

1 Scripts for Processing CPMD Output

This appendix provides descriptions of useful scripts I have written for processing CPMD output. They can be downloaded from <http://atrey.karlin.mff.cuni.cz/~verunka/CPMD>. The scripts can be compiled by typing

```
cc -lm filename.c -o filename
```

where `filename` is the name of the script (the same as in the following list) and `-lm` is an optional argument which links mathematical libraries when required, if they are not already linked automatically by the compiler. To get online help, run the script without any parameters.

cpmd2car prints out atomic coordinates from a geometry optimisation in cartesian coordinates (*file.car*). *Car* files are understood by Cerius2, Insight and WebLab Viewer Pro. The script reads in a CPMD output file, searches for the phrase 'FINAL RESULTS' and prints out the atomic coordinates, therefore it is possible to extract any structure produced during a geometry optimisation simply by adding a line 'FINAL RESULTS' just before the section with coordinates of the particular optimisation step. To use, type

```
cpmd2car cpmd_output_file > file.car
```

cpmd2xtl prints out atomic coordinates from a geometry optimisation in fractional coordinates (*file.xtl*). *Xtl* files are understood by Cerius2 and Insight, but not by WebLab Viewer Pro. The script is an update for CPMD 3.4.1 of a script written by Christian Tuma for CPMD 3.0. It reads in a CPMD output file, searches for the phrase 'FINAL RESULTS' and prints out the fractional coordinates, therefore it is possible to extract any structure produced during a geometry optimisation simply by adding a line 'FINAL RESULTS' just before the section with coordinates of the particular optimisation step. To use, type

```
cpmd2xtl cpmd_output_file > file.xtl
```

distance monitors changes in the distance between any two different atoms in the system during a geometry optimisation run. To use, type

```
distance cpmd_output_file atom_no_1 atom_no_2
```

where the numbers of atoms, `atom_no_1` and `atom_no_2` correspond to numbering in the CPMD input/output. Note that the script does not check whether the system really contains sufficient number of atoms, so that a request for atoms 1 and $N+1$ in a system of N atoms will produce *some* number, but this number will not have any meaning.

angle monitors changes in the angle between any three different atoms in the system during a geometry optimisation run. To use, type

```
angle cpmd_output_file atom_no_1 atom_no_2 atom_no_3
```

where the numbers of atoms, `atom_no_1`, `atom_no_2` and `atom_no_3` correspond to numbering in the CPMD input/output. Note that the script does not check whether the system really contains sufficient number of atoms, so that a request for atoms 1, N and $N+1$ in a system of N atoms will produce *some* number, but this number will not have any meaning.

cpmd2mkl formats the output of vibrational analysis by finite differences into a Molekel 4.2 (Flükiger et al., 2002, Portmann and Lüthi, 2002) format. To use, place the CPMD output file and the corresponding VIBEIGVEC file in the same directory and type

```
cpmd2mkl cpmd_output_file > file.mkl
```

The Molekel file can be edited by hand, so that the numbers of modes are replaced by symmetry of the modes. A problem can occur if the file is produced under UNIX/Linux and read into Molekel running under Windows: owing to the incompatibilities of UNIX and Windows standards for line break characters, it may be necessary to open and save the *.mkl* file (without any changes) in a Windows text editor (WordPad will work fine), so that the line breaks are set correctly according to the Windows standard.

dos constructs total density of states. This script is based on *dosscript* and *dos98.f* written by Schwarz and Wittenberg for the *phi98* code. To use, type

```
dos cpmd_output_file function width > file.tab
```

where `function` is the broadening function and possible types are `l` for Lorentz function and `g` for Gauss function. The `width` parameter specifies width of the broadening function in electronvolts. If it is omitted, default values are used, 0.1 eV for Gaussian and 0.05 eV for Lorentz function. The output file consists of two columns, energy in eV and total density of states in arbitrary units.

bandstructure extracts information about energy levels from CPMD output. The output file contains two columns: the k -point number and the energies of the bands (in eV). To use, type in one line:

```
bandstructure cpmd_output_file x_axis_shift Fermi_level_shift  
no_skipped_states > file.tab
```

The `x_axis_shift` is introduced so that results of calculations on different k -point paths can be appended to one file. For example, when having three files of bandstructure calculation on the same system, each of them with a different k -point path, `bands_GX.out`, `bands_XY.out` and `bands_YG.out`, each with 50 k -points along the path, one can type

```
bandstructure bands_GX.out 0 0 0 > bands.tab
bandstructure bands_XY.out 51 0 0 >> bands.tab
bandstructure bands_YG.out 101 0 0 >> bands.tab
```

The file `bands.tab` will then contain the bandstructure of the system between the Γ and X, X and Y and Y and Γ points of the reciprocal cell, which can be plotted as the standard bandstructure graph.

`Fermi_level_shift` specifies possible shift along the y axis. It can either be set "manually" so that the highest occupied state touches the zero level of the energy or, after a minor modification of the CPMD code, the shift can be read from the CPMD output: in file *wrener.F* from the CPMD 3.5.3 source add a line

```
INCLUDE 'simul.inc'
```

at the beginning of subroutine WREIGEN and lines

```
WRITE(6, '(1X,A,T43,F20.10,A3)')
&      'VPLOC REF = ',VPLOC*(2*RY), ' EV'
```

at the end of subroutine WREIGEN, after the command which prints out the chemical potential. These changes will cause a part of the potential, VP_{loc} , to be printed out, which shifts the Fermi level to zero. (Thanks to Justin Toh for this modification.)

`no_skipped_states` specifies number of low lying states to be skipped in the output. This is particularly useful with the semicore pseudopotentials, which produce a number of very low lying states, which are usually of no interest when plotting the bandstructure.

Defaults for `x_axis_shift`, `Fermi_level_shift` and `no_skipped_states` are zero. Note that to specify `Fermi_level_shift` and/or `no_skipped_states`, one must also specify the preceding arguments, even if they were zero, because the script assumes that the first argument after `cpmd_output_file` is the `x_axis_shift`, the second is the `Fermi_level_shift`, etc.

vib2math reads in the file VIBEIGVEC produced by CPMD in a vibrational analysis run and rewrites the normal vectors in a format which can be read into

Mathematica. Each normal mode vector is labeled fn , n being an integer number running from one to the number of modes, in order in which the vectors are written in the VIBEIGVEC file. To use, place the file VIBEIGVEC in the working directory and type

```
vib2math no_normal_modes > file.tab
```

where `no_normal_modes` is the number of normal modes in the file VIBEIGVEC (it is $6 \times$ number of atoms in the calculation, because also the acoustic branches are taken into account) and `file.tab` is the file which will be read by Mathematica by using

```
<< "file.tab";
```

in the Mathematica Notebook.

References

Flükiger, P., Lüthi, H. P., Portmann, S., and Weber, J. (2000–2002). *MOLEKEL 4.2*.

Portmann, S. and Lüthi, H. P. (2002). MOLEKEL: An interactive molecular graphics tool. *Chimia*, 54:766–770.