Help for the AIM utility of the ABINIT package

The AIM utility allows to analyse charge densities produced by the ABINIT code. The AIM analysis (Atom-In-Molecule) has been proposed by Bader. Thanks to topological properties of the charge density, the space is partitioned in non-overlapping regions, each containing a nucleus. The charge density of each region is attributed to the corresponding nucleus, hence the concept of Atom-In-Molecule. References to the relevant litterature are to be provided.

The aim utility is generated in the same way as other ABINIT utilities (i.e. make allseq).

To run the program one needs to prepare two files:

files-file file which contains the name of the input files and the root for the names of the different output files.

input-file file which gives the values of the input variables

Except this file you need the output file of ABINIT with the valence density in real space ($*\_DEN$) and the core density files ($*.fc$) sorting from the program for generating pseudopotentials FHI. LDA core density files have been generated for the whole periodic table, and are available on the ABINIT web site. Since the densities are weakly dependent on the choice of the XC functional, and moreover, charge density analysis is mostly a qualitative tool, these files can be used for other functionals. Still, if really accurate values for the Bader charge analysis are needed, one should generate core density files with the same XC functional as for the valence density.

Usually, the grid in the real space for the valence density should be taken more fine than the one proposed by ABINIT. (For example for the lattice parameter 7-8 a.u. , ngfft at least 64 gives the precision of the Bader charge estimated to be better than 0.003 electrons). To run the program type:

$$aim \; << files.file >$$

The structure of the **files-file** is as follows:

*.in	# input-file
.den	# valence density (output of ABINIT)
$< root >$	# the root of the names of different output files
at1.fc	# core density files (in the same order as
...	# in the ABINIT files-file)
atN.fc	

The example of the files-file is shown in the file *./example/mgo.files.*

The **input-file** is made in the same manner as the input file of ABINIT - it suffices to give the name of the input variable following by the value(s) separated by one or more white character(s). The names are not case sensitive. All input variables have the meaningful default value, however, you have to specify what you want to calculate (default = nothing). The example of the simple input file for Oxygen in bulk MgO is in *./example/mgo.in.* There are also the corresponding output files in this directory.

The output files

The atomic units and cartesian coordinates are used for all output parameters. The names of the output files are of the form $< root > .suffixe$. There are three main output files:

***.crit** - the file with the CPs - they are listing in the order BCPs, RCPs and CCPs. The line of the output contains these informations:

$$position \quad \frac{eigen\ values}{of\ Hessian} \quad \frac{index\ of\ the}{bonded\ atom}^{1} \quad \Delta\rho_c \quad \rho_c,$$

where position is done with respect to the considered atom.

[1] BCPs only

***.surf** - the file with the Bader surface - there is the head of the form:

index of the atom position
ntheta thetamin thetamax
nphi phimin phimax

and the list of the Bader surface radiuses:

$$\theta \quad \phi \quad r(\theta,\phi) \quad W(\theta,\phi)^2.$$

The minimal and maximal radiuses are done at the last line.

***.out** - the central output file - there are many informations which are clearly described. Here is some additional information on the integration - there is separated integrations of the core and the valence density. In both cases the radial integration is performed using cubic splines and the angular ones by Gauss quadrature. However the principal part of the core density *of the considered atom* is integrated in the sphere of minimal Bader radius using spherical symmetry. The rest of the core density (of this atom, out of this sphere) together with all the core contributions of the neighbors are added to the valence density integration. In the ouput file, there is the result of the *complete* integration of the core density of the atom, then the two contributions (spherical integration vs. the others) and then the total Bader charge.

***.log** - the log file - a lot of informations but it is not very nice actually.

***.gp** - gnuplot script showing the calculated part of the Bader surface with lines.

The gnuplot scripts are made in the manner that one needs type only
load '*file*' (quotes are necessary).
Note, that this isn't considered as the visualization (it is only for working purpose)!

The list of the input variables

The atomic units and cartesian coordinates are used for all input parameters.

Driver variables

surf the calculus of the Bader surface

- **0(d)** not
- **-1** reading from the file "root".surf
- **1** calculated

crit the calculus of the critical points

- **0(d)** not
- **-1** reading from the file "root".crit
- **1** calculated (simplified version)
- **2** calculated (standard version)
- **3** calculated (the original version)

The original version searches all CPs starting from the center betwen two and three atoms (atom - neighbor(s)) by Newton-Raphson algorithm - without tests (not recommended) - don't use together with surface analyse! **The simplified and standard version** searches CP(3,-1) starting from the center of the pairs atom-neighbor; then CP(3,1) from the center between two CP(3,-1) and finally CP(3,3) from the center between two CP(3,1). The robust Popeliers's algorithm is used. The difference between two is based in the fact that **the standard version** makes the test if the CP is really on the Bader surface of the calculated atom for each CP, while

[2] the weight for the Gauss quadrature

the simplified version only for CP(3,-1). When CP analyse is rather fast (with respect surface determination) 2 is recommended. In all cases the number of neighbors considered is limited by distance cutoff (variable **maxatd**)

irho the integration of the charge of the Bader atom

- **0(d)** not calculated
- **1** calculated

ivol the integration of the volume of the Bader atom

- **0(d)** not calculated
- **1** calculated

rsur the determination of the rayon of the Bader surface for the angles specified in the input variable **rsurdir**.

- **0(d)** not calculated
- **1** calculated

follow follow the gradient path to the corresponding atom starting from the position specified in the input variable **foldep**.

- **0(d)** not calculated
- **1** calculated

denout,lapout the additional output of electronic density and of the laplacian of electronic density. The specification of the line (plane) in the real space must be given in the input variable **vpts** and grid in **ngrid**. It is also possible to get only the valence density or the core density (see **dltyp**).

- **0(d)** not
- **1** 1D distribution
- **2** 2D distribution

dltyp the specification of the contribution of the electronic density corresponding to the additional output (denout, lapout).

- **0(d)** total electronic density
- **1** only the valence density
- **2** only the core density

gpsurf the additional graphic output (gnuplot script) of the irreducible part of the calculated Bader surface.

- **0(d)** not
- **1** given

Input variables

atom the index of the investigated atom (1 - default)

nsa,nsb,nsc the number of the primitive cell to be constructed in each direction and each sense of the primitive translation to simulate lattice ((3,3,3) - default).

inpt the number of radial points used for integration of the Bader charge (100 - default) (not too sensitive).

ntheta,nphi the angular grid for the integration of the Bader charge (32×48 - default). Proposition : 20 for $[0, \pi/2]$, 32 for $[0, \pi]$ and 48 for $[0, 2\pi]$.

thetamin,thetamax the limit of the angular region of the Bader surface which is calculated ($[0, \pi]$ - default).

phimin,phimax the limit of the angular region of the Bader surface which is calculated ($[0, 2\pi]$ - default)

atrad the first estimation of the Bader radius (not too important - it is used only two times) (1.0 - default).

radstp the first step in searching the exact Bader radius (0.05 - default) (0.03 is well usually)

folstp the first step for following the gradient path (0.05 - default)

ratmin the first estimation of the least radius of the bassin of the atom (the distance in which the procedure following the gradient path annonces that the gradient path finishes in the corresponding atom) (1.0 - default). This parameter is very important for the speed of the calcul, but this first estimation is not usually used because program makes the new one based on the knowledge of CPs. In fact after the CP analyse, the new estimation is done by the product of adhoc parameter coff1 (0.98) (**adhoc.f90**) and the distance of the nearest bonding CP. If there is problem later, coff2 (0.95) is used instead.

scal the scaling of the cartesian coordinates for the calcul of the distances ($x'[i] = x[i]/scal[i]$) ($[1.0, 1.0, 1.0]$ - default) - not really usefull.

maxatd - the maximal distance for the atoms considering for searching CP (10.0 - default).

maxcpd - the maximal distance of CPs considering for the corresponding atom (5.0 - default).

rsurdir variable only for the case rsurf=1 i.e. for the calcul of the singular Bader radius (rsur) = corresponding angular coordinates ($[0.0, 0.0]$ - default).

foldep variable only for the case follow=1 i.e. for the procedure following only the gradient path (follow) - starting point ($[0.0, 0.0, 0.0]$ - default).

ngrid dependent variable for the additional output (denout, lapout) - the grid in the real space ($[30, 30]$ - default)

vpts dependent variable for the additional output (denout, lapout) - the points defining the basic vector(s) of the line or rectangle in real space (the first = the basic point) (all points are par default nulify).