Hybrid Manual

Amsterdam Modeling Suite 2026.1

www.scm.com

Apr 02, 2026

CONTENTS

CHAPTER

ONE

INTRODUCTION

The Hybrid engine allows the user to set up multi-scale or multi-layered simulations by combining the results of a set of other AMS engines for different regions of the molecular system. The interface is like that of any other AMS engine, with a range of settings as input, and a potential energy and gradients as the main output. As such the Hybrid engine can be used in any AMS application such as geometry optimization, molecular dynamics, etc.

The Hybrid engine can be used for a wide range of multi-scale simulations, involving many different layers of the molecular system and many different levels of accuracy. The most commonly used multi-scale setup is a combined quantum mechanical / molecular mechanical (QM/MM) scheme. Such a QM/MM scheme divides the system into two different regions; (i) a QM region that is described by a QM engine such as ADF or DFTB, and (ii) an MM region that is described by an MM engine such as the ForceField engine. Within the QM/MM definition the user can choose between mechanical embedding and electrostatic embedding, and the choice determines which sub-options and which sub-engines are available.

In the following, first the broad use of the Hybrid engine for a flexible linear combination of engine results is outlined. Second, the QM/MM feature with its two available options (mechanical embedding and electrostatic embedding) is described in detail.

See also:

Tutorial: Inorganic linker in organic framework

1.1 Linear Combination of Energy Terms

Longtime users of the SCM software may know this functionality as QUILD. It can be used to set up a multi-layer computation according to the ONIOM scheme developed by Morokuma, but it has applicability beyond only layered setups. In general, this flexible and straight-forward feature provides a linear combination of results from different engines computed for different regions of the molecular system.

$$E = \sum_{i}^{N} w_i \; E^{\text{engine(i)}}\big(\text{region(i)}\big)$$

The engines may be all the same, and the regions can overlap or even be identical. Each individual energy is scaled by a weight. Most commonly, the weights will sum to one (the energy is a weighted average of its components), but this is not a requirement of the engine.

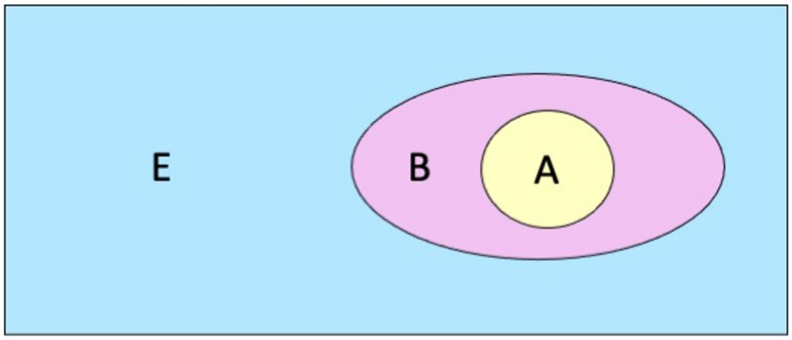
There are three main reasons to use such a linear combination of energy terms: to speed up a calculation, make it more accurate, or do some state averaging.

1) When the molecular system is large, efficiency may be improved by a layered setup, describing only a small region of interest with an accurate engine, while interaction with the environment is described with a more efficient engine.
2) DFT results can sometimes be improved by applying different functionals to different (or the same) regions.

3) Experimental results may be more accurately represented by a weighted average over several spin or excited states.

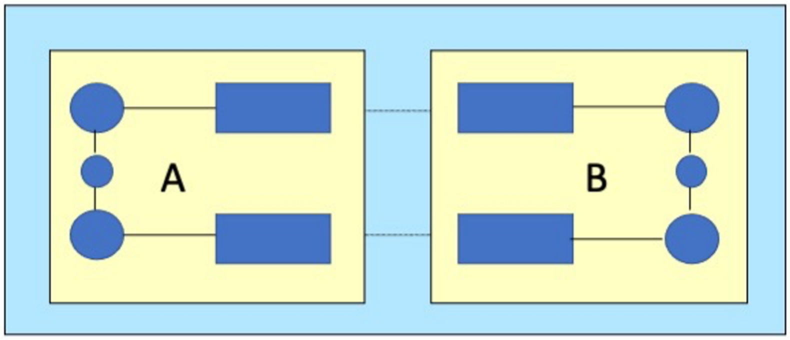
In a three-layered setup (Figure: active region of interest A in yellow, immediate surroundings B in pink, and long-range environment E in blue), the ONIOM energy expression contains five different terms, and uses three different engines (engines 1, 2, 3, in decreasing order of accuracy).

$$E = E^{\text{engine(3)}}(\text{A+B+C}) - E^{\text{engine(3)}}(\text{A+B}) + E^{\text{engine(2)}}(\text{A+B}) - E^{\text{engine(2)}}(\text{A}) + E^{\text{engine(1)}}(\text{A})$$



For systems like DNA, a GGA functional may best describe the intramolecular interactions (Figure: regions A and B), while the hydrogen bonds between regions are better described with an LDA functional. The corresponding energy expression contains five terms, and uses two different engine-setups (engines 1 and 2 for GGA and LDA respectively).

$$E = E^{\text{engine(1)}}(\text{A}) + E^{\text{engine(1)}}(\text{B}) + E^{\text{engine(2)}}(\text{A+B}) - E^{\text{engine(2)}}(\text{A}) - E^{\text{engine(2)}}(\text{B})$$



The Linear Combination of Energy Terms feature is enabled with the `Energy` block in the Hybrid engine input. Because of its simplicity the feature works with any combination of sub-engines.

Example

In this two-layer example, a water molecule at the DFTB level is embedded in an environment of a single water molecule at the UFF level. The energy is a linear combination of three different energy terms, with respective weights $w = (1, 1, -1)$.

$$E = E^{\text{uff}}(\text{QM+MM}) + E^{\text{dftb}}(\text{QM}) - E^{\text{uff}}(\text{QM})$$

```
Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM
        H -0.9986  0.2961 -0.3861  region=QM
        H -1.8623 -0.9560 -0.6510  region=QM
```

(continues on next page)

(continued from previous page)

```
        O  0.0121 -1.3731  0.5074  region=MM
        H  0.8930 -1.7879  0.3172  region=MM
        H -0.5625 -2.1395  0.7656  region=MM
    End
End

Engine Hybrid

    Energy
       Term region=QM EngineId=DFTB       factor=1.0
       Term region=*  EngineId=ForceField factor=1.0
       Term region=QM EngineId=ForceField factor=-1.0
    End

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type UFF
    EndEngine
EndEngine
```

Observe the following

- Regions are defined as usual via the `System` block
- The energy expression is defined in the `Energy` block, with multiple occurrences of the `Term` subkey
- The `Hybrid` engine block contains subblocks of other engines that are to be used
- The (sub) engines are referred to by an *EngineId* (page 11), usually the name of the engine, in this case DFTB and ForceField.

1.2 QM/MM

The Linear Combination of Energy Terms feature is very flexible, and can combine a multitude of layers and levels of accuracy. However, the most common multi-scale scheme is a two-layer scheme, with the active region of interest described at high accuracy, and the environment, which presumably has less impact on the desired outcome, described at lower accuracy. While the definition of high and low accuracy can vary (QM/QM', MM/MM', and even MM/QM combinations are possible), most often the high accuracy description is at electronic resolution (QM), while at low-accuracy the smallest particles described are atoms (MM).

1.2.1 Mechanical embedding

The mechanical embedding option is a short-cut to a subset of the possibilities provided by the flexible Linear Combination of Energy Terms described above. In this two-layer setup, the active region A is described with accurate engine (1), while the environment E (as well as the interactions between the two regions) is described with a more efficient engine (2). A subtractive scheme is used to compute the multi-scale energy and gradients. The energy expression involves a linear combination of only three energy terms.

$$E = E^{\text{engine(2)}}(\text{A+E}) + E^{\text{engine(1)}}(\text{A}) - E^{\text{engine(2)}}(\text{A})$$

As a result of this subtractive setup, any parameters required by engine(2) need to be provided for the full system. However, if engine(2) is the force field engine (which in QM/MM it will often be), then the corresponding energy terms are comprised of atom-pair interaction energies only. As a result, all contributions in energy-term 2 will cancel against the identical contributions in energy-term 1, which means that arbitrary parameters can be selected for region A energy contributions. The only remaining parameters of interest for engine(2) are those that describe the interaction inside region E and between regions A and E.

Example

As in the previous example, a water molecule at the DFTB level is embedded in an environment of a single water molecule at the UFF level. The computation is completely equivalent to the example for the Linear Combination of Energy Terms feature, with the simplified input as the only difference.

```
Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM
        H -0.9986  0.2961 -0.3861  region=QM
        H -1.8623 -0.9560 -0.6510  region=QM
        O  0.0121 -1.3731  0.5074  region=MM
        H  0.8930 -1.7879  0.3172  region=MM
        H -0.5625 -2.1395  0.7656  region=MM
    End
End

Engine Hybrid
    QMMM
       Embedding Mechanical
       mmEngineID ForceField
       qmEngineID DFTB
       qmRegion QM
    End

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type UFF
    EndEngine
EndEngine
```

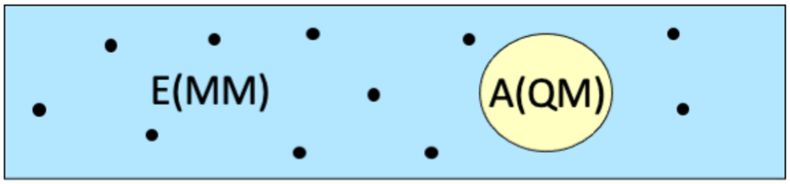
The (electrostatic) interactions between regions are described only at the lowest level of accuracy, so that the electron density in region A is not polarized by the charges in region B.

Supported engines

Like the Linear Combination of Energy Terms feature, the mechanical embedding option goes beyond QM/MM in that any combination of engines is implemented (QM/QM', MM/MM', MM/QM).

1.2.2 Electrostatic embedding

Electrostatic embedding is the more common application of two-layer QM/MM, where active region A is assumed to have an electron density, and this density is polarized by the point charges in environment region E.



A purely subtractive energy expression is no longer possible, as the interaction energy is not determined by a single description (QM or MM), but by a combination of both.

$$E = E^{\text{QM}}(\text{A}) + E^{\text{int}}(\text{A}|\text{E}) + E^{\text{MM}}(\text{E})$$

In practice, atomic point charges from the lower-level engine (usually an MM force field), are passed to the higher level QM engine, which computes the interaction of the electron density in region A with the point charges from region E.

$$E^{\text{int/elstat}}(\text{A}|\text{E}) = \int \rho^{\text{A}}(\boldsymbol{r})\ V^{\text{E}}(\boldsymbol{r}) d\boldsymbol{r} = \int \rho^{\text{A}}(\boldsymbol{r}) \sum_{i}^{N_{\text{E}}} \frac{q_i^{\text{E}}}{|\boldsymbol{r} - \boldsymbol{r}_i^{\text{E}}|} d\boldsymbol{r}$$

Here V^{E} is the potential from the E-region point charges, ρ_{A} is the charge density of the A-region sub-system, which is affected by the positions of the E-region point charges, N_{E} is the number of E-region atoms, q_i^{E} is the point charge of E-region atom i, and $\boldsymbol{r}_i^{\text{E}}$ is the position of E-region atom i.

The remaining non electrostatic interactions are calculated at the MM level by a subtractive scheme.

$$E^{\text{int/nonelstat}}(\text{A}|\text{E}) = E^{\text{MM}}_{\text{nonelstat}}(\text{A+E}) - E^{\text{MM}}_{\text{nonelstat}}(\text{A}) - E^{\text{MM}}_{\text{nonelstat}}(\text{E})$$

The nonelectrostatic energy $E^{\text{MM}}_{\text{nonelstat}}$ is obtained by simply setting the MM point charges to zero.

The final QM/MM energy consists of five terms, computed with two different engines.

1) The MM energy of region E.
2) The QM energy of region A, including interaction with point charges region E.
3) The MM energy of full system without electrostatics.
4) The MM energy of region E without electrostatics.
5) The MM energy of region A without electrostatics.

$$E = E^{MM}(E) + E^{QM}\left(\text{A}(V^{\text{E}})\right) + E^{\text{MM}}_{\text{nonelstat}}(\text{A+E}) - E^{\text{MM}}_{\text{nonelstat}}(\text{A}) - E^{\text{MM}}_{\text{nonelstat}}(\text{E})$$

Currently, only non-polarizable forcefields are supported, so that the MM charges q_i^{E} are not affected by the QM potential.

Example

Again, a water molecule at the DFTB level is embedded in an environment of a single water molecule at the UFF level, this time using electrostatic embedding.

```
Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM
        H -0.9986  0.2961 -0.3861  region=QM
        H -1.8623 -0.9560 -0.6510  region=QM
        O  0.0121 -1.3731  0.5074  region=MM
        H  0.8930 -1.7879  0.3172  region=MM
        H -0.5625 -2.1395  0.7656  region=MM
    End
End

Engine Hybrid
    QMMM
       Embedding Electrostatic
       mmEngineID ForceField
       qmEngineID DFTB
       qmRegion QM
    End

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type UFF
    EndEngine
EndEngine
eor
```

Supported engines

In the electrostatic embedding QM/MM setup of the Hybrid engine, the sub-engines involved can have one of two “roles”: the QM role and the MM role.

Engines supporting the QM role:

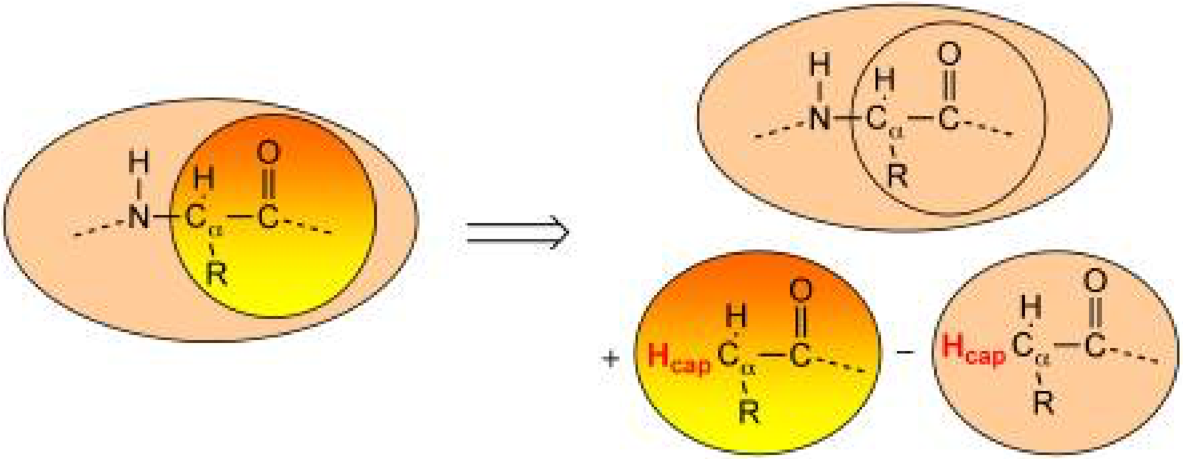
- adf
- band
- dftb

Engines supporting the MM role:

- forcefield

1.3 Capping Atoms

Whenever the boundary between two regions crosses a covalent bond (or better put, whenever the Hybrid engine notices that a sub-engine is assigned a system with dangling bonds), capping atoms - or link atoms - are assigned to satisfy the valence of the boundary atoms (see Figure below). By default, the engine adds hydrogen as capping atom, though a different (single) element can be selected by the user.



The capping atoms are added according to the AddRemove methodology [*1* (page 103)], in which the capping atoms follow the position of the real atoms in the total system. By default, the capping atoms are positioned along the vector of the dangling covalent bond, and at a distance that corresponds to the sum of the covalent radii of the capping atom and the boundary atom to which is has been attached. An alternative option can be selected that places the capping atom again along the vector of the dangling bond, but at a distance to the connected boundary atom that is a fraction of the bond between the two boundary atoms in the full system.

Capping can be disabled (via the *Energy* (page 12) or *QMMM* (page 14) block) and options can be set in the *Capping* (page 16) block. An element as well as force field atom-type and charge can be assigned to the capping atom. When more than one capping atom is present in the system, they will all have the same element, type, and charge.

1.3.1 Position of the Boundary

By default two sanity checks are performed on user-defined boundaries, and the following choices are not accepted by the engine.

- A boundary across bonds with a bond order larger than 1.25. The user can prevent this by using the `AllowHighBondOrders` subkey, see the *Capping* (page 16) documentation.
- A (QM) sub-region with more than one capping atom representing the same MM-boundary atom. The hybrid engine will not accept this, unless the user overrides it, see the `CheckCapping` subkey of *Capping* (page 16).

The GUI (amsinput) has an option (Make selection cappable, from the Select menu) to automatically extend a selection, so that the two checks will pass.

1.3.2 QM/MM Partitioning Examples

In a QM/MM simulation the basic question is how to partition the system into the QM and MM regions A and E. When studying an active site of a catalyst, for example, one must decide where to put the QM/MM boundary. Putting the boundary too close to the reaction center will jeopardize the chemical realism of the model. On the other hand, if one places the boundary too far away, the computational expense of the QM calculation may become problematic. Each system is different in this respect and the user must perform the proper tests to validate the appropriateness of the QM/MM partitioning used. We strongly suggest that the reader examines the literature on QM/MM methods and understands the basic limitations of the approach.

Below we give examples of the QM/MM partitioning choices that are by default prohibited by the engine. For comparison, we also give some representative examples of QM/MM partitioning that the engine does allow. In the examples, the region enclosed in the dotted polygon represents the A-region and the atoms labeled with 'LI' are the MM boundary atoms to be replaced by a capping - or link - atom.

In the examples in Figure 1, the boundary crosses double, triple, or aromatic bonds, so that a simple capping atom cannot satisfy the valence of the QM fragment and the electronic structure of the QM sub-system becomes drastically different from that of the full system.

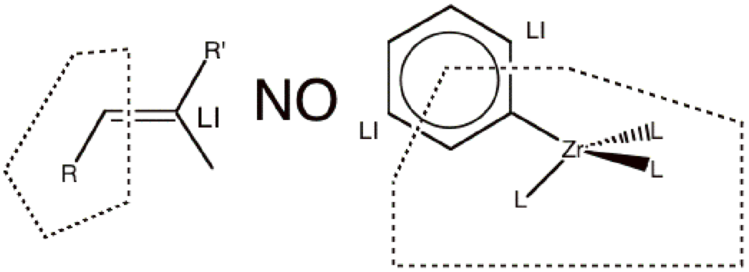
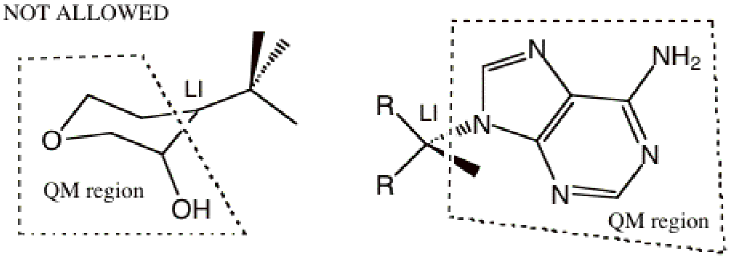


Figure 1 Examples of partitioning that should not be used because the link bonds are double or aromatic bonds.

Next, figure 2a depicts examples of partitioning that are not allowed because the MM boundary atom has a covalent bond to more than one QM atom. An MM boundary atom can only be bonded to one QM atom. Figure 2b shows the opposite, which is allowed. In other words, one QM atom can be bonded to more than one MM boundary atom. Note that there is no limit to the number of capping atoms that can be placed, just that each MM boundary atom can only be bonded to one QM atom.

a NOT ALLOWED



b ALLOWED

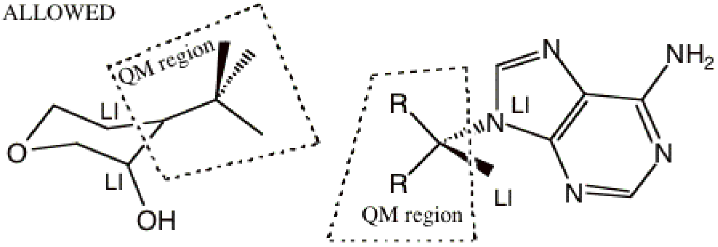


Figure 2 a) Examples of partitioning that are not allowed because the MM boundary atom has a covalent bond to more than one QM atom. b) The allowed reverse of the examples shown in (a). An MM boundary atom can **only** bond to one QM atom.

Then, figure 3 provides some representative examples of partitioning that the program does allow. Example **a** shows a typical solute-solvent QM/MM partitioning where there are no link bonds at all. Example **b** depicts two separate molecules each possessing a QM and a MM region. We emphasize that any number of molecules and link bonds can be used. Example **c** seems very similar to the earlier example in Figure 1. The difference is that the ring in Figure 3cd is not aromatic and consequently the link bonds in example **d** cross *single* bonds. Example **d** shows a single molecule, with two QM regions separated by an MM region. For this example, two equivalent pedagogic representations of the sample partitioning are displayed. Example **e** is a representative organometallic complex.

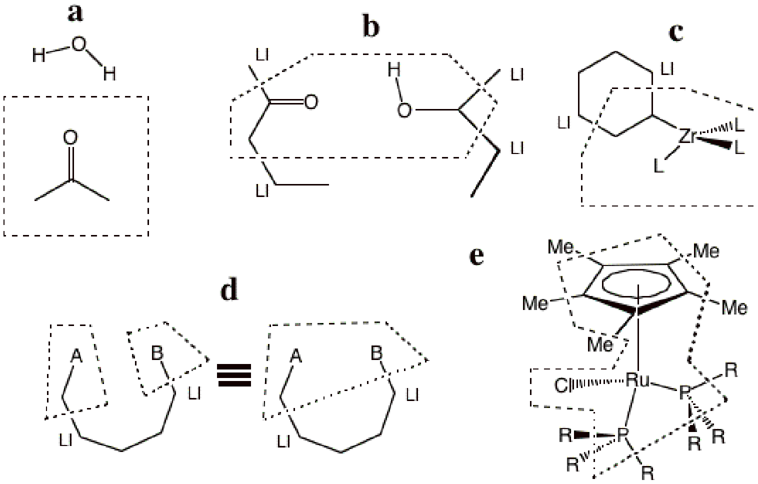


Figure 3 Representative examples of QM/MM partitioning that can be used in the Hybrid engine.

1.3.3 Electrostatic Embedding

Electrostatic embedding involving capping atoms can, without additional measures, result in unphysical behavior, such as nuclear fusion or extreme repulsion. The problem is the point charge of the MM boundary atom, which is located very close to the QM capping atom. To remedy this, our default implementation sets the charge at the position of the MM boundary atom to zero, and redistributes it over the remaining MM atoms while keeping the total charge (in the MM region) constant. While this avoids plain pathological behavior, the user should note that this alters the dipole moment of the MM region, which may have consequences for the behavior of the system.

In the electrostatic embedding setup, the sub-region E that is passed to the MM-engine is by default not provided with capping atoms, since the MM engine can handle an un-capped system. To compute the electrostatic embedding energy for a system with capping atoms, the energy expression needs to be slightly adjusted.

1) The MM energy of region E, without capping atoms.
2) The QM energy of capped A-region (A_C), including interaction with point charges region E.
3) The MM energy of full system without electrostatics.
4) The MM energy of region E without electrostatics and without capping atoms.
5) The MM energy of capped A-region (A_C) without electrostatics.

$$E = E^{MM}(E) + E^{QM}\big(\mathrm{A}_C(V^{\mathrm{E}})\big) + E^{\mathrm{MM}}_{\mathrm{nonelstat}}(\mathrm{A{+}E}) - E^{\mathrm{MM}}_{\mathrm{nonelstat}}(\mathrm{A}_C) - E^{\mathrm{MM}}_{\mathrm{nonelstat}}(\mathrm{E})$$

CHAPTER
TWO

HYBRID ENGINE OPTIONS

2.1 Sub-engines and EngineIDs

Inside the Hybrid engine input block one or more sub-engine blocks can be defined. These have exactly the same format as regular engine blocks. There is, however, one extra feature: the `EngineId`. Optionally, a unique name can be added to the engine definition as an extra string, serving as an identifier. By default the identifier is simply the engine name. The extra string allows the user to select the same engine multiple times, each time with different settings.

This is an example, where we use the same engine (ADF) with two different spin polarizations, using results corresponding to the lowest energy.

```
Engine Hybrid
    Energy # we want results from the engine that yields the lowest energy to be this␣
↪engine's result
       DynamicFactors UseLowestEnergy
       # The user-supplied factors are irrelevant here so we omit them
       Term EngineId=adf-singlet Region=*
       Term EngineId=adf-triplet Region=*
    End

    Engine ADF adf-singlet   # here adf-singlet is the EngineID
      SpinPolarization 0
      Unrestricted False
    EndEngine

    Engine ADF adf-triplet
      SpinPolarization 2
      Unrestricted True
    EndEngine

EndEngine
```

EngineIDs are for instance used in the technical example *QUILD* (page 85), which tests that EngineIDs are case-insensitive.

```
Engine header
```

`Engine`

Type
Block

Recurring
True

Description
The input for the computational (sub) engine. The header of the block determines the type of the engine. An optional second word in the header serves as the EngineID, if not present it defaults to the engine name. Currently it is not allowed to have a Hybrid engine as a sub engine.

2.2 Linear Combination of Energy Terms

The block `Energy` triggers a QUILD-like setup, allowing the energy to be defined as a linear combination of energy terms. Each energy term can be computed for a different subset of the molecular system, and with a different engine.

See the *basic QUILD example* (page 2). As you can see capping can be enabled per energy term, and the user can set a charge per term (for the corresponding region). One can also switch off the periodic boundary conditions for individual terms, using the `PBC` keyword. The latter can be used in a QM/MM setup (three energy terms), to have the interactions across the periodic boundary computed fully with the MM engine.

```
Energy
   DynamicFactors [Default | UseLowestEnergy | UseHighestEnergy]
   Term
      Charge float
      EngineID string
      Factor float
      PBC Yes/No
      Region string
      UseCappingAtoms Yes/No
   End
End
```

`Energy`

Type
Block

Description
This block is there to construct the energy.

`DynamicFactors`

Type
Multiple Choice

Default value
Default

Options
[Default, UseLowestEnergy, UseHighestEnergy]

GUI name
Adjust factors

Description
Default - use factors as set in the corresponding Term blocks;

UseLowestEnergy - set all factors to 0 except for that of the engine with the lowest energy, which is set to 1;

UseHighestEnergy - set all factors to 0 except for that of the engine with the highest energy, which is set to 1.

The last two options make sense only for non-QMMM hybrid calculation (that is, if the QMMM block is not present) and only when using engines whose energies can be compared directly.

`Term`

Type
Block

Recurring
True

Description
This block is there to construct the energy term. Can have multiple occurrences

`Charge`

Type
Float

Default value
0.0

Description
Net charge to be used for this energy term.

`EngineID`

Type
String

Description
Identifier for the engine

`Factor`

Type
Float

Default value
1.0

Description

`PBC`

Type
Bool

Default value
Yes

Description
If the full system has a periodic lattice, apply it to the region

`Region`

Type
String

Description
Identifier for the region

`UseCappingAtoms`

Type
Bool

Default value
Yes

Description
Whether to use capping for broken bonds

2.3 QM/MM

The alternative to the `Energy` block is the `QMMM` block, which triggers a two-layer computation. The embedding type can be selected with the `Embedding` key, for which mechanical or electrostatic can be selected. The former option triggers a specific linear combination of three energy terms, and can therefore also be set up using the `Energy` block (see *basic QUILD example* (page 2)).

See the *basic electrostatic embedding example* (page 5). Capping can be disabled, and charges can be set for the QM and MM regions.

```
QMMM
   Embedding [Mechanical | Electrostatic]
   MMCharge float
   MMEngineID string
   QMCharge float
   QMEngineID string
   QMRegion string
   UseCappingAtoms Yes/No
End
```

QMMM

Type
Block

Description
This block is there to identify the QMMM engines.

Embedding

Type
Multiple Choice

Default value
Electrostatic

Options
[Mechanical, Electrostatic]

Description
Determines how the QM region is embedded into the MM region.

Mechanical embedding embedding can also be achieved using the Energy%Terms keywords, but the common case of a two region mechanical QM/MM embedding is easier to set up using this keyword.

MMCharge

Type
Float

Default value
0.0

Description
Net charge to be used for the MM region.

`MMEngineID`

Type
String

Description
Identifier for the MM engine

`QMCharge`

Type
Float

Default value
0.0

Description
Net charge to be used for the QM region.

`QMEngineID`

Type
String

Description
Identifier for the QM engine

`QMRegion`

Type
String

Description
Identifier for the QM region. The rest of the system is considered the MM region.

`UseCappingAtoms`

Type
Bool

Default value
Yes

Description
Whether to use capping for broken bonds.

2.4 Committee

It is also possible to run the hybrid engine in `Committee` mode. When using the hybrid engine as a committee, the average is taken of each engine, meaning that all the regions and factors must be the same. Additionally, the spread of results of the different engines is understood as an uncertainty. The standard deviation of the different engine energies and forces are computed and reported as the uncertainty.

```
Committee
   Enabled Yes/No
End
```

`Committee`

Type
Block

Description
Settings for using the hybrid engine as a committee. The factors and region for each engine must be the same. When committee is enabled the standard deviation is also reported as the uncertainty.

`Enabled`

Type
Bool

Default value
No

Description
Enable committee

2.5 Capping

Whether of not capping is enabled is set inside the `Energy` and `QMMM` blocks. If enabled then the user can influence the position and type of the capping atom with the `Capping` sub-block.

```
Capping
   AllowHighBondOrders Yes/No
   AtomicInfoForCappingAtom string
   CappingElement string
   CheckCapping Yes/No
   Distance float
   Option [Fractional | Fixed]
End
```

`Capping`

Type
Block

Description
This block is about capping details. Capping occurs with hydrogen atoms when a bond is broken between an atom inside the region and one outside.

`AllowHighBondOrders`

Type
Bool

Default value
No

Description
Allows capping of interregional aromatic, double and triple bonds. This is normally not a good idea, since the capping is done with hydrogen atoms.

`AtomicInfoForCappingAtom`

Type
String

Default value
ForceField.Type=H_ ForceField.Charge=0.0

Description
The AtomicInfo for the capping atoms. Typically a string like ForceField.Type=X much like forcefield info is entered in the System block for normal atoms.

`CappingElement`

Type
String

Default value
H

Description
The element to be used for capping. The hydrogen atom has the advantage that it is very small.

`CheckCapping`

Type
Bool

Default value
Yes

Description
The same outside atom can be involved in multiple capping coordinate definitions. This is not a good idea, and this will not be accepted by using this check.

`Distance`

Type
Float

Default value
-1.0

Description
A negative value means automatic. In that case the sum of covalent radii is used

`Option`

Type
Multiple Choice

Default value
Fixed

Options
[Fractional, Fixed]

GUI name
Capping option

Description
The capping atom is always along the broken bond vector.

The bond distance between the capping atom and the two atoms are obtained from covalent radii, let us call them D1H and D2H.

With option=Fractional the capping is on the bond vector with the fraction D1H/(D1H+D2H).

With the Fixed option it at the distance D1H from atom 1. A distance of zero always means the coordinate of the inside atom.

For a specific application of QM/MM with capping atoms see this *example* (page 83).

2.6 Restarts

In a molecular dynamics run or geometry optimization, the geometries at subsequent steps are often very similar. Generally, efficiency can be gained by providing the engines with information from the previous step ("restart"), as this might speed up the SCF or charge equilibration procedure, if applicable. To the forcefield engine, this might avoid re-loading of the database, guessing bonds, etc. at every step. By default all sub-engines are provided with restart information. It can be switched off with the `RestartSubEngines` key.

```
RestartSubEngines Yes/No
```

`RestartSubEngines`

Type
Bool

Default value
Yes

Description
Save all the results of the subengines and pass those in a next geometry step or MD step.

2.7 Charges per region

The user can specify charges per region associated with each energy term.

Depending on the setup the charges can be set with the `Energy%Term%Charge` or the `QMMM%mmCharge` and `QMMM%mmCharge` keys.

For a QM engine the charge for a region determines the number of electrons in the region defined in the energy term. For the ForceField engine, charges are specified per atom, and they should add up to the charge specified for the region.

2.7.1 Linear Combination of Energy Terms

When the Linear Combination of Energy Terms feature is selected, but using the `Energy` block, the energy is a linear combination of independent calculations. It is in this spirit that the total charge is considered to be

$$Q = \sum_{i}^{N} w_i \, Q^{\text{engine(i)}}\big(region(i)\big)$$

In the QMMM setup the total charge is the sum of the charge of the mm region and the qm region.

When `Energy%DynamicFactors` is set to anything other than `Default` the factors specified in the Term blocks are not used.

2.7.2 Electrostatic Embedding

In general, the charges for sub-regions should be consistent with the charge specified for the total system. However, in the case of an electrostatic embedding computation with capping atoms, the sum of charges of the subsystems used in the computation of the five energy terms is allowed to deviate from the total system charge.

$$E = E^{MM}(E) + E^{QM}\big(\mathrm{A}_C(V^{\mathrm{E}})\big) + E^{\mathrm{MM}}_{\text{nonelstat}}(\mathrm{A{+}E}) - E^{\mathrm{MM}}_{\text{nonelstat}}(\mathrm{A}_C) - E^{\mathrm{MM}}_{\text{nonelstat}}(\mathrm{E})$$

The MM region that will be passed to the MM engine (term 1) will often have a fractional charge, due to un-capped dangling bonds. The fractional charge of the QM-region however (term 2), should be corrected by the capping atom charges, to yield a chemical system that optimally resembles the full system. As a result, the sub-region charges do not need to add up to the total charge of the system.

CHAPTER

THREE

PDB2ADF: TRANSFORM PDB FILE TO QMMM INPUT FILE

The pdb2adf utility program was written by Marcel Swart.

3.1 Overview

3.1.1 General description

Starting from the ADF2005.01 version the utility pdb2adf is available in the official release. Previously this utility could be found on the contributed software page. Starting from AMS2020 the default is to make an amsified ADF input, which can not be used with previous versions. One can get the old style input if the environment variable SCM_PDB2ADF is set to OLD (only to be used with ADF<=2019). One can get the NEWQMMM style input input if the environment variable SCM_PDB2ADF is set to NEW (only to be used with ADF<=2019).

The pdb2adf utility was written to read a PDB file (https://www.pdb.org), which contains the atomic coordinates of a protein structure, and transform it into an ADF inputfile, particularly for use with QMMM calculations. It can also be used for setting up a solvent shell around a solute molecule.

The PDB files are generally used for protein structures, and are formatted according to certain rules, see: https://www.wwpdb.org/docs.html, and the part about the official PDB format below.

For every residue/molecule present in the PDB file, there should be a fragment file available, either in the general AMS library ($AMSHOME/atomicdata/pdb2adf directory), or in the local directory where the pdb2adf program is being called. Fragment files in the local directory take higher priority than those in the general AMS library. The fragment files are formatted, based loosely on AMBER parameter files, and contain information about the residues; e.g., the atoms present, with their general and forcefield atomnames, atomic charges, connections to other atoms for creating their positions when not found on the PDB file, etc.; see part about fragment files below. Available in the AMS library are fragment files for amino acid residues, including those at the N- or C-terminal residue, three solvents (water, methanol, chloroform), some ions that are present frequently in protein structures (copper, fluoride), etc.

Also present in the AMS library are solvent box files that can be used to place a layer of solvents surrounding the protein, or a solute. Available are the three solvents mentioned above.

After reading the PDB and corresponding fragment files, the program tries to figure out which atoms are missing, and will add those; it uses the information provided on the fragment files to do so. For certain amino acid residues, there are several protonation states possible, e.g. histidine can be protonated at the N-delta position, at the N-epsilon position, or on both. The default option is to choose the fully charged option for aspartate (Asp), glutamate (Glu), lysine (Lys) residues, and decide for each histidine (His) and cysteine (Cys) residue individually what the protonation state should be. In those individual cases, the distances of neighboring molecules/residues are given that may help determine the protonation state. See the protein example below.

After all that is setup properly, a list is given with residue names/numbers, from which you can choose those that should be placed in the QM system; afterwards, for each of the selected QM residues, a choice should be made where to cut-off the QM part. The most appropriate point to cut-off seems to be at the C-alpha position, except when dealing with a proline (Pro). The latter residue is cyclic, e.g. the sidechain is connected to the C-alpha carbon ! For that residue, it may be better to include the C-alpha, H-alpha, and backbone carbonyl group of the preceding residue in the QM part.

The program will try to use to replace the ".pdb" extension of the PDB file by ".pdb2adf" for the AMS inputfile to be made; for convenience, the program also writes out an ".p2a.pdb" file with the complete system as it being made by the program. This file can then be visualized by conventional viewer programs (such as iMol, VMD, Molekel, AMSview) for visual inspection if everything has been carried out correctly.

Given below are two examples, one for the application of a protein, the other how to set up a solvent shell run.

3.1.2 Things to notice

- The QMMM implementation in AMS2020 is new. It uses the AMS Hybrid engine.
- By default pdb2adf makes an AMS Hybrid Engine input format.
- The pdb2adf program uses AMBER parameter files, and is setup to work with the AMBER force field, version AMBER95, which is designed for and works well for biosystems.
- For questions, remarks, contact: support@scm.com.

For ADF<=2019 only:

- The old style QMMM input format is used if the environment variable SCM_PDB2ADF is set to OLD.
- The NEWQMMM style QMMM input format is used if the environment variable SCM_PDB2ADF is set to NEW.

3.1.3 Official PDB format

Columns	Data Type	Field	Definition
1 - 6	Record name	'ATOM' or 'HETATM'	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	borderleft for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
73 - 76	LString(4)	segID	Segment identifier, left-justified.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

Typical examples from PDB-files:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

ATOM     76  O   GLY    A9       6.671  55.354  35.873  1.00 14.75           A
ATOM     77  N   ASN   A10       6.876  53.257  36.629  1.00 16.09           A
```

(continues on next page)

(continued from previous page)

```
ATOM     62  O   GLY A   9       6.791  55.214  35.719  1.00 15.61      4AZU 153
ATOM     63  N   ASN A  10       6.892  53.135  36.555  1.00 12.64      4AZU 154
```

The pdb2adf utility is flexible, and should be able to read most PDB files, even those with incomplete or erroneous line formats. From every ATOM/HETATM line, it tries to read:

- atom number
- atom name
- residuename
- chain identifier
- residue number
- X,Y,Z coordinates

Hints for proper formatting:

- always group together atoms that belong to one residue
- always give the atom name on columns 13-16
- when specifying a chain-id use only letters (or a blank)

3.1.4 Contents of fragment file

Given below is the contents of the fragment file for water. The first line is a comment line, the only important parameter is the NOCONNECT keyword, which indicates that the program should not try to make any connections to other residues/molecules. Then follow three lines, that define the orientation in space of the residue; they are not used for general fragments, but are relevant and important for amino acid residues and DNA nucleotides. Finally, for each atom in the molecule, there should be a line with its number in the fragment; its name to be used in PDB files; the AMBER forcefield atomtype; a dummy atomname; connections and coordinates (bond, angle, dihedral angle) to other atoms in the molecule that can be used to give the position of the atom if it is not present in the PDB file; the atomic charge; and after the exclamation mark (!) the connections to other atoms in this fragment, or other fragments in case of amino acid residues/DNA nucleotides. The current version does not use the latter connections yet, but the next version will probably use them.

```
HOH  Water molecule  NOCONNECT
   1   DUMM  DU    M      0   0   0       0.0000      0.0000      0.0000
   2   DUMM  DU    M      1   0   0       1.4490      0.0000      0.0000
   3   DUMM  DU    M      2   1   0       1.5220    111.1000      0.0000
   4   O     OW    O      0   0   0       0.0000      0.0000      0.0000  -0.8340  ! ␣
↪5  6
   5   H1    HW    H      4   0   0       0.9572      0.0000      0.0000   0.4170  ! ␣
↪4
   6   H2    HW    H      4   5   0       0.9572    104.5200      0.0000   0.4170  ! ␣
↪4
```

3.1.5 Contents of solvent box files

The first line is a comment line, followed by a line with the total number of atoms in the solvent box and the dimensions of the box (in Angstroms); then for each atom in the box, the atom name, which must match the PDB atomname, and the Cartesian coordinates, again in Angstroms.

3.2 Usage of pdb2adf

3.2.1 Short description

The program works interactively, and should be straightforwardly to use. However, for some of the stages in the output a short description is given below.

```
                                   P D B 2 A D F - program
                                      version 2008.01
                               Written by: Marcel Swart, 2008

                          This program uses AMBER parameter files
                             see: https://amber.scripps.edu

Do you want a logfile to be written (Y/n) ?
```

This option exists to create a logfile of what pdb2adf does. However, it should normally be used only for debugging purposes.

```
Ignoring atom on line:
ATOM    974   OH LYS A 128     -10.073 42.775 15.690 1.00 38.79     5AZU1065
```

This is a warning that the atom on that particular line is ignored, should normally occur only few times (less than ten). Depends also on how well the PDB file follows the PDB format rules.

```
Data Processed:
     Nat:       2519
    Nmol:        196
 NChains:          1
```

Information about what has been read on the PDB file: the total number of atoms (Nat), number of molecules/residues (Nmol) and number of protein chains (Nchains).

```
Please wait, making connection tables
```

At this point, the connections between the atoms are being made by looking at atom distances. It may take a while, depending on the size of the system.

```
Do you want to make separate files for each chain (Y/n) ?
```

You have the option to make different inputfiles for different protein chains, but you can also make one inputfile for all of them together.

```
Found the following terminal amino acid residues : (C-term) 128 (N-term) 1
Do you want to use these as terminal residues (Y/n) ?
```

Info is given about the C- and N-terminal residue of each chain. Reported for making sure they are chosen correctly. Note, if the C- and N-terminal residues are connected (rarely the case probably), enter N here.

```
Multiple AMBER options for HIS :
  0         Decide every time differently
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

Suggested option: 0
```

For a number of residues (His, Glu, Asp, Lys and Cys) there is more than one option available in the AMBER95 force field, depending on the protonation state (His, Glu, Asp and Lys) or the existence of a sulphur bridge/connection to a metal atom (Cys). The default is to choose a different option for the His and Cys residues, and use one option for Glu, Asp and Lys (fully charged). However, if wanted you can make a choice for all residues.

```
Multiple AMBER options for CYS    3 (    3) :
  1   CYS   Cysteine (SH)
  2   CYM   Deprotonated Cysteine (S-)
  3   CYX   Cystine (S-S bridge)

  Connections and Nearest Atoms for SG CYS    3 SG  ( P2A #   41 PDB#   20 )
      Dist   P2A Nr   PDB Nr   Label                    Near      Dist   P2A Nr   PDB Nr   Label
 1    1.82       38       19   CB   CYS    3   CB        1        3.79     2382      980   O  ␣
↪HOH  151   O
 2    2.02      461      193   SG   CYS   26   SG        2        3.80       22        0   HC ␣
↪GLN    2
                                                         3        4.04     2391      983   O  ␣
↪HOH  154   O
                                                         4        4.15      509      206   O  ␣
↪GLN   28   O
                                                         5        4.18      522        0   HA ␣
↪PHE   29
Suggestion: 3
```

The options for Cys3 are given, with information about the atoms bonded to the SG sulphur atom (on the left), as well as the closest five non-bonded atoms (on the right). This information may help you decide which choice to make for this particular residue. Also given (on the bottom) is the suggested choice, which is based, in this case, on the presence of a sulphur bridge.

```
Multiple AMBER options for HIS   46 (   46) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS   46 ND1 ( P2A #  844 PDB#  347 )
      Dist   P2A Nr   PDB Nr   Label                    Near      Dist   P2A Nr   PDB Nr   Label
 1    1.37      843      346   CG   HIS   46   CG        1        2.62     2166        0   H1 ␣
↪MET  121
 2    1.33      846      349   CE   HIS   46   CE1       2        3.23     2080      863   ND ␣
↪HIS  117   ND1
 3    2.04     2318      959   CU   CU   130   CU        3 HB     3.33     2163      900   S  ␣
↪MET  121   SD
                                                         4        3.40     2164      901   CT ␣
↪MET  121   CE
                                                         5        3.57     2082      865   CE ␣
↪HIS  117   CE1

  Connections and Nearest Atoms for NE HIS   46 NE2 ( P2A #  848 PDB#  350 )
      Dist   P2A Nr   PDB Nr   Label                    Near      Dist   P2A Nr   PDB Nr   Label
```

(continues on next page)

(continued from previous page)

```
 1    1.32     846      349  CE  HIS   46  CE1       1 HB    2.70      162       67  O  ␣
↪ASN   10  O
 2    1.37     850      348  CD  HIS   46  CD2       2       2.83      814        0  H1 ␣
↪MET   44
                                                     3       3.23     2166        0  H1 ␣
↪MET  121
                                                     4       3.52      822      332  O  ␣
↪MET   44  O
                                                     5       3.74      813      334  CT ␣
↪MET   44  CG
Suggestion: 2
```

For His residues, the information is given for both the delta- and the epsilon nitrogen atoms. Also indicated (by HB) is the presence of a hydrogen bond with another atom. The definition used here is that two atoms are hydrogen bonded if they are both non-carbon/non-hydrogen atoms, and the distance between them is less than the sum of the van der Waals radii of the atoms. It is a simple definition, but seems to be effective. In this case, as the N(delta) is bonded to copper, the proton should be attached to the N(epsilon).

```
Making choice for which molecules should be QM, which MM
```

Now we come to the part where the division in the QM and MM systems is made.

```
Residues belonging to chain  0
Option  Molecule    Option  Molecule    Option  Molecule    Option  Molecule   ␣
↪Option  Molecule
     1: ALA    1        28: GLN   28        55: ASP   55        82: ALA   82      ␣
↪109: ALA  109
     2: GLN    2        29: PHE   29        56: LYS   56        83: HIS   83      ␣
↪110: TYR  110
etc
```

All molecules/residues belonging to chain 0 are given, with an option number.

```
Give option number of molecules to be put in QM region (or 'c' to continue):
Note: by specifying a negative number a molecule is removed from the QM region
```

Here you are asked to enter the option numbers of the residues you want to put in the QM system.

```
Putting GLY   45 in QM region
Putting HIS   46 in QM region
```

In this case, Gly45 and His46 have been put in the QM system.

```
Make a choice for the QM/MM treatment of GLY   45
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 2
Give choice:
```

A choice should be made for where to cut-off the QM system. Normally this is done at the C(alpha) position, and you should simply choose the Suggestion.

```
Solvent molecules (SOL/HOH) belonging to this chain:
    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17␣
↪   18   19   20
   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37␣
↪   38   39   40
   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57␣
↪   58   59   60
   61   62   63   64   65   66

Give the number of the molecule to be put in QM region (or 'c' to continue):
```

Also water molecules can be put in the QM system.

```
Box Shape options:
 1  Spherical box
 2  Cubic box
Make a choice:
```

Type of box to be used.

```
Maximum atomic distance (Angs) from center       25.62
Give boxsize (def.:    28.62 Angs)
```

Size of box to be used to put a layer of solvent molecules around the system. Max. dist. is the maximal distance of any protein atom from the center of mass of the protein. Usually you should choose a boxsize at least 6 Angstrom larger (so at least two solvent molecules are surrounding the system).

…

```
Using BOXSIZE value of  30.0000
Adding atoms for box     1  Added (Box):     0 (Total):     0  Excl. (1):   648  Excl.
↪  (2):     0
Adding atoms for box     2  Added (Box):     9 (Total):     9  Excl. (1):   639  Excl.
↪  (2):     0
Adding atoms for box    63  Added (Box):     3 (Total):  7635  Excl. (1):   645  Excl.
↪  (2):     0
Adding atoms for box    64  Added (Box):     0 (Total):  7635  Excl. (1):   648  Excl.
↪  (2):     0
Writing inputfile for chain   1
```

A total amount of 7635 atoms (2545 water molecules) has been added.

```
Inputfile(s) written, everything processed, work has been done.
Thank you for using the PDB2ADF program.

================================
Normal ending of PDB2ADF program
================================
```

ADF inputfile(s) have been written, the PDB-file has been processes. Everything is done.

3.2.2 An example on protein structure

The idea of this example is to make an adf-input file using a PDB of an azurin (`1DYZ.pdb.txt`). The result of this example should be that in the adf-input file the active site of azurin (Figure 1) is in the QM part, and the rest of the protein is in the MM part, and that the solvent water is added (in a box), which is also in the MM part.

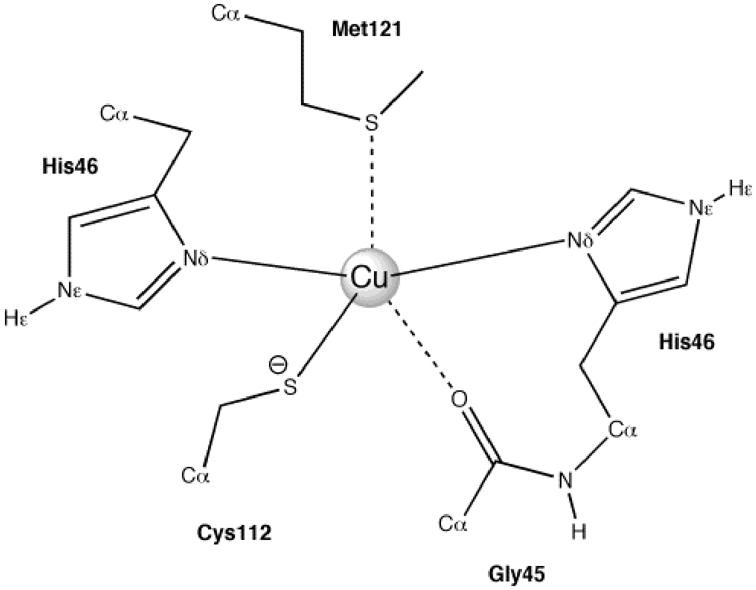


Figure 1: the active site of azurin

Usage of pdb2adf

The program works interactively. Given below in **bold** are the parts that the user has to type. In cases where the user agrees with the suggestion given by the program, the user can press the Enter key indicated with **Enter**.

```
                              P D B 2 A D F - program
                                 version 2008.01
                          Written by: Marcel Swart, 2008

                     This program uses AMBER parameter files
                        see: https://amber.scripps.edu

Please give name of PDB-file
```

1DYZ.pdb.txt

```
Do you want a logfile to be written (Y/n) ?
```

Enter

```
 read fragments

Data Processed:
     Nat:       2519
    Nmol:        196
 NChains:          1

Please wait, making connection tables
```

(continues on next page)

(continued from previous page)

```
Now finding nearby atoms
Assigning chain ID to all residues
Completing residues for which only option is available

Found the following terminal amino acid residues : (C-term)   129 (N-term)     1
Do you want to use these as terminal residues (Y/n) ?
```

Enter

```
Refinding nearby atoms (including atoms added in residue completion)

Multiple AMBER options for HIS :
  0         Decide every time differently
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

Suggested option: 0
```

Enter

```
Using 0: Decide every time differently

Multiple AMBER options for GLU :
  0         Decide every time differently
  1   GLU   Glutamic acid (COO-)
  2   GLH   Neutral Glutamic acid (COOH)

Suggested option: 1
```

Enter

```
Using 17   GLU   Glutamic acid (COO-)

Multiple AMBER options for ASP :
  0         Decide every time differently
  1   ASP   Aspartic acid (COO-)
  2   ASH   Neutral Aspartatic acid (COOH)

Suggested option: 1
```

Enter

```
Using 18   ASP   Aspartic acid (COO-)

Multiple AMBER options for LYS :
  0         Decide every time differently
  1   LYS   Charged Lysine (NH3+)
  2   LYN   Neutral Lysine (NH2)

Suggested option: 1
```

Enter

```
Using 19   LYS   Charged Lysine (NH3+)
```

(continues on next page)

(continued from previous page)

```
Multiple AMBER options for CYS :
  0         Decide every time differently
  1   CYS   Cysteine (SH)
  2   CYM   Deprotonated Cysteine (S-)
  3   CYX   Cystine (S-S bridge)

Suggested option: 0
```

Enter

```
Using 0: Decide every time differently
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
           Making Choices for Chain      0
 - - - - - - - - - - - - - - - - - - - - - - - - - - -

Multiple AMBER options for CYS    3 (    3) :
  1   CYS   Cysteine (SH)
  2   CYM   Deprotonated Cysteine (S-)
  3   CYX   Cystine (S-S bridge)

  Connections and Nearest Atoms for SG CYS    3 SG  ( P2A #   41 PDB#   20 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr  Label
 1    1.82      38      19  CB  CYS    3  CB       1       3.79    2382     980  O  ␣
 ↪HOH  151  O
 2    2.02     461     193  SG  CYS   26  SG       2       3.80      22       0  HC ␣
 ↪GLN    2
                                                   3       4.04    2391     983  O  ␣
 ↪HOH  154  O
                                                   4       4.15     509     206  O  ␣
 ↪GLN   28  O
                                                   5       4.18     522       0  HA ␣
 ↪PHE   29
Suggestion: 3
```

Enter

```
Multiple AMBER options for CYS   26 (   26) :
  1   CYS   Cysteine (SH)
  2   CYM   Deprotonated Cysteine (S-)
  3   CYX   Cystine (S-S bridge)

  Connections and Nearest Atoms for SG CYS   26 SG  ( P2A #  461 PDB#  193 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr  Label
 1    1.82     458     192  CB  CYS   26  CB       1       3.41     522       0  HA ␣
 ↪PHE   29
 2    2.02      41      20  SG  CYS    3  SG       2       3.43     411     168  O  ␣
 ↪ASP   23  O
                                                   3       3.60    2322     960  O  ␣
 ↪HOH  131  O
                                                   4       3.91     403     169  CB ␣
 ↪ASP   23  CB
                                                   5       4.15     387       0  HC ␣
 ↪VAL   22
Suggestion: 3
```

Enter

```
Multiple AMBER options for HIS   32 (   32) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS   32 ND1 ( P2A #  581 PDB#  244 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr Label
 1    1.39     580     243  CG  HIS   32  CG       1       3.41     545       0 HC ␣
↪THR   30
 2    1.33     583     246  CE  HIS   32  CE1      2       3.43      76      33 O  ␣
↪ALA    5  O
                                                   3       3.58      90      40 OH ␣
↪THR    6  OG1
                                                   4       3.99      91       0 HO ␣
↪THR    6
                                                   5       4.17      68       0 H  ␣
↪ALA    5

  Connections and Nearest Atoms for NE HIS   32 NE2 ( P2A #  585 PDB#  247 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr Label
 1    1.31     583     246  CE  HIS   32  CE1      1       2.86     544       0 HC ␣
↪THR   30
 2    1.37     587     245  CD  HIS   32  CD2      2       3.00     545       0 HC ␣
↪THR   30
                                                   3       3.14    1677       0 HO ␣
↪SER   94
                                                   4       3.42     542     229 CT ␣
↪THR   30  CG2
                                                   5       3.65    1676     688 OH ␣
↪SER   94  OG
Suggestion: 1
```

3

```
Multiple AMBER options for HIS   35 (   35) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS   35 ND1 ( P2A #  649 PDB#  271 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr Label
 1    1.38     648     270  CG  HIS   35  CG       1       2.46     682       0 H  ␣
↪GLY   37
 2    1.32     651     273  CE  HIS   35  CE1      2       2.69    1604       0 H1 ␣
↪GLY   89
                                                   3       3.31     681     282 N  ␣
↪GLY   37  N
                                                   4       3.56    1602     653 CT ␣
↪GLY   89  CA
                                                   5       3.67     152       0 H1 ␣
↪ASN   10

  Connections and Nearest Atoms for NE HIS   35 NE2 ( P2A #  653 PDB#  274 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr Label
 1    1.33     651     273  CE  HIS   35  CE1      1 HB    2.91     822     332 O  ␣
↪MET   44  O
 2    1.37     655     272  CD  HIS   35  CD2      2       3.24     814       0 H1 ␣
```

(continues on next page)

(continued from previous page)

```
↪MET   44
                                                   3       3.24     850     348  CD ␣
↪HIS   46  CD2
                                                   4       3.34    1593       0  H1 ␣
↪GLY   88
                                                   5       3.75     848     350  NE ␣
↪HIS   46  NE2
Suggestion: 2
```

3

```
Multiple AMBER options for HIS   46 (   46) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS   46 ND1 ( P2A #  844 PDB#  347 )
      Dist  P2A Nr  PDB Nr  Label                 Near      Dist  P2A Nr  PDB Nr  Label
 1    1.37     843      346  CG  HIS   46  CG       1       2.62    2166       0  H1 ␣
↪MET  121
 2    1.33     846      349  CE  HIS   46  CE1      2       3.23    2080     863  ND ␣
↪HIS  117  ND1
 3    2.04    2318      959  CU  CU   130  CU       3 HB    3.33    2163     900  S  ␣
↪MET  121  SD
                                                   4       3.40    2164     901  CT ␣
↪MET  121  CE
                                                   5       3.57    2082     865  CE ␣
↪HIS  117  CE1

  Connections and Nearest Atoms for NE HIS   46 NE2 ( P2A #  848 PDB#  350 )
      Dist  P2A Nr  PDB Nr  Label                 Near      Dist  P2A Nr  PDB Nr  Label
 1    1.32     846      349  CE  HIS   46  CE1      1 HB    2.70     162      67  O  ␣
↪ASN   10  O
 2    1.37     850      348  CD  HIS   46  CD2      2       2.83     814       0  H1 ␣
↪MET   44
                                                   3       3.23    2166       0  H1 ␣
↪MET  121
                                                   4       3.52     822     332  O  ␣
↪MET   44  O
                                                   5       3.74     813     334  CT ␣
↪MET   44  CG
Suggestion: 2
```

Enter

```
Multiple AMBER options for HIS   83 (   83) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS   83 ND1 ( P2A # 1494 PDB#  613 )
      Dist  P2A Nr  PDB Nr  Label                 Near      Dist  P2A Nr  PDB Nr  Label
 1    1.39    1493      612  CG  HIS   83  CG       1       2.67    1317       0  HC ␣
↪VAL   73
 2    1.33    1496      615  CE  HIS   83  CE1      2       3.63    1315     542  CT ␣
↪VAL   73  CG2
                                                   3       3.74    1310       0  HC ␣
```

(continues on next page)

(continued from previous page)

```
↪VAL   73
                                                         4       3.82    1316       0  HC ␣
↪VAL   73
                                                         5       3.86    1313       0  HC ␣
↪VAL   73

  Connections and Nearest Atoms for NE HIS   83 NE2 ( P2A # 1498 PDB#  616 )
      Dist  P2A Nr  PDB Nr  Label                   Near     Dist  P2A Nr  PDB Nr  Label
 1    1.32    1496     615  CE  HIS   83  CE1        1       3.09    1313       0  HC ␣
↪VAL   73
 2    1.38    1500     614  CD  HIS   83  CD2        2       3.44    1317       0  HC ␣
↪VAL   73
                                                         3       3.88    2385     981  O  ␣
↪HOH  152  O
                                                         4       3.93    1311     541  CT ␣
↪VAL   73  CG1
                                                         5       4.03    1309     540  CT ␣
↪VAL   73  CB
Suggestion: 2
```

3

```
Multiple AMBER options for CYS  112 (  112) :
  1   CYS   Cysteine (SH)
  2   CYM   Deprotonated Cysteine (S-)
  3   CYX   Cystine (S-S bridge)

  Connections and Nearest Atoms for SG CYS  112 SG  ( P2A # 2001 PDB#  828 )
      Dist  P2A Nr  PDB Nr  Label                   Near     Dist  P2A Nr  PDB Nr  Label
 1    1.82    1998     827  CB  CYS  112  CB         1       2.53     858       0  H  ␣
↪ASN   47
 2    2.14    2318     959  CU  CU   130  CU         2       2.65    2023       0  H  ␣
↪PHE  114
                                                         3       3.00    2028       0  HC ␣
↪PHE  114
                                                         4       3.29     868       0  H  ␣
↪ASN   47
                                                         5       3.39    2027       0  HC ␣
↪PHE  114
Suggestion: 2
```

Enter

```
Multiple AMBER options for HIS  117 (  117) :
  1   HID   Histidine Delta Hydrogen
  2   HIE   Histidine Epsilon Hydrogen
  3   HIP   Histidine E & D Hydrogens

  Connections and Nearest Atoms for ND HIS  117 ND1 ( P2A # 2080 PDB#  863 )
      Dist  P2A Nr  PDB Nr  Label                   Near     Dist  P2A Nr  PDB Nr  Label
 1    1.37    2079     862  CG  HIS  117  CG         1       2.82    2028       0  HC ␣
↪PHE  114
 2    1.34    2082     865  CE  HIS  117  CE1        2       3.23     844     347  ND ␣
↪HIS   46  ND1
 3    1.99    2318     959  CU  CU   130  CU         3       3.26    2031       0  HA ␣
↪PHE  114
                                                         4       3.27     832     340  O  ␣
```

(continues on next page)

(continued from previous page)

```
↪GLY   45  O
                                                   5       3.43     846     349  CE ␣
↪HIS   46  CE1

  Connections and Nearest Atoms for NE HIS  117 NE2 ( P2A # 2084 PDB#  866 )
      Dist  P2A Nr  PDB Nr  Label                 Near     Dist  P2A Nr  PDB Nr  Label
 1    1.31    2082     865  CE  HIS  117  CE1      1       2.57     209       0  H1 ␣
↪MET   13
 2    1.37    2086     864  CD  HIS  117  CD2      2       2.65    2031       0  HA ␣
↪PHE  114
                                                   3 HB    2.74    2406     988  O  ␣
↪HOH  159  O
                                                   4       3.34    2030     841  CA ␣
↪PHE  114  CD1
                                                   5       3.41     204       0  H1 ␣
↪MET   13
Suggestion: 2
```

Enter

```
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
          Making Choices for Chain      1
 - - - - - - - - - - - - - - - - - - - - - - - - - - -

Completing residues with multiple options available, and solvent molecules

Checking positions of newly added atoms

Making choice for which molecules should be QM, which MM

Residues belonging to chain  0
Option  Molecule    Option  Molecule    Option  Molecule    Option  Molecule    ␣
↪Option  Molecule
     1: ALA    1        28: GLN   28        55: ASP   55        82: ALA   82      ␣
↪109: ALA  109
     2: GLN    2        29: PHE   29        56: LYS   56        83: HIS   83      ␣
↪110: TYR  110
     3: CYS    3        30: THR   30        57: GLN   57        84: THR   84      ␣
↪111: PHE  111
     4: GLU    4        31: MET   31        58: ALA   58        85: LYS   85      ␣
↪112: CYS  112
     5: ALA    5        32: HIS   32        59: VAL   59        86: VAL   86      ␣
↪113: SER  113
     6: THR    6        33: LEU   33        60: ALA   60        87: ILE   87      ␣
↪114: PHE  114
     7: VAL    7        34: LYS   34        61: THR   61        88: GLY   88      ␣
↪115: PRO  115
     8: GLU    8        35: HIS   35        62: ASP   62        89: GLY   89      ␣
↪116: GLY  116
     9: SER    9        36: VAL   36        63: GLY   63        90: GLY   90      ␣
↪117: HIS  117
    10: ASN   10        37: GLY   37        64: MET   64        91: GLU   91      ␣
↪118: TRP  118
    11: ASP   11        38: LYS   38        65: GLY   65        92: SER   92      ␣
↪119: ALA  119
    12: ALA   12        39: MET   39        66: ALA   66        93: ASP   93      ␣
↪120: MET  120
```

(continues on next page)

(continued from previous page)

```
   13: MET   13         40: ALA   40         67: GLY   67         94: SER   94         ␣
↪121: MET  121
   14: GLN   14         41: LYS   41         68: LEU   68         95: VAL   95         ␣
↪122: LYS  122
   15: TYR   15         42: VAL   42         69: ALA   69         96: THR   96         ␣
↪123: GLY  123
   16: ASN   16         43: ALA   43         70: GLN   70         97: PHE   97         ␣
↪124: THR  124
   17: VAL   17         44: MET   44         71: ASP   71         98: ASP   98         ␣
↪125: LEU  125
   18: LYS   18         45: GLY   45         72: TYR   72         99: VAL   99         ␣
↪126: LYS  126
   19: GLU   19         46: HIS   46         73: VAL   73        100: SER  100         ␣
↪127: LEU  127
   20: ILE   20         47: ASN   47         74: LYS   74        101: LYS  101         ␣
↪128: GLY  128
   21: VAL   21         48: LEU   48         75: ALA   75        102: ILE  102         ␣
↪129: SER  129
   22: VAL   22         49: VAL   49         76: GLY   76        103: ALA  103         ␣
↪130: CU   130
   23: ASP   23         50: LEU   50         77: ASP   77        104: ALA  104
   24: LYS   24         51: THR   51         78: THR   78        105: GLY  105
   25: SER   25         52: LYS   52         79: ARG   79        106: GLU  106
   26: CYS   26         53: ASP   53         80: VAL   80        107: ASN  107
   27: LYS   27         54: ALA   54         81: ILE   81        108: TYR  108

Give option number of molecules to be put in QM region (or 'c' to continue):
Note: by specifying a negative number a molecule is removed from the QM region
```

45 46 112 117 121 130

```
Putting GLY   45 in QM region
Putting HIS   46 in QM region
Putting CYS  112 in QM region
Putting HIS  117 in QM region
Putting MET  121 in QM region
Putting CU   130 in QM region

Give option number of molecules to be put in QM region (or 'c' to continue):
Note: by specifying a negative number a molecule is removed from the QM region
```

c

```
Make a choice for the QM/MM treatment of GLY   45
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 2
Give choice:
```

Enter

```
Make a choice for the QM/MM treatment of HIS   46
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 1
Give choice:
```

Enter

```
Make a choice for the QM/MM treatment of CYS  112
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 3
Give choice:
```

Enter

```
Make a choice for the QM/MM treatment of HIS  117
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 3
Give choice:
```

Enter

```
Make a choice for the QM/MM treatment of MET  121
 0:  Put completely in QM region
 1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
 2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
 3:  Cut off at C-alpha (put NH and CO in MM region)
 4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
 5:  Put only part of sidechain in QM region

Suggestion: 3
Give choice:
```

Enter

```
Make a choice for the QM/MM treatment of CU   130
 0:  Put completely in QM region
 1:  Put only part of molecule in QM region

Suggestion: 0
Give choice:
```

Enter

```
Total formal charge on molecule CU    130      2.0000

Solvent molecules (SOL/HOH) belonging to this chain:
    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17␣
↪  18   19   20
   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37␣
↪  38   39   40
   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57␣
↪  58   59   60
   61   62   63   64   65   66

Give the number of the molecule to be put in QM region (or 'c' to continue):
```

c

```
Residues belonging to chain  1

Do you want to add solvent to your system (Y/n) ?
```

Enter

```
Solvent (box) available:
   1:  HOH      HOH  Water molecule
   2:  MOH      MOH  Methanol molecule
   3:  CHL      CHL  Chloroform molecule
```

1

```
Reading contents of solvent box p2abox.HOH

Box Shape options:
 1  Spherical box
 2  Cubic box
Make a choice:
```

1

```
Writing inputfile for chain   0

Using total charge   1.0 and total spin   1.0

Maximum atomic distance (Angs) from center       25.62
Give boxsize (def.:    28.62 Angs)
```

30.0

```
Using BOXSIZE value of  30.0000
Adding atoms for box     1  Added (Box):     0 (Total):     0  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box     2  Added (Box):     9 (Total):     9  Excl. (1):   639  Excl.
↪ (2):     0
Adding atoms for box     3  Added (Box):     3 (Total):    12  Excl. (1):   645  Excl.
↪ (2):     0
Adding atoms for box     4  Added (Box):     0 (Total):    12  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box     5  Added (Box):     6 (Total):    18  Excl. (1):   642  Excl.
```

(continues on next page)

(continued from previous page)

```
↪ (2):     0
Adding atoms for box    6  Added (Box):   228 (Total):   246  Excl. (1):   420  Excl.
↪ (2):     0
Adding atoms for box    7  Added (Box):   219 (Total):   465  Excl. (1):   429  Excl.
↪ (2):     0
Adding atoms for box    8  Added (Box):     9 (Total):   474  Excl. (1):   639  Excl.
↪ (2):     0
Adding atoms for box    9  Added (Box):     0 (Total):   474  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box   10  Added (Box):   225 (Total):   699  Excl. (1):   423  Excl.
↪ (2):     0
Adding atoms for box   11  Added (Box):   216 (Total):   915  Excl. (1):   432  Excl.
↪ (2):     0
Adding atoms for box   12  Added (Box):     6 (Total):   921  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box   13  Added (Box):     0 (Total):   921  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box   14  Added (Box):     6 (Total):   927  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box   15  Added (Box):    12 (Total):   939  Excl. (1):   636  Excl.
↪ (2):     0
Adding atoms for box   16  Added (Box):     0 (Total):   939  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box   17  Added (Box):    12 (Total):   951  Excl. (1):   636  Excl.
↪ (2):     0
Adding atoms for box   18  Added (Box):   210 (Total):  1161  Excl. (1):   438  Excl.
↪ (2):     0
Adding atoms for box   19  Added (Box):   219 (Total):  1380  Excl. (1):   429  Excl.
↪ (2):     0
Adding atoms for box   20  Added (Box):     3 (Total):  1383  Excl. (1):   645  Excl.
↪ (2):     0
Adding atoms for box   21  Added (Box):   216 (Total):  1599  Excl. (1):   417  Excl.
↪ (2):    15
Adding atoms for box   22  Added (Box):   381 (Total):  1980  Excl. (1):     3  Excl.
↪ (2):   264
Adding atoms for box   23  Added (Box):   261 (Total):  2241  Excl. (1):     3  Excl.
↪ (2):   384
Adding atoms for box   24  Added (Box):   183 (Total):  2424  Excl. (1):   423  Excl.
↪ (2):    42
Adding atoms for box   25  Added (Box):   189 (Total):  2613  Excl. (1):   426  Excl.
↪ (2):    33
Adding atoms for box   26  Added (Box):   186 (Total):  2799  Excl. (1):     3  Excl.
↪ (2):   459
Adding atoms for box   27  Added (Box):   351 (Total):  3150  Excl. (1):     3  Excl.
↪ (2):   294
Adding atoms for box   28  Added (Box):   222 (Total):  3372  Excl. (1):   420  Excl.
↪ (2):     6
Adding atoms for box   29  Added (Box):     9 (Total):  3381  Excl. (1):   639  Excl.
↪ (2):     0
Adding atoms for box   30  Added (Box):   162 (Total):  3543  Excl. (1):   429  Excl.
↪ (2):    57
Adding atoms for box   31  Added (Box):   219 (Total):  3762  Excl. (1):   426  Excl.
↪ (2):     3
Adding atoms for box   32  Added (Box):     6 (Total):  3768  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box   33  Added (Box):     6 (Total):  3774  Excl. (1):   642  Excl.
↪ (2):     0
```

(continues on next page)

(continued from previous page)

```
Adding atoms for box    34  Added (Box):   219 (Total):  3993  Excl. (1):   426  Excl.
↪ (2):     3
Adding atoms for box    35  Added (Box):   216 (Total):  4209  Excl. (1):   432  Excl.
↪ (2):     0
Adding atoms for box    36  Added (Box):     6 (Total):  4215  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    37  Added (Box):   219 (Total):  4434  Excl. (1):   426  Excl.
↪ (2):     3
Adding atoms for box    38  Added (Box):   279 (Total):  4713  Excl. (1):     6  Excl.
↪ (2):   363
Adding atoms for box    39  Added (Box):   231 (Total):  4944  Excl. (1):     0  Excl.
↪ (2):   417
Adding atoms for box    40  Added (Box):   195 (Total):  5139  Excl. (1):   432  Excl.
↪ (2):    21
Adding atoms for box    41  Added (Box):   231 (Total):  5370  Excl. (1):   414  Excl.
↪ (2):     3
Adding atoms for box    42  Added (Box):   324 (Total):  5694  Excl. (1):     0  Excl.
↪ (2):   324
Adding atoms for box    43  Added (Box):   408 (Total):  6102  Excl. (1):     6  Excl.
↪ (2):   234
Adding atoms for box    44  Added (Box):   204 (Total):  6306  Excl. (1):   435  Excl.
↪ (2):     9
Adding atoms for box    45  Added (Box):     6 (Total):  6312  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    46  Added (Box):   177 (Total):  6489  Excl. (1):   435  Excl.
↪ (2):    36
Adding atoms for box    47  Added (Box):   219 (Total):  6708  Excl. (1):   429  Excl.
↪ (2):     0
Adding atoms for box    48  Added (Box):     6 (Total):  6714  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    49  Added (Box):     0 (Total):  6714  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box    50  Added (Box):     3 (Total):  6717  Excl. (1):   645  Excl.
↪ (2):     0
Adding atoms for box    51  Added (Box):     6 (Total):  6723  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    52  Added (Box):     0 (Total):  6723  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box    53  Added (Box):     9 (Total):  6732  Excl. (1):   639  Excl.
↪ (2):     0
Adding atoms for box    54  Added (Box):   222 (Total):  6954  Excl. (1):   426  Excl.
↪ (2):     0
Adding atoms for box    55  Added (Box):   213 (Total):  7167  Excl. (1):   426  Excl.
↪ (2):     9
Adding atoms for box    56  Added (Box):     6 (Total):  7173  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    57  Added (Box):     3 (Total):  7176  Excl. (1):   645  Excl.
↪ (2):     0
Adding atoms for box    58  Added (Box):   219 (Total):  7395  Excl. (1):   423  Excl.
↪ (2):     6
Adding atoms for box    59  Added (Box):   219 (Total):  7614  Excl. (1):   429  Excl.
↪ (2):     0
Adding atoms for box    60  Added (Box):     6 (Total):  7620  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box    61  Added (Box):     0 (Total):  7620  Excl. (1):   648  Excl.
↪ (2):     0
Adding atoms for box    62  Added (Box):    12 (Total):  7632  Excl. (1):   636  Excl.
```

(continues on next page)

(continued from previous page)

```
↪ (2):     0
Adding atoms for box    63  Added (Box):     3 (Total):  7635  Excl. (1):   645  Excl.
↪ (2):     0
Adding atoms for box    64  Added (Box):     0 (Total):  7635  Excl. (1):   648  Excl.
↪ (2):     0

Total spin   1.0

Writing inputfile for chain   1
There are no atoms in this chain, ignoring it

Inputfile(s) written, everything processed, work has been done.
Thank you for using the PDB2ADF program.

================================
Normal ending of PDB2ADF program
================================
```

Contents of the 1DYZ.pdb2adf file generated by pdb2adf

The file is not given completely, since it contains more than 9000 atoms.

```
#! /bin/sh

$AMSBIN/ams << eor
System
  Charge   1.0
  Atoms
    N     -1.1930    25.6890    17.1840 region=MM ForceField.Charge=.141400   ␣
↪ForceField.Type=N3   !     1 ALA      1 N
    H     -0.3133    25.1929    17.1970 region=MM ForceField.Charge=.199700   ␣
↪ForceField.Type=H    !     2 ALA      1 H1
    H     -1.3738    25.1438    18.0148 region=MM ForceField.Charge=.199700   ␣
↪ForceField.Type=H    !     3 ALA      1 H2
    H     -1.5170    24.8559    16.7138 region=MM ForceField.Charge=.199700   ␣
↪ForceField.Type=H    !     4 ALA      1 H3
    C     -1.4820    27.1340    16.8960 region=MM ForceField.Charge=.096200   ␣
↪ForceField.Type=CT   !     5 ALA      1 CA
    H     -2.1350    27.2082    16.0264 region=MM ForceField.Charge=.088900   ␣
↪ForceField.Type=HP   !     6 ALA      1 HA
    C     -2.1950    27.7860    18.0880 region=MM ForceField.Charge=-.059700  ␣
↪ForceField.Type=CT   !     7 ALA      1 CB
    H     -1.5602    27.7210    18.9717 region=MM ForceField.Charge=.030000   ␣
↪ForceField.Type=HC   !     8 ALA      1 HB1
    H     -2.3971    28.8331    17.8627 region=MM ForceField.Charge=.030000   ␣
↪ForceField.Type=HC   !     9 ALA      1 HB2
    H     -3.1350    27.2677    18.2776 region=MM ForceField.Charge=.030000   ␣
↪ForceField.Type=HC   !    10 ALA      1 HB3
    C     -0.1820    27.8790    16.5880 region=MM ForceField.Charge=.616300   ␣
↪ForceField.Type=C    !    11 ALA      1 C
    O      0.8890    27.4920    17.0690 region=MM ForceField.Charge=-.572200  ␣
↪ForceField.Type=O    !    12 ALA      1 O
    N     -0.2890    28.9420    15.7940 region=MM ForceField.Charge=-.415700  ␣
↪ForceField.Type=N    !    13 GLN      2 N
    ...
```

(continues on next page)

(continued from previous page)

```
    H    11.6901     6.5638    30.5231 region=MM ForceField.Charge=.271900    ␣
↪ForceField.Type=H    !   690 GLY     45 H
    C    11.3760     8.5410    29.7530 region=QM ForceField.Charge=-.025200   ␣
↪ForceField.Type=CT   !   691 GLY     45 CA
    H    10.9114     9.3322    30.3413 region=QM ForceField.Charge=.069800    ␣
↪ForceField.Type=H1   !   692 GLY     45 HA2
    H    12.4602     8.6423    29.8009 region=QM ForceField.Charge=.069800    ␣
↪ForceField.Type=H1   !   693 GLY     45 HA3
    C    10.9630     8.7450    28.3090 region=QM ForceField.Charge=.597300    ␣
↪ForceField.Type=C    !   694 GLY     45 C
    O    10.8510     7.7910    27.5300 region=QM ForceField.Charge=-.567900   ␣
↪ForceField.Type=O    !   695 GLY     45 O
    N    10.6890     9.9800    27.9260 region=QM ForceField.Charge=-.415700   ␣
↪ForceField.Type=N    !   696 HIS     46 N
    H    10.7572    10.7382    28.5898 region=QM ForceField.Charge=.271900    ␣
↪ForceField.Type=H    !   697 HIS     46 H
    C    10.2900    10.2500    26.5530 region=QM ForceField.Charge=-.058100   ␣
↪ForceField.Type=CT   !   698 HIS     46 CA
    H    10.5517     9.3991    25.9240 region=QM ForceField.Charge=.136000    ␣
↪ForceField.Type=H1   !   699 HIS     46 HA
    C     8.7770    10.5120    26.4440 region=QM ForceField.Charge=-.007400   ␣
↪ForceField.Type=CT   !   700 HIS     46 CB
    H     8.5050    11.3473    27.0893 region=QM ForceField.Charge=.036700    ␣
↪ForceField.Type=HC   !   701 HIS     46 HB2
    H     8.5229    10.7532    25.4118 region=QM ForceField.Charge=.036700    ␣
↪ForceField.Type=HC   !   702 HIS     46 HB3
    C     7.9110     9.3590    26.8430 region=QM ForceField.Charge=.186800    ␣
↪ForceField.Type=CC   !   703 HIS     46 CG
    N     8.0710     8.0910    26.3490 region=QM ForceField.Charge=-.543200   ␣
↪ForceField.Type=NB   !   704 HIS     46 ND1
    C     7.1230     7.3010    26.8370 region=QM ForceField.Charge=.163500    ␣
↪ForceField.Type=CR   !   705 HIS     46 CE1
    H     7.0894     6.2496    26.5516 region=QM ForceField.Charge=.143500    ␣
↪ForceField.Type=H5   !   706 HIS     46 HE1
    N     6.3580     8.0230    27.6330 region=QM ForceField.Charge=-.279500   ␣
↪ForceField.Type=NA   !   707 HIS     46 NE2
    H     5.5568     7.6742    28.1395 region=QM ForceField.Charge=.333900    ␣
↪ForceField.Type=H    !   708 HIS     46 HE2
    C     6.8210     9.3110    27.6620 region=QM ForceField.Charge=-.220700   ␣
↪ForceField.Type=CW   !   709 HIS     46 CD2
    H     6.3141    10.0588    28.2719 region=QM ForceField.Charge=.186200    ␣
↪ForceField.Type=H4   !   710 HIS     46 HD2
    C    10.9790    11.4950    26.0450 region=MM ForceField.Charge=.597300    ␣
↪ForceField.Type=C    !   711 HIS     46 C
    ...
    C    11.0290     8.8020    20.9600 region=QM ForceField.Charge=.035000    ␣
↪ForceField.Type=CT   !  1648 CYS    112 CA
    H    11.3902     9.8061    21.1823 region=QM ForceField.Charge=.048000    ␣
↪ForceField.Type=H1   !  1649 CYS    112 HA
    C    10.0620     8.3640    22.0630 region=QM ForceField.Charge=-.736000   ␣
↪ForceField.Type=CT   !  1650 CYS    112 CB
    H     9.2477     9.0845    22.1402 region=QM ForceField.Charge=.244000    ␣
↪ForceField.Type=H1   !  1651 CYS    112 HB3
    H     9.6557     7.3817    21.8218 region=QM ForceField.Charge=.244000    ␣
↪ForceField.Type=H1   !  1652 CYS    112 HB2
    S    10.8340     8.2410    23.7100 region=QM ForceField.Charge=-.736000   ␣
↪ForceField.Type=SH   !  1653 CYS    112 SG
```

(continues on next page)

(continued from previous page)

```
   C     10.1650     3.3080    22.4340 region=QM ForceField.Charge=-.058100  ␣
↪ForceField.Type=CT   !  1710 HIS    117 CA
   H       9.2929     2.7403    22.7584 region=QM ForceField.Charge=.136000   ␣
↪ForceField.Type=H1   !  1711 HIS    117 HA
   C     10.1750     4.6030    23.2620 region=QM ForceField.Charge=-.007400  ␣
↪ForceField.Type=CT   !  1712 HIS    117 CB
   H     11.1220     5.1220    23.1143 region=QM ForceField.Charge=.036700   ␣
↪ForceField.Type=HC   !  1713 HIS    117 HB2
   H       9.3551     5.2459    22.9418 region=QM ForceField.Charge=.036700   ␣
↪ForceField.Type=HC   !  1714 HIS    117 HB3
   C     10.0160     4.3980    24.7440 region=QM ForceField.Charge=.186800   ␣
↪ForceField.Type=CC   !  1715 HIS    117 CG
   N       9.7040     5.4090    25.6080 region=QM ForceField.Charge=-.543200  ␣
↪ForceField.Type=NB   !  1716 HIS    117 ND1
   C       9.6570     4.9300    26.8540 region=QM ForceField.Charge=.163500   ␣
↪ForceField.Type=CR   !  1717 HIS    117 CE1
   H       9.4228     5.5952    27.6851 region=QM ForceField.Charge=.143500   ␣
↪ForceField.Type=H5   !  1718 HIS    117 HE1
   N       9.9280     3.6450    26.8000 region=QM ForceField.Charge=-.279500  ␣
↪ForceField.Type=NA   !  1719 HIS    117 NE2
   H       9.9617     3.0260    27.5974 region=QM ForceField.Charge=.333900   ␣
↪ForceField.Type=H    !  1720 HIS    117 HE2
   C     10.1580     3.2710    25.4990 region=QM ForceField.Charge=-.220700  ␣
↪ForceField.Type=CW   !  1721 HIS    117 CD2
   H     10.3982     2.2340    25.2644 region=QM ForceField.Charge=.186200   ␣
↪ForceField.Type=H4   !  1722 HIS    117 HD2
   C       6.0350     6.2800    19.5280 region=QM ForceField.Charge=-.023700  ␣
↪ForceField.Type=CT   !  1778 MET    121 CA
   H       4.9702     6.5113    19.5559 region=QM ForceField.Charge=.088000   ␣
↪ForceField.Type=H1   !  1779 MET    121 HA
   C       6.6730     6.7710    20.8330 region=QM ForceField.Charge=.034200   ␣
↪ForceField.Type=CT   !  1780 MET    121 CB
   H       7.7511     6.6157    20.7919 region=QM ForceField.Charge=.024100   ␣
↪ForceField.Type=HC   !  1781 MET    121 HB2
   H       6.4641     7.8329    20.9631 region=QM ForceField.Charge=.024100   ␣
↪ForceField.Type=HC   !  1782 MET    121 HB3
   C       6.1560     6.0500    22.0720 region=QM ForceField.Charge=.001800   ␣
↪ForceField.Type=CT   !  1783 MET    121 CG
   H       5.0693     6.1257    22.1101 region=QM ForceField.Charge=.044000   ␣
↪ForceField.Type=H1   !  1784 MET    121 HG2
   H       6.4453     5.0000    22.0292 region=QM ForceField.Charge=.044000   ␣
↪ForceField.Type=H1   !  1785 MET    121 HG3
   S       6.7760     6.6970    23.6140 region=QM ForceField.Charge=-.273700  ␣
↪ForceField.Type=S    !  1786 MET    121 SD
   C       6.0690     8.3070    23.6050 region=QM ForceField.Charge=-.053600  ␣
↪ForceField.Type=CT   !  1787 MET    121 CE
   H       4.9825     8.2271    23.5709 region=QM ForceField.Charge=.068400   ␣
↪ForceField.Type=H1   !  1788 MET    121 HE1
   H       6.3654     8.8396    24.5086 region=QM ForceField.Charge=.068400   ␣
↪ForceField.Type=H1   !  1789 MET    121 HE2
   H       6.4202     8.8537    22.7299 region=QM ForceField.Charge=.068400   ␣
↪ForceField.Type=H1   !  1790 MET    121 HE3
   CU      9.5640     7.3450    25.1750 region=QM ForceField.Charge=2.000000  ␣
↪ForceField.Type=CU   !  1915 CU     130 CU
   ...
   O     31.1328    34.4612    22.6903 region=MM ForceField.Charge=-.834000  ␣
↪ForceField.Type=OW   !  9746 HOH  2545 O
```

(continues on next page)

(continued from previous page)

```
    H      31.8908    34.5740    22.1167 region=MM ForceField.Charge=.417000   ␣
↪ForceField.Type=HW   !  9747 HOH  2545 H1
    H      30.6706    35.2981    22.6446 region=MM ForceField.Charge=.417000   ␣
↪ForceField.Type=HW   !  9748 HOH  2545 H2
  End

  BondOrders
        1      5 1.0
        1      2 1.0
        1      3 1.0
        1      4 1.0
        5      7 1.0
        5     11 1.0
        5      6 1.0
        7      8 1.0
        7      9 1.0
        7     10 1.0
       11     12 1.0
       11     13 1.0
       13     15 1.0
       13     14 1.0
       15     17 1.0
       15     28 1.0
       15     16 1.0
       17     20 1.0
       17     18 1.0
       17     19 1.0
     ...
     9746   9747 1.0
     9746   9748 1.0
  End
End

Task GeometryOptimization

GeometryOptimization
  MaxIterations 100
  Convergence Gradients=1e-3
End
Engine Hybrid
  QMMM QMRegion=QM QMEngineID=ADF MMEngineID=ForceField

  Capping
    AtomicInfoForCappingAtom ForceField.Type=H1 ForceField.Charge=0.0
  End
  Engine ADF
    Title QM/MM calculation setup by pdb2adf: M.Swart et al., 2020
    Symmetry NOSYM

    Eprint
      SFO NOEIG NOOVL
    End

    XC
      GGA BP86
    End
```

(continues on next page)

(continued from previous page)

```
    Basis
      type TZP
      core small
    End

    SCF
      diis ok=0.01
      Converge 1.0e-5 1.0e-5
      Iterations 99
    End
    Unrestricted
    SpinPolarization   1.0
  EndEngine
  Engine ForceField
    Type Amber95
    ForceFieldFile $AMSHOME/atomicdata/ForceFields/amber95.ff
  EndEngine
EndEngine
eor
```

3.2.3 An example on solvent shell run

The idea of this example is to make an adf-input file using a PDB file of water (`hoh.pdb.txt`), in the solvent methanol. The water molecule in the adf-input file should be in the QM part, and the solvent methanol (in a box) is in MM part.

Contents of the hoh.pdb file

```
TITLE     PDB-FILE CORRESPONDING TO pdb2adf-GENERATED ADF-INPUTFILE
REMARK    Written by M. Swart, March 2005
HETATM    1 H1   HOH     1       1.716  26.282  11.239  1.00  0.00       1DYZ H
HETATM    2 O    HOH     1       2.439  25.795  11.634  1.00  0.00       1DYZ O
HETATM    3 H2   HOH     1       3.140  26.440  11.729  1.00  0.00       1DYZ H
END
```

Usage of pdb2adf

The program works interactively. Given below in **bold** are the parts that the user has to type. In cases where the user agrees with the suggestion given by the program, the user can press the Enter key indicated with **Enter**.

```
                                   P D B 2 A D F - program
                                      version 2008.01
                                 Written by: Marcel Swart, 2008

                            This program uses AMBER parameter files
                               see: https://amber.scripps.edu
Please give name of PDB-file
```

hoh.pdb.txt

```
Do you want a logfile to be written (Y/n) ?
```

Enter

```
 read fragments

Data Processed:
     Nat:          3
    Nmol:          1
 NChains:          0

Please wait, making connection tables
Now finding nearby atoms
Assigning chain ID to all residues
Completing residues for which only option is available

Refinding nearby atoms (including atoms added in residue completion)

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           Making Choices for Chain      0
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Completing residues with multiple options available, and solvent molecules

Checking positions of newly added atoms

Making choice for which molecules should be QM, which MM

Residues belonging to chain  0

Solvent molecules (SOL/HOH) belonging to this chain:
    1

Give the number of the molecule to be put in QM region (or 'c' to continue):
```

1

```
Putting HOH    1 in QM region

Give the number of the molecule to be put in QM region (or 'c' to continue):
```

c

```
Do you want to add solvent to your system (Y/n) ?
```

Enter

```
Solvent (box) available:
   1:  HOH     HOH  Water molecule
   2:  MOH     MOH  Methanol molecule
   3:  CHL     CHL  Chloroform molecule
```

2

```
Reading contents of solvent box p2abox.MOH

Box Shape options:
 1  Spherical box
 2  Cubic box
Make a choice:
```

1

```
Writing inputfile for chain   0

Using total charge   0.0 and total spin   0.0

Maximum atomic distance (Angs) from center        0.92
Give boxsize (def.:    15.00 Angs)
```

14.0

```
Using BOXSIZE value of  14.0000
Adding atoms for box     1  Added (Box):    84 (Total):    84  Excl. (1):   660  Excl.
↪ (2):     6
Adding atoms for box     2  Added (Box):   102 (Total):   186  Excl. (1):   642  Excl.
↪ (2):     6
Adding atoms for box     3  Added (Box):   102 (Total):   288  Excl. (1):   642  Excl.
↪ (2):     6
Adding atoms for box     4  Added (Box):   108 (Total):   396  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box     5  Added (Box):   120 (Total):   516  Excl. (1):   630  Excl.
↪ (2):     0
Adding atoms for box     6  Added (Box):    96 (Total):   612  Excl. (1):   654  Excl.
↪ (2):     0
Adding atoms for box     7  Added (Box):   108 (Total):   720  Excl. (1):   642  Excl.
↪ (2):     0
Adding atoms for box     8  Added (Box):   102 (Total):   822  Excl. (1):   642  Excl.
↪ (2):     6

Inputfile(s) written, everything processed, work has been done.
Thank you for using the PDB2ADF program.

================================
Normal ending of PDB2ADF program
================================
```

Contents of the hoh.pdb2adf file generated by pdb2adf

The file is not given completely, since it contains more than 800 atoms.

```
#! /bin/sh

$AMSBIN/ams << eor
System
  Charge   0.0
  Atoms
    O      2.4390    25.7950    11.6340 region=QM ForceField.Charge=-.834000  ␣
↪ForceField.Type=OW   !     1 HOH       1 O
    H      1.7160    26.2820    11.2390 region=QM ForceField.Charge=.417000   ␣
↪ForceField.Type=HW   !     2 HOH       1 H1
    H      3.1400    26.4400    11.7290 region=QM ForceField.Charge=.417000   ␣
↪ForceField.Type=HW   !     3 HOH       1 H2
    C    -10.0667    22.2493    11.7437 region=MM ForceField.Charge=.116600   ␣
↪ForceField.Type=CT   !     4 MOH     1 C1
    H    -10.2077    21.5053    10.9597 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !     5 MOH     1 HC1
    H    -10.5047    21.8683    12.6667 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !     6 MOH     1 HC2
```

(continues on next page)

(continued from previous page)

```
    H    -10.5167    23.2103    11.4977 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !     7 MOH     1 HC3
    O     -8.7387    22.3983    12.0617 region=MM ForceField.Charge=-.649700  ␣
↪ForceField.Type=OH   !     8 MOH     1 O1
    H     -8.3007    22.6943    11.2607 region=MM ForceField.Charge=.421500   ␣
↪ForceField.Type=HO   !     9 MOH     1 HO1
    C     -0.2827    19.0253     2.2847 region=MM ForceField.Charge=.116600   ␣
↪ForceField.Type=CT   !    10 MOH     2 C1
    H     -0.5357    18.2063     2.9567 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !    11 MOH     2 HC1
    H      0.7633    19.2913     2.4407 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !    12 MOH     2 HC2
    H     -0.9267    19.8753     2.5107 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !    13 MOH     2 HC3
    O     -0.4997    18.6373     0.9467 region=MM ForceField.Charge=-.649700  ␣
↪ForceField.Type=OH   !    14 MOH     2 O1
    H      0.1123    17.9313     0.7287 region=MM ForceField.Charge=.421500   ␣
↪ForceField.Type=HO   !    15 MOH     2 HO1
    ...
    C      6.1721    28.5021    18.9485 region=MM ForceField.Charge=.116600   ␣
↪ForceField.Type=CT   !   820 MOH   137 C1
    H      7.1011    27.9431    18.8355 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !   821 MOH   137 HC1
    H      6.3621    29.4771    19.3985 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !   822 MOH   137 HC2
    H      5.4711    27.9401    19.5645 region=MM ForceField.Charge=.037200   ␣
↪ForceField.Type=H1   !   823 MOH   137 HC3
    O      5.5611    28.7181    17.7095 region=MM ForceField.Charge=-.649700  ␣
↪ForceField.Type=OH   !   824 MOH   137 O1
    H      5.2631    27.8621    17.3935 region=MM ForceField.Charge=.421500   ␣
↪ForceField.Type=HO   !   825 MOH   137 HO1
  End

  BondOrders
        1      2 1.0
        1      3 1.0
        4      5 1.0
        4      6 1.0
        4      7 1.0
        4      8 1.0
        8      9 1.0
       10     11 1.0
       10     12 1.0
       10     13 1.0
       10     14 1.0
       14     15 1.0
      ...
      820    821 1.0
      820    822 1.0
      820    823 1.0
      820    824 1.0
      824    825 1.0
  End
End

Task GeometryOptimization
```

(continues on next page)

(continued from previous page)

```
GeometryOptimization
  MaxIterations 100
  Convergence Gradients=1e-3
End
Engine Hybrid
  QMMM QMRegion=QM QMEngineID=ADF MMEngineID=ForceField

  Capping
    AtomicInfoForCappingAtom ForceField.Type=H1 ForceField.Charge=0.0
  End
  Engine ADF
    Title QM/MM calculation setup by pdb2adf: M.Swart et al., 2020
    Symmetry NOSYM

    Eprint
      SFO NOEIG NOOVL
    End

    XC
      GGA BP86
    End

    Basis
      type TZP
      core small
    End

    SCF
      diis ok=0.01
      Converge 1.0e-5 1.0e-5
      Iterations 99
    End
  EndEngine
  Engine ForceField
    Type Amber95
    ForceFieldFile $AMSHOME/atomicdata/ForceFields/amber95.ff
  EndEngine
EndEngine
eor
```

CHAPTER

FOUR

EXAMPLES

Examples of the Hybrid engine are scattered over the examples directory. Here we pick a few.

We do not repeat here all functionality that is available from the AMS driver level, see the AMS Examples.

4.1 Example: QMMM with various forcefields

`Download qmmm_water.run`

```
#!/bin/sh

# This example shows you how you can use the forcield engine in a qmmm setup
# Both the regions and the atom typing and charges (if any) go via te AMS system block

# UFF
# ===

AMS_JOBNAME=uff $AMSBIN/ams <<eor

Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM
        H -0.9986  0.2961 -0.3861  region=QM
        H -1.8623 -0.9560 -0.6510  region=QM
        O  0.0121 -1.3731  0.5074  region=MM
        H  0.8930 -1.7879  0.3172  region=MM
        H -0.5625 -2.1395  0.7656  region=MM
    End
End

Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type UFF
    EndEngine
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor


# Amber
# =====

AMS_JOBNAME=amber $AMSBIN/ams <<eor

Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM ForceField.Charge=-0.8340  ForceField.
↪Type=OW
        H -0.9986  0.2961 -0.3861  region=QM ForceField.Charge=0.4170   ForceField.
↪Type=HW
        H -1.8623 -0.9560 -0.6510  region=QM ForceField.Charge=0.4170   ForceField.
↪Type=HW
        O  0.0121 -1.3731  0.5074  region=MM ForceField.Charge=-0.8340  ForceField.
↪Type=OW
        H  0.8930 -1.7879  0.3172  region=MM ForceField.Charge=0.4170   ForceField.
↪Type=HW
        H -0.5625 -2.1395  0.7656  region=MM ForceField.Charge=0.4170   ForceField.
↪Type=HW
    End
    BondOrders
         1 2 1.0
         1 3 1.0
         4 5 1.0
         4 6 1.0
    End
End

Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type Amber95
        ForceFieldFile $AMSHOME/atomicdata/ForceFields/amber95.ff
    EndEngine
EndEngine
eor

# Tripos
# ======

AMS_JOBNAME=tripos $AMSBIN/ams <<eor

Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM ForceField.Charge=-0.8340   ForceField.
```

(continues on next page)

(continued from previous page)

```
↪Type=O.3
        H -0.9986  0.2961 -0.3861  region=QM  ForceField.Charge=0.4170    ForceField.
↪Type=H
        H -1.8623 -0.9560 -0.6510  region=QM  ForceField.Charge=0.4170    ForceField.
↪Type=H
        O  0.0121 -1.3731  0.5074  region=MM  ForceField.Charge=-0.8340   ForceField.
↪Type=O.3
        H  0.8930 -1.7879  0.3172  region=MM  ForceField.Charge=0.4170    ForceField.
↪Type=H
        H -0.5625 -2.1395  0.7656  region=MM  ForceField.Charge=0.4170    ForceField.
↪Type=H
    End
    BondOrders
         1 2 1.0
         1 3 1.0
         4 5 1.0
         4 6 1.0
    End
End

Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type Tripos5.2
        ForceFieldFile $AMSRESOURCES/ForceFields/tripos5.2.ff
    EndEngine
EndEngine
eor
```

4.2 Example: Mechanical embedding QUILD

Download QUILD_water.run

```
#!/bin/sh

# This example shows you how you can use the forcefield with mechanical embedding


AMS_JOBNAME=uff $AMSBIN/ams <<eor

Task GeometryOptimization

System
    Atoms
        O -1.8782  0.0294 -0.7574  region=QM
        H -0.9986  0.2961 -0.3861  region=QM
        H -1.8623 -0.9560 -0.6510  region=QM
        O  0.0121 -1.3731  0.5074  region=MM
        H  0.8930 -1.7879  0.3172  region=MM
```

(continues on next page)

(continued from previous page)

```
        H -0.5625 -2.1395  0.7656  region=MM
    End
End

Engine Hybrid

    Energy
       Term region=QM EngineId=DFTB       factor=1.0
       Term region=*  EngineId=ForceField factor=1.0
       Term region=QM EngineId=ForceField factor=-1.0
    End

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
        Type UFF
    EndEngine
EndEngine
eor
```

4.3 Example: Hybrid engine with charged regions

Download HybridWithCharges.run

```
#!/bin/sh

# not needed, just slighly faster
export NSCM=1

report=report.txt



printf "Here we treat H3O+ as qm and OH- as the MM region (Optimizing without regions␣
↪gives two H2O molecules)\n" > $report
printf "We do this with both mechanical and electrostatic embedding\n" >> $report
printf  "\n%15s %10s %10s %10s\n" "embedding" "charge" "d(O-O)" "charges" >> $report

for charge in 0.0 1.0
do

if [ "$charge" = "0.0" ]; then
chargeOinOHm=0.0
chargeHinOHm=0.0
chargeOinH3Op=0.0
chargeHinH3Op=0.0
else
chargeOinOHm=-1.123
chargeHinOHm=0.123
chargeOinH3Op=-0.5
chargeHinH3Op=0.5
fi
```

(continues on next page)

(continued from previous page)

```
export AMS_JOBNAME=quild.charge=$charge

rm -rf $AMS_JOBNAME.results

"$AMSBIN/ams" << eor

Task GeometryOptimization

Properties Charges=yes

GeometryOptimization
  Convergence Gradients=1.0e-6
End

System
    Atoms
        O -1.527946410885647 -0.2107366711137158 -0.0008116899510243671 region=QM ␣
↪ForceField.Charge=$chargeOinH3Op
        H -0.8459142126057956 0.3517312394359257 0.4094504676540848     region=QM ␣
↪ForceField.Charge=$chargeHinH3Op
        H -1.834953147575289 0.1051014241823828 -0.8704652381864062     region=QM ␣
↪ForceField.Charge=$chargeHinH3Op
        H -1.328032016244278 -1.164422847242489 0.02894848344144469     region=QM ␣
↪ForceField.Charge=$chargeHinH3Op
        O 0.6370858511871781 -0.3378071707560572 -0.0006181020627287671 region=MM ␣
↪ForceField.Charge=$chargeOinOHm
        H 1.318474396634582 0.2241299231185073 0.4092568796869673       region=MM ␣
↪ForceField.Charge=$chargeHinOHm
    End
    GuessBonds True
End

Engine Hybrid
    Energy
       Term Factor=1.0  Region=*  EngineID=ForceField
       Term Factor=-1.0 Region=QM EngineID=ForceField     Charge=$charge
       Term Factor=1.0  Region=QM EngineID=DFTB           Charge=$charge
    End

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
    EndEngine
EndEngine

eor

ddd=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#5`
eee=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -k AMSResults%Charges#5.3f`

printf "%15s %10s %10s %10s %10s %10s %10s %10s %10s\n" "mechanical"  $charge $ddd
↪$eee >> $report

export AMS_JOBNAME=qmmm.charge=$charge
```

(continues on next page)

(continued from previous page)

```
rm -rf $AMS_JOBNAME.results

"$AMSBIN/ams" << eor

Properties Charges=yes

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1.0e-6
End

System
    Atoms
        O -1.527946410885647 -0.2107366711137158 -0.0008116899510243671 region=QM   ␣
↪ForceField.Charge=$chargeOinH3Op
        H -0.8459142126057956 0.3517312394359257 0.4094504676540848     region=QM   ␣
↪ForceField.Charge=$chargeHinH3Op
        H -1.834953147575289 0.1051014241823828 -0.8704652381864062     region=QM   ␣
↪ForceField.Charge=$chargeHinH3Op
        H -1.328032016244278 -1.164422847242489 0.02894848344144469     region=QM   ␣
↪ForceField.Charge=$chargeHinH3Op
        O 0.6370858511871781 -0.3378071707560572 -0.0006181020627287671 region=MM   ␣
↪ForceField.Charge=$chargeOinOHm
        H 1.318474396634582 0.2241299231185073 0.4092568796869673       region=MM   ␣
↪ForceField.Charge=$chargeHinOHm
    End
    GuessBonds True
End

Engine Hybrid
    QMMM QMRegion=QM QMEngineID=DFTB MMEngineID=ForceField QMCharge=$charge MMCharge=-
↪$charge

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
    EndEngine
EndEngine

eor

ddd=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#5`
eee=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -k AMSResults%Charges#5.3f`

printf "%15s %10s %10s %10s %10s %10s %10s %10s %10s\n" "electrostatic"   $charge
↪$ddd $eee >> $report

done


printf "\n* Using charges shortens the O-O distance\n" >> $report
printf "* In this case the results (mechanical vs. electrostatic) are quite similar␣
↪as apparently the OH does not polarize the QM region much\n" >> $report
```

(continues on next page)

(continued from previous page)

```



echo "start of report"
cat $report
echo "end of report"






report=report2.txt

printf "\nNow we add an extra OH- to the mm region and get a total charge of -1\n" >
↪$report
printf "We do this with mechanical and electrostatic embedding\n" >> $report
printf "We look at two distances: d(O1-O5) and d(O1-O7) \n" >> $report
printf "Atom O1 is in the H3O+ and atoms O5 and O7 are in the two OH- molecules\n" >>
↪$report
printf  "\n%15s %15s %10s %10s %10s %10s\n" "embedding"  "optim" "d(1-5)" "d(1-7)"
↪"energy" >> $report

charge=1.0
chargeOinOHm=-1.123
chargeHinOHm=0.123
chargeOinH3Op=-0.5
chargeHinH3Op=0.5

for embedding in mechanical electrostatic
do

for optim in FIRE # Quasi-Newton
do

export AMS_JOBNAME=embedding=$embedding.optim=$optim

rm -rf $AMS_JOBNAME.results

"$AMSBIN/ams" << eor

Task GeometryOptimization

Properties Charges=yes

GeometryOptimization
  Method $optim
  MaxIterations 3000
  Convergence Gradients=1.0e-6
End

System
    Atoms
        O 0.9019652567984636 -1.133079116834755   0.01338426553857459  region=QM  ␣
↪ForceField.Charge=$chargeOinH3Op
        H 0.1122251167578682 -1.036551903399635   0.5668491423154995   region=QM  ␣
↪ForceField.Charge=$chargeHinH3Op
```

(continues on next page)

(continued from previous page)

```
        H 1.037136681303829  -0.2320347366030556 -0.3773644469587724   region=QM  ␣
↪ForceField.Charge=$chargeHinH3Op
        H 1.678241221654873  -1.266912785246295   0.5779953693196539   region=QM  ␣
↪ForceField.Charge=$chargeHinH3Op
        O -1.130580450693341   0.6009421414132099 -0.02453852439122078  region=MM  ␣
↪ForceField.Charge=$chargeOinOHm
        H -1.671378074377012   1.410809444490273  -0.2141830902463049   region=MM  ␣
↪ForceField.Charge=$chargeHinOHm
        O 3.346891191122751  -0.05485781804516161 0.01059240308504993  region=MM  ␣
↪ForceField.Charge=$chargeOinOHm
        H 4.099773764135065   0.5660034244354222 -0.1683355405307263   region=MM  ␣
↪ForceField.Charge=$chargeHinOHm
    End
    BondOrders
         1 3 1.0
         1 4 1.0
         2 1 1.0
         5 6 1.0
         7 8 1.0
    End
    Charge -1.0
End

Engine Hybrid

    QMMM qmRegion=QM qmCharge=1.0 mmCharge=-2.0 qmEngineID=dftb mmEngineID=forcefield␣
↪Embedding=$embedding

    Engine DFTB
        Model GFN1-xTB
    EndEngine

    Engine ForceField
    EndEngine
EndEngine

eor


d15=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#5`
d17=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#7`
eee=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -k "AMSResults%Energy"`


printf "%15s %15s %10s %10s %10.4f\n" $embedding $optim $d15 $d17 $eee >> $report

done
done


printf "\n* Very flat PES as function of these two distances\n" >> $report
printf "\n* Electrostatic embeddiding gives a bit shorter distances\n" >> $report


echo "start of report"
cat $report
echo "end of report"
```

4.4 Example: Loading MM charges for regions

In this example we consider an OH- with an H3O+ fragment. As the charges on the fragments are kept fixed, the formation of two water molecules is avoided.

First we “estimate” the charges for the two fragments with a DFTB calculation.

These charges are then loaded for the correct regions in the total system. Observe that this is done in the `System` block, see the System definition section of the AMS manual.

We do this first for a QUILD-like setup (mechanical embedding), and next for a QMMM calculation with electrostatic coupling.

Note that this example makes use of the python scripting tools PLAMS and PISA, available in the AMS Python Stack.

Download LoadCharges.py

```
#!/usr/bin/env amspython

from scm.base import ChemicalSystem
from scm.input_classes import drivers, engines
from scm.plams import AMSJob, JobRunner, config, Settings
import copy

# Here we treat H3O+ as qm and OH- as the MM region (Optimizing without regions gives␣
↪two H2O molecules)
# We do this with a QUILD setup (mechanical embedding) and electrostatic embedding␣
↪(QMMM)
# We obtain the charges from a DFTB calculation
# In this case the results (QUILD vs. QMMM) are quite similar as apparently the OH␣
↪does not polarize the QM region much

# First we do two DFTB calculations on the two fragments
config.default_jobrunner = JobRunner(parallel=True)
settings = Settings()
settings.runscript.nproc = 1

h30 = ChemicalSystem(
    """
System
    Atoms
        O -1.527946410885647 -0.2107366711137158 -0.0008116899510243671
        H -0.8459142126057956 0.3517312394359257 0.4094504676540848
        H -1.834953147575289 0.1051014241823828 -0.8704652381864062
        H -1.328032016244278 -1.164422847242489 0.02894848344144469
    End
    Charge 1.0
    GuessBonds True
End
"""
)

oh = ChemicalSystem(
    """
System
    Atoms
        O 0.6370858511871781 -0.3378071707560572 -0.0006181020627287671
        H 1.318474396634582 0.2241299231185073 0.4092568796869673
    End
```

(continues on next page)

(continued from previous page)

```
    Charge -1.0
    GuessBonds True
End
"""
)

fragments = {"H3O+": h30, "OH-": oh}

results_dftb = {}
for name, mol in fragments.items():
    driver_dftb = drivers.AMS()
    driver_dftb.Task = "SinglePoint"
    driver_dftb.Properties.Charges = True
    driver_dftb.GeometryOptimization.Convergence.Gradients = 1.0e-6
    driver_dftb.Engine = engines.DFTB()
    job = AMSJob(molecule=mol, settings=settings + Settings({"input": driver_dftb}),␣
↪name=name)
    results_dftb[name] = job.run()

# Now we run it in a QUILD-like setup (mechanical embedding) and in a QMMM-like setup
h20_2 = ChemicalSystem(
    """
System
    Atoms
        O -1.527946410885647 -0.2107366711137158 -0.0008116899510243671 region=QM
        H -0.8459142126057956 0.3517312394359257 0.4094504676540848     region=QM
        H -1.834953147575289 0.1051014241823828 -0.8704652381864062     region=QM
        H -1.328032016244278 -1.164422847242489 0.02894848344144469     region=QM
        O 0.6370858511871781 -0.3378071707560572 -0.0006181020627287671 region=MM
        H 1.318474396634582 0.2241299231185073 0.4092568796869673       region=MM
    End
    GuessBonds True
End
"""
)

qm_charges = results_dftb["H3O+"].get_charges()
mm_charges = results_dftb["OH-"].get_charges()
h20_2.enable_atom_attributes("forcefield")

for i, charge in zip(h20_2.get_atoms_in_region("QM"), qm_charges):
    h20_2.atoms[i].forcefield.charge = charge

for i, charge in zip(h20_2.get_atoms_in_region("MM"), mm_charges):
    h20_2.atoms[i].forcefield.charge = charge

results = {}
for name in ["quild", "qmmm"]:
    driver = copy.deepcopy(driver_dftb)
    driver.Task = "GeometryOptimization"
    driver.Engine = engines.Hybrid()

    driver.Engine.Engine[0] = engines.DFTB()
    driver.Engine.Engine[0].Model = "GFN1-xTB"
    driver.Engine.Engine[1] = engines.ForceField()

    if name == "quild":
```

(continues on next page)

(continued from previous page)

```
        driver.Engine.Energy.Term[0].Factor = 1.0
        driver.Engine.Energy.Term[0].Region = "*"
        driver.Engine.Energy.Term[0].EngineID = "ForceField"

        driver.Engine.Energy.Term[1].Factor = -1.0
        driver.Engine.Energy.Term[1].Region = "QM"
        driver.Engine.Energy.Term[1].EngineID = "ForceField"
        driver.Engine.Energy.Term[1].Charge = 1.0

        driver.Engine.Energy.Term[2].Factor = 1.0
        driver.Engine.Energy.Term[2].Region = "QM"
        driver.Engine.Energy.Term[2].EngineID = "DFTB"
        driver.Engine.Energy.Term[2].Charge = 1.0

    elif name == "qmmm":
        driver.Engine.QMMM.QMRegion = "QM"
        driver.Engine.QMMM.QMEngineID = "DFTB"
        driver.Engine.QMMM.MMEngineID = "ForceField"
        driver.Engine.QMMM.QMCharge = 1.0
        driver.Engine.QMMM.MMCharge = -1.0

    job = AMSJob(molecule=h20_2, settings=settings + Settings({"input": driver}),␣
↪name=name)
    results[name] = job.run()

results = {k: (v.get_charges(), v.get_main_system()) for k, v in results.items()}

print("start of report")
print("method distance charges")
for name, (charges, system) in results.items():
    print(
        f"{name} {system.get_distance(0, 4):5.3f} {charges[0]:5.3f} {charges[1]:5.3f}
↪{charges[2]:5.3f} {charges[3]:5.3f} {charges[4]:5.3f} {charges[5]:5.3f}"
    )
print("end of report")
```

4.5 Example: Molecular gun with the hybrid engine

In this example we are going to really stretch the use of the Hybrid Engine, and shoot bullets (treated with a QM engine) at a surface described at the MM level.

The choice of bullets are HF molecules and the target is a two dimensional BN sheet, that looks like a graphene sheet, with half of the C atoms turned into N and the other half into B atoms. In a BN sheet the atoms have of course a small charge, which we pre calculate with a QM engine (DFTB).

It is important to understand the role of bonds in this example, because the number of atoms is not constant during the simulations, as bullets are fired (and hence appear), ricochet off the surface and hence disappear after a while. They may as well stick to, or penetrate into the surface, but this is beyond the hybrid engine concept.

In this example there are fixed bond orders withing the target and within the bullets. This is because we specify GuessBonds in the two system blocks (target and bullet). When a bullet is added its bonds are automatically added. The hybrid engine itself will never guess bonds and always use what is specified on input. No bonds are ever formed between the bullet and the surface (the QM and MM regions).

Note that this example makes use of the python scripting tools PLAMS and PISA, available in the AMS Python Stack.

Download HybridGun.py

```
#!/usr/bin/env amspython

# In this example we use the hybrid engine in a molecular gun MD application, shooting HF molecules at a BN surface

# The BN slab represents the MM region and the "bullets" are the QM region
# The regions are defined in the xyz files using end of line strings (atom attributes)

from scm.base import ChemicalSystem, Units
from scm.input_classes import drivers, engines
from scm.plams import AMSJob, JobRunner, config, Settings
import os

# Load chemical systems from xyz files
struct_dir = os.path.join(os.getenv("AMSHOME"), "examples", "Hybrid", "HybridGun", "molecules")
bn_slab = ChemicalSystem.from_xyz(os.path.join(struct_dir, "BNSlab.xyz"))
hf = ChemicalSystem.from_xyz(os.path.join(struct_dir, "HF.xyz"))
systems = {"bn_slab": bn_slab, "hf": hf}

# First we do two dftb calculations to get a guess of the charges to be used by the force field.
# Setup jobs and run in parallel
config.default_jobrunner = JobRunner(parallel=True)
settings = Settings()
settings.runscript.nproc = 1

driver_dftb = drivers.AMS()
driver_dftb.Task = "SinglePoint"
driver_dftb.Engine = engines.DFTB()

results_dftb = {}
for name, system in systems.items():
    job_dftb = AMSJob(molecule=system, settings=settings + Settings({"input": driver_dftb}), name=name)
    results_dftb[name] = job_dftb.run()

# Load forcefield charges from the dftb calculations onto the chemical systems
for name, system in systems.items():
    results = results_dftb[name]
    charges = results.get_charges()
    system.enable_atom_attributes("forcefield")
    for atom, charge in zip(system.atoms, charges):
        atom.forcefield.charge = charge
    system.guess_bonds()
    print(f"Total Energy (hartree) {results.get_energy('hartree', 'dftb'):.8f}")
    print(f"Total Energy (eV) {results.get_energy('eV', 'dftb'):.8f}")

# Now we can run our MD simulation using both mechanical and electrostatic embedding
# (because electrostatic embedding is more expensive we limit here the number of steps)
results_md = {}
for embedding, steps in [("mechanical", 1400), ("electrostatic", 300)]:

    driver_md = drivers.AMS()
    driver_md.Task = "MolecularDynamics"
```

(continues on next page)

(continued from previous page)

```
    driver_md.RNGSeed = [-1341016088, 83513668, 1764626453, -87803069, -1149690266,␣
↪1963370818, -1393571175, 1985130742]
    driver_md.Constraints.Atom = 1

    driver_md.MolecularDynamics.NSteps = steps
    driver_md.MolecularDynamics.Trajectory.SamplingFreq = 20
    driver_md.MolecularDynamics.InitialVelocities.Temperature = 300
    driver_md.MolecularDynamics.Preserve.Momentum = "No"
    driver_md.MolecularDynamics.Preserve.AngularMomentum = "No"
    driver_md.MolecularDynamics.AddMolecules.System = "HF"
    driver_md.MolecularDynamics.AddMolecules.Frequency = 159
    driver_md.MolecularDynamics.AddMolecules.CoordsBox = [0, 3, 0, 8.57, 6, 7]
    driver_md.MolecularDynamics.AddMolecules.VelocityDirection = [0.45752820, 0, -0.
↪5540656]
    driver_md.MolecularDynamics.AddMolecules.Velocity = 0.07
    driver_md.MolecularDynamics.AddMolecules.Rotate = "Yes"
    driver_md.MolecularDynamics.AddMolecules.MinDistance = 3.0
    driver_md.MolecularDynamics.RemoveMolecules.Formula = "*"
    driver_md.MolecularDynamics.RemoveMolecules.Frequency = 101
    driver_md.MolecularDynamics.RemoveMolecules.SinkBox.FractionalCoordsBox = [0, 1,␣
↪0, 1, 8, 1000]

    driver_md.Engine = engines.Hybrid()
    driver_md.Engine.QMMM.QMRegion = "QM"
    driver_md.Engine.QMMM.QMEngineID = "DFTB"
    driver_md.Engine.QMMM.MMEngineID = "ForceField"
    driver_md.Engine.QMMM.Embedding = embedding

    driver_md.Engine.Engine[0] = engines.DFTB()
    driver_md.Engine.Engine[1] = engines.ForceField()
    driver_md.Engine.Engine[1].NonBondedCutoff = 50 * Units.conversion_factor("bohr",
↪"angstrom")

    job_md = AMSJob(
        molecule={"": systems["bn_slab"], "HF": systems["hf"]},
        settings=settings + Settings({"input": driver_md}),
        name=f"SinkBox.embedding={embedding}",
    )
    results_md[embedding] = job_md.run()

for result in results_md.values():
    res = result.read_rkf_section("MDResults")
    header = "{:<20} {:>16} {:>16} {:>16} {:>16}".format("Measure", "Min", "Max",
↪"Mean", "StdDev")
    print(header)
    print("-" * len(header))
    print(
        f"{'Potential Energy':<20} {res['MinPotentialEnergy']:>16.8f} {res[
↪'MaxPotentialEnergy']:>16.8f} {res['MeanPotentialEnergy']:>16.8f} {res[
↪'StdDevPotentialEnergy']:>16.8f}"
    )
    print(
        f"{'Kinetic Energy':<20} {res['MinKineticEnergy']:>16.8f} {res[
↪'MaxKineticEnergy']:>16.8f} {res['MeanKineticEnergy']:>16.8f} {res[
↪'StdDevKineticEnergy']:>16.8f}"
    )
    print(
```

(continues on next page)

(continued from previous page)

```
        f"{'Total Energy':<20} {res['MinTotalEnergy']:>16.8f} {res['MaxTotalEnergy']:>
↪16.8f} {res['MeanTotalEnergy']:>16.8f} {res['StdDevTotalEnergy']:>16.8f}"
    )
    print(
        f"{'Temperature':<20} {res['MinTemperature']:>16.8f} {res['MaxTemperature']:>
↪16.8f} {res['MeanTemperature']:>16.8f} {res['StdDevTemperature']:>16.8f}"
    )
```

4.6 Example: The effect of specifying atom types, or not

Whether or not you specify the ForceField.Type on input (via atom attributes) makes a difference for the hybrid engine using a ForceField sub engine.

If you do, then for all regions these atom types will be used.

If you do not specify them, then for each region independently the atom typing will done automatically (if possible).

`Download AtomAttributes.run`

```
#!/bin/bash

export NSCM=1

# Here we show the role played by the  (ForceField.Type) atom attribute

# on purpose we specify a strange type for the carbon atoms "C_2" (nonsensical for␣
↪any of the regions)

# we also show that capitalization does not matter for the keys ForceField.Charge and␣
↪Type.
# For the values, such as C_R the case matters


# In the first run we specify in the input ForceField.Type
# The result is that for all regions this will be used, and the type for C is always␣
↪C-2
# --------------------
# region mol.  C-type
# --------------------
#  MM    CH3   C_2
#  *     C2H6  C_2
#  QM    CH4   C_2
#---------------------


export AMS_JOBNAME=type=inp.cap=none

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task GeometryOptimization

Properties Gradients=yes

```

(continues on next page)

(continued from previous page)

```
System
    Atoms
        C -1.559601 -1.234340  0.000641 region=QM forcefield.charge=-0.27  forcefield.
↪type=C_2
        H -1.898371 -0.303860  0.503021 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H -2.034545 -2.108050  0.494609 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H -1.869847 -1.205955 -1.065139 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        C -0.047661 -1.348892  0.094039 region=MM ForceField.Charge=-0.27  ForceField.
↪Type=C_2
        H  0.427282 -0.475182 -0.399929 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H  0.291107 -2.279373 -0.408341 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H  0.262583 -1.377277  1.159819 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
    End
    GuessBonds True
End


Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
    EndEngine

    Engine ForceField
    EndEngine

EndEngine

eor

#  Not specifying the types, they will be guessed independently for all regions

# -------------------
# region mol.  C-type
# -------------------
#  MM    CH3   C_R
#  *     C2H6  C_3
#  QM    CH4   C_3
#--------------------

export AMS_JOBNAME=type=none.cap=none

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task GeometryOptimization

Properties Gradients=yes

System
```

(continues on next page)

(continued from previous page)

```
    Atoms
        C -1.559601 -1.234340  0.000641 region=QM forcefield.charge=-0.27
        H -1.898371 -0.303860  0.503021 region=QM ForceField.Charge=0.09
        H -2.034545 -2.108050  0.494609 region=QM ForceField.Charge=0.09
        H -1.869847 -1.205955 -1.065139 region=QM ForceField.Charge=0.09
        C -0.047661 -1.348892  0.094039 region=MM ForceField.Charge=-0.27
        H  0.427282 -0.475182 -0.399929 region=MM ForceField.Charge=0.09
        H  0.291107 -2.279373 -0.408341 region=MM ForceField.Charge=0.09
        H  0.262583 -1.377277  1.159819 region=MM ForceField.Charge=0.09
    End
    GuessBonds True
End


Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
    EndEngine

    Engine ForceField
    EndEngine

    GuessAttributesOnce False

EndEngine

eor


# The last two runs are mostly a technical test
# We change the capping setup, but that influences only the capping atom

export AMS_JOBNAME=type=inp.cap=inp

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task GeometryOptimization

Properties Gradients=yes

System
    Atoms
        C -1.559601 -1.234340  0.000641 region=QM forcefield.charge=-0.27  forcefield.
↪type=C_2
        H -1.898371 -0.303860  0.503021 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H -2.034545 -2.108050  0.494609 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H -1.869847 -1.205955 -1.065139 region=QM ForceField.Charge=0.09   ForceField.
↪Type=H_
        C -0.047661 -1.348892  0.094039 region=MM ForceField.Charge=-0.27  ForceField.
↪Type=C_2
        H  0.427282 -0.475182 -0.399929 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
```

(continues on next page)

(continued from previous page)

```
        H  0.291107 -2.279373 -0.408341 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
        H  0.262583 -1.377277  1.159819 region=MM ForceField.Charge=0.09   ForceField.
↪Type=H_
    End
    GuessBonds True
End


Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
    EndEngine

    Engine ForceField
    EndEngine

    Capping
        CappingElement Li
        AtomicInfoForCappingAtom ForceField.Type=Li
    End

EndEngine

eor

export AMS_JOBNAME=type=none.cap=inp

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task GeometryOptimization

Properties Gradients=yes

System
    Atoms
        C -1.559601 -1.234340  0.000641 region=QM forcefield.charge=-0.27
        H -1.898371 -0.303860  0.503021 region=QM ForceField.Charge=0.09
        H -2.034545 -2.108050  0.494609 region=QM ForceField.Charge=0.09
        H -1.869847 -1.205955 -1.065139 region=QM ForceField.Charge=0.09
        C -0.047661 -1.348892  0.094039 region=MM ForceField.Charge=-0.27
        H  0.427282 -0.475182 -0.399929 region=MM ForceField.Charge=0.09
        H  0.291107 -2.279373 -0.408341 region=MM ForceField.Charge=0.09
        H  0.262583 -1.377277  1.159819 region=MM ForceField.Charge=0.09
    End
    GuessBonds True
End


Engine Hybrid
    QMMM qmRegion=QM qmEngineID=DFTB mmEngineID=ForceField

    Engine DFTB
    EndEngine
```

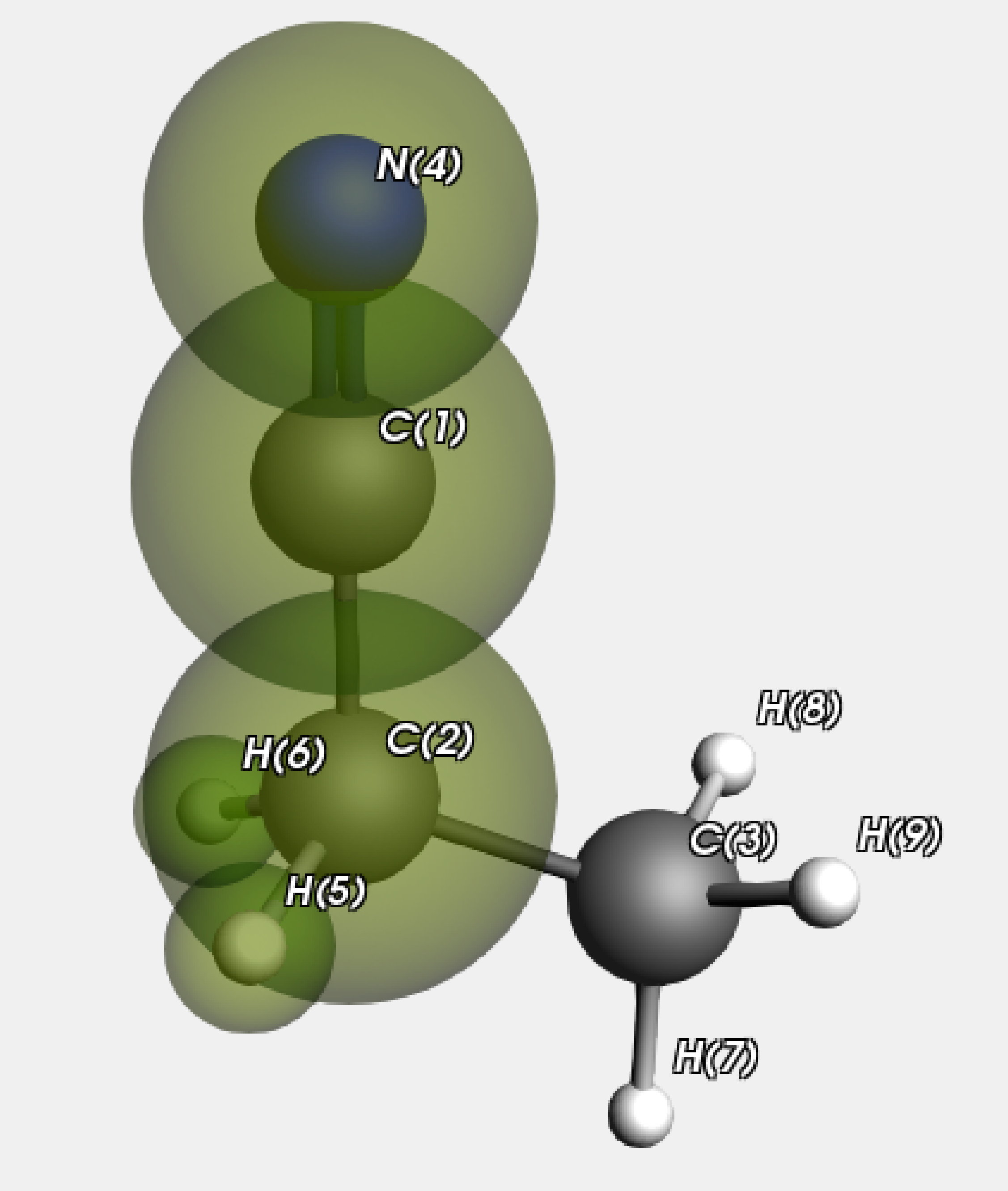
(continues on next page)

(continued from previous page)

```
    Engine ForceField
    EndEngine

    Capping
        CappingElement Li
        AtomicInfoForCappingAtom ForceField.Type=Li
    End

    GuessAttributesOnce False

EndEngine

eor
```

4.7 Example: The role of specifying the atom types

Now we look at a Propanenitrile molecule, the QM region is highlighted.



We run this with and without specifying the atom types on input. In principle this makes a difference for the MM type for atom “C(3)” in the MM sub calculation on atoms 3,7,8, and 9. If specified it will be C_3 (as it is in the whole Propanenitrile molecule), but if not it will be guessed as C_R. In practice there is no effect for this calculation.

Let us have a look at the report generated by the example, that pretty much explains what is done

`Download report Propanenitrile.txt`

```
We first check how bad the MM method is compared to the QM method for some distances␣
↪in the QM region.

Here are the distances (Angstrom) as obtained with a QM and an MM method
```

(continues on next page)

(continued from previous page)

```
Distance             QM          MM    Err(MM)
-------------------------------------------
C(1)-C(2)        1.456      1.467      0.011
C(1)-N(4)        1.147      1.157      0.010
C(2)-H(5)        1.095      1.110      0.015

Can we get better results for the QM region with the hybrid engine?

Even though UFF has automatic atom typing, it still matters (in principle) whether we␣
↪specify it on input or not:
    * Without typing for each region the types are automatically guessed
    * With typing the types are always as on input (for all regions)

The only difference is in the C type for the MM region.

Here are the distances (Angstrom) as obtained with a QM and an Hybrid method without␣
↪explicit typing:

Distance             QM          MM    Err(MM)
-------------------------------------------
C(1)-C(2)        1.456      1.456      0.000
C(1)-N(4)        1.147      1.147     -0.001
C(2)-H(5)        1.095      1.092     -0.003

Here are the distances (Angstrom) as obtained with a QM and an Hybrid method with␣
↪explicit typing:

Distance             QM          MM    Err(MM)
-------------------------------------------
C(1)-C(2)        1.456      1.456      0.000
C(1)-N(4)        1.147      1.147     -0.001
C(2)-H(5)        1.095      1.092     -0.003

Here are some observations for this example:
    * The hybrid engine does better than pure MM
    * The subtle issue whether or not we specify the types has negligible effect
```

Note that this example makes use of the python scripting tools PLAMS and PISA, available in the AMS Python Stack.

`Download Propanenitrile.py`

```python
#!/usr/bin/env amspython

from scm.base import ChemicalSystem
from scm.input_classes import drivers, engines
from scm.plams import AMSJob, JobRunner, config, Settings

config.default_jobrunner = JobRunner(parallel=True)
settings = Settings()
settings.runscript.nproc = 1

mol = ChemicalSystem(
    """
System
    Atoms
        C -0.02116 1.01286 0.0 region=qm
```

(continues on next page)

(continued from previous page)

```
        C 0.01258 -0.45034 0.0 region=qm
        C 1.44394 -1.0175 0.0
        N -0.03362 2.17616 0.0 region=qm
        H -0.54281 -0.80179 0.88302 region=qm
        H -0.54281 -0.80179 -0.88302 region=qm
        H 1.40659 -2.11445 0.0
        H 1.99584 -0.68766 -0.88907
        H 1.99584 -0.68766 0.88907
    End
    GuessBonds true
End
"""
)

results = {}
for engine in [engines.DFTB(), engines.ForceField()]:

    driver = drivers.AMS()
    driver.Engine = engine
    driver.Task = "GeometryOptimization"
    job = AMSJob(molecule=mol, settings=settings + Settings({"input": driver}),␣
↪name=engine.name)
    results[driver.Engine.name] = job.run()

mol_qm = results["DFTB"].get_main_system()
mol_mm = results["ForceField"].get_main_system()
bonds = {"C(1)-C(2)": (0, 1), "C(1)-N(4)": (0, 3), "C(2)-H(5)": (1, 4)}

print("begin report")
print(
    """We first check how bad the MM method is compared to the QM method for some␣
↪distances in the QM region.
Here are the distances (Angstrom) as obtained with a QM and an MM method:"""
)

header = "{:<10} {:>10} {:>10} {:>10}".format("Distance", "QM", "MM", "Err(MM)")
print(header)
print("-" * len(header))

for bond, (i, j) in bonds.items():
    dist_qm = mol_qm.get_distance(i, j)
    dist_mm = mol_mm.get_distance(i, j)
    print(f"{bond:<10} {dist_qm:>10.3f} {dist_mm:>10.3f} {dist_mm - dist_qm:>10.3f}")

print(
    """Can we get better results for the QM region with the hybrid engine?
Even though UFF has automatic atom typing, it still matters (in principle) whether we␣
↪specify it on input or not:
    * Without typing for each region the types are automatically guessed
    * With typing the types are always as on input (for all regions)
The only difference is in the C type for the MM region."""
)

results_typing = {}
for typing in [False, True]:
    driver = drivers.AMS()
    driver.Engine = engine
```

(continues on next page)

(continued from previous page)

```
    driver.Task = "GeometryOptimization"
    driver.Engine = engines.Hybrid()
    driver.Engine.QMMM.QMRegion = "qm"
    driver.Engine.QMMM.QMEngineID = "DFTB"
    driver.Engine.QMMM.MMEngineID = "ForceField"
    driver.Engine.Engine[0] = engines.DFTB()
    driver.Engine.Engine[1] = engines.ForceField()

    if typing:
        atom_types = results["ForceField"].get_atom_types()
        mol.enable_atom_attributes("forcefield")
        mm_atoms = mol.get_atoms_outside_region("qm")
        mol.set_atoms_in_region(mm_atoms, "mm")
        for atom, atom_type in zip(mol.atoms, atom_types):
            atom.forcefield.type = atom_type

    job = AMSJob(molecule=mol, settings=settings + Settings({"input": driver}), name=f
↪"hybrid.types={typing}")
    results_typing[typing] = job.run()

systems_typing = {k: v.get_main_system() for k, v in results_typing.items()}

for typing, system in systems_typing.items():
    print(
        f"Here are the distances (Angstrom) as obtained with a QM and an Hybrid␣
↪method {'with' if typing else 'without'} explicit typing:"
    )

    header = "{:<10} {:>10} {:>10} {:>10}".format("Distance", "QM", "MM", "Err(MM)")
    print(header)
    print("-" * len(header))

    for bond, (i, j) in bonds.items():
        dist_qm = mol_qm.get_distance(i, j)
        dist_hybrid = system.get_distance(i, j)
        print(f"{bond:<10} {dist_qm:>10.3f} {dist_hybrid:>10.3f} {dist_hybrid - dist_
↪qm:>10.3f}")

print(
    """Here are some observations for this example:
    * The hybrid engine does better than pure MM
    * The subtle issue whether or not we specify the types has negligible effect
"""
)
print("end report")
```

4.8 Example: Mixing DFT functionals

We consider a system of two weakly bonded molecules, namely NH3 and N2. We will use a GGA for the intra molecular interactions and LDA for the intermolecular one.

We look at two bond lengths, an N-H bond within the NH3 molecule, this is the "intra" bond.

The other is the bond from the N in NH3 to an N atom in the N2 molecule: the "inter" bond.

First we run the whole system with LDA and GGA, and finally with the hybrid engine.

The result for the hybrid calculation is that the "inter" bond has the value of the GGA calculation, whereas the "intra" one is equal to the LDA calculated one.

The energy expression used in the hybrid calculation is

$$E^{\text{hybrid}} = E^{\text{LDA/*}} + E^{\text{GGA/NH3}} - E^{\text{LDA/NH3}} + E^{\text{GGA/N2}} - E^{\text{LDA/N2}}$$

Remember that the region * indicates the whole system, i.e. NH3 + N2.

`Download MixingDFTFunctionals.run`

```
#!/bin/sh

intera1=1
intera2=5

intraa1=5
intraa2=6

report=adf.report

echo "We compare for two weakly coupled systems an inter and intra bond length" >
↪$report
echo "using several methods" >> $report
echo "" >> $report
echo "method    d-inter   d-intra" >> $report

bas=TZP

system=AmmoniaN2

export AMS_JOBNAME=$system.lda

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization
    Convergence Gradients=1e-5
end

System
  Atoms
    N      -1.57871800      -0.04661100       0.00000000 region=one
    H      -2.15862100       0.13639600      -0.80956500 region=one
    H      -2.15862100       0.13639600       0.80956500 region=one
```

(continues on next page)

(continued from previous page)

```
    H      -0.84947100       0.65819300       0.00000000 region=one
    N       1.57871800       0.04661100       0.00000000 region=two
    N       1.03629999      -1.31580113      -0.10254699 region=two
  End
  GuessBonds True
end


Engine adf
   Basis Type=$bas
EndEngine


EOF

dInter=`$AMSBIN/amsreport $AMS_JOBNAME.results/adf.rkf -r distance#$intera1#$intera2`
dIntra=`$AMSBIN/amsreport $AMS_JOBNAME.results/adf.rkf -r distance#$intraa1#$intraa2`

echo "lda    $dInter $dIntra" >> $report



export AMS_JOBNAME=$system.gga

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization
    Convergence Gradients=1e-5
end

LoadSystem
  File $system.lda.results/adf.rkf
End


Engine adf
   xc gga=pbe
   Basis Type=$bas
EndEngine


EOF

dInter=`$AMSBIN/amsreport $AMS_JOBNAME.results/adf.rkf -r distance#$intera1#$intera2`
dIntra=`$AMSBIN/amsreport $AMS_JOBNAME.results/adf.rkf -r distance#$intraa1#$intraa2`

echo "gga    $dInter $dIntra" >> $report



export AMS_JOBNAME=$system.hybrid

rm -rf $AMS_JOBNAME.results
```

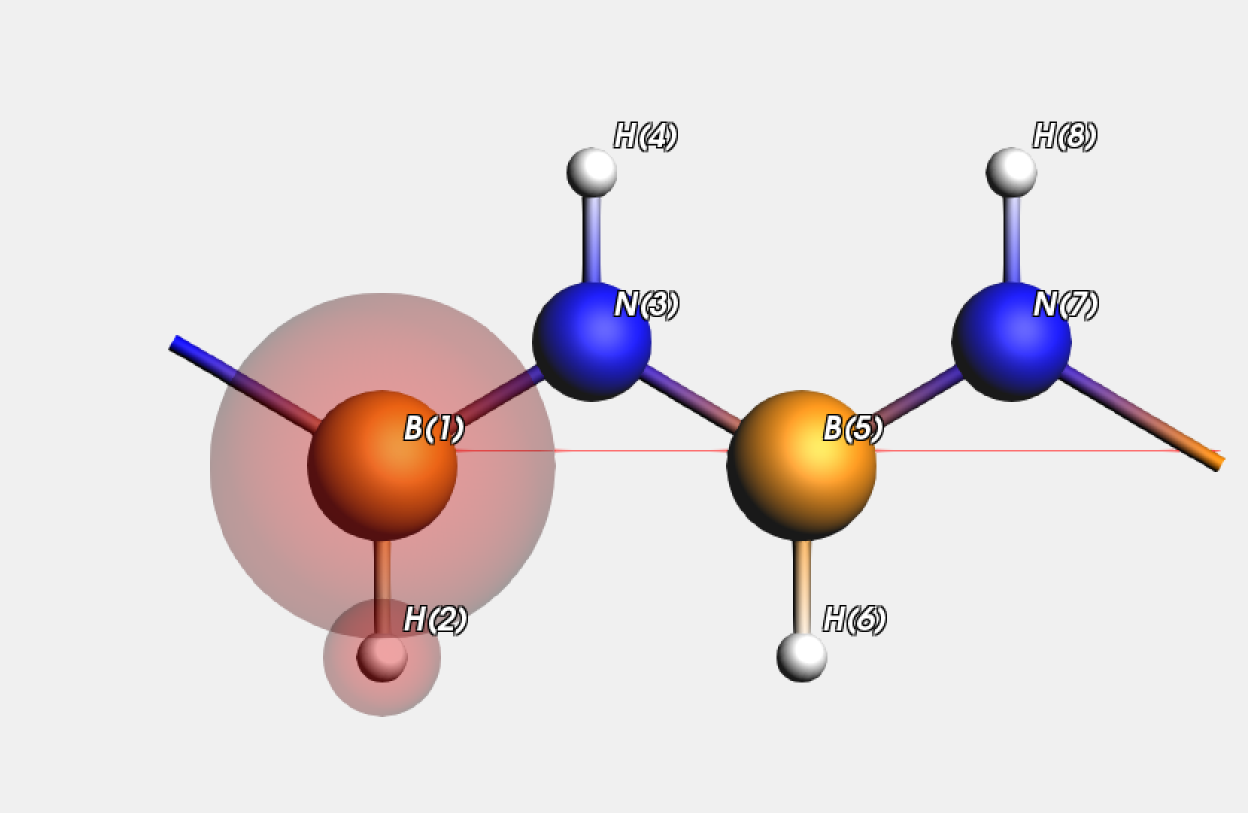
(continues on next page)

(continued from previous page)

```
$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization
    Convergence Gradients=1e-5
end

LoadSystem
  File $system.lda.results/adf.rkf
End

Engine Hybrid
    Energy
        Term Factor=1.0   Region=*    EngineID=adf-lda
        Term Factor=-1.0  Region=one  EngineID=adf-lda
        Term Factor=1.0   Region=one  EngineID=adf-gga
        Term Factor=-1.0  Region=two  EngineID=adf-lda
        Term Factor=1.0   Region=two  EngineID=adf-gga
    End

    Engine adf adf-lda
        Basis Type=$bas
    EndEngine

    Engine adf adf-gga
       xc gga=pbe
        Basis Type=$bas
    EndEngine

 EndEngine

EOF

dInter=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#$intera1#
↪$intera2`
dIntra=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#$intraa1#
↪$intraa2`

echo "hybrid $dInter $dIntra" >> $report

echo "start of report"
cat $report
echo "end of report"
```

4.9 Example: Using capping atoms in a periodic system

Here we look at a polyethylene-like (infinite) chain, the carbons being alternatingly substituted with B and N atoms.



We will use a bunch of different QM systems in a QMMM setup, and check what happens. Inevitably we need to break B-N bonds, and hence capping atoms are used. (In the picture the QM region according to variant one is shown.) We also show the results obtained for the system with the pure QM and MM methods.

Let us have a look at the report generated by the example, that pretty much explains what is done

`Download report PeriodicCapping.txt`

```
We optimize the lattice and test several distances

The system can be cut in several variations into a a QM and an MM part breaking a B-N␣
↪bond

      variation             QM atoms
           var1            B(1),H(2)
           var2            B(5),H(6)
           var3            N(3),H(4)
           var4            N(7),H(8)
           var5  B(1),H(2),N(3),H(4)
           var6  B(5),H(6),N(7),H(8)

Variation one is equivalent to variation two, and variation 3 should be equivalent␣
↪with variation 4

Variation five is equivalent to variation six
```

(continues on next page)

(continued from previous page)

```
Here are the distances (Angstrom) as obtained with a QM and an MM method
  distance          qm          mm
 B(1)-H(2)      1.182      1.185
 B(5)-H(6)      1.182      1.185
 N(3)-H(4)      1.007      1.045
 N(7)-H(8)      1.007      1.045
 B(1)-N(3)      1.431      1.508
 B(5)-N(7)      1.431      1.508

Now we try the hybrid engine with several variations for the QM region

Two capping methods are tried as well.

      variation         capping          energy       B(1)-H(2)       B(5)-H(6)       ␣
↪N(3)-H(4)       N(7)-H(8)       B(1)-N(3)       B(5)-N(7)
           var1           fixed       -2.901499           1.184           1.185       ␣
↪    1.045           1.045           1.508           1.508
           var1      fractional       -2.787165           1.198           1.182       ␣
↪    1.044           1.044           1.673           1.505
           var2           fixed       -2.901499           1.185           1.184       ␣
↪    1.045           1.045           1.508           1.508
           var2      fractional       -2.787165           1.182           1.198       ␣
↪    1.044           1.044           1.505           1.673
           var3           fixed       -4.791110           1.184           1.184       ␣
↪    0.993           1.044           1.508           1.507
           var3      fractional       -4.733046           1.184           1.184       ␣
↪    0.997           1.045           1.657           1.506
           var4           fixed       -4.791110           1.184           1.184       ␣
↪    1.044           0.993           1.507           1.508
           var4      fractional       -4.733046           1.184           1.184       ␣
↪    1.045           0.997           1.506           1.657
           var5           fixed       -6.741093           1.189           1.187       ␣
↪    1.003           1.045           1.390           1.489
           var5      fractional       -6.648313           1.198           1.183       ␣
↪    1.003           1.045           1.405           1.505
           var6           fixed       -6.741093           1.187           1.189       ␣
↪    1.045           1.003           1.489           1.390
           var6      fractional       -6.648313           1.183           1.198       ␣
↪    1.045           1.003           1.505           1.405

Here are some observations
     * generally the fixed capping seems a bit better

Here are some remarks
     * Starting from the initial very bad structure the fixed capping fails␣
↪completely for variant 5 and 6
       (not if you use as qm engine band and as mm engine dftb)
     * A reasonable starting geometry can avoid strange collapses
     * The more the two engines disagree about the capped QM region, the stronger the␣
↪capping forces
```

Download PeriodicCapping.run

```
#!/bin/sh

export NSCM=1
```

(continues on next page)

(continued from previous page)

```
report=report.txt

STRUCTDIR=$AMSHOME/examples/Hybrid/PeriodicCapping/systems

# ensure that not a comma is used for decimals in the printf function
LC_NUMERIC=en_US.UTF-8

export AMS_JOBNAME=reference

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization OptimizeLattice=yes Method=FIRE MaxIterations=300

System
   GeometryFile $STRUCTDIR/var1.xyz
   GuessBonds true
end

Engine DFTB
EndEngine

EOF

aaa1qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#1#2`
aaa2qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#5#6`
bbb1qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#3#4`
bbb2qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#7#8`
ccc1qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#1#3`
ccc2qm=`$AMSBIN/amsreport $AMS_JOBNAME.results/dftb.rkf -r distance#5#7`

printf "We optimize the lattice and test several distances\n" > $report

printf "\nThe system can be cut in several variations into a a QM and an MM part␣
↪breaking a B-N bond\n" >>$report

printf "\n%15s %20s\n" "variation" "QM atoms" >>$report
printf "%15s %20s\n" "var1" "B(1),H(2)" >>$report
printf "%15s %20s\n" "var2" "B(5),H(6)" >>$report
printf "%15s %20s\n" "var3" "N(3),H(4)" >>$report
printf "%15s %20s\n" "var4" "N(7),H(8)" >>$report
printf "%15s %20s\n" "var5" "B(1),H(2),N(3),H(4)" >>$report
printf "%15s %20s\n" "var6" "B(5),H(6),N(7),H(8)" >>$report

printf "\nVariation one is equivalent to variation two, and variation 3 should be␣
↪equivalent with variation 4\n" >>$report
printf "\nVariation five is equivalent to variation six\n" >>$report


export AMS_JOBNAME=cheap

rm -rf $AMS_JOBNAME.results
```

(continues on next page)

(continued from previous page)

```
$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization OptimizeLattice=yes Method=FIRE MaxIterations=300

System
   GeometryFile $STRUCTDIR/var1.xyz
   GuessBonds true
end

Engine ForceField
EndEngine

EOF

aaa1mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#1#2`
aaa2mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#5#6`
bbb1mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#3#4`
bbb2mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#7#8`
ccc1mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#1#3`
ccc2mm=`$AMSBIN/amsreport $AMS_JOBNAME.results/forcefield.rkf -r distance#5#7`

printf "\nHere are the distances (Angstrom) as obtained with a QM and an MM method\n"␣
↪>> $report
printf "%10s %10s %10s\n"   "distance"  "qm"  "mm" >> $report
printf "%10s %10.3f %10.3f\n"   "B(1)-H(2)"  $aaa1qm $aaa1mm >> $report
printf "%10s %10.3f %10.3f\n"   "B(5)-H(6)"  $aaa2qm $aaa2mm >> $report
printf "%10s %10.3f %10.3f\n"   "N(3)-H(4)"  $bbb1qm $bbb1mm >> $report
printf "%10s %10.3f %10.3f\n"   "N(7)-H(8)"  $bbb2qm $bbb2mm >> $report
printf "%10s %10.3f %10.3f\n"   "B(1)-N(3)"  $ccc1qm $ccc1mm >> $report
printf "%10s %10.3f %10.3f\n"   "B(5)-N(7)"  $ccc2qm $ccc2mm >> $report


printf "\nNow we try the hybrid engine with several variations for the QM region\n" >>
↪ $report
printf "\nTwo capping methods are tried as well.\n" >>$report


printf "\n%15s %15s %15s %15s %15s %15s %15s %15s %15s\n" "variation" "capping"
↪"energy" "B(1)-H(2)" "B(5)-H(6)"  "N(3)-H(4)" "N(7)-H(8)" "B(1)-N(3)"  "B(5)-N(7)" >
↪> $report

for system in var1 var2 var3 var4 var5 var6
do

# This calc in only needed to start from a reasonable guess

export AMS_JOBNAME=$system.cheap

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams<<EOF

Task GeometryOptimization

GeometryOptimization OptimizeLattice=yes Method=FIRE MaxIterations=300
```

(continues on next page)

(continued from previous page)

```
System
   GeometryFile $STRUCTDIR/$system.xyz
   GuessBonds true
end

Engine ForceField
EndEngine

EOF


for embedding in electrostatic
do

for capping in fixed fractional
do

export AMS_JOBNAME=$system.embedding=$embedding.capping=$capping.go

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams<<EOF


Task GeometryOptimization

GeometryOptimization OptimizeLattice=yes Method=FIRE MaxIterations=100

LoadSystem
 File $system.cheap.results
End

Engine Hybrid

        Capping AllowHighBondOrders=true Option=$capping

  QMMM qmRegion=qm qmEngineID=dftb mmEngineID=ForceField Embedding=$embedding

    Engine Band
    EndEngine

    Engine DFTB
    EndEngine

    Engine ForceField
    EndEngine

 EndEngine

EOF

aaa1=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#2`
aaa2=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#5#6`
bbb1=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#3#4`
bbb2=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#7#8`
ccc1=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#1#3`
```

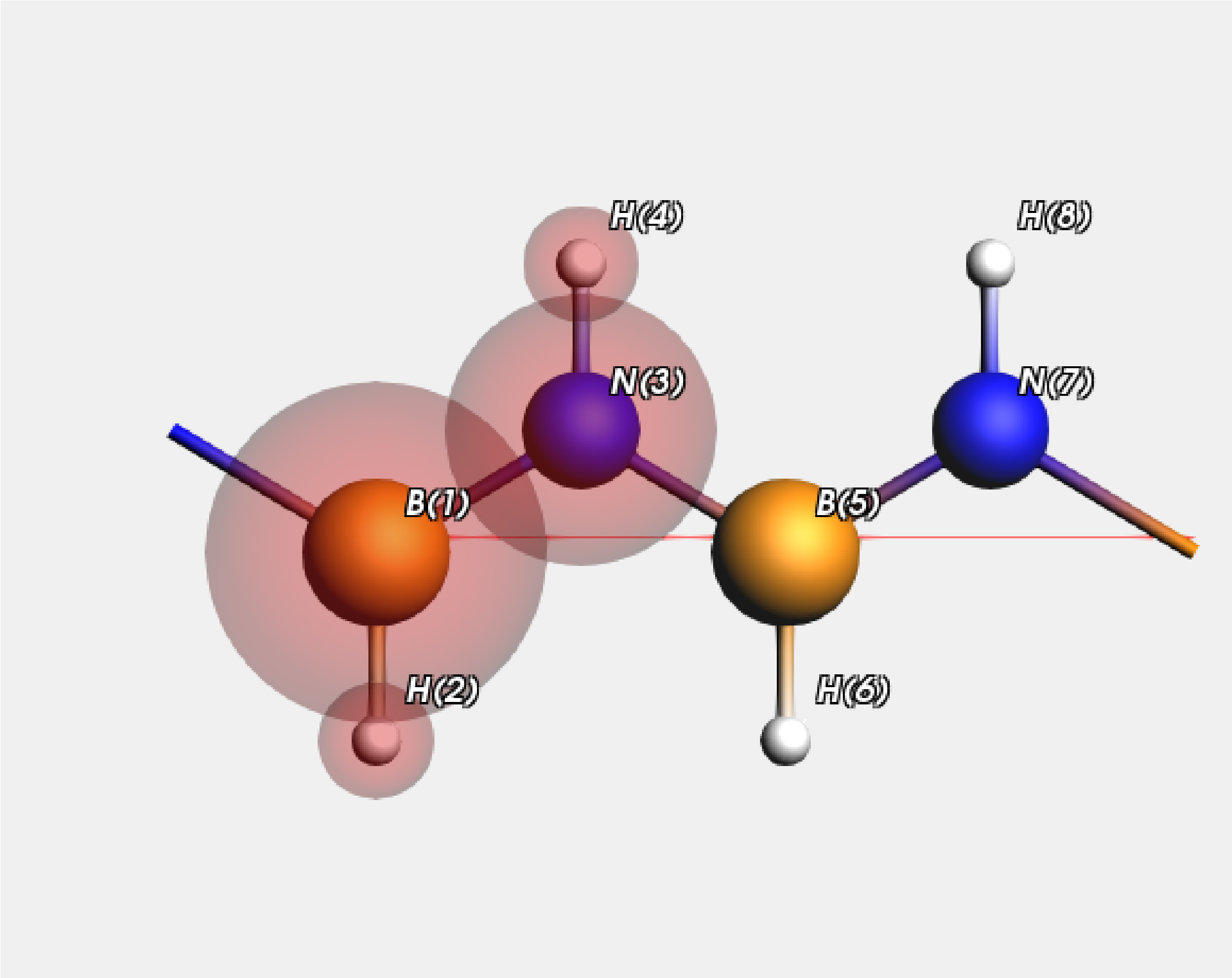
(continues on next page)

(continued from previous page)

```
ccc2=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -r distance#5#7`
xxx=`$AMSBIN/amsreport $AMS_JOBNAME.results/hybrid.rkf -k "AMSResults%Energy"`

printf "%15s %15s %15.6f %15.3f %15.3f %15.3f %15.3f %15.3f %15.3f\n" $system
↪$capping $xxx $aaa1 $aaa2 $bbb1 $bbb2 $ccc1  $ccc2 >> $report

done
done
done

printf "\nHere are some observations\n" >>$report
printf "     * generally the fixed capping seems a bit better\n" >>$report

printf "\nHere are some remarks\n" >>$report
printf "     * Starting from the initial very bad structure the fixed capping fails␣
↪completely for variant 5 and 6\n" >>$report
printf "       (not if you use as qm engine band and as mm engine dftb)\n" >>$report
printf "     * A reasonable starting geometry can avoid strange collapses\n" >>$report
printf "     * The more the two engines disagree about the capped QM region, the␣
↪stronger the capping forces\n" >>$report

echo "begin report"
cat $report
echo "end report"
```

4.10 Example: Using capping atoms in a periodic system with charges

Now we look at a BN chain and use charges for the MM calculation.



Let us have a look at the report generated by the example, that pretty much explains what is done

Download report PeriodicCappingWithCharges.txt

```
We optimize the lattice and test several distances.

We divide the system in such a way that there are two equivalent, and hence neutral␣
↪regions.

Here are the distances (Angstrom) as obtained with a QM and an MM method:

Distance            QM          MM    Err(MM)
-------------------------------------------
B-H              1.182       1.181     -0.001
N-H              1.007       1.042      0.034
B-N              1.431       1.499      0.068

Of course the force field results do not exactly match the QM results, the error␣
↪displayed in the last column.

Now we try the hybrid engine, can we improve the bonds in the QM region?

We start from the geometry calculated with the (cheap) forcefield.
```

(continues on next page)

(continued from previous page)

```
In this table we show the errors in bond lengths (in the QM region) of the hybrid␣
↪method with respect to the QM method:

Embedding                    Capping       Energy      B-H      N-H      B-N
-------------------------------------------------------------------------
mechanical                     fixed    -6.844721    0.004   -0.007   -0.044
mechanical                fractional    -6.745243    0.012   -0.007   -0.042
electrostatic                  fixed    -6.749110    0.002   -0.004   -0.036
electrostatic             fractional    -6.652361    0.010   -0.004   -0.026

Here are some observations
     * the B-H distance is a bit worse than with a plain forcefield, especially with␣
↪fractional capping
     * the N-H distance is much better than with the plain forcefield
     * the B-N distance is a bit better than with the plain forcefield, now too short.
↪ Fractional capping works best.
     * Electrostatic embedding is doing slightly better than mechanical embedding,␣
↪the biggest improvement is on the B-N bond
```

Note that this example makes use of the python scripting tools PLAMS and PISA, available in the AMS Python Stack.

`Download PeriodicCappingWithCharge.py`

```
#!/usr/bin/env amspython

from scm.base import ChemicalSystem, Units
from scm.input_classes import drivers, engines
from scm.plams import AMSJob, JobRunner, config, Settings
import os
import copy

# Load molecule and run initial geometry optimization
config.default_jobrunner = JobRunner(parallel=True)
settings = Settings()
settings.runscript.nproc = 1

struct_dir = os.path.join(os.getenv("AMSHOME"), "examples", "Hybrid", "PeriodicCapping
↪", "systems")

mol = ChemicalSystem.from_xyz(os.path.join(struct_dir, "var5.xyz"))
mol.guess_bonds()

driver_geom_opt = drivers.AMS()
driver_geom_opt.Task = "GeometryOptimization"
driver_geom_opt.Engine = engines.DFTB()
driver_geom_opt.GeometryOptimization.OptimizeLattice = True
driver_geom_opt.GeometryOptimization.Method = "FIRE"
driver_geom_opt.GeometryOptimization.MaxIterations = 300

job_dftb = AMSJob(molecule=mol, settings=settings + Settings({"input": driver_geom_
↪opt}), name="reference")
results_dftb = job_dftb.run()

# Get charges and add them to molecule
mol.enable_atom_attributes("forcefield")
charges = results_dftb.get_charges()
for atom, charge in zip(mol.atoms, charges):
```

(continues on next page)

(continued from previous page)

```
    atom.forcefield.charge = charge

# Run cheap calculation using forcefield engine
driver_geom_opt.Engine = engines.ForceField()
driver_geom_opt.Engine.NonBondedCutoff = 50 * Units.conversion_factor("bohr",
↪"angstrom")
job_ff = AMSJob(molecule=mol, settings=settings + Settings({"input": driver_geom_opt}
↪), name="cheap")
results_ff = job_ff.run()
mol = results_ff.get_main_system()

print("begin report")
print(
    """We optimize the lattice and test several distances.
We divide the system in such a way that there are two equivalent, and hence neutral␣
↪regions.
Here are the distances (Angstrom) as obtained with a QM and an MM method:
"""
)

header = "{:<10} {:>10} {:>10} {:>10}".format("Distance", "QM", "MM", "Err(MM)")
print(header)
print("-" * len(header))

mol_qm = results_dftb.get_main_system()
mol_mm = results_ff.get_main_system()
for bond, i, j in [("B-H", 0, 1), ("N-H", 2, 3), ("B-N", 0, 2)]:
    dist_qm = mol_qm.get_distance(i, j)
    dist_mm = mol_mm.get_distance(i, j)
    print(f"{bond:<10} {dist_qm:>10.3f} {dist_mm:>10.3f} {dist_mm - dist_qm:>10.3f}")

# Run hybrid engine
results = {}
for embedding in ["mechanical", "electrostatic"]:
    for capping in ["fixed", "fractional"]:
        driver = copy.deepcopy(driver_geom_opt)
        driver.Properties.Gradients = True
        driver.GeometryOptimization.MaxIterations = 100

        driver.Engine = engines.Hybrid()
        driver.Engine.Capping.AllowHighBondOrders = True
        driver.Engine.Capping.Option = capping
        driver.Engine.QMMM.QMRegion = "qm"
        driver.Engine.QMMM.QMEngineID = "DFTB"
        driver.Engine.QMMM.MMEngineID = "ForceField"
        driver.Engine.QMMM.Embedding = embedding

        driver.Engine.Engine[0] = engines.BAND()
        driver.Engine.Engine[1] = engines.DFTB()
        driver.Engine.Engine[2] = engines.ForceField()
        driver.Engine.Engine[2].NonBondedCutoff = 50 * Units.conversion_factor("bohr",
↪ "angstrom")

        job = AMSJob(
            molecule=mol,
            settings=settings + Settings({"input": driver}),
            name=f"system.embedding={embedding}.capping={capping}.go",
```

(continues on next page)

(continued from previous page)

```
        )

        results[(embedding, capping)] = job.run()

hybrid_results = {k: (v.get_energy(engine="hybrid"), v.get_main_system()) for k, v in␣
↪results.items()}

print(
    """Of course the force field results do not exactly match the QM results, the␣
↪error displayed in the last column.
Now we try the hybrid engine, can we improve the bonds in the QM region?
We start from the geometry calculated with the (cheap) forcefield.
In this table we show the errors in bond lengths (in the QM region) of the hybrid␣
↪method with respect to the QM method:
"""
)

header = "{:<16} {:>16} {:>12} {:>8} {:>8} {:>8}".format("Embedding", "Capping",
↪"Energy", "B-H", "N-H", "B-N")
print(header)
print("-" * len(header))

for (embedding, capping), (energy, mol_hybrid) in hybrid_results.items():
    errs = [mol_hybrid.get_distance(i, j) - mol_qm.get_distance(i, j) for i, j in [(0,
↪ 1), (2, 3), (0, 2)]]
    print(f"{embedding:<16} {capping:>16} {energy:>12.6f} {errs[0]:>8.3f} {errs[1]:>8.
↪3f} {errs[2]:>8.3f}")

print(
    """Here are some observations
     * the B-H distance is a bit worse than with a plain forcefield, especially with␣
↪fractional capping
     * the N-H distance is much better than with the plain forcefield
     * the B-N distance is a bit better than with the plain forcefield, now too short.
↪ Fractional capping works best.
     * Electrostatic embedding is doing slightly better than mechanical embedding,␣
↪the biggest improvement is on the B-N bond
"""
)
print("end report")
```

4.11 Example: QMMM with capping atoms

This is an example of QMMM using capping atoms. Capping atoms are added automatically when bonds are broken (between the QM and MM region). Because the amber forcefield is used the `AtomicInfoForCappingAtom` needs to be set, as the default type “H” is not an AMBER type.

`Download QMMM_Butane.run`

```
#! /bin/sh

"$AMSBIN/ams" <<eor

Task GeometryOptimization
```

(continues on next page)

(continued from previous page)

```
GeometryOptimization
    Convergence Gradients=1E-4
End

System
    Atoms
        C  0.0000  0.0000  0.0000  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=CT
        H  1.0910  0.0000  0.0000  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -0.3598  1.0300  0.0000  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        C -0.5021 -0.7074 -1.2586  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=CT
        H -0.1397 -1.7383 -1.2662  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -1.5949 -0.7053 -1.2662  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -0.1307 -0.1841 -2.1433  region=QM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        C -0.5195 -0.7318  1.2374  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=CT
        C -0.0376 -0.0530  2.5227  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=CT
        H  1.0524 -0.0554  2.5580  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -0.3994  0.9752  2.5580  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -0.4230 -0.5961  3.3860  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -0.1631 -1.7630  1.2286  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=HC
        H -1.6105 -0.7355  1.2286  region=MM  ForceField.Charge=0.0  ForceField.
↪Type=HC
    End
    BondOrders
        1 2  1.0
        1 3  1.0
        1 8  1.0
        1 4  1.0
        8 14 1.0
        8 13 1.0
        8 9  1.0
        4 5  1.0
        4 6  1.0
        4 7  1.0
        9 12 1.0
        9 11 1.0
        9 10 1.0
    End
End

Engine Hybrid
    QMMM
        QMRegion QM
        QMEngineID ADF
```

(continues on next page)

(continued from previous page)

```
        MMEngineID ForceField
    End

    Capping
        AtomicInfoForCappingAtom ForceField.Type=HC ForceField.Charge=0.0
    End

    Engine ADF
      Basis
        Type DZP
      End
      Relativity
        Level None
      End
    EndEngine

    Engine ForceField
      Type Amber95
    EndEngine
EndEngine

eor
```

4.12 Example*: Mechanical embedding (QUILD)

This is a fairly technical test, testing the case insensitivity of region and engine names. Furthermore gradients are tested.

Download `qmmm_water.run`

```
#!/bin/sh

hybridenginename=hybrid     # this is a temporary hack
report=report.txt

echo "Start of the report" > $report


echo "------------------------------" >> $report
echo "Start of the region names test" >> $report
echo "------------------------------" >> $report

export AMS_JOBNAME=regionnamestest

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task SinglePoint

Properties Gradients=yes

System
  Atoms
     C 0.0  0.0 0.0  region=A
```

(continues on next page)

(continued from previous page)

```
     O 1.13 0.0 0.0  region=A
     O -1.13 0.0 0.0 region=B
  End
  BondOrders
       1 2 2.0
       1 3 2.0
  End
End

Engine Hybrid
   Energy
      Term factor=1.0   region=*     engineID=ForceField
      Term factor=1.0   region=A       engineID=dftB-2
      Term factor=-1.0  region=A       engineID=ForceField
   End

   Capping
      AllowHighBondOrders True # Because we cut through double bonds here ...
      AtomicInfoForCappingAtom ForceField.Type=H_   # Remove ForceField.Charge␣
↪because we do not specify charges for other atoms
   End

   Engine ForceField
   EndEngine
   Engine dftb dftb-1
     Model GFN1-xTB
   EndEngine
   Engine dftb dFtb-2
     Model SCC-DFTB
     ResourcesDir QUASINANO2015
   EndEngine
   Engine mopac
   EndEngine

EndEngine
eor

echo "Energy for region names test" >> $report
$AMSBIN/amsreport $AMS_JOBNAME.results/$hybridenginename.rkf -k "AMSResults%Energy"  ␣
↪>> $report

echo "----------------------------" >> $report
echo "End of the region names test" >> $report
echo "----------------------------" >> $report



echo "--------------------------" >> $report
echo "Start of the gradient test" >> $report
echo "--------------------------" >> $report

# strange capitalization is on purpose

for capping in fractionAl fiXed
do

for num in true false
```

(continues on next page)

(continued from previous page)

```
do

export AMS_JOBNAME=gradtest.capping=$capping.num=$num

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams <<eor

Task SinglePoint


Properties Gradients=yes

EngineDebugging IgnoreGradientsRequest=$num NeverQuiet=false

NumericalDifferentiation NuclearStepSize=1.0e-4

System
  Atoms
     C 0.0  0.0 0.0  region=a
     O 1.13 0.0 0.0  region=a
     O -1.13 0.0 0.0 region=b
  End
  BondOrders
       1 2 2.0
       1 3 2.0
  End
End

Engine Hybrid
   Capping
      Option $capping
      AllowHighBondOrders True # Because we cut through double bonds here ...
      AtomicInfoForCappingAtom ForceField.Type=H_   # Remove ForceField.Charge␣
↪because we do not specify charges for other atoms
   End
   Energy
      Term factor=1.0   region=*     engineID=ForceField
      Term factor=1.0   region=a       engineID=dftb-2
      Term factor=-1.0  region=a       engineID=ForceField
   End

   Engine ForceField
   EndEngine
   Engine dftb dftb-1
     Model GFN1-xTB
   EndEngine
   Engine dftb dftb-2
     Model SCC-DFTB
     ResourcesDir QUASINANO2015
   EndEngine
   Engine mopac
   EndEngine

EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

echo "gradients for capping=$capping num=$num:" >> $report
$AMSBIN/amsreport $AMS_JOBNAME.results/$hybridenginename.rkf -k "AMSResults%Gradients##3"   >> $report

done
done

echo "-------------------------" >> $report
echo "End of the gradient test" >> $report
echo "-------------------------" >> $report



echo "----------------------------" >> $report
echo "Start of the singleton test" >> $report
echo "----------------------------" >> $report

echo "Calculate the same system twice with the same engine but with different settings. The energy should be non-zero" >> $report

for engine in dftb band adf mopac
do
export AMS_JOBNAME=testsingleton.engine.$engine

rm -rf $AMS_JOBNAME.results

$AMSBIN/ams << eor

Task SinglePoint

System
   GeometryFile $AMSHOME/atomicdata/Molecules/TestMols/Acetamide.xyz
End

Engine Hybrid
   AllowSanityCheckWarnings true
   Energy
      Term factor=1.0   region=*       engineID=$engine-1
      Term factor=-1.0  region=*       engineID=$engine-2
   End

   Engine band band-1
      xc gga=pbe
   EndEngine
   Engine band band-2
      basis type=SZ
   EndEngine
   Engine adf adf-1
      xc gga=pbe
   EndEngine
   Engine adf adf-2
      basis type=SZ
   EndEngine
   Engine mopac mopac-1
      Model PM6
```

(continues on next page)

(continued from previous page)

```
   EndEngine
   Engine mopac mopac-2
      Model PM7
   EndEngine
   Engine DFTB dftb-1
      Model GFN1-xTB
   EndEngine
   Engine DFTB dftb-2
      Model SCC-DFTB
      ResourcesDIR QUASINANO2015
   EndEngine

EndEngine

eor


echo "Hopfully nonzero energy for engine=$engine" >> $report
$AMSBIN/amsreport $AMS_JOBNAME.results/$hybridenginename.rkf -k "AMSResults%Energy" ␣
↪>> $report

done

echo "--------------------------" >> $report
echo "End of the singleton test" >> $report
echo "--------------------------" >> $report


cat $report
```

4.13 Example: pdb2adf transforms a PDB file to a QMMM input file

`Download pdb2adf.run`

```
#! /bin/sh

# This example shows how to use the utility pdb2adf,
# which creates an amsified ADF input file (ADF>=2020) from a PDB file,
# for a subsequent QM/MM calculation using ADF.

# -------------------------
# First create the PDB file
# -------------------------

cat << eor > chymotrypsin.pdb
HEADER    COMPLEX (SERINE PROTEASE/INHIBITOR)     12-MAR-97   1AFQ
TITLE     CRYSTAL STRUCTURE OF BOVINE GAMMA-CHYMOTRYPSIN COMPLEXED
TITLE    2 WITH A SYNTHETIC INHIBITOR
REMARK
REMARK    Adaptation of original PDB file by M. Swart, March 2005
REMARK    only coordinates of GAMMA-CHYMOTRYPSIN are kept;
REMARK    rest has been deleted.
REMARK
ATOM      1  N   CYS A   1      13.717  20.021  22.754  1.00 13.46      PROA N
```

(continues on next page)

(continued from previous page)

```
ATOM      2  CA  CYS A   1      14.211  18.932  23.617  1.00 13.34      PROA C
ATOM      3  C   CYS A   1      13.597  19.033  25.005  1.00 13.34      PROA C
ATOM      4  O   CYS A   1      12.953  20.026  25.329  1.00 13.48      PROA O
ATOM      5  CB  CYS A   1      15.734  19.018  23.753  1.00 13.44      PROA C
ATOM      6  SG  CYS A   1      16.298  20.647  24.361  1.00 13.30      PROA S
ATOM      7  N   GLY A   2      13.801  17.985  25.813  1.00 13.44      PROA N
ATOM      8  CA  GLY A   2      13.369  17.952  27.214  1.00 13.65      PROA C
ATOM      9  C   GLY A   2      11.904  18.088  27.631  1.00 13.87      PROA C
ATOM     10  O   GLY A   2      11.669  18.375  28.799  1.00 13.63      PROA O
ATOM     11  N   VAL A   3      10.947  17.887  26.732  1.00 14.18      PROA N
ATOM     12  CA  VAL A   3       9.559  17.968  27.090  1.00 14.86      PROA C
ATOM     13  C   VAL A   3       8.875  16.684  26.624  1.00 15.04      PROA C
ATOM     14  O   VAL A   3       8.529  16.546  25.452  1.00 14.91      PROA O
ATOM     15  CB  VAL A   3       8.861  19.211  26.437  1.00 15.00      PROA C
ATOM     16  CG1 VAL A   3       7.403  19.299  26.880  1.00 15.08      PROA C
ATOM     17  CG2 VAL A   3       9.585  20.486  26.805  1.00 15.27      PROA C
ATOM     18  N   PRO A   4       8.754  15.691  27.519  1.00 15.39      PROA N
ATOM     19  CA  PRO A   4       8.121  14.407  27.206  1.00 16.09      PROA C
ATOM     20  C   PRO A   4       6.675  14.535  26.769  1.00 16.35      PROA C
ATOM     21  O   PRO A   4       5.957  15.387  27.275  1.00 16.43      PROA O
ATOM     22  CB  PRO A   4       8.219  13.635  28.527  1.00 15.91      PROA C
ATOM     23  CG  PRO A   4       9.369  14.297  29.244  1.00 16.02      PROA C
ATOM     24  CD  PRO A   4       9.166  15.742  28.928  1.00 15.56      PROA C
ATOM     25  N   ALA A   5       6.262  13.690  25.827  1.00 16.87      PROA N
ATOM     26  CA  ALA A   5       4.874  13.703  25.351  1.00 17.53      PROA C
ATOM     27  C   ALA A   5       4.020  13.055  26.437  1.00 17.86      PROA C
ATOM     28  O   ALA A   5       2.862  13.413  26.643  1.00 17.81      PROA O
ATOM     29  CB  ALA A   5       4.740  12.936  24.027  1.00 17.34      PROA C
ATOM     30  N   ILE A   6       4.615  12.104  27.143  1.00 18.37      PROA N
ATOM     31  CA  ILE A   6       3.941  11.419  28.239  1.00 18.99      PROA C
ATOM     32  C   ILE A   6       4.553  11.993  29.514  1.00 19.59      PROA C
ATOM     33  O   ILE A   6       5.726  11.771  29.807  1.00 19.17      PROA O
ATOM     34  CB  ILE A   6       4.190   9.909  28.190  1.00 19.29      PROA C
ATOM     35  CG1 ILE A   6       3.631   9.335  26.886  1.00 19.66      PROA C
ATOM     36  CG2 ILE A   6       3.552   9.232  29.399  1.00 19.38      PROA C
ATOM     37  CD1 ILE A   6       3.977   7.887  26.674  1.00 20.59      PROA C
ATOM     38  N   GLN A   7       3.760  12.742  30.265  1.00 20.52      PROA N
ATOM     39  CA  GLN A   7       4.262  13.374  31.468  1.00 21.69      PROA C
ATOM     40  C   GLN A   7       4.683  12.459  32.597  1.00 22.02      PROA C
ATOM     41  O   GLN A   7       3.954  11.535  32.978  1.00 21.89      PROA O
ATOM     42  CB  GLN A   7       3.259  14.392  31.997  1.00 22.77      PROA C
ATOM     43  CG  GLN A   7       3.369  15.749  31.349  1.00 24.86      PROA C
ATOM     44  CD  GLN A   7       2.467  16.774  32.004  1.00 25.86      PROA C
ATOM     45  OE1 GLN A   7       1.660  17.417  31.337  1.00 27.24      PROA O
ATOM     46  NE2 GLN A   7       2.601  16.934  33.325  1.00 26.82      PROA N
ATOM     47  N   PRO A   8       5.898  12.675  33.125  1.00 22.41      PROA N
ATOM     48  CA  PRO A   8       6.345  11.830  34.231  1.00 22.84      PROA C
ATOM     49  C   PRO A   8       5.524  12.215  35.459  1.00 23.50      PROA C
ATOM     50  O   PRO A   8       5.069  13.359  35.575  1.00 23.62      PROA O
ATOM     51  CB  PRO A   8       7.821  12.219  34.389  1.00 22.58      PROA C
ATOM     52  CG  PRO A   8       7.864  13.641  33.894  1.00 22.61      PROA C
ATOM     53  CD  PRO A   8       6.972  13.583  32.678  1.00 22.34      PROA C
ATOM     54  N   VAL A   9       5.267  11.244  36.323  1.00 24.00      PROA N
ATOM     55  CA  VAL A   9       4.516  11.478  37.543  1.00 24.57      PROA C
ATOM     56  C   VAL A   9       5.471  11.122  38.665  1.00 24.99      PROA C
ATOM     57  O   VAL A   9       5.927   9.982  38.759  1.00 24.82      PROA O
ATOM     58  CB  VAL A   9       3.273  10.580  37.613  1.00 24.64      PROA C
```

(continues on next page)

(continued from previous page)

```
ATOM     59  CG1 VAL A   9       2.596  10.725  38.969  1.00 24.88      PROA C
ATOM     60  CG2 VAL A   9       2.308  10.935  36.488  1.00 24.78      PROA C
ATOM     61  N   LEU A  10       5.827  12.119  39.464  1.00 25.78      PROA N
ATOM     62  CA  LEU A  10       6.752  11.921  40.568  1.00 26.56      PROA C
ATOM     63  C   LEU A  10       6.043  11.958  41.914  1.00 26.89      PROA C
ATOM     64  O   LEU A  10       5.187  12.847  42.105  1.00 27.07      PROA O
ATOM     65  CB  LEU A  10       7.857  12.973  40.501  1.00 26.93      PROA C
ATOM     66  CG  LEU A  10       8.721  12.752  39.255  1.00 27.39      PROA C
ATOM     67  CD1 LEU A  10       9.351  14.055  38.769  1.00 27.66      PROA C
ATOM     68  CD2 LEU A  10       9.768  11.681  39.555  1.00 27.46      PROA C
ATOM     69  OXT LEU A  10       6.329  11.066  42.743  1.00 27.55      PROA O
TER      70      LEU A  10
END
eor

# ---------------------------------------------------------------------------------------
↪--
# then run program to create ADF inputfile
# The program works interactively. The input described here are answers to the␣
↪questions
# that were asked interactively.
# In cases where the user agrees with the suggestion given by the program,
# the user can press the **Enter** key, which is shown here with an empty line.
# ---------------------------------------------------------------------------------------
↪--

$AMSBIN/pdb2adf << eor
chymotrypsin.pdb




3 4 5
c
5
3 4 15 16
c


Y
1
1
17.5
eor

# ---------------------------------------------------------------------------------------
↪--
# Questions asked were:
#  Q1: Please give name of PDB-file
#  A1: chymotrypsin.pdb
#  Q2: Do you want a logfile to be written (Y/n) ?
#  A2: Enter
#  Q3: ..
#      Found the following terminal amino acid residues : (C-term)    10 (N-term)    ␣
↪1
#      Do you want to use these as terminal residues (Y/n) ?
#  A3: Enter
```

(continues on next page)

(continued from previous page)

```
#  Q4: Multiple AMBER options for CYS :
#        0         Decide every time differently
#        1   CYS   Cysteine (SH)
#        2   CYM   Deprotonated Cysteine (S-)
#        3   CYX   Cystine (S-S bridge)
#      Suggested option: 0
#  A4: Enter
#  Q5: Multiple AMBER options for CYS    1 (    1) :
#      ...
#      Suggestion: 1
#  A5: Enter
#  Q6: ...
#      Option  Molecule    Option  Molecule    Option  Molecule    Option  Molecule  ␣
↪  Option  Molecule
#           1: CYS    1          4: PRO    4          7: GLN    7        10: LEU   10
#           2: GLY    2          5: ALA    5          8: PRO    8
#           3: VAL    3          6: ILE    6          9: VAL    9
#      Give option number of molecules to be put in QM region (or 'c' to continue):
#      Note: by specifying a negative number a molecule is removed from the QM region
#  A6: 3 4 5
#  Q7:    ...
#      Give option number of molecules to be put in QM region (or 'c' to continue):
#      Note: by specifying a negative number a molecule is removed from the QM region
#  A7: c
#  Q8: Make a choice for the QM/MM treatment of VAL    3
#       0:  Put completely in QM region
#       1:  Cut off at C-alpha (put NH in QM region, CO in MM region)
#       2:  Cut off at C-alpha (put NH in MM region, CO in QM region)
#       3:  Cut off at C-alpha (put NH and CO in MM region)
#       4:  Cut off at C-alpha (put NH and CO in QM region,  sidechain in MM region)
#       5:  Put only part of sidechain in QM region
#      Suggestion: 2
#      Give choice:
#  A8: 5
#  Q9: Atoms belonging to molecule VAL     3
#         1: N     MM                6: HB    MM               11: CG2   MM          ␣
↪    16: O     MM
#         2: H     MM                7: CG1   MM               12: HG21  MM
#         3: CA    MM                8: HG11  MM               13: HG22  MM
#         4: HA    MM                9: HG12  MM               14: HG23  MM
#         5: CB    MM               10: HG13  MM               15: C     MM
#      Give option number of atoms to be put in QM region (or 'c' to continue):
#      (Note: a range can be entered as 3-21, while a negative number removes an atom)
#  A9: 3 4 15 16
# Q10: ...
#      Give option number of atoms to be put in QM region (or 'c' to continue):
#      (Note: a range can be entered as 3-21, while a negative number removes an atom)
# A10: c
# Q11: Make a choice for the QM/MM treatment of PRO    4
#      ...
#      Suggestion: 2
# A11: Enter
# Q12: Make a choice for the QM/MM treatment of ALA    5
#      ...
#      Suggestion: 1
# A12: Enter
# Q13: Do you want to add solvent to your system (Y/n) ?
```

(continues on next page)

(continued from previous page)

```
# A13: Y
# Q14: Solvent (box) available:
#         1:  HOH     HOH  Water molecule
#         2:  MOH     MOH  Methanol molecule
#         3:  CHL     CHL  Chloroform molecule
# A14: 1
# Q15: Box Shape options:
#       1  Spherical box
#       2  Cubic box
# A15: 1
# Q16: Give boxsize (def.:    16.71 Angs)
# A16: 17.5
# ----------------------------------------------------------------------------------
↪--

# ----------------------------
# now for checking correctness
# ----------------------------

echo ""
echo "ADF inputfile made by pdb2adf"
echo ""

cat chymotrypsin.pdb2adf

rm chymotrypsin.pdb chymotrypsin.pdb.log chymotrypsin.p2a.pdb chymotrypsin.pdb2adf
```

CHAPTER

FIVE

KEYWORDS

5.1 Links to manual entries

hybrid:

- *Capping* (page 16)
- *Committee* (page 15)
- *Energy* (page 12)
- *Engine* (page 11)
- *QMMM* (page 14)
- *RestartSubEngines* (page 18)

5.2 Summary of all keywords

5.2.1 Engine Hybrid

`AllowSanityCheckWarnings`

Type
Bool

Default value
No

Description
Sanity checks will be performed on the setup. If this option is on, only warnings are printed. If not the program will stop on warnings.

`Capping`

Type
Block

Description
This block is about capping details. Capping occurs with hydrogen atoms when a bond is broken between an atom inside the region and one outside.

`AllowHighBondOrders`

Type
Bool

Default value
No

Description
Allows capping of interregional aromatic, double and triple bonds. This is normally not a good idea, since the capping is done with hydrogen atoms.

`AtomicInfoForCappingAtom`

Type
String

Default value
ForceField.Type=H_ ForceField.Charge=0.0

Description
The AtomicInfo for the capping atoms. Typically a string like ForceField.Type=X much like forcefield info is entered in the System block for normal atoms.

`CappingElement`

Type
String

Default value
H

Description
The element to be used for capping. The hydrogen atom has the advantage that it is very small.

`CheckCapping`

Type
Bool

Default value
Yes

Description
The same outside atom can be involved in multiple capping coordinate definitions. This is not a good idea, and this will not be accepted by using this check.

`Distance`

Type
Float

Default value
-1.0

Description
A negative value means automatic. In that case the sum of covalent radii is used

`Option`

Type
Multiple Choice

Default value
Fixed

Options
[Fractional, Fixed]

GUI name
Capping option

Description
The capping atom is always along the broken bond vector.

The bond distance between the capping atom and the two atoms are obtained from covalent radii, let us call them D1H and D2H.

With option=Fractional the capping is on the bond vector with the fraction D1H/(D1H+D2H).

With the Fixed option it at the distance D1H from atom 1. A distance of zero always means the coordinate of the inside atom.

`Committee`

Type
Block

Description
Settings for using the hybrid engine as a committee. The factors and region for each engine must be the same. When committee is enabled the standard deviation is also reported as the uncertainty.

`Enabled`

Type
Bool

Default value
No

Description
Enable committee

`Energy`

Type
Block

Description
This block is there to construct the energy.

`DynamicFactors`

Type
Multiple Choice

Default value
Default

Options
[Default, UseLowestEnergy, UseHighestEnergy]

GUI name
Adjust factors

Description
Default - use factors as set in the corresponding Term blocks;

UseLowestEnergy - set all factors to 0 except for that of the engine with the lowest energy, which is set to 1;

UseHighestEnergy - set all factors to 0 except for that of the engine with the highest energy, which is set to 1.

The last two options make sense only for non-QMMM hybrid calculation (that is, if the QMMM block is not present) and only when using engines whose energies can be compared directly.

`Term`

Type
Block

Recurring
True

Description
This block is there to construct the energy term. Can have multiple occurrences

`Charge`

Type
Float

Default value
0.0

Description
Net charge to be used for this energy term.

`EngineID`

Type
String

Description
Identifier for the engine

`Factor`

Type
Float

Default value
1.0

Description

`PBC`

Type
Bool

Default value
Yes

Description
If the full system has a periodic lattice, apply it to the region

`Region`

Type
String

Description
Identifier for the region

`UseCappingAtoms`

Type
Bool

Default value
Yes

Description
Whether to use capping for broken bonds

`Engine`

Type
Block

Recurring
True

Description
The input for the computational (sub) engine. The header of the block determines the type of the engine. An optional second word in the header serves as the EngineID, if not present it defaults to the engine name. Currently it is not allowed to have a Hybrid engine as a sub engine.

`GuessAttributesOnce`

Type
Bool

Default value
Yes

Description
If any ForceField subengines are defined, and if automatic atom typing is possible, then the atom typing is done at the level of the Hybrid engine, and not of the ForceField subengines. This ensures that the same atom types and charges are used in each subsystem, so that pair energy terms that should cancel, will cancel. If set to False, then for each energy term the atom types and charges of the subsystem will be determined separately.

`QMMM`

Type
Block

Description
This block is there to identify the QMMM engines.

`Embedding`

Type
Multiple Choice

Default value
Electrostatic

Options
[Mechanical, Electrostatic]

Description
Determines how the QM region is embedded into the MM region.

Mechanical embedding embedding can also be achieved using the Energy%Terms keywords, but the common case of a two region mechanical QM/MM embedding is easier to set up using this keyword.

`MMCharge`

Type
Float

Default value
0.0

Description
Net charge to be used for the MM region.

`MMEngineID`

Type
String

Description
Identifier for the MM engine

`QMCharge`

Type
Float

Default value
0.0

Description
Net charge to be used for the QM region.

`QMEngineID`

Type
String

Description
Identifier for the QM engine

`QMRegion`

Type
String

Description
Identifier for the QM region. The rest of the system is considered the MM region.

`UseCappingAtoms`

Type
Bool

Default value
Yes

Description
Whether to use capping for broken bonds.

`RestartSubEngines`

Type
Bool

Default value
Yes

Description
Save all the results of the subengines and pass those in a next geometry step or MD step.

`TweakRequestForSubEngines`

Type
Bool

Default value
Yes

Description
Only request what is really needed, gradients and charges.

CHAPTER

SIX

REFERENCES

1. M. Swart, AddRemove: A new link model for use in QM/MM studies. International Journal of Quantum Chemistry 91, 177 (2003) (https://doi.org/10.1002/qua.10463)