

It is also possible to approximate light atoms a, b (below a certain atomic number) as non-relativistic:

$$\begin{aligned}\mathbf{X}_a &= \mathbf{R}_a = \mathbf{I} \\ \mathbf{W}_{ab} &= \mathbf{W}_{aB} = \mathbf{0} \\ \mathbf{h}_{ab}^+ &= \mathbf{V}_{ab} + \mathbf{T}_{ab} \\ \mathbf{h}_{aB}^+ &= (\mathbf{V}_{aB} + \mathbf{T}_{aB}\mathbf{X}_B) \mathbf{R}_B\end{aligned}$$

Unlike the one center approximation (which is also available for X2C), all nuclei are included in the potential operator \hat{V} .

The DLU approximation can be requested via the simple input keyword `!DLU-X2C`. Additional options are available in the `%rel` block (see also [Section 2.12.9](#)):

```
%rel
  DLU          false # default - not used
               true  # turn DLU on
  LightAtomThresh 0 # (default) highest atomic number treated as non-relativistic
end
```

X2C Derivatives and Properties

As discussed in [Section 2.12.7](#), when computing properties with relativistic methods, derivatives need to be taken of the correct Hamiltonian, namely \mathbf{h}^+ (eq (2.50)) in the X2C case. These include contributions due to the derivatives of \mathbf{R} and \mathbf{X} , which are often small. Therefore, it is possible to neglect them and save some computational time via the following option:

```
%rel
  PictureChange 2 # compute the full relativistic Hamiltonian derivative
                 1 # neglect derivatives of R and X
                 0 # (default) use the non-relativistic operator
end
```

The same setting is applied to all properties for which the X2C correction is implemented. Currently, these are: geometric gradients, electric dipoles, quadrupoles, and polarizabilities, electric field gradients, EPR hyperfine couplings, and NMR shieldings and spin–spin couplings. For the Hessian, the X2C correction is implemented in a semi-numeric fashion. The DLU approximation is applied throughout, if requested, and reduces the computational effort dramatically. Note that for magnetic properties, the restricted magnetic balance (RMB) for the small component basis functions is used whenever GIAOs are requested (e.g. NMR shielding), while the restricted kinetic balance (RKB) is used otherwise (e.g. NMR coupling). The spin–orbit coupling integrals used for various properties only include a relativistic correction to the one-electron term, as is the case for DKH.

For second derivative properties whose number is proportional to the number of atoms, e.g. the DSO term of NMR couplings, first derivatives of various intermediate quantities required for the full X2C second derivative are stored on disk. The storage requirements can be reduced via the `StorageLevel` keyword and the missing intermediates will be recomputed on-the-fly, which of course increases the computation time.

```
%rel
  StorageLevel 0 # do not store anything - not always available
               1 # store integral derivatives
               2 # 1 + derivatives of R and X (default, minimum for DLU)
               3 # 2 + derivatives of L
               4 # 3 + derivatives of S-tilde
               5 # 4 + derivatives of D and M
end
```

2.12.6 Basis Sets in Relativistic Calculations

The different scalar relativistic potentials have different shapes in the core region. Consequently, each one of them requires specialized all electron basis sets that are optimized for the Hamiltonian at hand. Such basis sets have been designed, both as DKH and ZORA reconstructions of the *Ahlrichs* and *Karlsruhe* basis sets for elements up to Kr (for which the original non-relativistic versions are all-electron), and as purpose-built segmented all-electron relativistically contracted (SARC) basis sets for elements beyond Kr.[50, 51, 52, 53, 54, 55] For X2C calculations, the “x2c-XVPall” basis sets and their variants are available.[57, 58] All available built-in choices are listed in the section *Relativistic Basis Sets*.

An uncontracted all-electron basis set of sufficient size will always work. Likewise, uncontracted fitting basis sets in all forms of RI calculations are always appropriate. Use the *!Decontract keyword* to decontract the chosen basis set and make it suitable for any relativistic Hamiltonian, as well as comparisons between them.

If large, uncontracted basis sets are used in scalar relativistic calculations, there is a distinct danger of variational collapse. This behavior is related to the fact that the relativistic orbitals will diverge for a point nucleus. ORCA features the Gaussian finite nucleus model of Dyall and Visscher for DKH and X2C. We recommend to always use this feature (*FiniteNuc*) in relativistic calculations (see [Section 2.12.8](#)).

Given the fact relativistic all-electron calculations on heavy element compounds feature very steep core basis functions, numeric integration, such as in DFT and COSX, may be challenging. ORCA features automatic procedures that adapt the integration grids for the presence of steep basis functions. However, in case you experience strange results, the numeric integration is one potential source of problem. The cure is to go to larger integration grids and, in particular, increase the *radial integration accuracy* (*IntAcc*).

2.12.7 Picture-Change Effects

Irrespective of which Hamiltonian has been used in the determination of the wave function, the calculation of properties requires some special care. This can be understood in two ways: First of all, we changed from the ordinary Schrödinger Hamiltonian to a more complicated Hamiltonian. As properties are defined as derivatives of the energy, it is clear that a new Hamiltonian will yield a new expression for the energy and thus a new and different expression for the property in question. Another way of seeing this is that through the transformation U , we changed not only the Hamiltonian but also the wave function. To obtain the property at hand as the expectation value of the property operator with the wave function, we have to make sure that property operator and wave function are actually given in the same space. This is done through a transformation of either the property operator or the wave function.

In any case, the difference between the non-relativistic and (quasi) relativistic property operator, evaluated over the (quasi) relativistic wave function, is called the *picture-change effect*. From what was said above, this is clearly not a physical effect. It describes how consistently the quasi relativistic calculation is carried out. A fully consistent calculation requires the determination of the wave function on the (quasi) relativistic level as well, as the use of the (quasi) relativistic property operator. This is obtained by enabling the *PictureChange* option in the *%rel* block:

```
%rel PictureChange 1 end # or 2 - see below
```

It may be that the (quasi) relativistic and non-relativistic property operator do produce similar results. In this case, a calculation with picture-change turned off (*PictureChange=0*) may be a good approximation. This is often the case for valence shell-dominated properties, such as the electric dipole and polarizability. On the other hand, property calculations involving operators that carry inverse powers of the electron-nucleus distance can be wildly inaccurate if picture change effects are ignored. It is therefore highly recommended to turn on picture-change in all (quasi) relativistic property calculations!

For DKH2, the fully consistent picture-change effects are obtained using the same transformation order for the property operator as for the one-electron Hamiltonian, i.e. setting *PictureChange=2*, while with *PictureChange=1* only first-order changes on the property operators are taken into account, which reduces the computational cost. However, since this is in no way a significant reduction, this choice is not recommended.

A similar argument applies to X2C, where *PictureChange=2* is the formally correct approach and *PictureChange=1* is an approximate treatment (see *X2C Derivatives and Properties*).

By default, ORCA tries to choose the most appropriate value of the *PictureChange* variable depending on the requested properties, as described below (and issues a warning at the start of the output), but explicit user input is

always respected.

- For geometric perturbations with DKH or ZORA the *one-center approximation* is activated and picture-change corrections are not needed. With X2C, `PictureChange=1` is chosen, because it significantly reduces the computational cost.
- For other properties, in particular core-sensitive ones (i.e. in EPR/NMR), the full picture-change correction is applied.
- Certain combinations of relativistic Hamiltonian and property operator are not implemented, in which case ORCA prints a warning, or aborts entirely if picture change effects are expected to be significant.

2.12.8 Finite Nucleus Model

Composite particles like nuclei have, as opposed to elementary particles, a certain spatial extent. While the point-charge approximation for nuclei is in general very good in nonrelativistic calculations, in relativistic calculations it might lead to non-negligible errors. A finite-nucleus model is implemented in ORCA – the Gaussian nucleus model of Ref. [138] – and is available for DKH and X2C (but not ZORA) calculations in the ORCA program package. It is accessible from the `%rel` block via

```
%rel FiniteNuc true end
```

By default, this keyword is set to `false`. If the keyword is set to `true`, finite-nucleus effects are considered in the following integrals:

- nucleus potential V and its nuclear derivatives;
- relativistic potential integral $V^{(p)}$ in DKH and X2C;
- one-electron part of the spin-orbit (SOC) integrals;
- electric-field gradient (EFG);
- contact density integral;
- Fermi-contact and spin-dipole terms of the EPR hyperfine coupling; and NMR indirect spin-spin coupling tensors;
- nucleus-orbit coupling integral;
- angular-momentum integral;
- NMR shielding integrals over GIAOs.

2.12.9 Keywords

Table 2.54: Simple input keywords for relativistic methods.

Keyword	Description
DKH1	Activate the first-order DKH Hamiltonian
DKH2, DKH	Activate the second-order DKH Hamiltonian
ZORA	Activate the ZORA Hamiltonian
IORA	Activate the IORA Hamiltonian
X2C	Activate the X2C Hamiltonian
DLU-X2C	Activate the X2C Hamiltonian in the DLU approximation
RelDLU	Enable the <i>DLU approximation</i> (X2C only)
Rel1C	Enable the <i>one-center approximation</i>
RelFull	Disable the one-center and DLU approximations

Table 2.55: `%rel` block input keywords relativistic methods.

Keyword	Options	Description
Method	DKH ZORA IORA X2C	Scalar relativistic Hamiltonian
ModelPot	<VeN>, <VC>, <VXa>, <VLDA>, <VPC>	Flags for model potential (1: included, 0: not included). <VeN>: nuclear attraction; <VC>: Coulomb (ZORA/IOA only); <VXa>: Xalpha (ZORA/IOA only); <VLDA>: VWN-5 correlation (ZORA/IOA only); <VPC>: point charges (X2C only)
ModelDens	rhoDKH rhoZORA rhoHF	Fitted atomic density used for Coulomb model potential and for PModel guess
OneCenter	false	If true, use the <i>one-center approximation</i>
C	137.0359895	Speed of light used (in a.u.). Synonyms: VELIT, VELOCITY
PictureChange	0, 1, 2	<i>Picture change</i> for properties. 0: none; 1: enable; 2: full (DKH/X2C). Default: determine automatically
FiniteNuc	false	Whether to use the <i>finite nucleus model</i> (DKH or X2C)
PrintLevel	1	How much relativistics-related output to print (0-4). Alias for %output Print[P_Rel]
DKH-specific		
Order	2	Order of DKH Hamiltonian (1 or 2)
fpFWtrafo	true	Kind of Foldy-Wouthuysen transformation used for EPR with DKH (see Section 2.12.4). true: fpFW; false: fπFW
ZORA-specific		
Xalpha	0.7	Default value for Xalpha potential. Has effect only if VXa is included
ScaleZORA	false	Whether to scale the orbital energies (single points only)
X2C-specific		
DLU	false	If true, use the <i>DLU approximation</i> for X2C (mutually exclusive with OneCenter)
LightAtomThresh	0	Highest atomic number treated as non-relativistic in the <i>DLU approximation</i>
StorageLevel	2	X2C intermediate storage level (0-5) – see Section 2.12.5

2.13 Implicit Solvation

Implicit solvation models play an important role in quantum chemistry. Without resorting to placing multiple solvation shells of solvent molecules implicit solvent models are able to mimic the effect of a specific solvent on the solute.

The implicit solvent models available in ORCA are

1. C-PCM[139, 140] : The Conductor-like Continuum Polarization Model
2. SMD[141] : The Solvation Model based on Density
3. OpenCOSMO-RS[142] : Interface to the open source implementation of the COSMO-RS model
4. ALPB/ddCOSMO/CPCM-X : The solvation models available in XTB

Regarding C-PCM and SMD, they are natively implemented in ORCA. With these models, various types of calculations can be performed using a polarizable continuum with a realistic van der Waals cavity as summarized below:

- Energies of molecules in solution with a finite dielectric constant ϵ at the HF and DFT level, as well as for post-HF methods.
- Optimization of molecular structures in solution using HF or any DFT method with analytic gradients.
- Calculation of vibrational frequencies using the analytic Hessian for HF or any DFT method, provided that the same calculation is available in vacuum.
- Calculation of solvent effects on response properties like polarizabilities through coupled-perturbed SCF theory. For magnetic response properties, such as the g-tensor, the C-PCM response vanishes.
- Calculations of solvent shifts on transition energies using the time-dependent DFT or CIS method. The refractive index of the solvent needs to be provided in addition to the dielectric constant.
- First order perturbation estimate of solvent effects on state and transition energies in multireference perturbation and configuration-interaction calculations.

A detailed overview of the available implicit solvation methods and their usage is provided in Sections *The Conductor-like Polarizable Continuum Model (C-PCM)*, *The SMD Solvation Model*, *OpenCOSMO-RS*, *Extended Tight-Binding: GFN0-xTB*, *GFN-xTB*, *GFN2-xTB*, and *ONIOM Methods: General Overview*.

2.13.1 The Conductor-like Polarizable Continuum Model (C-PCM)

The conductor-like polarizable continuum model (C-PCM) is an implementation of the conductor-like apparent surface charge methods. In these models the solute is placed in a cavity of roughly molecular shape. The solvent reaction field is described by apparent polarization charges on the cavity surface, which are in turn determined by the solute. These charges can be treated as punctual (point charges) or be modelled as spherical Gaussians [143]. The cavity in ORCA is constructed differently depending on how the charges are treated. In the case of using point charges, the cavity is generated through the GEPOL[144, 145, 146] algorithm, either as solvent-excluding surface (SES), or solvent-accessible surface (SAS). When Gaussian charges are considered, the user can choose between a scaled vdW surface or the GEPOL SES, and the charge positions are determined following a Lebedev quadrature approach. This scheme is known as Gaussian Charge Scheme[140] and more details on how to use it are given in Section *Use of the Gaussian Charge Scheme*.

Within C-PCM the molecular Hamiltonian of the isolated system is perturbed by the solvent:

$$\hat{H} = \hat{H}^0 + \hat{V}$$

where \hat{H}^0 is the Hamiltonian of the isolated molecule, whereas \hat{V} describes the solute – solvent interactions. The SCF procedure leads to the variational minimization of the free energy of the solute, G :

$$G = \langle \Psi | \hat{H}^0 | \Psi \rangle + \frac{1}{2} \langle \Psi | \hat{V} | \Psi \rangle$$

Using the conductor-like boundary condition the electrostatic potential can be determined by

$$V(\vec{r}) + \sum_i^{N_q} V_{q_i}(\vec{r}) = 0$$

where V and V_{q_i} are the electrostatic potential due to the solute and to the polarization charges, \vec{r} is a point on the cavity surface, and N_q is the total number of solvation charges. The vector of polarization charge can then be determined by

$$\mathbf{A}\mathbf{Q} = -\mathbf{V} \quad (2.52)$$

where the vector \mathbf{V} contains the electrostatic potential due to the solute at the position of the charges. The elements of the matrix \mathbf{A} have a different functional form depending on how the charges are treated. If we use point charges:

$$\begin{aligned} A_{ii} &= 1.07 \sqrt{\frac{4\pi}{S_i}} \\ A_{ij} &= \frac{1}{r_{ij}} \end{aligned} \quad (2.53)$$

in which S_i is the area of the surface element i , and $r_{ij} = |\vec{r}_i - \vec{r}_j|$. When Gaussian charges are considered:

$$\begin{aligned} A_{ii} &= \frac{\zeta_i \sqrt{2/\pi}}{F_i} \\ A_{ij} &= \frac{\text{erf}(\zeta_{ij} r_{ij})}{r_{ij}} \end{aligned}$$

Here, ζ_i is the exponent of the Gaussian charge i (i belongs to sphere I). This quantity is calculated as $\zeta_i = \zeta / (R_I \sqrt{w_i})$, where R_I is the radius of sphere I , w_i is the weight of the Lebedev point i , and ζ is a width parameter optimized for each particular Lebedev grid [143]. On the other hand, $\zeta_{ij} = \zeta_i \zeta_j / \sqrt{\zeta_i^2 + \zeta_j^2}$. The function

F_i , known as switching function, measures the contribution of the Gaussian charge i to the solvation energy. This function is calculated as

$$F_i = \prod_{J, i \neq J}^{\text{atoms}} g(\vec{r}_i, \vec{R}_J)$$

where $g(\vec{r}_i, \vec{R}_J)$ is the elementary switching function. In ORCA we use the improved Switching/Gaussian (ISWIG) function for $g(\vec{r}_i, \vec{R}_J)$ proposed in ref. [147]:

$$g(\vec{r}_i, \vec{R}_J) = 1 - \frac{1}{2} \{ \text{erf} [\zeta_i (R_J - r_{iJ})] + \text{erf} [\zeta_i (R_J + r_{iJ})] \}$$

If $g(\vec{r}_i, \vec{R}_J) < 10^{-7}$ the value of g is set equal to 0.

If we consider a solvent with a dielectric constant ε , eq. (2.52) reads as

$$\mathbf{A}\mathbf{Q} = -f(\varepsilon)\mathbf{V} \quad (2.54)$$

where $f(\varepsilon) = (\varepsilon - 1)/(\varepsilon + x)$ is a scaling function, and x is in the range 0-2. In C-PCM x is equal to 0.

The C-PCM model can be used via

```
! CPCM(solvent)
```

where `solvent` is one of the available solvents in Table 2.56

Table 2.56: List of available solvents for the C-PCM, SMD, openCOSMO-RS, and ALPB in ORCA. The data for the dielectric constant used within C-PCM is that at 293.15,[148] except for ammonia, which has a boiling point of 239.81 K. For the rest of solvation models, see the corresponding sources.[141][149] Notice that ALPB is only available in the context of XTb calculations. The list of solvents for ddCOSMO and CPCM-X is, on the other hand, provided in Table 3.24.

Solvent	C-PCM	SMD	COSMO-RS	ALPB
1,1,1-trichloroethane	X	X		
1,1,2-trichloroethane	X	X		
1,2,4-trimethylbenzene	X	X		
1,2-dibromoethane	X	X	X	
1,2-dichloroethane	X	X	X	
1,2-ethanediol	X	X		
1,4-dioxane / dioxane	X	X	X	X
1-bromo-2-methylpropane	X	X		
1-bromooctane / bromooctane	X	X	X	
1-bromopentane	X	X		
1-bromopropane	X	X		
1-butanol / butanol	X	X	X	
1-chlorohexane / chlorohexane	X	X	X	
1-chloropentane	X	X		
1-chloropropane	X	X		
1-decanol / decanol	X	X	X	
1-fluorooctane	X	X	X	
1-heptanol / heptanol	X	X	X	
1-hexanol / hexanol	X	X	X	
1-hexene	X	X		
1-hexyne	X	X		
1-iodobutane	X	X		
1-iodohexadecane / hexadecyl iodide	X	X	X	
1-iodopentane	X	X		
1-iodopropane	X	X		

continues on next page

Table 2.56 – continued from previous page

Solvent	C-PCM	SMD	COSMO-RS	ALPB
1-nitropropane	X	X		
1-nonanol / nonanol	X	X	X	
1-octanol / octanol	X	X	X	X
1-pentanol / pentanol	X	X	X	
1-pentene	X	X		
1-propanol / propanol	X	X	X	
2,2,2-trifluoroethanol	X	X		
2,2,4-trimethylpentane / isooctane	X	X	X	
2,4-dimethylpentane	X	X		
2,4-dimethylpyridine	X	X		
2,6-dimethylpyridine	X	X	X	
2-bromopropane	X	X		
2-butanol / secbutanol	X	X	X	
2-chlorobutane	X	X		
2-heptanone	X	X		
2-hexanone	X	X		
2-methoxyethanol / methoxyethanol	X	X	X	
2-methyl-1-propanol / isobutanol	X	X	X	
2-methyl-2-propanol	X	X		
2-methylpentane	X	X		
2-methylpyridine / 2methylpyridine	X	X	X	
2-nitropropane	X	X		
2-octanone	X	X		
2-pentanone	X	X		
2-propanol / isopropanol	X	X	X	
2-propen-1-ol	X	X		
e-2-pentene	X	X		
3-methylpyridine	X	X		
3-pentanone	X	X		
4-heptanone	X	X		
4-methyl-2-pentanone / 4methyl2pentanone	X	X	X	
4-methylpyridine	X	X		
5-nonanone	X	X		
acetic acid / aceticacid	X	X	X	
acetone	X	X	X	X
acetonitrile / mecn / ch3cn	X	X	X	X
acetophenone	X	X	X	
ammonia	X		X	
aniline	X	X	X	X
anisole	X	X	X	
benzaldehyde	X	X	X	X
benzene	X	X	X	X
benzonitrile	X	X	X	
benzyl alcohol / benzylalcohol	X	X	X	
bromobenzene	X	X	X	
bromoethane	X	X	X	
bromoform	X	X	X	
butanal	X	X		
butanoic acid	X	X		
butanone	X	X	X	
butanonitrile	X	X		
butyl ethanoate / butyl acetate / butylacetate	X	X	X	
butylamine	X	X		
n-butylbenzene / butylbenzene	X	X	X	
sec-butylbenzene / secbutylbenzene	X	X	X	

continues on next page

Table 2.56 – continued from previous page

Solvent	C-PCM	SMD	COSMO-RS	ALPB
tert-butylbenzene / tbutylbenzene	X	X	X	
carbon disulfide / carbondisulfide / cs2	X	X	X	X
carbon tetrachloride / ccl4	X	X	X	
chlorobenzene	X	X	X	
chloroform / chcl3	X	X	X	X
a-chlorotoluene	X	X		
o-chlorotoluene	X	X		
conductor	X			
m-cresol / mcresol	X	X	X	
o-cresol	X	X		
cyclohexane	X	X	X	
cyclohexanone	X	X	X	
cyclopentane	X	X		
cyclopentanol	X	X		
cyclopentanone	X	X		
decalin	X	X	X	
cis-decalin	X	X		
n-decane / decane	X	X	X	
dibromomethane	X	X		
dibutylether	X	X	X	
o-dichlorobenzene / odichlorobenzene	X	X	X	
e-1,2-dichloroethene	X	X		
z-1,2-dichloroethene	X	X		
dichloromethane / ch2cl2 / dcm	X	X	X	X
diethyl ether / diethylether	X	X	X	X
diethyl sulfide	X	X		
diethylamine	X	X		
diiodomethane	X	X		
diisopropyl ether / diisopropylether	X	X	X	
cis-1,2-dimethylcyclohexane	X	X		
dimethyl disulfide	X	X		
n,n-dimethylacetamide / dimethylacetamide	X	X	X	
n,n-dimethylformamide / dimethylformamide / dmf	X	X	X	X
dimethylsulfoxide / dmsol	X	X	X	X
diphenylether	X	X	X	
dipropylamine	X	X		
n-dodecane / dodecane	X	X	X	
ethanethiol	X	X		
ethanol	X	X	X	X
ethyl acetate / ethylacetate / ethanoate	X	X	X	X
ethyl methanoate	X	X		
ethyl phenyl ether / ethoxybenzene	X	X	X	
ethylbenzene	X	X	X	
fluorobenzene	X	X	X	
formamide	X	X		
formic acid	X	X		
furan / furane			X	X
n-heptane / heptane	X	X	X	
n-hexadecane / hexadecane	X	X	X	X
n-hexane / hexane	X	X	X	X
hexanoic acid	X	X		
iodobenzene	X	X	X	
iodoethane	X	X		
iodomethane	X	X		
isopropylbenzene	X	X	X	

continues on next page

Table 2.56 – continued from previous page

Solvent	C-PCM	SMD	COSMO-RS	ALPB
p-isopropyltoluene / isopropyltoluene	X	X		
mesitylene	X	X	X	
methanol	X	X	X	X
methyl benzoate	X	X		
methyl butanoate	X	X		
methyl ethanoate	X	X		
methyl methanoate	X	X		
methyl propanoate	X	X		
n-methylaniline	X	X		
methylcyclohexane	X	X		
n-methylformamide / methylformamide	X	X	X	
nitrobenzene / phno2	X	X	X	
nitroethane	X	X	X	
nitromethane / meno2	X	X	X	X
o-nitrotoluene / onitrotoluene	X	X		
n-nonane / nonane	X	X	X	
n-octane / octane	X	X	X	
n-pentadecane / pentadecane	X	X	X	
octanol(wet) / wetoctanol / woctanol				X
pentanal	X	X		
n-pentane / pentane	X	X	X	
pentanoic acid	X	X		
pentyl ethanoate	X	X		
pentylamine	X	X		
perfluorobenzene / hexafluorobenzene	X	X	X	
phenol	X		X	X
propanal	X	X		
propanoic acid	X	X		
propanonitrile	X	X		
propyl ethanoate	X	X		
propylamine	X	X		
pyridine	X	X	X	
tetrachloroethene / c2cl4	X	X	X	
tetrahydrofuran / thf	X	X	X	X
tetrahydrothiophene-s,s-dioxide / / tetrahydrothiophenedioxide / sulfolane	X	X		
tetralin	X	X	X	
thiophene	X	X		
thiophenol	X	X		
toluene	X	X	X	X
trans-decalin	X	X		
tributylphosphate	X	X	X	
trichloroethene	X	X		
triethylamine	X	X	X	
n-undecane / undecane	X	X	X	
water / h2o	X	X	X	X
xylene	X	X		
m-xylene	X	X		
o-xylene	X	X		
p-xylene	X	X		

The parameters related to each implicit solvation method, as well as to the solvents can be more accurately defined using the %cpcm block input. A complete list of the available options are described in [Complete Keyword List for the %cpcm Block](#).

The type of cavities surrounding the solute and how the solvation charges are treated in ORCA is explained in subsec-

tions *Use of the Gaussian Charge Scheme*, and *Use of the Point Charge Scheme*. ORCA supports two types of solvation charge schemes: (i) the point charge scheme, (ii) the Gaussian charge scheme. The default solvation scheme from ORCA 5.0 on is the Gaussian charge scheme with a vdW-type cavity (`surfacetype vdw_gaussian`). Older ORCA versions considered `surfacetype gepol_ses` and the point charge scheme as defaults. So, if one wants to reproduce the results obtained with ORCA versions older than ORCA 5.0, please use the tag `surfacetype gepol_ses` in the `%cpcm` block.

The different charge schemes are described in the subsections *Use of the Gaussian Charge Scheme*, and *Use of the Point Charge Scheme*. The availability of the analytical gradient and Hessian for the different combinations of charge scheme and surface is shown in Table 2.57.

Table 2.57: Available type of gradients and Hessians within the C-PCM in ORCA. The tag “YES” means that the feature is implemented, and “NO” that it isn’t. For clarity, we denote by “*” the default scheme in ORCA (`surfacetype vdw_gaussian`).

surfacetype	Charge type	Gradient		Hessian	
		Analytical	Numerical	Analytical	Numerical
gepol_ses	Point	Yes	Yes	Yes	Yes
gepol_sas	Point	Yes	Yes	Yes	Yes
vdw_gaussian^*	Gaussian	Yes	Yes	Yes	Yes
gepol_ses_gaussian	Gaussian	No	Yes	No	Yes

Note 1: If the user wants to turn off the C-PCM, one has to write the following tag in the simple input:

```
! NOCPCM
```

This is needed, for instance, in the context of concatenated calculations using the `$new_job` feature, where the previous calculation involved solvation, but one wants to turn it off for the next one.

Note 2: As stated above, the default for C-PCM in ORCA is the Gaussian charge scheme with a vdW-type cavity. The implementation is done following ref. [140]. Hence, if C-PCM is used in ORCA calculations, one must cite this paper together with ref. [139]. Ref. [140] must also be cited if SMD, or openCOSMO-RS are requested, since these type of calculations rely on C-PCM.

Use of the Gaussian Charge Scheme

The Gaussian charge scheme avoids the Coulomb singularity present in conventional point charge surface element models. This approach, when applied together with a switching function, results in a smooth solvation potential and, more importantly, on smooth derivatives of this quantity with respect to external perturbations. Then, it is highly recommended to adopt this approach within the C-PCM. The Gaussian charge scheme can be used with two types of solute cavity surfaces: (1) a scaled vdW surface, (2) a solvent-excluded surface (SES). To assign the radii for the different atoms we follow the scheme proposed in ref. [147]. That is, we use Bondi radii [150] for all elements, except for hydrogen where we adopt 1.1 Å. For 16 of the main-group elements in the periodic table, where Bondi’s radii are not defined, we adopt the radii proposed in ref. [151] by Mantina et al. This is the case for elements: Be, B, Al, Ca, Ge, Rb, Sr, Sb, Cs, Ba, Bi, Po, At, Rn, Fr, Ra. For the elements that are not covered neither by Bondi nor by Mantina, we consider a radius of 2 Å.

• Scaled vdW cavity

The Gaussian charge scheme with a scaled vdW-type cavity is now the default in ORCA, so one just needs to add the C-PCM tag in the `%cpcm` block in the input file. That would correspond to (although the user does not need to write that, as it is internally processed by ORCA):

```
%cpcm
  surfacetype vdw_gaussian
end
```

In this case, the radius R_I of atom I for the scaled vdW cavity is calculated as

$$R_I = f_{scal} R_I^{vdW}$$

where R_I^{vdW} is the vdW radius of atom I and f_{scal} is a scaling factor. This parameter is by default equal to 1.2, as suggested in ref. [147]. However, the user can modify its value through the `scale_gauss` tag in the `%cpcm` block in the input file. The number of C-PCM charges per atom is such that we have an approximate density of 5.0 charges/Å² on the surface of the cavity. This number corresponds approximately to 110 points on a hydrogen atom ($110/(4\pi \times 1.32^2) \approx 5.0 \text{ Å}^{-2}$). ORCA will choose then different levels of discretization (Lebedev grids) depending on the radii of the atoms. This scheme is called isodensity scheme. The threshold for the number of charges per unit of area on hydrogens and non-hydrogen atoms can be specified by the user via `thresh_h` and `thresh_noth`, respectively. Alternatively, the user can request a fixed number of Lebedev points per sphere in the cavity (independent of the radius of the sphere). This can be done via the `num_leb` tag. This parameter can adopt the following values: 50, 110, 194, 302, 434, 590, 770, 974, and 1202. ORCA versions older than ORCA 6.0 use this last scheme.

The analytical gradient, as well as the analytical Hessian are available for this solvation method.

- **Solvent-excluded surface**

The GEPOL-generated SES can be used together with the Gaussian charge scheme. In this case, the Gaussian charges are not only placed on the surface of the atomic spheres but also on the surface of the new spheres generated through the GEPOL algorithm. To use this approach we should modify the ORCA input file as follows

```
%cpcm
  surfacetype gepol_ses_gaussian
end
```

The SES is in general recommended when the solute is explicitly solvated by few solvation layers. In this case, the additional GEPOL spheres prevent the solvent to fill the space between explicit solvent molecules or between those molecules and the solute. The radius of the solvent sphere that rolls over the solute vdW-type surface to generate the SES is controlled by the keyword `rsolv`. This radius has a default value of 1.30 Å, but the user can change it by specifying another value for `rsolv` in the `%cpcm` block. The minimum radius for an added GEPOL sphere is controlled by `rmin`.

Neither the analytical gradient nor the analytical Hessian are available for this strategy. They should be computed numerically. Due to the interdependency between GEPOL spheres and the atoms present in our system, the analytical gradient computed using the SES does not converge as smooth as compared to that using the vdW-type cavity (see ref. [140]) and can lead to wrong minima. The Hessian is affected in the same way. Then, ORCA 6.0 does not support anymore the analytical gradient/Hessian for such surface to prevent inaccurate results.

Use of the Point Charge Scheme

Within this scheme, the solvation charges are treated as punctual and no switching function is used to accept/discard charges in the intersection between the spheres that form the solute cavity. These two facts lead to a discontinuous potential energy surface (numerical instabilities in the SCF energy and its derivatives). Then, the point charge scheme is not recommended within the C-PCM. If the user still wants to use this charge scheme it can be used together with two different type of surfaces for the solute cavity: (1) the SES, and (2) the SAS.

- **Solvent-excluded surface**

In the same way as for the Gaussian charge scheme with the SES, this surface is generated for the point charge scheme through the GEPOL algorithm. However, while for the case of `surfacetype gepol_ses_gaussian`, the surface is discretized using Lebedev-type grids, in the point charge scheme, the surface is divided in spherical triangles called “tesserae”. The level of tessellation is controlled by the tag `ndiv`. The radii used for the solute atoms in order to construct the GEPOL cavity are those optimized for the COSMO-RS model[152] for H, C, N, O, F, S, Cl, Br, and I, while for the rest of elements scaled Bondi radii are used.

In order to use the point charge scheme with the SES, we should add the following tag in the %cpcm block:

```
%cpcm
  surfacetype gepol_ses
end
```

- **Solvent-accessible surface**

Here, the surface of the solute cavity is generated by following the center of a sphere with radius `rsolv` (representing the solvent molecule) rolling over the surface of the vdW surface of the solute. The discretization of the resulting surface is done via tesserae, as done for the SES. In order to use the point charge scheme together with the SAS, one should add the following tag in the %cpcm block:

```
%cpcm
  surfacetype gepol_sas
end
```

- **How to circumvent the numerical instabilities in the point charge scheme**

Numerical instabilities are implicit in the point charge scheme due to the “punctual” nature of the charges and to the fact that no switching function is considered in the intersection between the spheres that form the solute cavity. At the same time, there is no way to predict, in advance if there will be discontinuities in the SCF energy and/or in its derivatives. However, if the user still wants to use this charge scheme, no matter with which type of surface (SES or SAS), there are two things that one can try to minimize the aforementioned problems:

- **Change `ndiv`** : The parameter `ndiv` controls the number of triangles per sphere in the solute cavity. By changing the number of triangles, it also changes the number of triangles in the intersection between spheres and, then, this can solve those situations where two point charges get too close making the elements A_{ij} in eq (2.53) to diverge (and the solvation charges to suddenly have very large values). However, this strategy can be a solution for a particular case, but will never ensure that for the same system with another geometry the discontinuities in the SCF energy do not show (as it does not prevent point charges to be too close from each other).
- **Increase `pmin`** : This parameter removes those charges that are at a distance lower than `pmin` from each other. The default value for `pmin` is of 0.1 a.u. (≈ 0.0529 Å). Then, by increasing `pmin` one removes all pairs of charges that are too close from each other in the intersection between the spheres. **Note:** Be careful when increasing `pmin`. Although this prevents sudden jumps in the SCF energy, it can lead to “biasing” the solute-solvent interaction, as one is removing a significant number of charges that represent the effect of the solvent.

Calculation of the free energy of solvation within the C-PCM

The solvation free energy, ΔG_{solv} , is defined as the free energy of transfer of a solute from the gas phase to the condensed phase. This quantity can be written as (see eq 27 in ref. [140])

$$\Delta G_{solv} = E_{solv}(\vec{R}_l, \vec{R}_v) + \Delta G_{el} + \Delta G_{cav} + \Delta G_{disp} \quad (2.55)$$

The first term, $E_{solv}(\vec{R}_v, \vec{R}_l)$, corresponds to the difference between the liquid-phase expectation value of the gas-phase Hamiltonian, $E(\vec{R}_l)$, and the gas-phase potential energy surface $E(\vec{R}_v)$

$$E_{solv}(\vec{R}_v, \vec{R}_l) = E(\vec{R}_l) - E(\vec{R}_v)$$

The second term in eq (2.55) accounts for the electronic-polarization contribution to ΔG_{solv} and is calculated from the charges spread over the surface of the solute cavity (vdW-type or the SES). Finally, ΔG_{cav} is the cavitation energy, that is, the reversible work required to create a cavity inside the bulk of the solvent in order to accommodate the solute, while ΔG_{disp} accounts for the changes in the dispersion energy occurring when solvating the solute. The sum of these last two terms correspond to the non-electrostatic contribution, ΔG_{nel} , to ΔG_{solv}

$$\Delta G_{nel} = \Delta G_{cav} + \Delta G_{disp}$$

In ORCA, if a system is solvated within the C-PCM (defining `!CPCM(solvent name)` in the input file, within either the point charge scheme or the Gaussian charge scheme) there is no “non-electrostatic solvation contribution”

neither to the SCF energy, nor to its derivatives. In order to have a rough idea of the value of ΔG_{nel} , one can estimate this quantity through empirical equations available in the literature obtained from experimental data. For instance, ΔG_{nel} can be estimated from the free energy of hydration for linear-chain alkanes[153]

$$\Delta G_{nel} = 1.321 + 0.0067639 \times \text{SASA} \quad (2.56)$$

where SASA is the solvent-accessible surface area. However, although eq (2.56) may give a good estimation of ΔG_{nel} for organic molecules in water, it is a bad approximation for non-electrostatic solvation effects when considering solvents different than water (as well as for many solutes in water). There are two alternatives to this approximation that one can adopt in order to include non-electrostatic solvation effects in ORCA calculations:

1. The use of the SMD model in combination with the C-PCM (see section *The SMD Solvation Model*). This is the most adopted strategy within the QM community.
2. The use of Gaussian charges to compute the electrostatic contribution to ΔG_{solv} together with an equation for ΔG_{nel} calculated from the contribution of the solute atoms to the SASA.[140]

Focusing on the second strategy, the cavitation energy, ΔG_{cav} is calculated through the equation proposed by Pierotti and Claverie.[154, 155] In this case, ΔG_{cav} is expanded in powers of the ratio $\bar{\mathbf{R}}$ between the radius of the solute sphere R_I and that of the solvent spheres R_S . Then, the cavitation energy for the solute sphere centered at atom I reads as,

$$\Delta G_{cav,I} = -\ln(1-y) + \left(\frac{3y}{1-y}\right)\bar{\mathbf{R}} + \left[\frac{3y}{1-y} + \frac{9}{2}\left(\frac{y}{1-y}\right)^2\right]\bar{\mathbf{R}}^2 + \frac{yP}{\rho_S k_B T}\bar{\mathbf{R}}^3$$

with $y = 8\pi\rho_S R_S^3/6$, being ρ_S the density of the solvent, $\bar{\mathbf{R}} = R_I/R_S$, T the temperature and P the pressure. If we consider the solvent-accessible surface (SAS), then, the total ΔG_{cav} is the sum of the $\Delta G_{cav,I}$ weighted by a factor that depends on the exposed SASA $_I$ of sphere I

$$\Delta G_{cav} = \sum_{i=1}^{N_{sph}} \frac{\text{SASA}_I}{4\pi R_I^2} \Delta G_{cav,I} = \sum_{i=1}^{N_{el}} \frac{\text{SASA}_i}{4\pi R_I^2} \Delta G_{cav,i} \quad (2.57)$$

Here, the summation over the total number of spheres (N_{sph}) that conform the SAS is replaced by a summation over the total number of surface elements (N_{el}) in which the SAS is divided into.

With respect to ΔG_{disp} , this quantity is assumed to depend linearly on the contribution of each surface element to the SASA

$$\Delta G_{disp} = \sum_{i=1}^{N_{el}} \sigma_I \text{SASA}_i \quad (2.58)$$

Here, the factor σ_I is the atomic surface tension of the sphere I to which the surface element i belongs to.

This strategy, that considers eqs (2.57) and (2.58) to compute ΔG_{nel} assumes the use of Gaussian charges for the calculation of ΔG_{el} in eq (2.55). When a vdW-type cavity is used to account for electrostatic solvation effects, then the solvation approach is called GVDW_nel, while when a SES-type cavity is used we refer to the solvation approach as GSES_nel (“G” for Gaussian charges and “nel” because of the inclusion of non-electrostatic solvation effects). Details on the GVDW_nel and GSES_nel models can be found in ref. [140].

The GVDW_nel and GSES_nel models have been parametrized for solutes containing H, C, N and O atoms in the following solvents: benzene, chloroform, cyclohexane, octanol, toluene and water. The parameters for the σ_I atomic surface tensions in eq (2.58) together with the radius of the solvent molecules R_S used to generate the SAS are provided in ref. [140]. The use of the GVDW_nel and GSES_nel is controlled by the string `cds_cpcm` to add in the `%cpcm` block. This flag should be equal to 2 within these two models (by default, when we use C-PCM together with the Gaussian Charge Scheme without non-electrostatic solvation effects, `cds_cpcm` is equal to 0). In order to calculate ΔG_{solv} for the GVDW_nel scheme one has to do the following steps:

1. Optimize the geometry of the solute (anisole in this case) in vacuum. For a DFT calculation at the B3LYP/def2-TZVP level of theory (with RIJCOSX), our input file would read like:

```
! B3LYP def2-TZVP D3BJ tightopt

%maxcore 4000

* xyz 0 1
C 0.66588156410505 1.19833878437922 -0.02647672759883
C 1.95470174000504 0.69483871574862 -0.01927812843331
C 2.17893173780041 -0.68097630146538 0.00933914777704
C 1.09270919891774 -1.54252176637089 0.03141990372830
C -0.21073409652496 -1.05179390481975 0.02471602253843
C -0.42517201252444 0.32533458015547 -0.00499572635290
H 0.47565631823749 2.26336213776820 -0.05012843684387
H 2.79210971512409 1.38085300778245 -0.03749594945361
H 3.18771596563920 -1.07149825338966 0.01433233618265
H 1.25061585302875 -2.61357041387417 0.05495238973649
H -1.03877793934640 -1.74453554370491 0.04346162651229
O -1.65366310927299 0.90979347913477 -0.01693978807213
C -2.79791686738707 0.07370821312998 0.01117694976431
H -2.83142307710305 -0.58944196707467 -0.85867919998175
H -3.65619850989594 0.74176224021803 -0.00958531946332
H -2.82883648080292 -0.53125300761732 0.92288089996024
*
```

2. Use the resulting structure (`anisoле.xyz`) as input structure for the subsequent geometry optimization in solution. In this case, we consider water as the solvent. The input file looks like:

```
! B3LYP def2-TZVP D3BJ tightopt CPCM(Water)

%cpcm
  cds_cpcm 2
end

%maxcore 4000

* xyzfile 0 1 anisoле.xyz
```

3. Search for the FINAL SINGLE POINT ENERGY in the output file for the solvated system (we call it E_s):

```
-----
FINAL SINGLE POINT ENERGY      -346.723542928703
-----
```

4. Search for the FINAL SINGLE POINT ENERGY in the output file for the system in vacuum (we call it E_0):

```
-----
FINAL SINGLE POINT ENERGY      -346.719657761697
-----
```

5. Calculate $\Delta G_{solv} = E_s - E_0 = -0.003885$ a.u. If we convert it to kcal/mol and round it to two decimal digits, we have $\Delta G_{solv} = -2.44$ kcal/mol. This quantity is in very good agreement with its experimental counterpart (-2.45 kcal/mol[141]).
6. In case the user is interested in the value of ΔG_{nel} , this quantity is printed in the CPCM Solvation Model Properties block:

```
CPCM Solvation Model Properties:
Surface-charge      : -0.02580688416088
Corrected charge    : 0.00000000000000
Outlying charge corr. : 0.00005046206545 Eh 0.00137 eV
Free-energy (cav+disp) : 0.00272597402638 Eh 0.07418 eV
```

Here, ΔG_{nel} corresponds to `Free-energy (cav+disp)` and is calculated through eqs (2.57) and (2.58). The term called `Surface-charge` corresponds to the solute net charge that lies outside the C-PCM cavity. It is basically the sum of the C-PCM charges (without the scaling by $f(\epsilon)$). The effect of this excess of charge on the energy and the C-PCM charges themselves can be corrected by the so-called outlying charge correction. In ORCA, this correction is calculated through a Lagrangian-based algorithm,[156] but only printed for information purposes. That is, the outlying charge effect is neither added to the SCF energy nor to its derivatives. In any case, the corrected total charge is printed in `Corrected charge`, while the correction term for the energy is printed in `Outlying charge corr..` To further help the user, ORCA also prints a file with extension `.cpcm_corr`, where the corrected C-PCM charges are provided.

If we want to use, instead, the `GSES_nel` model, one just needs to add `surfacetyp gepol_ses_gaussian` in the `%cpcm` block.

Note: The analytical Hessian is not available for the `GVDW_nel` and `GSES_nel` models as the second derivative of ΔG_{nel} with respect to nuclear displacements is not implemented.

Options for Computing the CPCM terms

In CPCM calculations, there are three major steps. The first step is the calculation of the electrostatic potential created by the quantum object on the surface grid. This step is characterized by:

$$\begin{aligned} V(\mathbf{r}_g) &= \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}_g|} - \sum_{\mu\nu} P_{\mu\nu} \int \frac{\mu(\mathbf{r})\nu(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|} d\mathbf{r} \\ &= \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}_g|} - \sum_{\mu\nu} P_{\mu\nu} (\mu\nu|g) \end{aligned}$$

The first sum is over atomic nuclei, the second over basis functions. The second step in the calculation consists of converting the surface potential into surface screening charges \mathbf{q} . We are not writing it out here, since this step is slightly different for each variant of CPCM. In a nutshell, it involves solving a linear equation system of the dimension of the number of surface segments which is usually done by Cholesky decomposition.

The third, step is to sum the interaction of the surface charges with the quantum electrons into the Fock matrix. This step is given by:

$$\begin{aligned} F_{\mu\nu}^{(CPCM)} &= - \sum_g q_g \int \frac{\mu(\mathbf{r})\nu(\mathbf{r})}{|\mathbf{r} - \mathbf{r}_g|} d\mathbf{r} \\ &= - \sum_g q_g (\mu\nu|g) \end{aligned}$$

Here the sum is over surface segments with surface screening charges q_g .

The time consuming step in these calculations is the calculation of the three-center electrostatic integrals $(\mu\nu|g)$. By default, these integrals are calculated using efficient analytic integration provided by SHARK. However, there are the following alternatives available:

- 1) The integrals can be evaluated by numeric integration
- 2) The potential and Fock matrix contributions can be evaluated using the RI density and the RI approximation respectively
- 3) The potential and Fock matrix elements can be evaluated using multipole approximations to the integrals.

These options are triggered by:

```
%cpcm UseAnalyticOnRebuild true # if true, full Fock matrix builds
                                # are done with analytic integrals
                                # (strongly recommended)

Vopt v_analytic
# = v_multipole
```

(continues on next page)

(continued from previous page)

```
# = v_RI
# = v_numeric
Fopt f_analytic
# = f_multipole
# = f_RI
# = f_numeric
```

Note

- Our experience indicates that analytic integration is still the safest and most accurate options.
- Analytic integration will win out over numerical integration or multipole approximations
- Some savings can be realized with the RI approximation, but it may also introduce non-negligible errors.

2.13.2 Bug Fixes and Extensions

- Relativistic GIAO pVp integrals with finite nucleus models for NMR
- Quadrupole Derivative Integrals for efficient native XTB Gradients
- Prevent errors in general contraction calculations by automatic switching to PGC
- LeanSCF now always calculates a final Fock matrix at the end in order to guarantee that the energy, orbitals, orbital energies, density and orbitals are all in sync.

2.13.3 The Conductor-like Screening Solvation Model (COSMO)

Note

The COSMO solvation model has been removed from ORCA v4.0.0 !!!

Please use the C-PCM solvation model in combination with the COSMO epsilon function if required!

As a short form to use the C-PCM model with the COSMO epsilon function, you can specify the solvent via `!CPCMC(solvent)`.

2.13.4 The SMD Solvation Model

The SMD solvation model has been proposed by the Cramer and Truhlar groups,^[141] and is based on the quantum mechanical charge density of a solute molecule interacting with a continuum description of the solvent. In the model the full solute electron density is used without defining partial atomic charges and the solvent is not represented explicitly but rather as a dielectric medium with the surface tension at the solute–solvent boundary. SMD is a universal solvation model, in the sense that it is applicable to any charged or uncharged solute in any solvent or liquid medium for which a few key descriptors are known. In particular, these descriptors are the dielectric constant, refractive index, bulk surface tension, and acidity and basicity parameters. Neglecting the concentration contribution, the model separates the observable solvation free energy into two main components,

$$\Delta G_S = \Delta G_{\text{ENP}} + \Delta G_{\text{CDS}}. \quad (2.59)$$

In ORCA, the first component is the bulk electrostatic contribution arising from a self-consistent reaction field treatment that involves the electrostatic interaction using the Conductor-like Polarizable Continuum Model (C-PCM). However, the radii are set to “intrinsic atomic Coulomb radii”. The second component, called the cavity-dispersion solvent-structure (CDS) term, is the contribution resulting from short-range interactions between the solute and solvent molecules in the first solvation shell. This contribution is a sum of terms that are proportional (with geometry-dependent proportionality constants called atomic surface tensions) to the solvent-accessible surface areas of the

individual atoms of the solute. The CDS contribution to the free energy of solvation is given by

$$\Delta G_{\text{CDS}} = \sum_k^{\text{atoms}} \sigma_k A_k(\mathbf{R}, R_{Z_k} + r_s) + \sigma^{[M]} \sum_k^{\text{atoms}} A_k(\mathbf{R}, R_{Z_k} + r_s),$$

where σ_k and $\sigma^{[M]}$ are the atomic surface tension of atom k and the molecular surface tension, respectively, and A_k is the solvent-accessible surface area (SASA). The SASA depends on the geometry \mathbf{R} , the set R_{Z_k} of all atomic van der Waals radii, and the solvent radius r_s , which is added to each of the atomic van der Waals radii. In the program Bondi radii are used for CDS contribution.

More details can be found in the original paper of Marenich et al. [141], which should be cited in publications using results of SMD calculations (together with ref. [140], describing the implementation of the C-PCM in ORCA, which is used by SMD).

SMD can be employed in single point calculations and geometry optimizations, using single-determinant SCF (HF and DFT) and CASSCF methods. In post-SCF methods the result is corrected in the reference wave function. The SMD solvation model is invoked in the input file via

```
! SMD(solvent)
```

where `solvent` is one of the 179 solvents in the SMD library (see Table Table 2.56). Alternatively, one can request the SMD model via the `%cpcm` block by writing:

```
%cpcm
  smd true          # turn on SMD
  SMDsolvent "solvent" # specify the name of solvent from the list
end
```

Independently on the way the user invokes the SMD model, ORCA automatically sets a number of default SMD parameters for the chosen solvent. If required, the user can also manually specify the solvent descriptors used in an SMD calculation in the `%cpcm` block.

```
%cpcm
  soln      # index of refraction at optical frequencies at 293 K
  soln25    # index of refraction at optical frequencies at 298 K
  sola      # Abraham's hydrogen bond acidity
  solb      # Abraham's hydrogen bond basicity
  solg      # relative macroscopic surface tension
  solc      # aromaticity, fraction of non-hydrogenic solvent atoms
             # that are aromatic carbon atoms
  solh      # electronegative halogenicity, fraction of non-hydrogenic
             # solvent atoms that are F, Cl, or Br
end
```

Let's consider the following input for a water molecule solvated by water,

```
! B3LYP def2-SVP tightscf SMD(water)

* xyz 0 1
O  -0.00000018976103    0.00606010894837    0.00000000004527
H   0.76098169249695   -0.58891312953082   -0.000000000000022
H  -0.76098151333900   -0.58891299029372   -0.000000000000022
*
```

Before the SCF part starts, the program prints the SMD information. This part reads as:

```
-----
CPCM SOLVATION MODEL
-----
CPCM parameters:
  Epsilon          ...      78.3550
  Refrac           ...      1.3328
```

(continues on next page)

(continued from previous page)

```

Rsolv                ...      1.3000
Surface type         ... GAUSSIAN VDW
Discretization scheme ... Constant charge density
  Threshold for H atoms ...      5.0000 (charges/Ang^2)
  Threshold for non-H atoms ...      5.0000 (charges/Ang^2)
Epsilon function type ...      CPCM
Solvent:             ... WATER
SMD-CDS solvent descriptors:
  Soln                ...      1.3328
  Soln25              ...      1.3323
  Sola                ...      0.0000
  Solb                ...      0.0000
  Solg                ...      0.0000
  Solc                ...      0.0000
  Solh                ...      0.0000
Radii:
  Scheme              ... Element-dependent radii
  Radius for O used is 2.8724 Bohr (= 1.5200 Ang.)
  Radius for H used is 2.2677 Bohr (= 1.2000 Ang.)
Calculating surface ... done! ( 0.0s)
Cavity surface points ...      244
Cavity Volume        ...      130.0616
Cavity Surface-area  ...      129.3354
Calculating surface distance matrix ... done! ( 0.0s)
Performing Cholesky decomposition & store ... done! ( 0.0s)
Overall time for CPCM initialization ...      0.0s

```

After the SCF is converged, the SMD contribution to the total energy is printed (this term is labelled as “SMD CDS” term).

```

*****
*                               SUCCESS                               *
*                               SCF CONVERGED AFTER 10 CYCLES          *
*****

Recomputing exchange energy using gridx3 ... done ( 0.056 sec)
Old exchange energy : -1.791736873 Eh
New exchange energy : -1.791694663 Eh
Exchange energy change after final integration : 0.000042211 Eh
Total energy after final integration : -76.335230273 Eh

SMD CDS free energy correction energy : 1.44927 Kcal/mol
Total Energy after SMD CDS correction = -76.332920707 Eh
**** ENERGY FILE WAS UPDATED (water.en.tmp) ****

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

Total Energy : -76.33292070695327 Eh -2077.12437 eV

Components:
Nuclear Repulsion : 9.11286213981824 Eh 247.97359 eV
Electronic Energy : -85.43200537714571 Eh -2324.72305 eV
One Electron Energy: -123.06209110061025 Eh -3348.68974 eV
Two Electron Energy: 37.63008572346453 Eh 1023.96669 eV
CPCM Dielectric : -0.01612924659271 Eh -0.43890 eV
SMD CDS (GcDs) : 0.00230956608299 Eh 0.06285 eV

```

In ORCA, the user can also request the SMD18 model,^[157] which involves refined radii for Bromine and Iodine (the radii adopted for the construction of the cavity used in the C-PCM part). The SMD18 model is shown to accurately describe Halogen-bonding interactions in solution. It is requested via the SMD18 tag in the simple input:

```
! SMD(solvent) SMD18
```

or via the %cpcm block:

```
%cpcm
  SMD18 true # default = false
end
```

Notes:

- If one is interested in the calculation of free energies of solvation using the SMD model, one just needs to compute ΔG_S according to eq (2.59). However, here one should take into account that the SMD model considers the same concentration (1 mol/L) in both the gaseous and solution phases. Then, if a gas-phase standard state of 1 atm is considered, and we want to compare the calculated ΔG_S with its experimental counterpart, a concentration term equal to 1.89 kcal/mol has to be added to the calculated ΔG_S .

2.13.5 Dynamic Radii Adjustment for Continuum Solvation (DRACO)

The DRACO scheme is an approach that improves the performance of implicit solvation models, in particular the accuracy of the calculated solvation free energies.[158] It is based on a dynamic scaling of the original static radii used to describe the atoms/spheres that define the cavity of the solute.

In this approach, the original radii r_i of an atom i is scaled according to

$$r_{\text{iscale}} = f_{\text{iscale}} r_i \quad (2.60)$$

with the scaling factor f_{iscale} determined from

$$f_{\text{iscale}} = \text{erf}(a_Z(q_{\text{eff},i} - b_Z)) + 1 \quad (2.61)$$

Here, $q_{\text{eff},i}$ corresponds to the effective partial charge and is defined as

$$q_{\text{eff},i} = q_i + k_Z q_i \text{CN}_i \quad (2.62)$$

where q_i is the atomic partial charge. The parameters a_Z , b_Z and k_Z are element-specific parameters, and CN_i is the fractional coordination number.

For oxygen atoms, the f_{iscale} is corrected via a term that depends on the Abraham's hydrogen bond acidity (α):

$$f_{\text{iscale}}^{\text{O}} = f_{\text{iscale}} + c_{\text{O}}(0.43 - \alpha) \quad (2.63)$$

with c_{O} a parameter that is different for pure C-PCM or SMD. The correction in eq (2.63) is only applied for solvents with $\alpha < 0.43$.

DRACO is parametrized for C-PCM and SMD for the following solvents: acetonitrile, DMSO, methanol, and water. The element-specific parameters in eqs (2.61), (2.62), and (2.63) are available in <https://github.com/grimme-lab/DRACO>.

The use of DRACO with C-PCM or SMD is triggered via the following tags in the simple input:

```
! CPCM(solvent) DRACO
```

or

```
! SMD(solvent) DRACO
```

Alternatively, it can also be requested in the %cpcm block:

```
%cpcm
  DRACO true
end
```

In ORCA, the default scheme to calculate the partial charges in eq (2.62) is the electronegativity-equilibration (EEQ) charge model (D4 case),[159]. However, one can also request the Charge Extended Hückel (CEH) model.[160] The charge scheme within DRACO is controlled by the following tag in the %cpcm block:

```
%cpcm
  draco_charges ceh # default = eeq
end
```

If CEH charges are requested, ORCA needs to run an XTB calculation to generate them. Here, one should have at least the version 6.7.1 of XTB (older versions will not generate the CEH charges, see section *Program Components*). Regarding the fractional coordination number in eq (2.62), it is calculated as described in GFN2-XTB calculations.[161]

DRACO can be used in single-point energy calculations (with both CEH and EEQ charges) and in geometry optimizations (with the EEQ charges)[162]. The DRACO contributions to the analytical Hessian are not yet implemented. In this case, the user should request numfreq.

Notice that calculations where DRACO is requested involve a discretization scheme of the surface of the cavity based on a constant number of charges per sphere. The default is 194 Lebedev points. This contrast with the scheme used for C-PCM/SMD calculations without DRACO, where the default is the isodensity scheme, as discussed in Section *Use of the Gaussian Charge Scheme*. This choice is needed, specially for geometry optimizations. For DRACO, the radii of the spheres in the cavity changes in each optimization step. Then, if ORCA used by default the isodensity scheme for DRACO, the associated Lebedev grid per atom could change between two consecutive steps, leading to discontinuities in the total energy and its gradient. Nevertheless, the isodensity scheme can still be requested for calculations involving DRACO via the tag DRACOisodens in the %cpcm block:

```
%cpcm
  dracoisodens true # default = false
end
```

2.13.6 OpenCOSMO-RS

ORCA is interfaced to openCOSMO-RS,[142, 163] an open source implementation of the COSMO-RS model.[164, 165] This model is widely used in both academia and industry to predict fluid phase thermodynamics.

The main idea behind COSMO-RS is that the interaction between molecules in the liquid phase can be depicted as an ensemble of interacting surface segments. The properties on the surface segments are calculated via QM calculations of the solute and the solvent in a perfect conductor ($\epsilon = \infty$), and the interaction free energies between segments are functions of a set of descriptors. Among them, the most relevant one is the screening charge density (σ).

A complete description of the COSMO-RS model used in ORCA is provided in refs [142, 163], and the code is available in <https://github.com/TUHH-TVT>. In particular, the parametrization for ORCA 6.0 is called openCOSMO-RS 24a. The corresponding executable is shipped together with the ORCA 6.0 binaries and called openCOSMORS. ORCA will then call internally this executable whenever it is needed.

COSMO-RS calculations within ORCA are a special type of calculation. By requesting COSMO-RS, ORCA runs a set of single-point-energy calculations for both the solute and the solvent and then calls the openCOSMO-RS executable, which expects an input geometry for the solute and the solvent and calculates the free energy of solvation of the solute in the solvent. A workflow of a calculation requesting COSMO-RS in ORCA is the following:

1. Single-point-energy calculation of the solute in the gas-phase
2. Single-point-energy calculation of the solute in a conductor ($\epsilon = \infty$)
3. Single-point-energy calculation of the solvent in a conductor ($\epsilon = \infty$)
4. Do COSMO-RS via the openCOSMO-RS executable and compute solvation properties

Points 1 to 3 correspond to DFT calculations at the BP86/def2-TZVPD level. This is the level of theory used to parametrize COSMO-RS for ORCA 6.0. As it is shown later, the user can change the functional and basis set, but this is not recommended!