UCONN

The Intersection of Computational Chemistry and Experiment

Visualization Hints Using Jmol, PyMol, and VMD, and GaussView

Angelo R. Rossi
Department of Chemistry
The University of Connecticut
angelo.rossi@uconn.edu

Spring 2017

Last Updated: February 17, 2017 at 5:18pm

Visualizing Molecular Orbitals with Jmol

Actually, automatic plotting of orbitals with Jmol is really easy, considering that Jmol can directly read the output of many major quantum chemical programs.

Jmol does not have all of its functionality easily available using a menu, and so, it might be a good idea to have the Jmol scripting manual available (http://chemapps.stolaf.edu/jmol/docs/). Jmol is rather easy to use via the console, and has an advantage of script-based input where one can naturally automate repetitive tasks.

For visualizing molecular orbitals, here are a few of the possible commands:

```
background white
mo fill
mo cutoff 0.03
mo homo
```

At this point, you should probably play with the perspective, and then continue by saving all images of orbitals of interest: For example,

```
write image png "homo.png"
mo homo-1
write image png "homo1.png"
mo homo-2
write image png "homo2.png"
mo lumo
write image png "lumo.png"
mo lumo+1
write image png "lumo1.png"
```

Jmol also allows you to make linear combinations of MOs.

For example, if we combine the HOMO (orbital no. 183 of structure 1) with the HOMO-1 (orbital no. 182 of structure 1 for a given molecule, a molecular orbital is obtained,

```
mo [1 183 1 182]
```

which is localized on a specific moiety of the molecule.

Generating Cube Files with Gaussian Utilities

The first step is to use the **formchk** utility and create a **formatted** checkpoint file (**.fck**) from an **unformatted** checkpoint file (**.chk**), resulting from a Gaussian calculation.

A formatted checkpoint file will be needed for generating cube files for the ground state and each excited state.

The **cubegen** utility is then used to extract cube files from the formatted checkpoint files of interest.

Using the Gaussian formchk Utility

- **formchk excited.chk**

  A formatted checkpoint file, **excited.fck**, is created for an excited state calculation.

It is necessary to perform only a single excited-state calculation using the **td=(root=N, $\cdots$, $\cdots$)** and **density=all** keywords to create the **excited.chk** checkpoint file.

These keywords save the density of the ground state and the density for N^{th} excited state. If the **root** keyword is not present, then the default is to calculate the electron density of the first excited state.

Using the Gaussian cubgen Utility

- **cubegen 0 density=SCF excited.fck ground.cube 0 h**

  A cube file, **ground.cube**, is created for the ground state.

- **cubegen 0 density=CI excited.fck excited.cube 0 h**

  A cube file, **excited.cube**, is created for the excited state.

Creating Difference Density Cube Files

Using the Gaussian cubman Utility

The following commands for the **cubman** utility will produce an appropriate density difference cube file:

1. **cubman**
2. Enter **su** for subtract at the initial prompt.

   This operation subtracts two cube files to produce a new cube file.
3. name for the first input file (example: **excited.cube**)

   This corresponds to the cube file containing electron density of a specific excited state.

   Is the file formatted? Answer **y** for yes.
4. name for the second input file (example: **ground.cube**)

   This corresponds to the cube file containing electron density of the ground state.
5. Is the file formatted? Answer **y** for yes.
6. name for the output file (example: **excited-ground.cube**)

   This corresponds to a cube file containing electron density difference of the ground and excited state.

   Should it be formatted? Answer **y** for yes.

The above sequence of commands should create the file **excited-ground.cube**.

Follow the above procedure for each of the excited-state difference densities to be viewed with visualization software which can read cube files.

Visualizing Difference Densities with GaussView

GaussView can be used to read an electron density cube file, and create an isosurface rendering of the result.

1. Select **File** → **Open**, and select the formatted checkpoint file for the excited state that you want to visualize the difference density for in GaussView. Select **OK**.

2. Select **Results** → **Surfaces**, and a **Surfaces and Cubes** selection window will appear.

3. Select the **Cube Actions** drop down menu. Select **Load Cube**.

   Find the cube file that represents the difference density calculated with **cubman**. Select **OK**.

   A line of data will appear in the **Cubes Available** box.

   Select an appropriate isosurface value, such as 0.004.

   Adjust the isosurface value until you obtain a visualization of the difference densities that is acceptable.

   This will generate a graphical window with the molecule and difference density for the selected excited state.

   The blue represents **from where** the electrons came, and the purple represents **to where** the electrons move.

4. If visualization of the electron density changes is not ideal, then select the **Surface Actions** drop down menu followed by selecting **Remove Surface**. Change the isosurface value and save. Then choose the **Surface Actions** drop down menu, and select **New Surface**.

   Repeat these steps until the visualization for the difference density is acceptable.

Visualizing Gaussian Cube Files
VMD, PyMol, and Jmol

VMD

1. Load the file **homo.cube** directly into VMD with a file type of **Gaussian Cube**.

   The skeletal structure of the molecule will now be visible.
2. Choose **Representations**

   - Duplicate the cube file.
   - Choose **Isosurface**.
   - Using the duplicated cube files, fiddle with the **Isovalue** to get the right size orbital.

     Use one duplicated cube file for the +**Isovalue**

     Use the other duplicated cube file for the -**Isovalue**.

PyMol

1. Load the cube file generated by the Gaussian utility:

   load homo.cube

   homo is now a defined object, and can be used as part of PyMol commands.

   Note this command will not load the skeletal structure. A separate coordinate file (.xyz, .pdb, ... etc) should be loaded for the molecular structure.
2. Generate the molecular orbital:

   isosurface pos, homo, 0.003

   isosurface neg, homo, -0.003

   or

   isosurface elf, homo, 0.003

   isosurface elf, homo, -0.003

   Also, see the **Ramp New** command (https://pymolwiki.org/index.php/Ramp_New) which can possibly be used to generate surfaces.

Jmol

1. Open Jmol Script Console Window so that several commands can be executed.
2. The following command loads the cube file:

   load homo.cube (A skeletal structure should now be visible.)
3. The next two commands create positive and negative lobes of the molecular orbital:

   isosurface pos 0.003 "homo.cube"

   isosurface neg -0.003 "homo.cube"