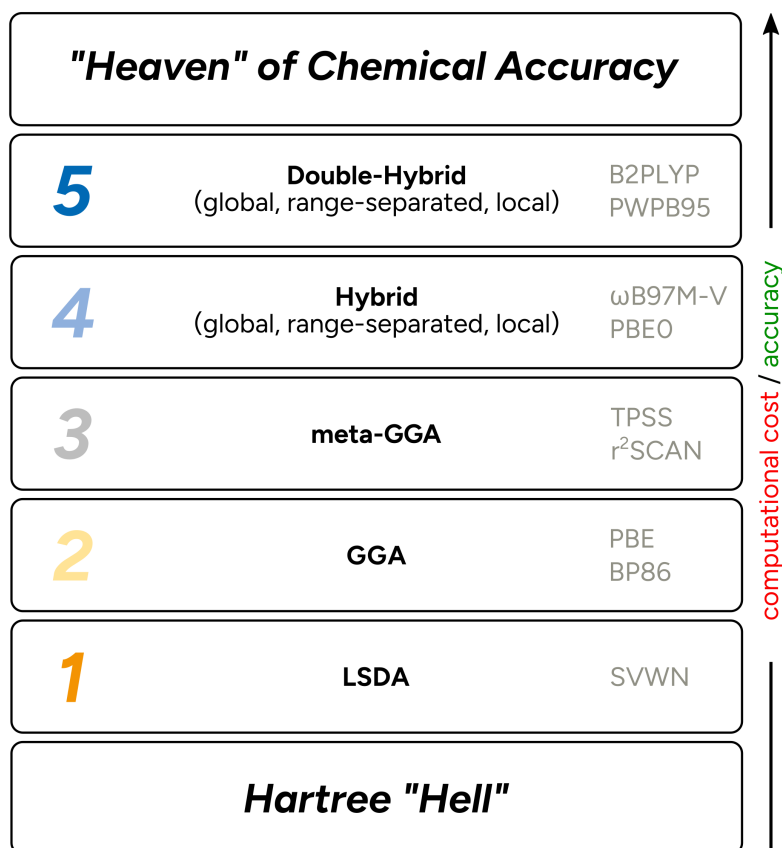


Fig. 3.1: Jacob's ladder picture of density functionals.

ORCA features various density functionals of each rung either implemented natively or via the *LibXC Functional Library*. These functionals can be combined with various *basis sets* and *relativistic approaches*. Further, ORCA features a useful toolkit of *resolution-of-identity (RI)* methods to speed-up DFT calculations.

For dispersion corrections that can be used with DFT and the respective available functional-specific parameters refer to the *Dispersion Corrections section*.

Also note that some functionals like the SCAN and Minnesota families may require enhanced *numerical integration grid settings*.

3.3.1 Basic Usage

All implemented DFT methods can be invoked via their respective keywords, e.g. !PBE using the simple input line:

```
! PBE
```

Warning

If only a functional is defined, various default settings will be used including a small def2-SVP basis set, an appropriate *RI* treatment and default *numerical integration grids*! To achieve higher accuracy, specifically the *basis set* should be adjusted!

Many functionals can also be invoked via the %method block even though double-hybrid functionals being not available due to technical reasons:

```
%method
  Method DFT          # equivalent to method DFTO
  Functional PBE
end
```

ORCA further provides some more utility to construct and modify functionals via the `%method` block. These expert options are described in the respective sections of each functional rung. Various other functionals can also be invoked via the *LibXC Functional Library*.

3.3.2 Local Functionals

Local exchange-correlation functionals are the most basic examples of functionals constructed according to equation (3.1):

$$E_{\text{XC}} = E_{\text{LSD}}^{\text{X}} + E_{\text{LSD}}^{\text{C}} \quad (3.1)$$

They typically use the original Slater exchange functional with different local correlation functionals like VWN-V[196] or PW-LDA[197]. All available local functionals that are available in ORCA can be found in Table 3.1.

Table 3.1: List of available local functionals.

Functional	Keywords	Description
HFS	HFS	Hartree–Fock–Slater Exchange only functional
LDA or LSD	LDA, LSD	Local density approximation (defaults to VWN5)
VWN-V [196]	VWN,VWN5	Vosko-Wilk-Nusair local density approx. parameter set “V”
VWN-III [196]	VWN3	Vosko-Wilk-Nusair local density approx. parameter set “III”
PW-LDA [197]	PWLDA	Perdew-Wang parameterization of LDA

3.3.3 Gradient Corrected Functionals: (meta-)GGAs

The next rung corresponds to gradient corrected functionals of the generalized gradient approximation (GGA) and the meta-generalized gradient approximation (meta-GGA) types. A simplified exchange-correlation energy expression for these types of functionals is given in (3.2). (meta-)GGAs are specifically useful for *geometry optimizations* and screening purposes due to their comparably small computational cost.

$$E_{\text{XC}} = E_{\text{LSD}}^{\text{X}} + E_{\text{GGA}}^{\text{X}} + E_{\text{LSD}}^{\text{C}} + E_{\text{GGA}}^{\text{C}} \quad (3.2)$$

Table 3.2: List of available (meta-)GGA functionals.

Functional	Keyword	Description
GGAs		
BP86 [198, 199]	BP86	Becke '88 exchange and Perdew '86 correlation
BLYP [200]	BLYP	Becke '88 exchange and Lee-Yang-Parr correlation
OLYP [201]	OLYP	Handy's "optimal" exchange and Lee-Yang-Parr correlation
GLYP [202]	GLYP	Gill's '96 exchange and Lee-Yang-Parr correlation
XLYP [203]	XLYP	The Xu and Goddard exchange and Lee-Yang-Parr correlation
PW91 [204]	PW91	Perdew-Wang '91 GGA functional
mPWPW [205]	MPWPW	Modified PW exchange and PW correlation
mPWLYP [205]	MPWLYP	Modified PW exchange and LYP correlation
PBE [206]	PBE	Perdew-Burke-Erzerhoff GGA functional
RPBE [207]	RPBE	"Modified" PBE
revPBE [208]	REVPBE	"Revised" PBE
RPW86PBE [209]	RPW86PBE	PBE correlation with refitted Perdew '86 exchange
PWP [204]	PWP	Perdew-Wang '91 exchange and Perdew '86 correlation
composite		
B97-3c [210]	B97-3C	Composite DFT method by Grimme et al.
meta-GGAs		
B97M-V [211]	B97M-V	Head-Gordon's DF B97M-V with VV10 nonlocal correlation
B97M-D3(BJ) [212]	B97M-D3BJ	Modified version of B97M-V with D3BJ correction by Najibi and Goerigk
B97M-D4 [213]	B97M-D4	Modified version of B97M-V with DFT-D4 correction by Najibi and Goerigk
SCAN [214]	SCANFUNC	Perdew's SCAN functional
rSCAN [215]	RSCAN	Regularized SCAN functional
r ² SCAN [216]	R2SCAN	Regularized and restored SCAN functional by Furness, Sun et al.
M06-L [217]	M06L	The Minnesota M06-L meta-GGA functional
TPSS [218]	TPSS	The TPSS meta-GGA functional
revTPSS [219, 220]	REVTSS	Revised TPSS meta-GGA functional
composite		
r ² SCAN-3c [221]	R2SCAN-3C	Composite DFT method by Grimme et al.

Customization of Basic Functionals

Basic exchange-correlation functionals can also be constructed from their components via the `%method` block using the `Exchange`, `Correlation`, and `LDASOpt` keywords. These control the exchange and correlation parts of the functional as well as the local part of the correlation functional, respectively. For example BP86 (!BP86) is constructed from Becke's '88 exchange via `X_BECKE` and Perdew's '86 correlation functional via `C_P86`. In ORCA, it further employs Perdew and Wang's LDA reparameterization[197] `PWLDA` for the local part of the correlation functional via `C_PWLDA`:

```
%method
  Method DFT
  Exchange      X_BECKE
  Correlation    C_P86
  LDASOpt        C_PWLDA
end
```

For a full list of available components see [Table 3.3](#).

The correct choice of the local part of the correlation functional is crucial to ensure reproducibility of the results. In ORCA, almost all functionals choose `PWLDA`[197] as the underlying LDA functional as it represents a slightly better fit to the uniform electron gas than VWN. For VWN (Vosko, Wilk, and Nusair[196]), several parameterizations exist.

In their classic paper give two sets of parameters – one in the main body (parametrization of RPA results; known as VWN-III) and one in their table 5 of correlation energies (parametrization of the Ceperley/Alder Monte Carlo results; known as VWN-V). The results from PW-LDA are very similar to those of VWN-V, whereas VWN-III yields quite different values. Over the years, the usage of different parameterizations for the same functional name throughout different quantum chemistry programs caused some confusion and deviating results. A prominent example is the different implementation of the B3LYP *hybrid functional* within ORCA and Gaussian. While ORCA uses VWN-V consistent with the TURBOMOLE program[222, 223, 224], the Gaussian program[225] employs VWN-III. Using the LDAOpt keyword, both variants can be constructed:

```
# consistent with TURBOMOLE
%method
  Method DFT
  Functional B3LYP
  LDAOpt C_VWN5
end
```

```
# consistent with Gaussian
%method
  Method DFT
  Functional B3LYP
  LDAOpt C_VWN3
end
```

For convenience both functionals can be invoked separately via the simple input keywords !B3LYP for the ORCA/TURBOMOLE variant and !B3LYP/G for the Gaussian variant.

Note

A special situation arises if LYP is the correlation functional [226] since LYP is *not* a correction to the correlation, but rather includes the full correlation. It is therefore used in the B3LYP functional as:

$$E_{\text{B3LYP}}^{\text{C}} = E_{\text{LSD}}^{\text{C}} + c(E_{\text{LYP}}^{\text{C}} - E_{\text{LSD}}^{\text{C}}) \quad (3.3)$$

Some functionals further incorporate empirical parameters that can be changed to improve agreement with experiment. Currently, there are a few parameters that can be changed (other than the parameters used in the *hybrid functionals*, which are described later).

The first of these parameters is α of Slater's X_α method. Theoretically, it has a value of 2/3 and this is used in the HFS and LSD functionals. However, the exchange contribution is underestimated by about 10% by this approximation (quite significant!) and a value around 0.70-0.75 is recommended for molecules.

The second parameter is the β for Becke's gradient-corrected exchange functional. Becke determined the value 0.0042 by fitting the exchange energies for rare gas atoms. There is some evidence that with smaller basis sets, a slightly smaller value such as 0.0039 gives improved results for molecules.

The next parameter is the value κ , which occurs in the PBE exchange functional. It has been given the value 0.804 by Perdew et al. in order to satisfy the Lieb-Oxford bound. Since then, other workers have argued that a larger value for this parameter (around 1.2) gives better energetics, which is explored in the revPBE functional. Note, however, that while revPBE gives slightly better energetics, it also gives slightly poorer geometries.

The last two parameters are also related to PBE. Within the PBE correlation functional, there is β_C (not to be confused with the β exchange parameter in Becke's exchange functional), whose original value is $\beta_C = 0.066725$. Modified variants exist with different β_C values, e.g., the PBEsol functional and the *PBEh-3c composite method*. Furthermore, the μ parameter in the PBE exchange functional may be modified. In the original formulation, it is related to β_C via $\mu = \beta_C \frac{\pi^2}{3}$, but has been modified in later variants as well.

Table 3.3: %method block input keywords to modify basic exchange-correlation functionals.

Keyword	Options	Description
Exchange	X_NOX	No exchange
	X_SLATER	Slater's local exchange
	X_BECKE	Becke's '88 exchange
	X_wB88	Short-range Becke '88 exchange for range-separated functionals
	X_G96	Gill '96 gradient exchange
	X_PW91	Perdew-Wang (PW) '91 gradient exchange
	X_mPW	Adamo-Barone modification of PW exchange
	X_PBE	Perdew-Burke-Ernzerhof (PBE) exchange
	X_RPBE	Revised PBE exchange
	X_OPTX	Hoe/Cohen/Handy's optimized exchange
	X_X	Xu/Goddard exchange
	X_TPSS	TPSS meta-GGA exchange
	X_B97D	Grimme's modified exchange for the B97-D GGA
	X_B97BECKE	Becke's original exchange for the B97 hybrid
	X_SCAN	Perdew's constrained exchange for the SCAN meta-GGA
	X_RSCAN	Perdew's constrained exchange for the rSCAN meta-GGA
	X_R2SCAN	Perdew's constrained exchange for the r ² SCAN meta-GGA
Correlation	C_NOC	No correlation
	C_VWN5	Local Vosko-Wilk-Nusair correlation (parameter set "V")
	C_VWN3	Local Vosko-Wilk-Nusair correlation (parameter set "III")
	C_PWLDA	Local PW correlation
	C_P86	Perdew '86 correlation
	C_PW91	PW '91 correlation
	C_PBE	PBE correlation
	C_LYP	LYP correlation
	C_TPSS	TPSS meta-GGA correlation
	C_B97D	Grimme's modified correlation for the B97-D GGA
	C_B97BECKE	Becke's original correlation for the B97 hybrid
	C_SCAN	Perdew's constrained correlation for the SCAN meta-GGA
	C_RSCAN	Perdew's constrained correlation for the rSCAN meta-GGA
	C_R2SCAN	Perdew's constrained correlation for the r ² SCAN meta-GGA
LDAOpt	C_NOC	Sets the local part of any correlation functional to no correlation
	C_PWLDA	Sets the local part of any correlation functional to PWLDA
	C_VWN5	Sets the local part of any correlation functional to VWN5
	C_VWN3	Sets the local part of any correlation functional to VWN3
Specific Parameters		
XAlpha	<real>	Modifies Slater's α parameter (default 2/3)
XBeta	<real>	Modifies Becke's β parameter (default 0.0042)
XKappa	<real>	Modifies the PBE(exchange) κ parameter (default 0.804)
XMuePBE	<real>	Modifies the PBE(exchange) κ parameter (default 0.21952)
CBetaPBE	<real>	Modifies the PBE(correlation) beta parameter (default 0.066725)

3.3.4 Hybrid DFT

The idea of replacing a portion of the DFT exchange by exact exchange from Hartree-Fock theory (HFX) was introduced by Becke in 1993[227, 228]. This scheme is typically referred to as hybrid DFT and the most simple formulation of a one-parameter hybrid functional is depicted in equation (3.4)

$$E_{\text{hyb}}^{\text{XC}} = aE_{\text{HF}}^{\text{X}} + (1 - a)E_{\text{GGA}}^{\text{X}} + E_{\text{GGA}}^{\text{C}} \quad (3.4)$$

A more general description is given by the adiabatic connection model (ACM). In the ACM, hybrid functionals are constructed with three parameters according to equation (3.5):

$$E_{\text{XC}} = aE_{\text{HF}}^{\text{X}} + (1 - a)E_{\text{LSD}}^{\text{X}} + bE_{\text{GGA}}^{\text{X}} + E_{\text{LSD}}^{\text{C}} + cE_{\text{GGA}}^{\text{C}} \quad (3.5)$$

Here, E_{XC} is the total exchange/correlation energy, E_{HF}^X is the Hartree-Fock exchange, E_{LSD}^X is the local (Slater) exchange, E_{GGA}^X is the gradient correction to the exchange, E_{LSD}^C is the local, spin-density based part of the correlation energy, and E_{GGA}^C is the gradient correction to the correlation energy.

Over the years, various strategies to construct hybrid functionals have been developed. These range from physics motivated approaches relying on as few as possible parameters like the overall well-balanced PBE0 functional[229] to more empirical variants like the prominent three-parameter hybrid B3LYP[230]. Furthermore, various treatments of the HF exchange admixture were developed. The basic hybrids that use a static amount of HFX are typically referred to as *global hybrids* while functionals with variable amounts of HFX include *range-separated hybrids* and local hybrids. Various *global* and *range-separated hybrid functionals* are available in ORCA. Local hybrids may be implemented in a future version.

ORCA also provides a sophisticated infrastructure to create and *customize global* and *range-separated hybrid functionals*.

Note

As by theory hybrid functionals are significantly more costly than (meta-)GGAs, ORCA uses the sophisticated *RJJCOSX* algorithm as a default to speed up hybrid DFT calculations.

Global Hybrid Functionals

Global hybrid functionals employ a fixed amount of Hartree-Fock exact exchange. ORCA features various representatives with different compositions and amounts of HFX. The optimal amount of HFX is controversially discussed in the literature and many empirical functionals were constructed by fitting it to reproduce an experimental property. However, it has been argued from a theoretical viewpoint that the optimal mixing of HF exchange is 25% [231] and large amounts of HFX are known to be disadvantageous for transition metal complexes.

A list of natively implemented global hybrid functionals is given in Table 3.4.

Table 3.4: List of available global hybrid functionals.

Functional	Key-word	%HF	Description
hybrid GGAs			
B1LYP [232]	B1LYP	25	The one-parameter hybrid functional with Becke '88 exchange and Lee-Yang-Parr correlation (25% HF exchange)
B3LYP [230]	B3LYP	20	The popular B3LYP functional (20% HF exchange) as defined in the Turbo-Mole program system (LDAOPT C_VWN5)
B3LYP/G [230]	B3LYP/ G	20	The popular B3LYP functional (20% HF exchange) as defined in the Gaussian program system (LDAOPT C_VWN3)
O3LYP [233]	O3LYP	11.61	The Handy hybrid functional
X3LYP [203]	X3LYP	21.8	The Xu and Goddard hybrid functional
B1P86 [199]	B1P86	25	The one-parameter hybrid version of BP86
B3P86 [227]	B3P86	20	The three-parameter hybrid version of BP86
B3PW91 [227]	B3PW91	20	The three-parameter hybrid version of PW91
PW1PW [229]	PW1PW	25	One-parameter hybrid version of PW91
mPW1PW [205]	MPW1PW	25	One-parameter hybrid version of mPWPW
mPW1LYP [205]	MPW1LYP	25	One-parameter hybrid version of mPWLYP
PBE0 [229]	PBE0	25	One-parameter hybrid version of PBE
revPBE0 [234]	REVPBE0	25	“Revised” PBE0
revPBE38 [234]	REVPBE3	37.5	“Revised” PBE0 with 37.5% HF exchange
BHandHLYP [227]	BHANDH1	50	Half-and-half hybrid functional by Becke
hybrid meta-GGAs			
M06 [235]	M06	27	
M06-2X [235]	M062X	54	
PW6B95 [236]	PW6B95	28	
TPSSH [218]	TPSSH	10	The hybrid version of TPSS (10% HF exchange)
TPSS0 [237]	TPSS0	25	A 25% exchange version of TPSSH that yields improved energetics
r ² SCANh [238]	R2SCANh	10	Global hybrid variant of r ² SCAN with 10% HF exchange
r ² SCAN0 [238]	R2SCAN0	25	Global hybrid variant of r ² SCAN with 25% HF exchange
r ² SCAN50 [238]	R2SCAN5	50	Global hybrid variant of r ² SCAN with 50% HF exchange
composite			
PBEh-3c [239]	PBEH-3C	42	Composite DFT approach by Grimme et al.
B3LYP-3c [240]	B3LYP-3	20	Composite DFT approach by Grimme et al.

Customization of Global Hybrids

Global hybrid functional parameters can be customized via the `%method` block. For the customization of the basic exchange and correlation components see the [Customization of Basic Functionals](#) section. The important parameters for hybrid functionals are those used in the ACM model (cf. equation (3.5)). In this case, we will recreate the PBE0 functional (25% HFX) by using the ACM syntax:

```
! RIJCOSX
# PBE0
%method
  Method DFT
  Functional PBE
  ACM_A      0.25      # fraction of Hartree-Fock exchange
  ACM_B      0.75      # fraction of the GGA exchange
  ACM_C      1.00      # fraction of the GGA correlation
  ScalLDAC   1.00      # fraction of LDA correlation, must be equal to ACM_C
end
```

For simplicity, the ACM parameters can also be given as one string.

```
! RIJCOSX
# PBE0
%method
  Method DFT
  Functional PBE
  ACM      0.25, 1.00, 1.00 # as ACM   ACM-A, ACM-B, ACM-C
  ScalLDAC 1.00           # fraction of LDA correlation, must be equal to_
↪ACM_C
end
```

As the ACM formalism can be non-intuitive, ORCA provides an equivalent syntax to control the respective parameters.

```
! RIJCOSX
# PBE0
%method
  Method DFT
  Functional PBE
  ScalHFX      0.25      # fraction of Hartree-Fock exchange
  ScalDFX      0.75      # fraction of GGA exchange
  ScalGGAC     1.00      # fraction of GGA correlation
  ScalLDAC     1.00      # fraction of LDA correlation, must be equal to_
↪ScalGGAC
end
```

Warning

Note that `Functional` chooses specific predefined values for the variables `Exchange`, `Correlation`, `LDAOpt` and `ACM` described below. If given as a simple input keyword, in some cases, it will also activate a dispersion correction. You can explicitly give these variables instead or in addition to `Functional`. However, make sure that you specify these variables *after* you have assigned a value to `Functional` or the others will be reset to the values chosen by `Functional`.

Table 3.5: `%method` block input keywords to modify global hybrid functionals.

Keyword	Options	Description
ACM	<real, real, real>	Defines the ACM-A, ACM-B, and ACM-C parameters
ACM_A	<real>	Defines the ACM-A parameter (fraction of Hartree-Fock exchange)
ACM_B	<real>	Defines the ACM-B parameter (fraction of GGA exchange)
ACM_C	<real>	Defines the ACM-C parameter (fraction of GGA correlation)
ScalHFX	<real>	Controls the fraction of Hartree-Fock exchange, equiv. to ACM_A
ScalDFX	<real>	Controls the fraction of GGA exchange, equiv. to ACM_B
ScalGGAC	<real>	Controls the fraction of GGA correlation, equiv. to ACM_C

Range-Separated Hybrid Functionals

Unlike *global hybrids*, range-separated hybrid (RSH) split the two-electron operator used for the exchange into a short-range (SR) and a long-range (LR) regime wherein the short-range part is typically described by DFT exchange and the long-range part by Hartree-Fock exchange. In ORCA, all range-separated hybrids use the error function splitting approach by Hirao and coworkers to distinguish both regimes (cf. equation (3.6)).[241] This allows the definition of DFT functionals that dominate the short-range part by an adapted exchange functional of *LDA*, *GGA* or *meta-GGA* level and the long-range part by Hartree-Fock exchange.

$$r_{12}^{-1} = \underbrace{\operatorname{erfc}(\mu \cdot r_{12}) \cdot r_{12}^{-1}}_{\text{SR}} + \underbrace{\operatorname{erf}(\mu \cdot r_{12}) \cdot r_{12}^{-1}}_{\text{LR}} \quad (3.6)$$

where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt$ and $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$. Note that the splitting is only applied to the exchange contribution; all other contributions (one-electron parts of the Hamiltonian, the electron-electron Coulomb interaction and the approximation for the DFT correlation) are not affected. Later, Handy and coworkers[242] generalized the ansatz to:

$$r_{12}^{-1} = \underbrace{\frac{1 - [\alpha + \beta \cdot \operatorname{erf}(\mu \cdot r_{12})]}{r_{12}}}_{\text{SR}} + \underbrace{\frac{\alpha + \beta \cdot \operatorname{erf}(\mu \cdot r_{12})}{r_{12}}}_{\text{LR}} \quad (3.7)$$

This form of the splitting used in ORCA is shown visually (according to Handy and coworkers) in Fig. 3.2.

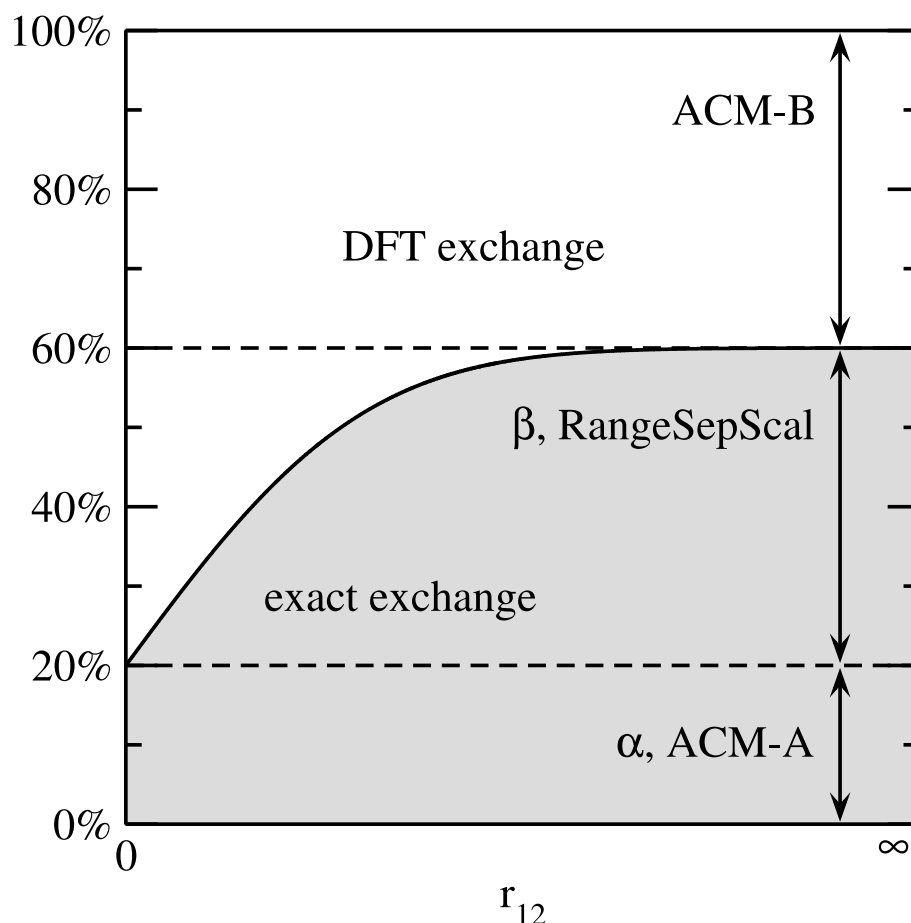


Fig. 3.2: Graphical description of the range-separation ansatz. The gray area corresponds to Hartree-Fock exchange. α and β follow Handy's terminology [242].

Important

Currently, RSH functionals can be used for integral-direct single-point calculations, calculations involving the first nuclear gradient (i.e. geometry optimizations), frequency calculations, TDDFT, TDDFT nuclear gradient, and EPR/NMR calculations. As with the *global hybrids*, *RIJCOSX* is used by default to drastically speed up the calculations. Alternatively, only RIJONX can be used.

A list of available range-separated hybrid functionals is given in [Table 3.6](#)

Table 3.6: List of available range-separated hybrid functionals. μ is the range-separation parameter.

Functional	Keyword	%HFX	μ / bohr ⁻¹	% fixed DFX
hybrid GGAs				
ω B97 [243]	WB97	0 – 100	0.40	—
ω B97X [243]	WB97X	15.7706 – 100	0.30	—
ω B97X-V [244]	WB97X-V	16.7 – 100	0.30	—
ω B97X-D3 [245]	WB97X-D3	19.5728 – 100	0.25	—
ω B97X-D3(BJ) [212]	WB97X-D3BJ	16.7 – 100	0.30	—
ω B97X-D4 [213]	WB97X-D4	16.7 – 100	0.30	—
ω B97X-D4rev[49]	WB97X-D4REV	16.7 – 100	0.30	—
CAM-B3LYP [242]	CAM-B3LYP	19 – 65	0.33	35%
LC-BLYP [246]	LC-BLYP	0 – 100	0.33	—
LC-PBE [241]	LC-PBE	0 – 100	0.47	—
hybrid meta-GGAs				
ω B97M-V [247]	WB97M-V	15 – 100	0.30	—
ω B97M-D3(BJ) [212]	WB97M-D3BJ	15 – 100	0.30	—
ω B97M-D4 [213]	WB97M-D4	15 – 100	0.30	—
ω B97M-D4rev [49]	WB97M-D4REV	15 – 100	0.30	—
ω r ² SCAN [248]	WR2SCAN	0 – 100	0.30	—
composite				
ω B97X-3c [49]	WB97X-3C	16.7 – 100	0.30	—

Customizing Range-Separated Hybrids

Range-separated hybrid functional parameters can also be customized via the `%method` block. For the customization of the basic exchange and correlation components see the *Customization of Basic Functionals* section. Basic hybrid-related keywords and the respective ACM model are described in the *Hybrid DFT* and *Customization of Global Hybrids* sections.

In this example, we will recreate the LC-PBE functional:

```
! RIJCOSX
# LC-PBE
%method
  Method DFT
  Exchange X_LRCPBE
  Correlation C_PBE
  LDAOpt C_PWLDA
  ACM_A      0.00 # fraction of static Hartree-Fock exchange
  ACM_B      0.00 # fraction of static GGA exchange
  ACM_C      1.00 # fraction of GGA correlation
  ScalLDAC   1.00 # fraction of LDA correlation, must be equal to ScalGGAC
  RangeSepEXX true # activates range-separated exchange
  RangeSepMu 0.47 # range-separation parameter, should be >0
  RangeSepScal 1.00 # fraction of variable HFX, should sum to 1 with ACM-A and
↪ACM-B
end
```

Which is equivalent to the `ScalX` syntax described before.

```
! RIJCOSX
# LC-PBE
%method
  Method DFT
  Exchange X_LRCPBE
  Correlation C_PBE
```

(continues on next page)