Bader analysis with SIESTA : How to Use

Dr Mohan L Verma

Computaional Nanoionics Research Lab, Deparment of Applied Physics,

FETSSGI, Shri Shankaracharya Technical Campus Junwani

Bhilai (Chhattisgarh) INDIA 490020

15 Jan 2015

In siesta-3.2 mannual, regarding *bader* implementation is given in details in page-72-73. So before starting *bader* analysis using siesta go though this.

For *bader* analysis following steps can be followed :

1. As per given in siesta-3.2 manual, Include one more flag in siesta input *.fdf file :

SaveBaderCharge .true.

During siesta run this will instruct the program to save charge density in form of *.BADER file in the working directory for further analysis of *bader* using bader analysis program.

2. Now use *grid2cube* utility from ~/siesta-3.2/Util/Grid/grid2cube (this binary file can be generated simply using "make" command in this directory) to convert this *.BADER into *.CUBE format which is an acceptable format for bader-analysis-program.

In order to use *grid2cube* use the instructions given in the top of *grid2cube.f* i.e.

The program grid2cube needs three input files:

(i) Main input text file (i.e. Input.bader), read by standard input. A sample of input file is:

```
C   --- begin input file ---
     h2o            #  The   label of the system, as in SIESTA SystemLabel
     bader          # the task viz  rho, toch, bader drho, ldos, vh or vt (in lowercase!!).
    4.0 6.0 5.0   #  a shift of the origin of coordinates (in bohr).
     2              an integer (nskip) that specifies the density of  grid points in the output. F
     unformatted
C   --- end input file ---
```

(ii) *SystemLabel.XV* file: this is a file generated by SIESTA i.e. in example above: *h2o.XV*. You should

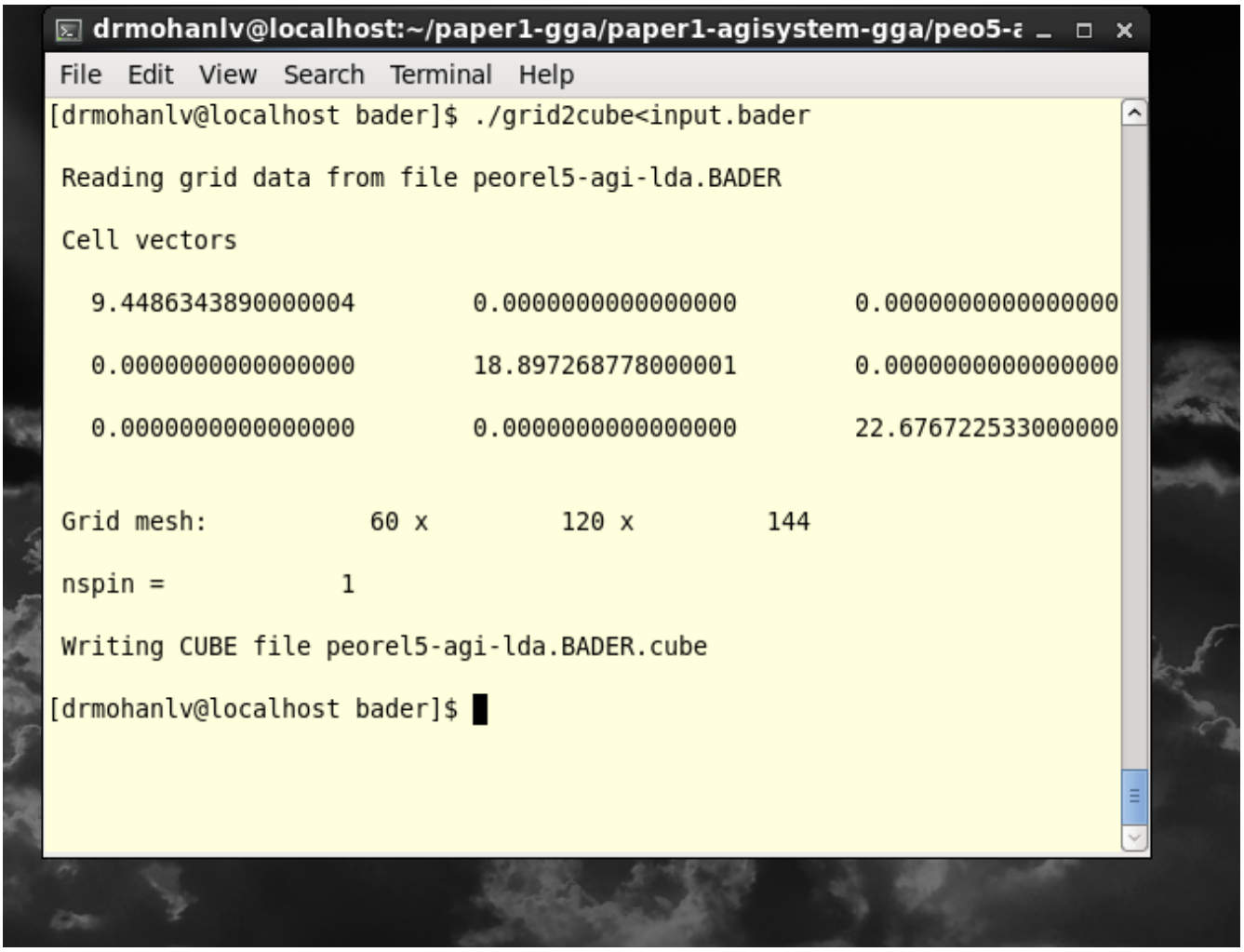
copy it from the directory with your SIESTA output files.

(iii) *SystemLabel.TASK* file: this is a file generated by SIESTA, with the values of the appropriate quantity on the grid. In example above: h2o.BADER. You should copy it from the directory with your SIESTA output files.

Now use a single command :

$./grid2cube<input.bader

This will generate *.CUBE file for further use.



3. Now get Graeme's BADER code from their website :

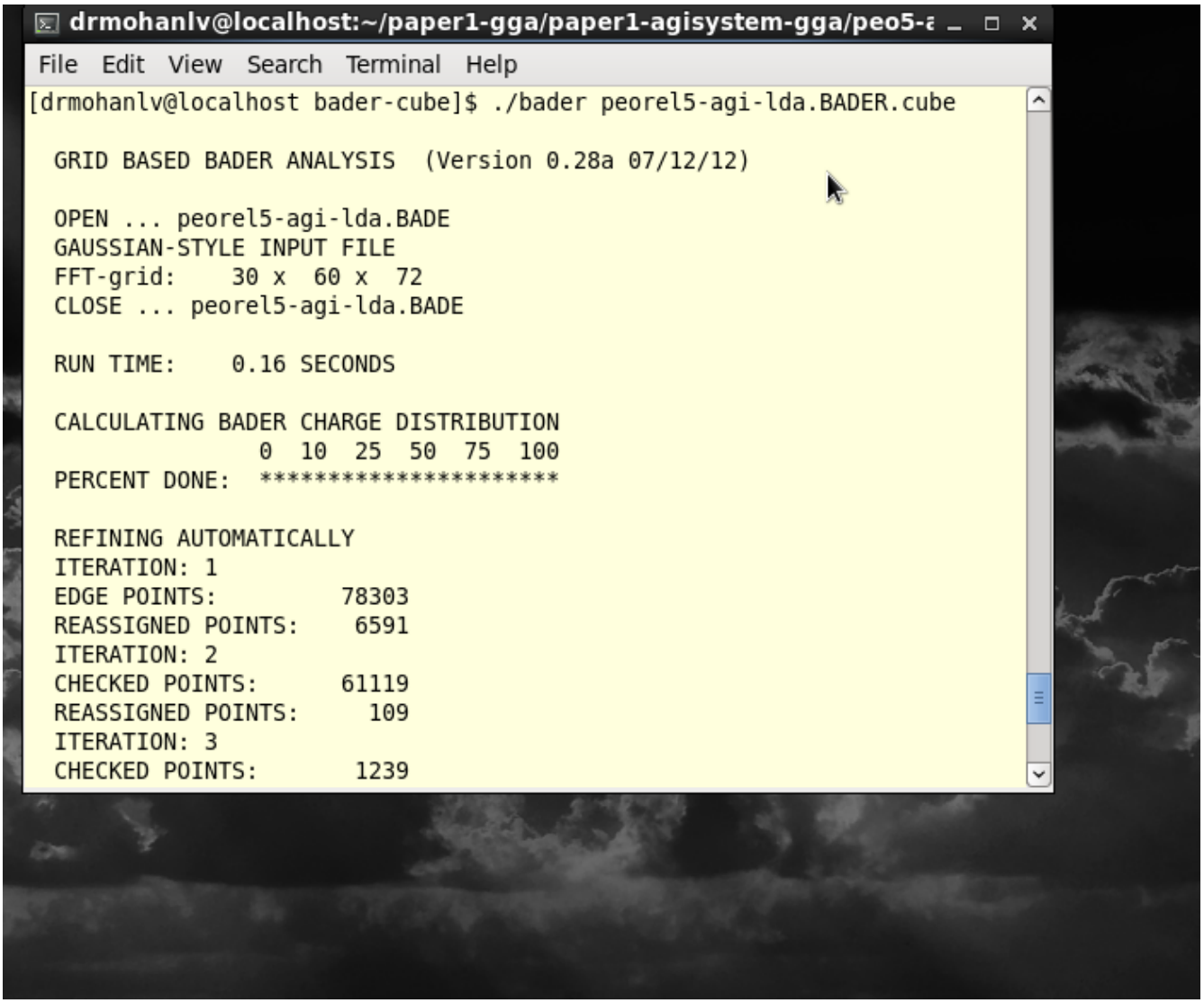
http://theory.cm.utexas.edu/henkelman/code/bader/

and compile by using necessary FORTRAN compiler and flags , this will generate "bader" binary file for further use.

The bader- program can be run with the command

./bader *.CUBE

which will generate following output files: ACF.dat, BCF.dat, AtomVolumes.dat.



ACF.dat contains the coordinates of each atom, the charge associated with it according to Bader partitioning, percentage of the whole according to Bader partitioning and the minimum distance to the surface. This distance should be compared to maximum cut-off radius for the core region if pseudo potentials have been used.

BCF.dat contains the coordinates of each Bader maxima, the charge within that volume, the nearest atom and the distance to that atom.

AtomVolumes.dat contains the number of each volume that has been assigned to each atom. These

numbers correspond to the number of the BvAtxxxx.dat files.

The following options can be used when running the Bader analysis program.

```
bader [ -c bader | voronoi ]
      [ -n bader | voronoi ]
      [ -b neargrid | ongrid ]
      [ -r refine_edge_method ]
      [ -ref reference_charge ]
      [ -vac off | auto | vacuum_density ]
      [ -p all_atom | all_bader ]
      [ -p sel_atom | sel_bader ] [volume list or range ]
      [ -p sum_atom | sum_bader ] [ volume list or range ]
      [ -p atom_index | bader_index ]
      [ -i cube | chgcar ]
      [ -h ] [ -v ]
      chargefile
```

The Bader volumes can be written using the print options.

```
bader [ -p all_atom | all_bader ] chargefile
  bader [ -p sel_atom | sel_bader ] [ volume list or range ] chargefile
  bader [ -p sum_atom | sum_bader ] [ volume list or range ] chargefile
bader [ -p atom_index | bader_index ] chargefile
```

-p none The default is to write no charge density files.

-p all_atom Combine all volumes associated with an atom and write to file. This is done for all atoms and written to files named BvAtxxxx.dat. The volumes associated with atoms are those for which the maximum in charge density within the volume is closest to the atom.

-p all_bader Write all Bader volumes (containing charge above threshold of 0.0001) to a file. The charge distribution in each volume is written to a separate file, named Bvolxxxx.dat. It will either be of a CHGCAR format or a CUBE file format, depending on the format of the initial charge density file. These files can be quite large, so this option should be used with caution.

-p sel_atom Write the selected atomic volumes, read from the subsequent list or range of volumes.

-p sel_bader Write the selected Bader volumes, read from the subsequent list or range of volumes.

-p sum_atom Write the sum of selected atomic volumes, read from the subsequent list of volumes.

-p sum_bader Write the sum of selected Bader volumes, read from the subsequent list of volumes.

-p atom_index Write the atomic volume index to a charge density file.

-p bader_index Write the Bader volume index to a charge density file.

For general information about Bader analysis see :

https://notendur.hi.is//~egillsk/stuff/annad/egillsk_bader_pres_150206.pdf

Some ifs and buts regarding bader analysis is mentioned in siesta3.2 manual and a blog

http://voznyy.elinity.com/blog/2008/01/bader-analysis-with-siesta/

So one should follow the conditions mentioned there in.

All the best.

Please give the feedback in drmohanlv@gmail.com

*****