

Table 3.17: DFT-D3(0) (!D3ZERO) parameters used for various functionals.

| Functional | s_6 | $r s_6$ | s_8 | α_6 |
|------------|--------|---------|---------|------------|
| HF | 1.0000 | 1.1580 | 1.7460 | 14 |
| BLYP | 1.0000 | 1.0940 | 1.6820 | 14 |
| BP86 | 1.0000 | 1.1390 | 1.6830 | 14 |
| B97-D | 1.0000 | 0.8920 | 0.9090 | 14 |
| REVPBE | 1.0000 | 0.9230 | 1.0100 | 14 |
| PBE | 1.0000 | 1.2170 | 0.7220 | 14 |
| RPBE | 1.0000 | 0.8720 | 0.5140 | 14 |
| TPSS | 1.0000 | 1.1660 | 1.1050 | 14 |
| B3LYP | 1.0000 | 1.2610 | 1.7030 | 14 |
| PBE0 | 1.0000 | 1.2870 | 0.9280 | 14 |
| PW6B95 | 1.0000 | 1.5230 | 0.8620 | 14 |
| TPSS0 | 1.0000 | 1.2520 | 1.2420 | 14 |
| B2PLYP | 0.6400 | 1.4270 | 1.0220 | 14 |
| B2GP-PLYP | 0.5600 | 1.5860 | 0.7600 | 14 |
| PWPB95 | 0.8200 | 1.5570 | 0.7050 | 14 |
| mPWLYP | 1.0000 | 1.2390 | 1.0980 | 14 |
| BPBE | 1.0000 | 1.0870 | 2.0330 | 14 |
| BHANDHLYP | 1.0000 | 1.3700 | 1.4420 | 14 |
| TPSSh | 1.0000 | 1.2230 | 1.2190 | 14 |
| REVPBE0 | 1.0000 | 0.9490 | 0.7920 | 14 |
| REVPBE38 | 1.0000 | 1.0210 | 0.8620 | 14 |
| RPW86PBE | 1.0000 | 1.2240 | 0.9010 | 14 |
| B3PW91 | 1.0000 | 1.1760 | 1.7750 | 14 |
| M06L | 1.0000 | 1.5810 | 0.0000 | 14 |
| M06 | 1.0000 | 1.3250 | 0.0000 | 14 |
| M062X | 1.0000 | 1.6190 | 0.0000 | 14 |
| wB97X-D3 | 1.0000 | 1.2810 | 1.0000 | 14 |
| CAM-B3LYP | 1.0000 | 1.3780 | 1.2170 | 14 |
| SCAN | 1.0000 | 1.3240 | 0.0000 | 14 |
| wB97X-2 | 0.5470 | 4.7330 | -0.1090 | 14 |
| PBE0DH | 0.8800 | 1.1280 | 0.0000 | 14 |
| PBE02 | 0.5400 | 1.2430 | -0.2280 | 14 |
| PBE-QIDH | 0.4000 | 1.1140 | 0.0540 | 14 |

3.4.2 Non-Local Dispersion Correction (VV10): DFT-NL

A different route to correct for London dispersion is followed by the van der Waals Density Functional (vdW-DF) as pioneered by Langreth and Lundquist [282]. These methods use only the electron density to include such dispersion/correlation effects and can be used in a self-consistent way also correction the molecular orbitals and not only energies. The vdW correlation functional VV10 of Vydrov and Van Voorhis [275] currently seems to be the most promising candidate for a general and accurate electronic structure method.

We use the term DFT-NL for any density functional in combination with the non-local part of the VV10 functional with an optimized parameter b , which will be defined below. The performance of these methods has been evaluated in Ref. [276] using the GMTKN30 [250, 283, 284] database and the S66 set [285]. The performance of weak hydrogen bonds were evaluated in Ref. [286].

DFT-NL and DFT-D3/D4 perform very similarly, but NL is to be preferred for metallic systems or when the basic electronic structure changes significantly (e.g. oxidations or ionizations). As a recent example, Janes and Iron showed that for functionals such as wB97X-V, including VV10 correlation results in very high quality reaction barriers when metals are involved [287].

The total exchange-correlation (XC) energy of VV10-type functionals is defined in eq. (3.18). It is composed of standard exchange (X) and correlation (C) parts and the non-local (NL) term, which covers (mainly) long-range

dispersive energy:

$$E_{\text{XC}}^{\text{DFT-NL}} = E_{\text{X}}^{(\text{hybrid})\text{GGA}} + E_{\text{C}}^{\text{GGA}} + E_{\text{C-NL}}^{\text{VV10}} \quad (3.18)$$

The NL term is given by the following double integral:

$$E_{\text{C-NL}}^{\text{VV10}} = \int dr \rho(r) \left[\beta + \frac{1}{2} \int dr' \rho(r') \varphi(r, r') \right] \quad (3.19)$$

where ρ is the total electron density, and the definitions of the kernel $\varphi(r, r')$ and β are as follows (in a.u.):

$$\varphi(r, r') = -\frac{3}{2gg'(g+g')} \quad (3.20)$$

$$\beta = \frac{1}{32} \left[\frac{3}{b^2} \right]^{3/4} \quad (3.21)$$

with

$$\begin{aligned} g(r) &= \omega_0(r) R^2 + \kappa(r) \\ R &= |r - r'| \\ \omega_0(r) &= \sqrt{C \left| \frac{\nabla \rho(r)}{\rho(r)} \right|^4 + \frac{4\pi}{3} \rho(r)} \\ \kappa(r) &= b \frac{3\pi}{2} \left[\frac{\rho(r)}{9\pi} \right]^{1/6} \end{aligned} \quad (3.22)$$

In the original definition, the short-range attenuation parameter b appearing in κ and β was fitted to the S22 set [288] of non-covalent interactions ($b = 5.9$ for the rPW86PBE GGA). The other parameter $C = 0.0093$, appearing in ω_0 , determines the long-range behavior, and was set to its original value. Other DFT-NL functionals are constructed analogously. For a detailed discussion of the derivation of the formulas and their physical meaning and basis, see the references given above.

The defined energy of the non-local DFT-NL exchange-correlation functional can be computed in two ways: as a post-SCF addition based on a converged SCF density or in a self-consistent treatment. In our implementation of the non-self-consistent DFT-NL functional, a self-consistent DFT computation is performed as the first step. In the second step, the optimized electron density from the DFT computation is taken as input for the energy calculation of the non-local part. In the self-consistent treatment, the orbitals and density are optimized in the presence of the full DFT + VV10 exchange-correlation potential. According to many test calculations, a self-consistent treatment is rarely necessary for normal energy evaluations. The computation of the double integral given in eq. (3.19) requires using an integration grid, just like for normal exchange-correlation functionals. This grid is built similarly to the regular grids available in the ORCA, see Sec. *Numerical Integration*.

Basic usage

The non-self-consistent variant of NL can be invoked by the `!NL` keyword.

```
!B3LYP NL
```

The self-consistent variant by `!SCNL`.

```
!B3LYP SCNL
```

In the current version of ORCA, NL parameters are implemented for various functionals and HF:

Table 3.18: Methods with available NL parameters. If no values are given, the default parameters are used.

| Method | b | C | $NLScal$ |
|--------------------------------------|---------|------|----------|
| Hartree-Fock | | | |
| HF | 3.9 | | |
| (meta-)GGA | | | |
| BLYP | 4.0 | | |
| BP86 | 4.4 | | |
| PBE | 6.4 | | |
| REVPBE | 3.7 | | |
| RPBE | 4.0 | | |
| RPW86PBE | 5.9 | | |
| SCANfunc | 6.4 | | |
| RSCAN | 10.8 | | |
| R2SCAN | 12.3 | | |
| TPSS | 5.0 | | |
| B97M-V | 6.0 | 0.01 | |
| Hybrid | | | |
| B3LYP | 4.8 | | |
| B3LYP/G | 4.8 | | |
| B3P86 | 5.3 | | |
| B3PW91 | 4.5 | | |
| mPW1PW | 5.3 | | |
| PBE0 | 6.9 | | |
| PW1PW | 7.7 | | |
| PW6B95 | 9.0 | | |
| REVPBE0 | 4.3 | | |
| REVPBE38 | 4.7 | | |
| TPSSH | 5.2 | | |
| TPSS0 | 5.5 | | |
| R2SCANh (see [238]) | 11.9 | | |
| R2SCAN0 (see [238]) | 11.4 | | |
| R2SCAN50 (see [238]) | 10.8 | | |
| Range-separated hybrid | | | |
| LRC-PBE | 6.4 | | |
| WR2SCAN (see [248]) | 9.2612 | | |
| WB97X-V | 6.0 | 0.01 | |
| WB97M-V | 6.0 | 0.01 | |
| Double-hybrid | | | |
| B2PLYP (see [289]) | 7.8 | | |
| DSD-BLYP (see [290]) | 12.0 | | |
| DSD-PBEP86 (see [290]) | 14.2 | | |
| DSD-BLYP/2013 | 12.0 | | |
| DSD-PBEP86/2013 | 14.2 | | |
| PWPB95 (see [290]) | 11.1 | | |
| PR2SCAN50 (see [248]) | 10.9207 | | 0.75 |
| KPR2SCAN50 (see [248]) | 10.6723 | | 0.70 |
| PR2SCAN69 (see [248]) | 9.0691 | | 0.5556 |
| Range-separated double-hybrid | | | |
| WPR2SCAN50 (see [248]) | 9.4149 | | 0.65 |
| WB97M(2) | 10.0 | 0.01 | 0.65904 |

Important

Head-Gordon's ω B97X-V[244], B97M-V [211], and ω B97M-V[247] functionals are specially fitted variants that use the non-local VV10 kernel to capture London-dispersion effects. They employ the non-self-consistent NL

correction by default and no additional NL keywords have to be used! A self-consistent treatment can be invoked by the !SCNL keyword.

i Some notes on the NL corrections in ORCA

- Self-consistent calculations are not possible with the Hartree-Fock method.
- Analytic gradients are available, thus geometry optimizations with numerical frequencies can be computed.
- TD-DFT calculations are not yet available.
- Any calculations that require second derivatives of the NL functional are not yet possible. These are needed for real type perturbations in the CP-SCF solutions, e.g. for analytic Hessians, dipole polarizabilities, and double-hybrid gradients.
- Strictly imaginary perturbations such as NMR shielding and EPR g-tensors (both also with GIAOs), and hyperfine couplings are available.

Custom parameters

For NL, the parameters b , C , and a global scaling parameter of the NL correlation energy term can be adjusted via the %method block. The global scaling parameter NLScal should be used to scale the NL correction for methods that already partly cover long-range correlation like double-hybrids.

```
!B3LYP NL

%method
  NLb      5.0
  NLC      0.0083
  NLScal   1.0
end
```

i Note

In order to improve the scaling for larger systems, a distance cutoff was also introduced, controlled by the vdWdistTCUT flag in the %method block. The default value is 10 Å, so two grid points more than 10 Å away from each other do not correlate via the VV10 potential. This is already very conservative and has practically zero effect on the final energy, but can be altered if needed.

Example

In the following, we compute the energy of the argon dimer at a distance of 3.76 Å with the def2-TZVP basis set and using the B3LYP hybrid functional as an example. Here, we choose the non-self-consistent variant of the DFT-NL dispersion correction.

```
!B3LYP NL def2-TZVP

*xyz 0 1
  Ar  0.0 0.0 0.0
  Ar  0.0 0.0 3.76
*
```

The DFT-NL output for this example is shown below:

```
-----
                        post-SCF DFT-NL dispersion correction
-----
```

```
SCF Energy:      -1054.960511496
NL   Energy:      0.209449910
SC+NL Energy:    -1054.751061586
NL done in   :           0.7 sec
-----
```

```
[...]
```

```
-----
TOTAL SCF ENERGY
-----
```

```
[...]
```

```
DFT components:
N(Alpha)      :      17.999996328440 electrons
N(Beta)       :      17.999996328440 electrons
N(Total)      :      35.999992656880 electrons
E(X)          :      -47.880920358714 Eh
E(C)          :      -1.761923687962 Eh
NL Energy, E(C,NL) :      0.209449910227 Eh
E(XC)         :      -49.433394136450 Eh
```

Here, we find the B3LYP total energy (SCF Energy), the non-local contribution (NL Energy), and their sum (SC+NL Energy), which is the final total energy. In the DFT components section, the non-local contribution is listed separately (NL Energy, E(C,NL)) in order to be consistent with the !SCNL output.

Keywords

Table 3.19: Simple input keywords for the NL correction.

| Keyword | Description |
|---------|---|
| NL | Activates non-self-consistent NL correction |
| SCNL | Activates self-consistent NL correction |

Table 3.20: %method block input keywords for the NL correction.

| Keyword | Options | Description |
|-------------|---------|---|
| NLb | <real> | Controls b parameter for NL |
| NLC | <real> | Controls C parameter for NL |
| NLScal | <real> | Controls the global scaling parameter for NL |
| vdWdistTCUT | <real> | Controls the correlation distance cutoff for NL |

3.5 Semiempirical Methods

Semi-empirical quantum mechanical methods (SQM) are very useful to deal with many of very large systems. They use tailored approximations to reduce the computational cost drastically while keeping a reasonable level of accuracy for large parts of the chemical space. They are specifically useful to simulate reaction paths with *NEB-TS* or to define a SQM region in *multi-scale approaches* like ONIOM.

3.5.1 Neglect of Differential Overlap: CNDO, INDO, NDDO

A number of methods based on the “neglect of differential overlap” [291, 292] are currently implemented for energies and analytic gradients including AM1[293], PM3[294], MNDO[295, 296, 297] and other variants of CNDO, INDO, and NDDO. The methods MNDO, AM1, and PM3 are available for main group elements only and arise from the work of the Dewar group. They have been optimized to reproduce molecular structure and energetics. The older CNDO/1,2 and INDO/1,2 were developed by the Pople group [298, 299, 300, 301, 302, 303, 304, 305, 306] and were designed to roughly mimic minimal basis *ab initio* calculations. The methods of the Zerner group (ZINDO/1,2 and ZINDO/S) are closely related to the older methods but have been well parameterized for transition metals too [307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317]. ZINDO/1 (and less so ZINDO/2) are suitable for geometry optimization. ZINDO/S gives good results for electronically excited states at moderate configuration interaction levels and is also successful for the calculation of electron and spin distributions in large transition metal complexes [312, 313, 314, 315, 316, 317]. The ZNDDO/1,2 methods have been implemented into ORCA as straightforward extensions of the corresponding INDO methods without changing any parameter. However, the methods benefit from the somewhat more accurate representation of the Coulomb interaction within the NDDO approximation [318, 319]. The preliminary experience with these methods is that they are better than the corresponding INDO methods for calculation of transition metal complex structures but on the whole have also similar deficiencies. The analytic gradients are available for all of these methods and can be used to produce reasonable molecular structures at low computational cost or to get preliminary insight in the behavior of the system under investigation. However, do not try to use ZINDO/S (or CNDO/S) for structure optimizations - it does not make sense and will lead to disastrous results because there is no accurate representation of nuclear repulsion in these methods.

Tip

Even though ORCA has no implementation of newer MNDO methods like PM6 or RM1, they can be used via ORCA's *External Optimizer feature* and *MOPAC*. This way, *MOPAC* can be interfaced to provide ORCA with energies and gradients via the respective wrapper script that can be downloaded from the [ORCA external tools GitHub repository](#).

Basic Usage

The NDDO-based MNDO, AM1, and PM3 methods can be invoked via the simple input keywords !MNDO, !AM1 and !PM3.

```
!MNDO
```

```
!AM1
```

```
!PM3
```

These and other methods can be further defined via the %method block, e.g. INDO with the INDO/1 parameter set as:

```
%method
  METHOD      INDO
  VERSION    INDO_1
end
```

Or by directly invoking the respective parameterization as METHOD

```
%method
  METHOD    INDO_1
end
```

Note

If you want you can also combine semiempirical methods with MP2 (energies only). For example use `Method AM1` and `DoMP2 true`. It is questionable if this makes the results of semiempirical calculations any better but at least it is possible in ORCA.

Custom Parameters

You can change the built-in semiempirical parameters in a straightforward fashion via the `%ndoparas` block. To do so, any element specific parameters can be defined via an array "P":

```
%ndoparas
  P[<atomic number>,<parameter number>] <real>
end
```

For example, we can set the resonance integral parameter BS (s shell beta) with the number 25 for Carbon (Z = 6) to 20 eV:

```
%ndoparas
  P[6,25] 20
end
```

A full list of parameters that can be modified can be found below. Most of them will only be interesting for expert users. The most commonly modified parameters are the Beta's (number 25 through 28). Note that most programs require a negative number here. In ORCA the resonance integrals are defined in a way that makes the Beta's positive.

Table 3.21: List of parameters that can be set in the `%ndoparas` block.

| Parameter | Number | Description |
|--|--------|--|
| <i>Core Integrals (in eV):</i> | | |
| US | 0 | |
| UP | 1 | |
| UD | 2 | |
| UF | 3 | |
| <i>Basis Set Parameters (double-zeta):</i> | | |
| NSH | 4 | Number of shells for the element |
| NZS | 5 | Number of Slater type orbitals for the s shell |
| ZS1 | 6 | First exponent |
| ZS2 | 7 | Second exponent |
| CS1 | 8 | First contraction coefficient |
| CS2 | 9 | Second contraction coefficient |
| NZP | 10 | Number of Slater type orbitals for the p shell |
| ZP1 | 11 | |
| ZP2 | 12 | |
| CP1 | 13 | |
| CP2 | 14 | |
| NZD | 15 | Number of Slater type orbitals for the d shell |
| ZD1 | 16 | |
| ZD2 | 17 | |
| CD1 | 18 | |
| CD2 | 19 | |
| NZF | 20 | Number of Slater type orbitals for the f shell |
| ZF1 | 21 | |
| ZF2 | 22 | |
| CF1 | 23 | |
| CF2 | 24 | |

continues on next page

Table 3.21 – continued from previous page

| Parameter | Number | Description |
|--|--------|---|
| <i>Resonance integral parameters (in eV):</i> | | |
| BS | 25 | s shell beta |
| BP | 26 | p shell beta |
| BD | 27 | d shell beta |
| BF | 28 | f shell beta |
| <i>Number of electrons in the ground state:</i> | | |
| NEL | 29 | Total number of electrons (integer) |
| NS | 30 | Fractional occupation number of the s shell |
| NP | 31 | Fractional occupation number of the p shell |
| ND | 32 | Fractional occupation number of the d shell |
| NF | 33 | Fractional occupation number of the f shell |
| <i>One center repulsion (gamma) integrals (in eV):</i> | | |
| GSS | 34 | |
| GSP | 35 | |
| GSD | 36 | |
| GSF | 37 | |
| GPP | 38 | |
| GPD | 39 | |
| GPF | 40 | |
| GDD | 41 | |
| GDF | 42 | |
| GFF | 43 | |
| <i>Slater Condon parameters (in eV):</i> | | |
| F2PP | 44 | |
| F2PD | 45 | |
| F2DD | 46 | |
| F4DD | 47 | |
| G1SP | 48 | |
| G1PD | 49 | |
| G2SD | 50 | |
| G3PD | 51 | |
| R1SPPD | 52 | |
| R2SDPP | 53 | |
| R2SDDD | 54 | |
| <i>The nuclear repulsion parameters for Dewar type models:</i> | | |
| NR1 | 55 | |
| NR2 | 56 | |
| NR3 | 57 | |
| NR4 | 58 | |
| NR5 | 59 | |
| NR6 | 60 | |
| NR7 | 61 | |
| NR8 | 62 | |
| NR9 | 63 | |
| NR10 | 64 | |
| NR11 | 65 | |
| NR12 | 66 | |
| NR13 | 67 | |
| <i>Nuclear attraction/repulsion parameter for MNDO/d:</i> | | |
| RHO | 68 | |
| <i>Spin orbit coupling parameters:</i> | | |
| SOCP | 69 | SOC for the p shell |
| SOCD | 70 | SOC for the d shell |
| SOCF | 71 | SOC for the f shell |

The “molecular” parameters are also set in the %ndoparas block using “INTFA” (“interaction factors”);

```
%ndoparas
  INTFA[PP_PI] 0.585
# The interaction factors exist for
# ss_sigma
# sp_sigma
# sd_sigma
# pp_sigma
# pd_sigma
# dd_sigma
# pp_pi
```

(continues on next page)

(continued from previous page)

```
# pd_pi
# dd_pi
# dd_delta
# the parameter entering the Coulomb integrals
# in INDO/S
FGAMMA 1.2
end
```

Keywords

Table 3.22: Simple input keywords for the NDO-based methods.

| Keyword | Description |
|---------|-------------------|
| AM1 | Activates AM1 |
| PM3 | Activates PM3 |
| MNDO | Activates MNDO |
| ZINDO/1 | Activates ZINDO/1 |
| ZINDO/2 | Activates ZINDO/2 |
| ZINDO/S | Activates ZINDO/S |
| ZNDDO/1 | Activates ZNDDO/1 |
| ZNDDO/2 | Activates ZNDDO/2 |

Table 3.23: %method block input keywords for the NDO-based methods.

| Keyword | Options | Description |
|---------|---------|--|
| METHOD | AM1 | Activates AM1 |
| | PM3 | Activates PM3 |
| | MNDO | Activates MNDO |
| | CNDO_1 | Activates CNDO/1 |
| | CNDO_2 | Activates CNDO/2 |
| | CNDO_S | Activates CNDO/S |
| | INDO_1 | Activates INDO/1 |
| | INDO_2 | Activates INDO/2 |
| | INDO_S | Activates INDO/S |
| | ZINDO_1 | Activates ZINDO/1 |
| | ZINDO_2 | Activates ZINDO/2 |
| | ZINDO_S | Activates ZINDO/S |
| | ZNDDO_1 | Activates ZNDDO/1 |
| | ZNDDO_2 | Activates ZNDDO/2 |
| | CNDO | Activates CNDO, requires a VERSION to be set |
| | INDO | Activates INDO, requires a VERSION to be set |
| | NDDO | Activates NDDO, requires a VERSION to be set |
| VERSION | CNDO_1 | Sets the parameter set for CNDO to CNDO_1 |
| | CNDO_2 | Sets the parameter set for CNDO to CNDO_2 |
| | CNDO_S | Sets the parameter set for CNDO to CNDO_S |
| | INDO_1 | Sets the parameter set for INDO to INDO_1 |
| | INDO_2 | Sets the parameter set for INDO to INDO_2 |
| | INDO_S | Sets the parameter set for INDO to INDO_S |
| | ZINDO_1 | Sets the parameter set for INDO to ZINDO_1 |
| | ZINDO_2 | Sets the parameter set for INDO to ZINDO_2 |
| | ZINDO_S | Sets the parameter set for INDO to ZINDO_S |
| | ZNDDO_1 | Sets the parameter set for NDDO to ZNDDO_1 |
| | ZNDDO_2 | Sets the parameter set for NDDO to ZNDDO_2 |
| | MNDO | Sets the parameter set for NDDO to MNDO |
| | AM1 | Sets the parameter set for NDDO to AM1 |
| | PM3 | Sets the parameter set for NDDO to PM3 |

3.5.2 Extended Tight-Binding: GFN0-xTB, GFN-xTB, GFN2-xTB

ORCA is interfaced to the xtb program by Grimme and coworkers, allowing the user to request all kinds of calculations using the popular GFN0-xTB, GFN-xTB and GFN2-xTB Hamiltonians.[161, 320, 321] From the technical side, the user has to provide the executable provided by the Grimme group. The xtb program package can be obtained free of charge from <https://github.com/grimme-lab/xtb/releases> and detailed information on the usage of the xtb standalone program and its utilities can be found at <https://xtb-docs.readthedocs.io/en/latest/contents.html>. Only the file `bin/xtb` is used by ORCA. The user should copy this file into the directory where the other ORCA executables are located, and rename it as `otool_xtb`.

Warning

Please use the 6.7.1 version (or any later version) of xtb; older versions are not fully compatible with ORCA anymore or are missing features, for example it may not be possible to invoke the solvation model! Additionally, Windows users should copy `libiomp5md.dll` from the XTB directory to the ORCA directory.

The xTB method can be invoked via the GFN0-xTB, GFN-xTB, and GFN2-xTB keywords. The GFN-FF generic force-field can be invoked via GFN-FF:

```
! GFN0-xTB # Synonym: XTB0
```

```
! GFN-xTB # Synonym: XTB1
```

```
! GFN2-xTB # Synonym: XTB2
```

```
! GFN-FF # Synonym: XTBFF
```

The following methods can be used in conjunction with XTB:

- Single Point Energy
- Energy and Gradient
- Optimization, using all kinds of constraints, relaxed surface scans, etc.
- Nudged-Elastic Band calculations
- Numerical Frequency Calculations
- Intrinsic Reaction Coordinate
- Molecular Dynamics Calculations
- QM/MM calculations

Note

- XTB0 is a non-self-consistent tight-binding method, and as such, its accuracy is generally inferior to XTB1 and XTB2 (and sometimes even XTBFF), despite that it is a few times faster than XTB1 and XTB2. From our experience, we only recommend XTB0 when both XTB1 and XTB2 exhibit qualitative failures for the system of interest.
- Please note that XTB0, XTB1 and XTB2 can also be used for the initial path generation or for the calculation of an initial TS structure on XTB level, both as input for the subsequent NEB calculation on a higher level of theory. For more details, please consult section *Nudged Elastic Band Method*.

Solvation in xTB

Three implicit solvation models can be requested in xTB calculations:

1. the analytical linearized Poisson-Boltzmann (ALPB)[322] solvation model
2. the domain decomposition COSMO (ddCOSMO)[323]
3. the extended conductor-like polarizable continuum model (CPCM-X).[149]

These models can be requested via the simple keyword input:

```
! ALPB(solvent) # use ALPB
```

```
! DDCOSMO(solvent) # use ddCOSMO
```

```
! CPCM(X)(solvent) # use CPCM(X)
```

where `solvent` is any of the solvents in Table 3.24.

They can further be evoked via the `%xtb` block, here exemplarily shown for ALPB:

```
%xtb
  DOALPB true
  ALPBSOLVENT "water"
end
```

Table 3.24: List of available solvents for the different implicit solvation methods that can be used with `xtb`.

| Solvent | ALPB | ddCOSMO | CPCM-X |
|--|------|---------|--------|
| 1,2,4-trimethylbenzene | | | X |
| 1,2-dichloroethane | | | X |
| 1,4-dioxane / dioxane | X | X | |
| 1-butanol / butanol | | | X |
| 1-chlorohexane / chlorohexane | | | X |
| 1-decanol / decanol | | | X |
| 1-fluorooctane | | | X |
| 1-heptanol / heptanol | | | X |
| 1-hexanol / hexanol | | | X |
| 1-iodohexadecane / hexadecyl iodide | | | X |
| 1-nonanol / nonanol | | | X |
| 1-octanol / octanol | X | X | X |
| 1-pentanol / pentanol | | | X |
| 1-propanol / propanol | | | X |
| 2,2,4-trimethylpentane / isooctane | | | X |
| 2,6-dimethylpyridine | | | X |
| 2-butanol / secbutanol | | | X |
| 2-methoxyethanol / methoxyethanol | | | X |
| 2-methyl-1-propanol / isobutanol | | | X |
| 2-methylpyridine / 2methylpyridine | | | X |
| 2-propanol / isopropanol | | | X |
| 4-methyl-2-pentanone / 4methyl2pentanone | | | X |
| acetic acid / aceticacid | | | X |
| acetone | X | X | |
| acetonitrile / mecn / ch3cn | X | X | X |
| acetophenone | | | X |
| aniline | X | X | X |
| anisole | | | X |
| benzaldehyde | X | X | |
| benzene | X | X | X |
| benzonitrile | | | X |
| benzyl alcohol / benzylalcohol | | | X |
| bromobenzene | | | X |
| bromoethane | | | X |
| bromoform | | | X |
| butanone | | | X |
| butyl ethanoate / butyl acetate / butylacetate | | | X |
| n-butylbenzene / butylbenzene | | | X |

continues on next page

Table 3.24 – continued from previous page

| Solvent | ALPB | ddCOSMO | CPCM-X |
|--|------|---------|--------|
| sec-butylbenzene / secbutylbenzene | | | X |
| tert-butylbenzene / tbutylbenzene | | | X |
| carbon disulfide / carbondisulfide / cs2 | X | X | X |
| carbon tetrachloride / ccl4 | | | X |
| chlorobenzene | | | X |
| chloroform / chcl3 | X | X | X |
| conductor | | X | |
| m-cresol / mcresol | | | X |
| cyclohexane | | | X |
| cyclohexanone | | | X |
| decalin | | | X |
| n-decane / decane | | | X |
| dibromomethane | | | X |
| dibutylether | | | X |
| o-dichlorobenzene / odichlorobenzene | | | X |
| dichloromethane / ch2cl2 / dcm | X | X | X |
| diethyl ether / diethylether | X | X | X |
| diisopropyl ether / diisopropylether | | | X |
| n,n-dimethylacetamide / dimethylacetamide | | | X |
| n,n-dimethylformamide / dimethylformamide / dmf | X | X | X |
| dimethylsulfoxide / dmsol | X | X | X |
| diphenylether | | | X |
| n-dodecane / dodecane | | | X |
| ethanol | X | X | X |
| ethyl acetate / ethylacetate / ethanoate | X | X | X |
| ethyl phenyl ether / ethoxybenzene | | | X |
| ethylbenzene | | | X |
| fluorobenzene | | | X |
| furan / furane | X | X | |
| n-heptane / heptane | | | X |
| n-hexadecane / hexadecane | X | X | X |
| n-hexane / hexane | X | X | X |
| iodobenzene | | | X |
| isopropylbenzene | | | X |
| p-isopropyltoluene / isopropyltoluene | | | X |
| mesitylene | | | X |
| methanol | X | X | X |
| n-methylformamide / methylformamide | | | X |
| nitrobenzene / phno2 | | | X |
| nitroethane | | | X |
| nitromethane / meno2 | X | X | X |
| o-nitrotoluene / onitrotoluene | | | X |
| n-nonane / nonane | | | X |
| n-octane / octane | | | X |
| n-pentadecane / pentadecane | | | X |
| octanol(wet) / wetoctanol / woctanol | X | X | |
| n-pentane / pentane | | | X |
| perfluorobenzene / hexafluorobenzene | | | X |
| phenol | X | X | |
| pyridine | | | X |
| tetrachloroethene / c2cl4 | | | X |
| tetrahydrofuran / thf | X | X | X |
| tetrahydrothiophene-s,s-dioxide / tetrahydrothiophenedioxide / sulfolane | | | X |
| tetralin | | | X |
| toluene | X | X | X |
| tributylphosphate | | | X |
| triethylamine | | | X |
| n-undecane / undecane | | | X |
| water / h2o | X | X | X |
| xylene | | | X |

Note

- If jobs are run over several nodes, the number of cores used by the XTB tool might be lower than requested via the pal keyword.

Keywords

Table 3.25: Simple input keywords for the extended tight-binding methods.

| Keyword | Description |
|---------------------|--|
| XTB | Activates GFN2-xTB |
| XTB0 | Activates GFN0-xTB |
| GFN0-XTB | Activates GFN0-xTB |
| XTB1 | Activates GFN1-xTB |
| GFN1-XTB | Activates GFN1-xTB |
| XTB2 | Activates GFN2-xTB |
| GFN2-XTB | Activates GFN2-xTB |
| XTBFF | Activates GFN-FF |
| GFN-FF | Activates GFN-FF |
| ALPB (<solvent>) | Activates ALPB solvation for a <solvent>, e.g. ALPB(Water) |
| DDCOSMO (<solvent>) | Activates DDCOSMO solvation for a <solvent>, e.g. DDCOSMO(Water) |
| CPCMX (<solvent>) | Activates CPCMX solvation for a <solvent>, e.g. CPCMX(Water) |

Table 3.26: %xtb block input keywords for the extended tight-binding methods.

| Keyword | Options | Description |
|-----------------|-------------|--|
| XTBINPUTSTRING | "<string>" | Parses string to the otool_xtb call |
| XTBINPUTSTRING2 | "<string>" | Parses string to the otool_xtb call |
| NPROCS | <integer> | Controls number of processors for the otool_xtb run, default is ORCA's PAL setting |
| MAXCORE | <integer> | Memory in MB reserved for the xtb run, default is ORCA's maxcore setting |
| ACCURACY | <real> | Accuracy setting for --acc in otool_xtb, default is ORCA's accuracy x 1.e6 |
| DOALPB | true/false | Activates ALPB solvation in xtb run |
| ALPBSOLVENT | "<solvent>" | Controls the <solvent> used for ALPB |
| DODDCOSMO | true/false | Activates DDCOSMO solvation in xtb run |
| DDCOSMOSOLVENT | "<solvent>" | Controls the <solvent> used for DDCOSMO |
| EPSILON | <real> | Controls the dielectric constant, only for ddCOSMO |
| DOCPCMX | true/false | Activates CPCMX solvation in xtb run |
| CPCMXSOLVENT | "<solvent>" | Controls the <solvent> used in CPCMX |
| ETEMP | <real> | Controls electronic temperature --etemp in otool_xtb |

3.5.3 Native GFN-xTB and GFN2-xTB

Added in version 6.1.0: Native implementation of GFN1-xTB and GFN2-xTB

ORCA features its own native implementation of GFN-xTB and GFN2-xTB. Unlike the interface to the xtb program, the native xTB implementation can use the *SCF* and *plotting* infrastructure.

The native xTB methods can be invoked via the !Native-GFN-xTB, !Native-GFN2-xTB simple keywords:

```
! Native-GFN-xTB # Synonym: Native-GFN1-xTB, Native-XTB1
```

```
! Native-GFN2-xTB # Synonym: Native-XTB2
```

Warning

For a faster SCF restart based on atomic charges and multipole moments, the native xTB methods write a `basename.xtbw` file. If such an `.xtbw` file is present, it will be used for the restart instead of the `.gbw` file.

Custom parameterization

The native implementations feature the use of custom semi-empirical parameters by reading a JSON parameter file with the name `basename.xtb.json`. This file has the same format as the JSON file in the `dxtb` program ([dxtb documentation](#)), and can be generated by writing the internal parameters. Reading and writing of semi-empirical parameters can be turned on and off via the `%method` block:

```
%method
  XTBParamFile "orca.xtb.json" # set parameter filename to orca.xtb.json
  WriteXTBParam true           # if true: write parameters to orca.xtb.json file
  WriteXTBParamDry false      # if true: write parameters to orca.xtb.json file and
  ↪exit
  ReadXTBParam false          # if true: read parameters from orca.xtb.json file
end
```

By default the parameter file will be named according to the basename of your calculation (`basename.xtb.json`), but other names can be chosen with the `XTBParamFile` option in the method block. For a dry run that just generates the parameter file from default parameters set `WriteXTBParamDry true`.

Warning

Please note that the native xTB implementation does not allow changes in the basis.

Spin-polarized native xTB

The energy expressions of the original GFNn-xTB methods are spin-independent. Therefore, calculations are typically performed using RHF/RKS-type wavefunctions, even for open-shell systems. Since spin does not enter the energy expression, electronic states with higher spin multiplicity will always be higher in energy compared to the respective low-spin states. This shortcoming is addressed by the spin-polarized xTB methods. The spin-polarized variants [324], termed `spGFN1-xTB` and `spGFN2-xTB`, employ a simple on-site spin-dependent term, which can stabilize high-spin states and allows for screening of potential high-spin configurations using xTB methods.

The native spin-polarized xTB methods can be invoked via the `!Native-spGFN1-xTB`, `!Native-spGFN2-xTB` simple keywords:

```
! Native-spGFN1-xTB # Synonym: Native-spGFN-xTB, Native-spXTB1
```

```
! Native-spGFN2-xTB # Synonym: Native-spXTB2
```

Note that the spin-polarization term will only affect high-spin states. The computational cost increases compared to RHF/RKS-type calculations, as the underlying UHF/UKS-type wavefunctions require two Fock matrix diagonalizations instead of one—approximately doubling the cost.

SCF with native xTB

The native xTB methods, by default, employ special SCF settings that mimic the SCF procedure of the `xtb` program and override all other ORCA SCF settings. These special settings take advantage of xTB-specific features, most notably, that the Fock matrix can be constructed from atomic charges and multipole moments alone. Due to these shortcuts, the special SCF settings are typically faster for xTB methods than the default ORCA SCF. However, the default SCF is more robust and may succeed in converging problematic cases where the `xtb` SCF fails.

To use the standard ORCA SCF infrastructure instead, set `UseXTBMixer` to `false` in the `%scf` block:

```
%scf
  UseXTBMixer false      # if true (default): special xTB SCF is used
end
```

With the ORCA default SCF infrastructure, challenging electronic structures can often be converged, but the results should be used with caution. In cases where SCF convergence has to be forced, the GFNn-xTB Hamiltonians are often not adequate to describe the system at hand. In such cases, Mulliken charges should be checked for sanity, and it might be helpful to slightly adjust the element-wise pair parameters via the parameter file described above.

Warning

Note that all SCF iterations are direct with xTB methods.

FOD with native xTB

To better account for static correlation, the xTB methods use *finite temperature SCF* by default, with an electronic temperature of 300 K. The corresponding *fractional occupation number weighted density (FOD)* is not printed by default, since 300 K is a too low temperature to use it as a diagnostic tool. The FOD printout can be enabled by setting XTBFOD to true in the %scf input block. The electronic temperature can be controlled—just as for other methods—via the SmearTemp keyword:

```
%scf
  XTBFOD true      # if true: Enable FOD printout and add FOD to density container
  SmearTemp 5000   # 5000 K is typically used for FOD plots with xTB methods
end
```

Solvation in native xTB

The native implementation of xTB includes the ALPB solvation model with the same solvents available as in the xtb program (*see above*). It is controlled with the same keywords as the external ALPB via the simple keyword input:

```
! ALPB(solvent) # use ALPB
```

In addition, the native implementations can be used with other *solvation models* available in ORCA, but one should consider that the CPCM and SMD models are quite expensive compared to an xTB calculation, and therefore the faster ALPB solvation model is recommended.

Keywords

Table 3.27: Simple input keywords for the native extended tight-binding methods.

| Keyword | Description |
|-------------------|--|
| NATIVE-XTB | Activates native GFN2-xTB |
| NATIVE-spXTB | Activates native spGFN2-xTB |
| NATIVE-XTB1 | Activates native GFN1-xTB |
| NATIVE-spXTB1 | Activates native spGFN1-xTB |
| NATIVE-GFN-XTB | Activates native GFN1-xTB |
| NATIVE-spGFN-XTB | Activates native spGFN1-xTB |
| NATIVE-GFN1-XTB | Activates native GFN1-xTB |
| NATIVE-spGFN1-XTB | Activates native spGFN1-xTB |
| NATIVE-XTB2 | Activates native GFN2-xTB |
| NATIVE-spXTB2 | Activates native spGFN2-xTB |
| NATIVE-GFN2-XTB | Activates native GFN2-xTB |
| NATIVE-spGFN2-XTB | Activates native spGFN2-xTB |
| ALPB(<solvent>) | Activates ALPB solvation for a <solvent>, e.g. ALPB(Water) |

Table 3.28: %method block input keywords for the native extended tight-binding methods.

| Keyword | Options | Description |
|------------------|---------------------|--|
| WRITEXTBPARAM | true/false | Write semi-empirical parameters to xtb.json file |
| WRITEXTBPARAMDRY | true/false | Write semi-empirical parameters to xtb.json file and abort |
| READXTBPARAM | true/false | Read semi-empirical parameters from xtb.json file |
| XTBPARAMFILE | "filename.xtb.json" | Specifies xtb.json filename to "filename.xtb.json" |

Table 3.29: %scf block input keywords for the native extended tight-binding methods.

| Keyword | Options | Description |
|-------------|------------|---|
| XTBFOD | true/false | Enable FOD printout |
| UseXTBMixer | true/false | Use special SCF settings similar to the ones in xtb |

3.6 Composite Methods (3c methods)

Composite *HF* and *DFT* methods utilize relatively small tailored *basis sets* and special corrections to achieve high accuracies at a fraction of the computational cost of a calculation approaching the basis set limit. The most prominent composite methods are the so-called “3c” methods by Grimme and co-workers.

3.6.1 HF-3c

HF-3c is a fast *Hartree-Fock* based method developed for computation of structures, vibrational frequencies and non-covalent interaction energies in huge molecular systems [325]. The starting point for evaluating the electronic energy is a standard HF calculation with a small Gaussian AO basis set. The used so-called MINIX basis set consists of different sets of basis functions for different groups of atoms as shown in table Table 3.30. In ORCA, HF-3c is available for all elements up to Pu ($Z = 94$).

Table 3.30: Composition of the MINIX basis set.

| element | basis |
|------------|------------------------|
| H-He, B-Ne | MINIS |
| Li-Be | MINIS+1(p) |
| Na-Mg | MINIS+1(p) |
| Al-Ar | MINIS+1(d) |
| K-Zn | SV |
| Ga-Kr | SVP |
| Rb-Rn | def2-SVP with def-ECPs |
| Fr-Lr | def-SVP with def-ECPs |

Three terms are added to correct the HF energy $E_{\text{tot}}^{\text{HF/MINIX}}$ in order to include London dispersion interactions, to account for the BSSE and to correct for basis set deficiencies, i.e. overestimated bond lengths. The corrected total energy is therefore calculated as

$$E_{\text{tot}}^{\text{HF-3c}} = E_{\text{tot}}^{\text{HF/MINIX}} + E_{\text{disp}}^{\text{D3(BJ)}} + E_{\text{BSSE}}^{\text{gCP}} + E_{\text{SRB}}. \quad (3.23)$$

The first correction term $E_{\text{disp}}^{\text{D3(BJ)}}$ is the atom-pair wise London dispersion energy from the *D3 dispersion correction* scheme[272] applying Becke-Johnson (BJ) damping [277, 278, 279] (see section *Grimme’s DFT-D3 and DFT-D4*). The second term $E_{\text{BSSE}}^{\text{gCP}}$ denotes the *geometrical counterpoise (gCP) correction* [124] to treat the BSSE (see section *Geometrical Counterpoise Correction (gCP)*). For the HF-3c method, the three usual D3 parameters s_8 , a_1 and a_2 were re-fitted using reference interaction energies of the complexes of the S66 test set [285]. This results in $s_8 = 0.8777$, $a_1 = 0.4171$ and $a_2 = 2.9149$. The parameter s_6 was set to unity as usual to enforce the correct asymptotic limit and the gCP correction was already applied in this fitting step.

The last term E_{SRB} is a short-ranged correction to deal with basis set deficiencies which occur when using small or minimal basis sets. It corrects for systematically overestimated covalent bond lengths for electronegative elements

and is calculated as a sum over all atom pairs:

$$E_{\text{SRB}} = -s \sum_A^{\text{Atoms}} \sum_{B \neq A}^{\text{Atoms}} (Z_A Z_B)^{3/2} \exp(-\gamma (R_{AB}^{0,\text{D3}})^{3/4} R_{AB})$$

Here, $R_{AB}^{0,\text{D3}}$ are the default cut-off radii as determined *ab initio* for the D3 scheme [272] and Z_A , Z_B are the nuclear charges. This correction is applied for all elements up to argon. The empirical fitting parameters $s = 0.03$ and $\gamma = 0.7$ were determined to produce vanishing HF-3c total atomic forces for B3LYP-D3(BJ)/def2-TZVPP equilibrium structures of 107 small organic molecules. More details can be found in the original publication [325].

The HF-3c method can only be invoked with a simple keyword:

```
! HF-3c
```

`! HF-3c` is a compound keyword and equals `! HF MINIX D3BJ GCP (HF/MINIX) PATOM`, hence no basis set etc. has to be specified. The `PATOM` guess is selected since the grid construction for the default guess can take more time than an actual SCF step. The guess can only be overwritten manually in the `%method` section.

The default mode for the integral handling is set to `Conventional`. The storing of the two-electron integrals on disk or in memory if possible leads to large computational savings. In case one want to use the `Direct` mode, this has to be specified in the `%scf` input section:

```
%scf
  SCFmode Direct
end
```

The output gives the used parameters and the correction itself for D3 and gCP separately. As the SRB correction is also calculated with the `otool_gcp`, the results are given in the gCP output section. The `Total correction to HF/MINIX` is the sum of all three corrections (D3, gCP and SRB) and `FINAL SINGLE POINT ENERGY` is the total HF-3c energy as given in equation (3.23).

```
-----
                        DFT DISPERSION CORRECTION
                        DFTD3 V2.1  Rev 6
                        USING Becke-Johnson damping
-----
The default Hartree-Fock  is recognized
Active option DFTDOPT          ...          4

molecular C6(AA) [au] = 1689.256597

      DFT-D V3
parameters
using HF/MINIX parameters
s6 scaling factor      :      1.0000
a1 scaling factor      :      0.4171
s8 scaling factor      :      0.8777
a2 scaling factor      :      2.9149
ad hoc parameters k1-k3 :      16.0000      1.3333      -4.0000

Edisp/kcal,au: -32.163184627631  -0.051255291794
E6  /kcal   : -18.007221978
E8  /kcal   : -14.155962649
% E8       :  44.012938437

-----
Dispersion correction          -0.051255292
-----
```

(continues on next page)

(continued from previous page)

```
-----
g C P - geometrical counterpoise correction
-----
```

```
Method: hf/minix
```

```
Parameters:  sigma      eta      alpha      beta
              0.1290    1.1526    1.1549    1.1763
Egcp:        0.0723150636 a.u.
Ebas:        -0.0636976872 a.u.
```

```
-----
gCP+bas correction      0.008617376
-----
```

```
-----
Total correction to HF/MINIX      -0.042637915
-----
```

```
-----
FINAL SINGLE POINT ENERGY      -163.002895262171
-----
```

For the elements up to Xe, the default initial guess is a Hückel guess. Beyond Xe, the guess mode is changed to HCore since no Hückel parameters for the respective ECP bases are available and other models are not implemented at the moment. For calculations with only lighter elements and therefore no ECPs, the ECP printouts in the output file can be ignored.

3.6.2 B97-3c

B97-3c[326] is another composite DFT method designed for thermochemistry, structures, and noncovalent interactions specifically also for transition metal chemistry and other stronger correlated systems. It is based on the B97 GGA including the *D3(BJ) dispersion correction* with three-body contribution, a short range bond length correction, and a modified, stripped-down triple- ζ basis termed def2-mTZVP, the computational cost of this method is between that of HF-3c and *PBEh-3c* (for large systems roughly two times more expensive than *HF-3c*). In ORCA, B97-3c is available for all elements up to Pu ($Z = 94$).

B97-3c can be invoked via simple input keyword:

```
! B97-3c
```

3.6.3 r^2 SCAN-3c

The r^2 SCAN-3c composite method[221] is available as robust “Swiss army knife” electronic structure method for thermochemistry, geometries and non-covalent interactions and has shown in preliminary tests consistent performance for both open and closed shell transition metal complexes. It is based on the r^2 SCAN[216] meta-GGA combined with the *D4 dispersion correction*[327] and the *geometrical counter poise-correction*[124]. The modified triple- ζ basis set, def2-mTZVPP, is larger and more consistent for the light main-group elements and almost as computationally efficient as the def2-mTZVP basis set of *B97-3c*. The computational cost of r^2 SCAN-3c is slightly larger than *B97-3c*. In ORCA, r^2 SCAN-3c is available for all elements up to Lr ($Z = 103$). It is invoked with the simple keyword

```
! r2SCAN-3c
```

3.6.4 PBEh-3c

PBEh-3c is a highly efficient electronic structure approach performing particularly well in the optimization of geometries and for interaction energies of non-covalent complexes.[239] Here, a global hybrid variant of the Perdew-Burke-Ernzerhof (PBE) functional with a relatively large amount of non-local Fock-exchange (42%) is employed with a valence-double-zeta Gaussian AO basis set (def2-mSVP). Basis set superposition errors (BSSE) and London dispersion effects are accounted for by the *gCP* and *D3* schemes, respectively (see above). The basis set is constructed such that:

Table 3.31: Composition of the def2-mSVP basis set.

| element | basis |
|-------------|---|
| H | def2-SV(P) (α scaled by 1.2) |
| He | def2-SVP(-p) |
| Li-Be,Na-Kr | def2-SV(P) |
| B,Ne | Ahlrichs' DZ + Polarization from def2-SVP |
| C-F | Ahlrichs' DZ + Polarization from 6-31G* |
| Rb-Rn | def2-SVP with def2 ECPs |
| Fr-Lr | def-SVP with def-ECPs |

For inter- and intramolecular BSSE the *gCP* expression from Eq. (2.25) is used but with a damping function (similar to the zero-damping in Eq. (3.14)). This damping improves the thermochemistry of the method significantly compared with the non-damped version. London dispersion effects are accounted for by the *DFT-D3* (BJ-damping) scheme including the three-body term. Compared to the related HF-3c approach, the PBEh-3c is somewhat more costly, however, it yields much better geometries. These are roughly of MP2-quality (or even better for non-covalent structures) but may be computed at much lower cost. Due to the moderate amount of non-local Fock exchange, the method is less prone to self-interaction errors (as in GGAs) but still applicable in cases when Hartree-Fock fails (strongly correlated systems). In ORCA, PBEh-3c is available for all elements up to Pu ($Z = 94$).

The PBEh-3c method may be invoked with the simple keyword:

```
! PBEh-3c
```

Identical to HF-3c, the default initial guess for all elements up to Xe is a Hückel guess. Beyond Xe, the guess mode is changed to HCore. For calculations with only lighter elements and therefore no ECPs, the ECP printouts in the output file can be ignored.

3.6.5 B3LYP-3c

B3LYP-3c is a method combination introduced by Grimme and co-workers to efficiently calculate gas-phase infrared spectra.[240] It combines the standard B3LYP functional with the *D3(BJ)-ATM dispersion correction*, a def2-mSVP basis set, and a *geometrical counterpoise correction*. In ORCA, B3LYP-3c is available for all elements up to Pu ($Z = 94$). B3LYP-3c can be invoked via simple input keyword:

```
! B3LYP-3c
```

or it can be constructed manually from its components via:

```
!B3LYP D3BJ GCP (DFT/SV(P)) def2-mSVP ABC
```

3.6.6 ω B97X-3c

The ω B97X-3c composite method[49] is based on the ω B97X-V functional and combines a tailored and molecule-optimized polarized valence double- ζ (vDZP) basis set and a specifically adapted D4 dispersion correction. The vDZP basis set employs large-core ECPs and shows only very small basis set superposition and incompleteness errors compared to conventional double- ζ basis sets. In thorough tests on standard benchmarks sets, the ω B97X-3c method was shown to be on par with well-performing hybrid DFT methods in a quadruple- ζ basis set at a fraction of their computational cost. ω B97X-3c is consistently available for all elements up to Rn ($Z = 1-86$).

It is invoked with the simple keyword:

```
! wB97X-3c
```

The vDZP basis set alone is utilized as follows (note that the corresponding large-core ECPs are called automatically):

```
! vDZP
```

3.6.7 Keywords

Table 3.32: Simple input keywords for the 3c composite methods.

| Keyword | Description |
|-----------|--|
| HF-3C | Invokes the <i>HF-3c</i> method |
| B97-3C | Invokes the <i>B97-3c</i> method |
| R2SCAN-3C | Invokes the <i>r²SCAN-3c</i> method |
| PBEH-3C | Invokes the <i>PBEh-3c</i> method |
| B3LYP-3C | Invokes the <i>B3LYP-3c</i> method |
| WB97X-3C | Invokes the <i>ωB97X-3c</i> method |

3.7 Analytic Density Functional Theory (ADFT)

The method that has come to be known as “analytic density functional theory” (ADFT) is a variant of DFT that avoids the very common use of numerical integration. Early approaches to a grid DFT were relying on diagonalizing the density matrix and applying the DFT functional to the eigenvalues of the density followed by back transformation. These methods work but have been abandoned in favor of numerical integration.

ORCA features an alternative approach, that is, however, still limited to the most elementary of all density functionals, namely the Hartree-Fock Slater (HFS) or, in a slightly parameterized form, the X-Alpha method. The method is based on fitting the Coulomb and exchange potentials to separate auxiliary basis sets. In the case of the Coulomb interaction, this is very well known in form of the RI-J approximation. For the exchange, the ADFT construction is based on a related but more tricky development given the nonlinear dependency of the Slater-exchange on the density.

The exchange energy is:

$$E_X^\sigma = -\frac{3}{4} \left(\frac{6}{\pi} \right)^{\frac{1}{3}} \int \rho_\sigma^{4/3}(\mathbf{r})$$

Leading to the exchange potential:

$$V_X^\sigma(\mathbf{r}) = -\left(\frac{6}{\pi} \right)^{\frac{1}{3}} \rho_\sigma^{1/3}(\mathbf{r})$$

$$C_X = -\left(\frac{6}{\pi} \right)^{\frac{1}{3}} = -1.24 \dots$$