

# The Howto of the Periodic Electrostatic Embedded Cluster Method (PEECM).

Asbjörn M. Burow\*

*Institut für Chemie, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099-D*

1<sup>st</sup> November 2011

In this manuscript, the usage of the periodic electrostatic embedded cluster method (PEECM) is presented. Theory was already presented in published work. In the following, I will describe how to perform a PEECM calculation. For this, its basic elements and the work flow are discussed. This includes the required structure models and files as well as the parameter/keyword setup. Finally, I will show how to take advantage of symmetry.

## I. Introduction

The periodic electrostatic embedded cluster method (PEECM) was published in Ref. 1. It is implemented in the TURBOMOLE program suite. This method embeds a quantum mechanical (QM) cluster into a periodic array of point charges (PC). The PEECM adds the contributions from the electrostatic interactions between the QM cluster and the PC to the equations of the calculation. So far, it is implemented within HF and DFT energy and gradient TURBOMOLE modules: `dscf`, `grad`, `ridft`, `rdgrad`, and `escf`. The setup for calculations using the PEECM is briefly discussed in the TURBOMOLE manual (section 6.6 in version 6.3).

## II. Setup of the model

### A. The basic picture

The model consists of two parts. The first part is the periodic array of PC and the second part is the QM cluster. Both are defined in different files (their names can be chosen arbitrarily, but in this manuscript I stick to certain conventional names). The periodic array of PC is defined in the `embed` file whereas the QM cluster is defined in the `coord` file. The `coord` file is the structure file of a regular TURBOMOLE calculation.

First, let's clarify which information is contained in both files from the physical point of view. The `embed` file contains the unit cell and the periodic directions. This sets up a periodic array of PC. If a QM cluster should be inserted into, on top of or close to this array then it is usually necessary to cut out a local region of the periodic array, i.e., creating some cavity or hollow. To do so, the `embed` file can contain a list of PC positions of the array (attention: these positions are not necessarily part of the defined unit cell). PC which are in this list are removed from the periodic array. The atoms of the QM cluster are added to the model. For this, the positions of these atoms are defined in the `coord` file. Each atom defined in the `coord` file list is inserted as QM atom into the model. The entire modelling procedure described in this paragraph is summarized in Fig. 1.

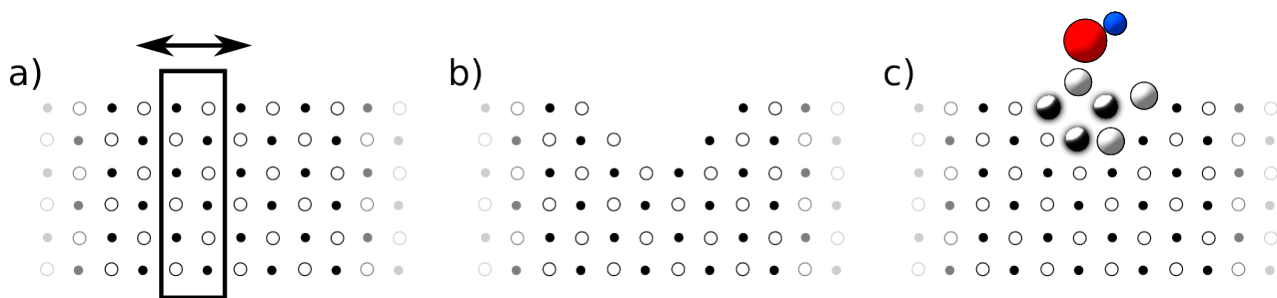
### B. Setup of the `embed` file

The `embed` file contains several sections and keywords. A simple `embed` file is presented in the Example 1. Each `embed` file starts with the `$embed` key. This invokes the PEECM.

This key is followed by three options in the next three lines: `periodic`, `wsicl`, and `lmaxmom`. The option `periodic` is followed by an integer number defining the number of (independent) periodic directions. Hence, the integer has the value 0, 1, 2, or 3. The option `wsicl` is followed by a real number which is used as a parameter for the evaluation of lattice sums of the electrostatic interactions between the QM cluster and the

---

\*email (office until end of 2011): [burowasb@chemie.hu-berlin.de](mailto:burowasb@chemie.hu-berlin.de), email (private): [a.burow@gmx.de](mailto:a.burow@gmx.de)



**Figure 1:** Model setup. a) Define a unit cell and periodic directions for the periodic array of PC (here a slab is created), b) Cut out some PC (remove the PC cluster), c) Insert a QM cluster. This cluster may consist of lattice atoms and adsorbants.

PC. This parameter determines the size of the region of analytical integration (the “near field” as it is called in the underlying multipole method, see Ref. 1). A “safe” value for `wsicl` is 4.0. The option `lmaxmom` is followed by an integer value (safe choice is 30) which is taken as the maximum order of the multipoles used for the multipole expansions of the lattice sums.

The next section of the `embed` file defines the unit cell shape, i.e., the three cell vectors **a**, **b**, and **c**. Starting a line with keyword `cell`, the internal cell parameters are given in the next line as real numbers in order *a*, *b*, *c*,  $\alpha$ ,  $\beta$ ,  $\gamma$ . These parameters are defined as

$$a = |\mathbf{a}|, b = |\mathbf{b}|, c = |\mathbf{c}|, \alpha = \angle(\mathbf{b}, \mathbf{c}), \beta = \angle(\mathbf{a}, \mathbf{c}), \gamma = \angle(\mathbf{a}, \mathbf{b}) \quad . \quad (1)$$

Note that you always have to give all six parameters even if you model systems of lower periodicity. By default the cell dimensions *a*, *b*, *c* are read in Bohr and the angles  $\alpha$ ,  $\beta$ ,  $\gamma$  are read in degree. If *a*, *b*, *c* are provided in Å then write `cell ang` instead of just `cell` in the line above. In the `embed` file, the cell must be defined using these internal cell parameters. The cell corner with fractional coordinates (0,0,0) is always fixed at the origin of a right-handed Cartesian coordinate system, i.e., the starting point of the cell vectors **a**, **b**, and **c** is identical to the Cartesian origin. The vector **a** shows along the positive direction of *x*. Vector **b** is in the *xy* plane while the component of **b** collinear to the *y* axis shows along the positive direction of *y*. The vector **c** forms a right-handed coordinate base together with **a** and **b**. In 3D systems (`periodic 3`) all vectors belong to periodic directions. In 2D systems (`periodic 2`) **a** and **b** belong to the periodic directions. In 1D systems (`periodic 1`) **a** belongs to the periodic direction.

The `cell` section is followed by the unit cell content using the `content` statement. This is followed by a list of PC positions in Cartesian coordinates and Bohr. Each entry in the list starts with a string defining the PC type and is followed by three real numbers. As shown in Example 1 it is convenient to use element names for the types of PC so that it is immediately clear which type of PC should represent which element. With `content ang` and `content frac` the list contains positions in Å and fractional coordinates, respectively. The parser of the TURBOMOLE program must know where the list of PC ends. Therefore, this list is finished with the `end` statement.

Next, there are removed PC from the defined periodic array in the `cluster` section. The `cluster` statement is followed by a list of PC which stops at the next `end` statement. The entries in this list are by default in Cartesian coordinates and in Bohr. Again, the `cluster` statement can be changed to `cluster ang` and `cluster frac`. Since the PC specified in the list should be removed from the periodic array, their positions must fit rather accurate to the corresponding positions defined for the periodic array. The list of removed PC is also called PC cluster.

Finally, the values of the PC are defined in the `charges` section. In this section, there must be an entry for each type of PC contained in the unit cell. This entry contains the string for the PC type followed by a real number representing the value of the PC in atomic units and including the charge sign. The list of entries is terminated by the `end` statement. The charges defined in the `charges` section must lead to a charge neutral unit cell. They need not yield a charge neutral PC cluster. To close the `$embed` section write the `$end` keyword at the end of the `embed` file.

### Example 1: The embed file for a bulk MgO model

```
$embed
periodic 3
wsicl 4.0
lmaxmom 30
cell ang
4.2 4.2 4.2 90.0 90.0 90.0
content frac
Mg 0.0000000000000000 0.0000000000000000 0.0000000000000000
O 0.5000000000000000 0.0000000000000000 0.0000000000000000
O 0.0000000000000000 0.5000000000000000 0.0000000000000000
O 0.0000000000000000 0.0000000000000000 0.5000000000000000
Mg 0.5000000000000000 0.5000000000000000 0.0000000000000000
Mg 0.5000000000000000 0.0000000000000000 0.5000000000000000
Mg 0.0000000000000000 0.5000000000000000 0.5000000000000000
O 0.5000000000000000 0.5000000000000000 0.5000000000000000
end
cluster frac
O 0.5000000000000000 0.0000000000000000 0.0000000000000000
Mg 0.0000000000000000 0.0000000000000000 0.0000000000000000
Mg 1.0000000000000000 0.0000000000000000 0.0000000000000000
Mg 0.5000000000000000 0.5000000000000000 0.0000000000000000
Mg 0.5000000000000000 -0.5000000000000000 0.0000000000000000
Mg 0.5000000000000000 0.0000000000000000 0.5000000000000000
Mg 0.5000000000000000 0.0000000000000000 -0.5000000000000000
end
charges
Mg 2.0
O -2.0
end
$end
```

### C. Setup of the coord file

The `coord` file is the standard structure file in a TURBOMOLE calculation. Hence, it is also used for molecular calculations. The `coord` file can contain several sections and keywords. A concise description of the `coord`

file is available in the TURBOMOLE manual. This Howto focus on the part of the `coord` file which is necessary to set up the QM cluster. The `coord` file starts with the `$coord` keyword. The next lines define the atoms of the QM cluster. Each line contains three real numbers as Cartesian coordinates in Bohr and a string for the element name of the atom. It may contain an `f` label in the end. This fixes the atom in structure optimizations. The list of atoms is terminated by the `$end` keyword. A small example which belongs to the `embed` file of Example 1 is presented in Example 2.

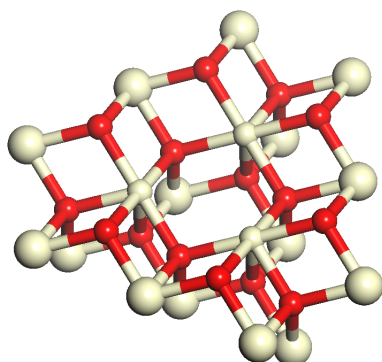
### Example 2: The `coord` file for a QM cluster

```
$coord
 3.96842486  0.0000000000000000  0.0000000000000000  o
 0.00000000  0.0000000000000000  0.0000000000000000  mg f
 7.93684972  0.0000000000000000  0.0000000000000000  mg f
 3.96842486  3.9684248600000000  0.0000000000000000  mg f
 3.96842486 -3.9684248600000000  0.0000000000000000  mg f
 3.96842486  0.0000000000000000  3.9684248600000000  mg f
 3.96842486  0.0000000000000000 -3.9684248600000000  mg f
$end
```

### III. Setup of the `control` file

If a program of the TURBOMOLE suite is invoked then this program usually collects the required input data by reading the `control` file. Thus, all input data should be accessible from the `control` file. Please read the TURBOMOLE manual how to set up the `control` file for the electronic structure calculation you want to perform. Standard procedures for this are available through scripts, e.g., `define`. With this, the basic `control` file can be created interactively. A `control` file created in this way contains already a link to the external `coord` file. The link looks like this: `$coord file=coord`. Therefore, the structure of the QM cluster is accessible by reading the `control` file. A similar link must be set manually for the `embed` file: `$embed file=embed`. The keyword `$embed` will start the PEECM during the calculation.

The `control` file is the place to assign basis functions, effective core potentials (ECP), and (if needed) modified nuclear charges to the atoms of the QM cluster. The syntax of these assignments is shown in Example 3. In the QM cluster, the electron density of the atoms is represented by a linear combination of basis function products (see quantum chemistry textbooks). Thus, the choice of basis set quality is a basic component in the electronic structure calculation. However, using PEECM the electron density may polarize at the border of the QM cluster due to the PC in the periodic array yielding artefacts in the description of chemical or physical properties of the model. Since the electron density is negatively charged, it is pushed away from the negative PC. Hence, the density should remain in spatial regions where the electrostatic field of the negative PC represents a sufficient approximation for the electrostatic field of anions. This is not the case for positive PC representing cations. Therefore, the atoms at the border of the QM cluster should not carry basis functions. This is achieved by replacement of positively charged border atoms by all electron ECP and treating negatively charged border atoms as PC counted to the periodic array.



**Figure 2:**  $\text{Ce}_{18}\text{O}_{15}^{40+}$  QM cluster. This cluster is defined in a `coord` file. According to the `control` setup given in Example 3 the white and red spheres represent Ce and O atoms, respectively. The larger spheres at the border of the QM cluster are treated with all electron ECP. This cluster is used to model an O vacancy on top a  $\text{CeO}_2(111)$  surface.

**Example 3:** Assignment of basis functions, ECP, and modified nuclear charges in the `control` file

The following `$atoms` setup is used for a  $\text{Ce}_{18}\text{O}_{15}^{40+}$  QM cluster (see Fig. 2). This QM cluster is used to model an O vacancy on top a  $\text{CeO}_2(111)$  surface. Basis functions are assigned to the cluster kernel  $\text{Ce}_3\text{O}_{15}^{18-}$ . The remaining Ce ions (19 to 33) are border cations of the QM cluster and thus treated with all electron ECP (each ECP replaces 54 electrons). The atom indices which follow the element labels correspond to the order of the atoms in the `coord` file.

```
$control
[...]
$atoms
ce 1-3 \
    basis =ce ecp-46-mwb \
    ecp =ce ecp-46-mwb
o 4-18 \
    basis =o def-SVP
ce 19-28 \
    basis =none \
    ecp =ce crenbs-ecp \
    charge= 57.80000000
ce 29-33 \
    basis =none \
    ecp =ce crenbs-ecp
$closed shells
a 1-87 ( 2 ) #i.e., the QM cluster contains 174 electrons
[...]
$end
```

For two reasons, it can be necessary to modify the nuclear charges of atoms in the QM cluster. First, in an

extended structure, a single reduced or oxidized ion belongs to a local defect, but is quite far away from the “principle” defect region. Since this ion is not periodically repeated, it must belong formally to the QM cluster. As part of the QM cluster we define it as “naked” PC (without basis functions) whose value is the modified nuclear charge. Second, there are no all electron ECP available for an element in some oxidation state but for another similar oxidation state. A very prominent example for this case are the reduced  $Ce^{z+}$  cations ( $z < 4$ ) described by  $Ce^{4+}$  ECP together with a change of nuclear charge (see Examples 3 and 4).

Whatever is modelled with the PEECM, the entire system consisting of the periodic PC array and the QM cluster must be charge neutral. This is determined by the PC removed from the periodic array, the QM cluster with its nuclear charges and ECP, and by the number of electrons which “are filled into” the QM cluster. The number of electrons is given in the `control` file (see Example 3). Please note that the interactive `define` script usually calculates wrong electron numbers if nuclear charges have been modified. This has to be corrected manually. The Example 4 shows how to check charge neutrality.

#### Example 4: How to check charge neutrality

The  $Ce_{18}O_{15}^{40+}$  QM cluster of Example 3 and Fig. 2 is considered. For the charge neutrality check of the entire system, the values of the PC in the periodic array and the removed PC cluster have to be known. They are defined in the `embed` file. In this example the periodic array for a  $CeO_2(111)$  surface consists of PC for  $Ce^{4+}$  and  $O^{2-}$  ions. The PC cluster removed from the periodic array is  $Ce_{18}O_{16}^{40+}$ , i.e.,  $18 \times Ce^{4+} + 16 \times O^{2-}$ . The  $Ce_{18}O_{15}^{40+}$  QM cluster is inserted. Please note that the QM cluster has one O atom less than the PC cluster. The missing O atom in the top layer of the surface creates a surface O vacancy (see Fig. 2). We will check now if the setup in Example 3 is done correctly to yield the  $Ce_{18}O_{15}^{40+}$  QM cluster and charge neutrality of the entire system. The check performed in the following can be applied vice versa to use the charge neutrality for the determination of the required setup.

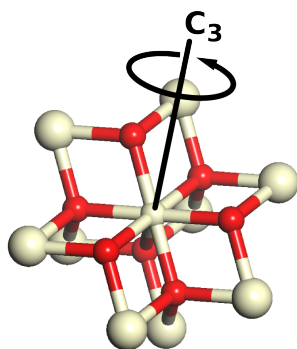
The charge of the cluster kernel  $Ce_3O_{15}^{18-}$  is calculated in two ways:

cluster kernel	$3 \times Ce^{4+}$	$15 \times O^{2-}$	sum
ion charges	$3 \times +4$	$15 \times -2$	-18
nuclear charges + ECP compensation	$3 \times (+58 - 46)$	$15 \times +8$	+156
elec. charges - ECP compensation	$-3 \times (54 - 46)$	$-15 \times 10$	-174

You may have noticed that the oxidation state of O atoms is  $-2$  (i.e.,  $O^{2-}$ ) and that the removal of the neutral O atom must leave two electrons in the O vacancy model. However, within the cluster kernel the oxidation states of all atoms are the same like in the periodic model ( $Ce^{4+}$  and  $O^{2-}$ ). This means, we assume here that the reduced sites are not within the kernel. Indeed the reduced sites are modelled in the all electron ECP part as shown in the next table:

all electron ECP part	$10 \times Ce^{3.8+}$	$5 \times Ce^{4+}$	sum
ion charges	$10 \times +3.8$	$5 \times +4$	+58
nuclear charges	$10 \times +57.8$	$5 \times +58$	+868
ECP compensation	$10 \times -54$	$5 \times -54$	-810

All electron ECP of  $Ce^{4+}$  are used for all Ce sites in the all electron ECP part. For the reduced  $Ce^{3.8+}$  sites, the nuclear charges are modified. The total charge of the QM cluster is  $-18 + 58 = +40$  which equals the charge of the removed PC cluster.



**Figure 3:**  $\text{Ce}_{10}\text{O}_7^{26+}$  QM cluster of Example 5. The white and red spheres represent Ce and O atoms, respectively. The larger spheres at the border of the QM cluster are treated with all electron ECP. The  $C_3$  axis goes through the central Ce atom. This QM cluster is used to model the orbital states of a  $\text{Ce}^{4+}$  ion on a  $\text{CeO}_2(111)$  surface.<sup>2</sup>

#### IV. Running the calculation

Start the electronic structure calculation as usual, e.g., with `ridft`. Information about the PEECM is written to the output file of calculation. This means you should check if the PEECM section appears in the output file and if the data look reasonable. You can also use the `jobex` script for structure optimizations. In this case the PEECM output section should appear in each output file of the different programs invoked, e.g., `ridft`, `rdgrad`.

#### V. Using symmetry

If the model (periodic PC array plus QM cluster) has several symmetry elements, it is possible to use the symmetry. However, since molecular QM methods are applied, only the point symmetry operations can be used. The point symmetry is used for the QM part of the calculation. For this, the symmetry plane/axis/point of the QM cluster must reside in the origin of the Cartesian coordinate system. Note that you have to define the unit cell and PC cluster such that the symmetry plane /axis/point has the correct position. By default TURBOMOLE uses symmetry operations created by standard orientations of symmetry generating elements. Therefore, never apply option `sy` and `desy` in the `define` script to set the symmetry of your model. `sy` and `desy` rotate the QM cluster back to standard orientations. Then the QM cluster fits no longer to the periodic part of the model.

Instead you have to set the symmetry manually in the `control` file as shown in Example 5. First, run the `define` and read the coordinates of the QM cluster. Then exit the `define` without any further settings (i.e., press \* and type “no” until you can quit `define`). This creates the basic `control` file. Second, define the symmetry point group and the generator orientations in this `control` file. The Table I shows the generators for all point groups of TURBOMOLE. Third, rerun `define`. The `define` procedure should immediately recognize the symmetry of the system. **Attention: For DFT calculations the grids for the numerical integration of the exchange-correlation term are not adapted to the redefined generator orientations.** This results in inaccuracies for the groups  $C_3$ ,  $C_{3v}$ ,  $C_{3h}$ ,  $C_6$ ,  $C_{6v}$ ,  $C_{6h}$ ,  $S_6$ ,  $D_6$ ,  $D_{6h}$ ,  $D_{6d}$ ,  $S_3$ ,  $D_{3d}$ ,  $D_3$ ,  $D_{3h}$ , if the orientations of the  $C_3$  and  $C_6$  axes are changed from default.

### Example 5: $C_{3v}$ treatment for the $CeO_2(111)$ surface

Lets use the  $C_{3v}$  symmetry for a Ce-centered  $Ce_{10}O_{7^{26+}}$  QM cluster embedded in the (111) surface of  $CeO_2$  (see Fig. 3). The generators of the  $C_{3v}$  symmetry are found in Table I. These are the  $C_3$  axis and a mirror plane  $\sigma$ . Hence, the translation constraint for the QM cluster is such that the  $C_3$  axis goes through the central Ce atom and the origin (see Fig. 3). The orientation can be defined by the user. Lets put the QM cluster such that its  $C_3$  axis is in the  $z$  axis (this is TURBOMOLE default and should not be changed in DFT calculations) and one mirror plane is in the  $yz$  plane. Then the coordinates of this QM cluster could be (the central Ce atom comes first),

```
$coord
  0.0000000000000  0.0000000000000  1.4892697159500  ce
  3.6479716300000  2.1061574025800  0.0000000000000  o
 -3.6479716300000  2.1061574025800  0.0000000000000  o
  0.0000000000000 -4.2123148051500  0.0000000000000  o
  0.0000000000000  4.2123148051500  2.9785394318900  o
  3.6479716300000 -2.1061574025800  2.9785394318900  o
 -3.6479716300000 -2.1061574025800  2.9785394318900  o
  0.0000000000000  0.0000000000000  5.9570788637900  o
  7.2959432600000  0.0000000000000  1.4892697159500  ce
  3.6479716300000  6.3184722077300  1.4892697159500  ce
 -3.6479716300000  6.3184722077300  1.4892697159500  ce
 -7.2959432600000  0.0000000000000  1.4892697159500  ce
  3.6479716300000 -6.3184722077300  1.4892697159500  ce
 -3.6479716300000 -6.3184722077300  1.4892697159500  ce
  0.0000000000000  4.2123148051500  7.4463485797300  ce
  3.6479716300000 -2.1061574025800  7.4463485797300  ce
 -3.6479716300000 -2.1061574025800  7.4463485797300  ce
$end
```

This is embedded in the (111) surface of a slab cell. Lets have a look at the corresponding embed file,

```
$embed
periodic 2
wsicl 4.0
lmaxmom 30
cell
  7.29594326 7.29594326 16.79943278 90.0000 90.0000 120.0000
content frac
O 0.66666667 0.33333333 0.00000
Ce 0.00000000 0.00000000 0.08865
O 0.33333333 0.66666667 0.17730
O 0.00000000 0.00000000 0.35460
Ce 0.33333333 0.66666667 0.44325
O 0.66666667 0.33333333 0.53190
O 0.33333333 0.66666667 0.70920
Ce 0.66666667 0.33333333 0.79784
O 0.00000000 0.00000000 0.88649
end
```



```

cluster frac
  Ce   0.00000000   0.00000000   0.08865
  O    0.66666667   0.33333333   0.00000
  O   -0.33333333   0.33333333   0.00000
  O   -0.33333333  -0.66666667   0.00000
  O    0.33333333   0.66666667   0.17730
  O    0.33333333  -0.33333333   0.17730
  O   -0.66666667  -0.33333333   0.17730
  O    0.00000000   0.00000000   0.35460
  Ce   1.00000000   0.00000000   0.08865
  Ce   1.00000000   1.00000000   0.08865
  Ce   0.00000000   1.00000000   0.08865
  Ce  -1.00000000   0.00000000   0.08865
  Ce   0.00000000  -1.00000000   0.08865
  Ce  -1.00000000  -1.00000000   0.08865
  Ce   0.33333333   0.66666667   0.44325
  Ce   0.33333333  -0.33333333   0.44325
  Ce  -0.66666667  -0.33333333   0.44325
end
charges
  Ce      4.0
  O      -2.0
end
$end

```

Invoking `define` with just the `coord` and the `embed` files in the directory, the coordinates can be read with command `a coord` in the first menu of `define`. Leaving `define` without any further setup leads to a basic `control` file in the current directory. Open this file with the editor of your choice and search the `$symmetry` keyword. After this keyword, there is written the symbol of the symmetry point group which is currently used for your model. This should be `c1`. Change this to `c3v`. Finally, the whole line reads

```
$symmetry c3v
```

The orientation of generators is defined with the keyword `$generators`. You can write this in any new line of the `control` file, but it is convenient to insert it just below the `$symmetry` keyword. The syntax to define the orientation of generators is as follows. For each generator of the point group defined with `$symmetry`, three decimal numbers must be given in one line separated by (one or more) spaces. This defines a direction vector for this generator (the three numbers need not yield a normalized vector). If the generator is an axis then the direction vector defines the direction along this axis. If the generator is a plane then the direction vector is perpendicular to the plane. The  $C_{3v}$  point group has 2 generators. First generator is a  $C_3$  axis and second generator is a  $\sigma$  plane (note that the order is important). Thus, in compliance with the orientation given in the beginning of this Example we have,

```

$symmetry c3v
$generators
  0. 0. 1. # First generator: C3 axis in z
  1. 0. 0. # Second generator: sigma plane in yz

```

Just adapt the `control` as shown, save, and leave the editor. Then rerun `define`. This prints the following information to the standard output,

**Table I:** Generators for symmetry point groups. For each point group, the given order of generators corresponds to the order required in the \$generators section of the control file.

Point Group	Generators	Point Group	Generators
$E \equiv C_1$	$C_1$	$D_{nh}$	$C_n, C_2, \sigma$
$C_i \equiv S_2$	$S_2$	$T$	$C_2, C_3$
$C_s \equiv C_{1h}$	$C_1, \sigma$	$T_d$	$S_4, C_3$
$C_n$	$C_n$	$T_h$	$C_2, C_3, \sigma$
$S_n$	$S_n$	$O$	$C_4, C_3$
$C_{nv}$	$C_n, \sigma$	$O_h$	$C_4, C_3, \sigma$
$D_n$	$C_n, C_2$	$I$	$C_5, C_3$
$D_{nd}$	$S_{2n}, C_2$	$I_h$	$C_5, C_3, \sigma$

```
FILE control ALREADY EXISTS
I WILL PLUG IN THE NEW DATA.
```

```
DATA WILL BE TAKEN FROM control BY DEFAULT
```

```
INPUT TITLE
```

```
*** USER DEFINED GENERATORS PROVIDED ***
```

```
Note: for DFT calculations do not change
orientations of C3 and C6 axes for the
following groups: C3, C3v, C3h, C6, C6v,
C6h, S6, D6, D6h, D6d, S3, D3d, D3, D3h,
as it will result in inaccurate results
```

```
Following generators have been provided:
```

```
c3, direction      0.00   0.00   1.00
mirror plane sigma, direction      1.00   0.00   0.00
```

```
SYMMETRY c3v AND CARTESIAN COORDINATES FOR 17 ATOMS
HAVE BEEN READ FROM THE DEFAULT INPUT FILE control .
```

```
keyword $intdef missing in file <control>
```

```
DEFINITIONS OF INTERNAL COORDINATES HAVE NOT BEEN READ.
SPECIFICATION OF BOND TOPOLOGY HAS BEEN READ.
```

```
DO YOU WANT TO CHANGE THE GEOMETRY DATA ? DEFAULT=n GOBACK=&
```

At this point, the setup procedure can be continued as usual.

## References

- [1] A. M. Burow, M. Sierka, J. Döbler, and J. Sauer, *J. Chem. Phys.* **130**, 174710 (2009).
- [2] J.-F. Jerratsch, X. Shao, N. Nilius, H.-J. Freund, C. Popa, M. V. Ganduglia-Pirovano, A. M. Burow, and J. Sauer, *Phys. Rev. Lett.* **106**, 246801 (2011).