**PyFIREBALL Beta.**

***The supercell class****.*

Originally the atom class was seen as the center of the PyFIREBALL toolbox, where all data is dealt with in an atom-centric philosophy. The supercell class is a further abstraction that makes dealing with atomLists that much easier. In essence, the supercell class is actually all the information from a FIREBALL-TG style working directory. The supercell class acts as a container for the total energy, the eigenvalues, the atomList, etc. The supercell class can be created by calling it’s initializer, which requires a number of arguments:

cell1 = supercell(directory, basisfile, atomList, lvs=[], xl=[], eigens = []), where all arguments have their standard meanings.

Alternatively, and more conveniently, the supercell can be initialized with a call to the “directory2Supercell()” global method, which reads in the FIREBALL directory and generates a supercell object:

cell1 = directory2Supercell(elements, directory)

here, “elements” is the dictionary of elements from the readInfo() method (which reads in the Fdata) and directory is the path to the FIREBALL directory where your study was run.

The power of a supercell object should be obvious- you can read in a number of directories for direct comparison, you can carry out any analysis of a single FIREBALL run and compare and contrast with others (for example, see the 3DOS tutorial in the Tutorials section), etc.

*supercell immutable variables:*

.ztot -> Total number of electrons in the cell

.HOMO -> The energy of the HOMO band in the cell

*supercell mutable variables:*

.directory -> Working directory for this supercell

.basisfile -> This supercell’s basis file

.lvs -> The LVS vectors for this cell

.xl -> The eXtended LVS vectors for reflections of the cell in space, required by the neighbor methods for atom()

.atomList -> The list of atom objects in this cell

.eigens -> energy eigenvalues of the cell

*supercell internal methods:*

.plotPDOS(elements, filename = 'supercellDOS') -> Generate total density of states plot with a contributions plot by each species in the cell

.genDOS (elements) -> Returns a DOS object with a contributions dictionary for each species in the cell.

.genDOSspec(element) -> Returns a DOS object of ONLY the PDOS of the species named in elements with a contributions dictionary of the shell/orbital combinations.

*global methods that apply to supercell():*