How to calculate Fermi-Softness *via* CP2K

QIAOSONG LIN

FROM DALIAN INSTITUTE OF CHEMICAL PHYSICS, CAS

FEB 2022

Contents

1. How to **install program** ?
2. How to **run CP2K program** for calculating Fermi-Softness ?
3. How to **get Total, Condensed and Local Fermi-Softness** ?
4. How to **visualize Local Fermi-Softness** ?

1. Install program

Anaconda3, bader, and FermiSoftness is necessary.

1. You can install Anaconda3 from the website:

Anaconda | Individual Edition (https://www.anaconda.com/products/individual#Downloads)

2. You can download bader from the website:

Bader Charge (http://theory.cm.utexas.edu/henkelman/code/bader/)

3. You can install FermiSoftness by pip, ASE and Numpy will be installed automatically:

$ *pip install FermiSoftness*

2. Run CP2K program

1. Build a Slab model.

2. Relaxation calculation.

3. Do a single point energy calculation with **gamma only**, and following key words should be in your input:

```
&DFT
......
   &PRINT
    &MO_CUBES
     STRIDE 1
     NHOMO -1
     NLUMO -1
    &END MO_CUBES
   &END PRINT
 &END DFT
```

3. Calculate Total, Condensed and Local Fermi-Softness

1. Generate input file runfs.py :

$ *python –c “from FermiSoftness import gen; gen(software=‘cp2k’);”*

2. Copy runfs.py to the path where the input of CP2K is.

3. Calculate Total, Condensed and Local Fermi-Softness

3. Modify the parameters in *runfs.py* :

```
filename='pt3y.out'
project_name='pt3y'
kbT=0.4                          # Electron temperature (eV)
dfdd_threshold=0.001             # Derivation of Fermi-Dirac distribution threshold
bader_dir='bader'                # Path of bader
band_gap={'VBM':[0.0],           # If band gap exists, set as E_VBM,E_CBM (Do not minus E_Fermi);
          'CBM':[0.0]}           # Otherwise set as 0.0 0.0 (eV)
```

- filename is the single point energy output file, project_name should be the same as the value after the PROJECT keyword in the input file
- If *bader* is in your *$PATH*, you don't need to change bader_dir

3. Calculate Total, Condensed and Local Fermi-Softness

4. Run "*python runfs.py*" in terminal.

5. Wait … it will take several minutes.

6. Finished. You can find files *FSCAR* and *LFS.cube* in current path. (If the wavefunction is spin polarization, you can find *_UP*, *_DW* for different spin. If band gap exists, you can find *_CB*, *_VB* for holes and electrons.)

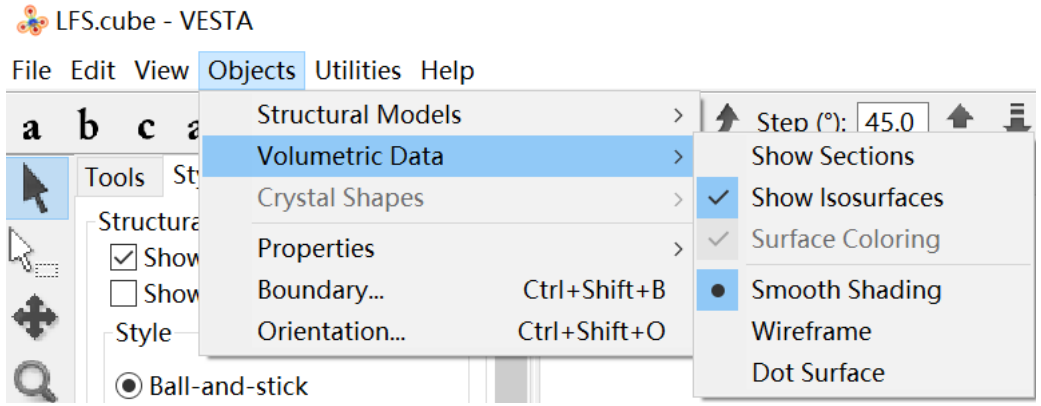
7. In *FSCAR*, you can find Total Fermi-Softness (NUMBER OF ELECTRONS) and Condensed Fermi-Softness (CHARGE), the grid data of Local Fermi-Softness were recorded in *LFS.cube*.

4. Visualize Local Fermi-Softness

1. Isosurfaces

Load *LFS.cube* by VESTA

- Objects->Volumetric Data-> Show Isosurfaces

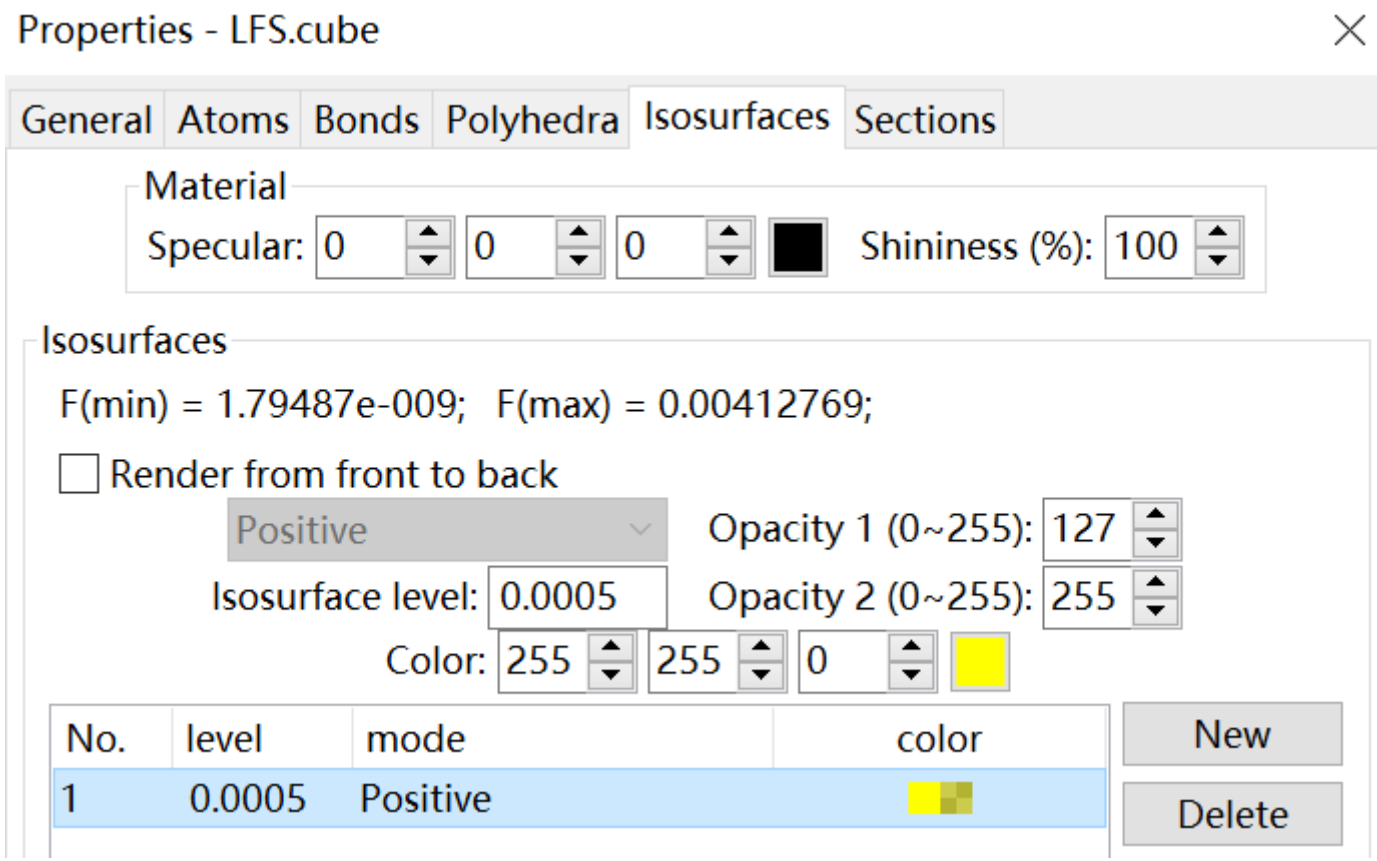


4. Visualize Local Fermi-Softness

1. Isosurfaces

Objects->Properties-> Isosurfaces

- Set appropriate isosurface level

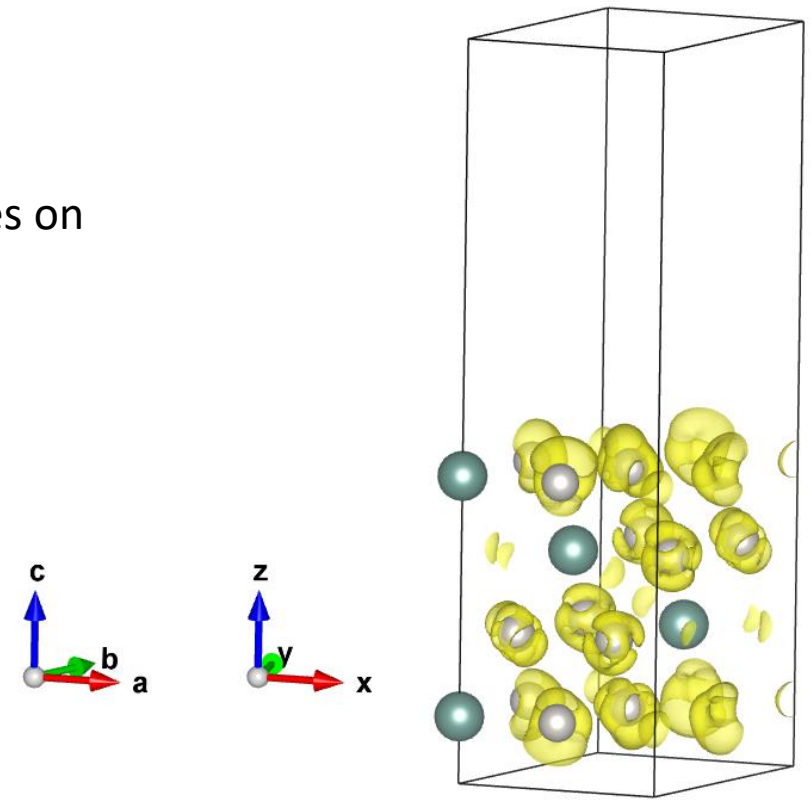


4. Visualize Local Fermi-Softness

1. Isosurfaces

Local Fermi-Softness mainly distributes on

Pt atoms in $Pt_3Y(111)$ surface.



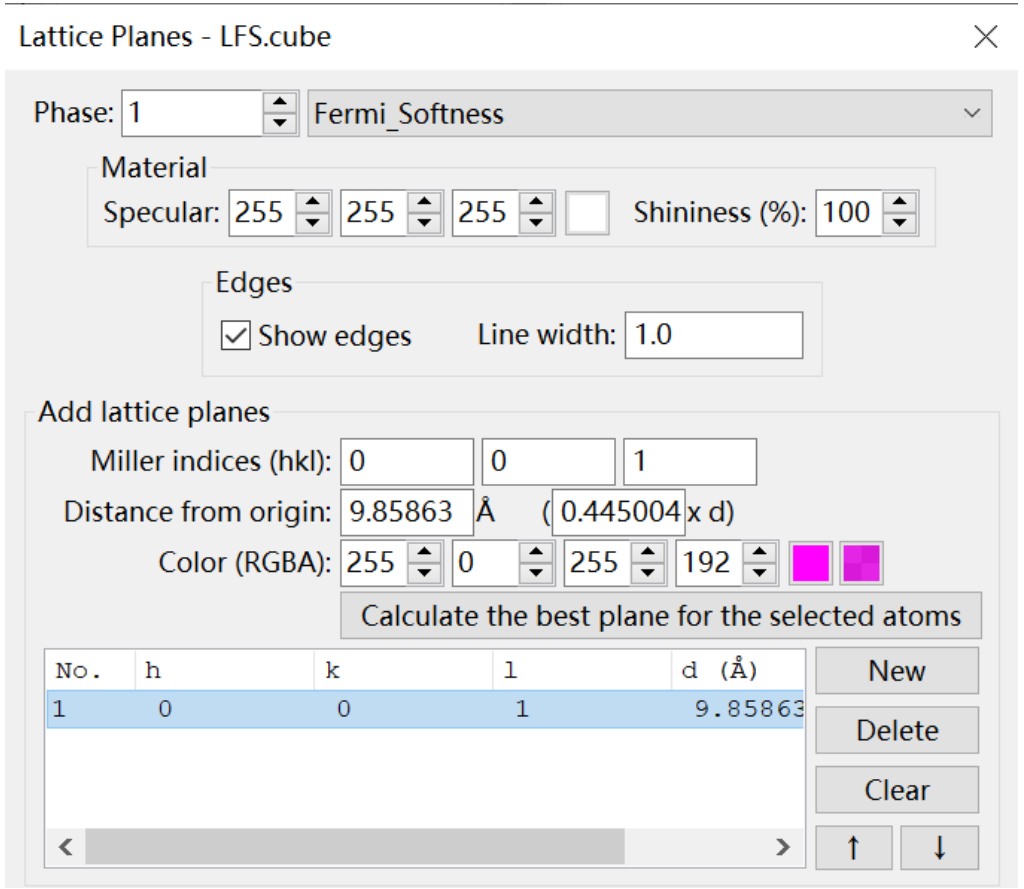
4. Visualize Local Fermi-Softness

2. Contour plante

Load *LFS.cube* by VESTA

Edit->Lattice Planes

- Set Miller indices (001)
- Set appropriate Distance from origin

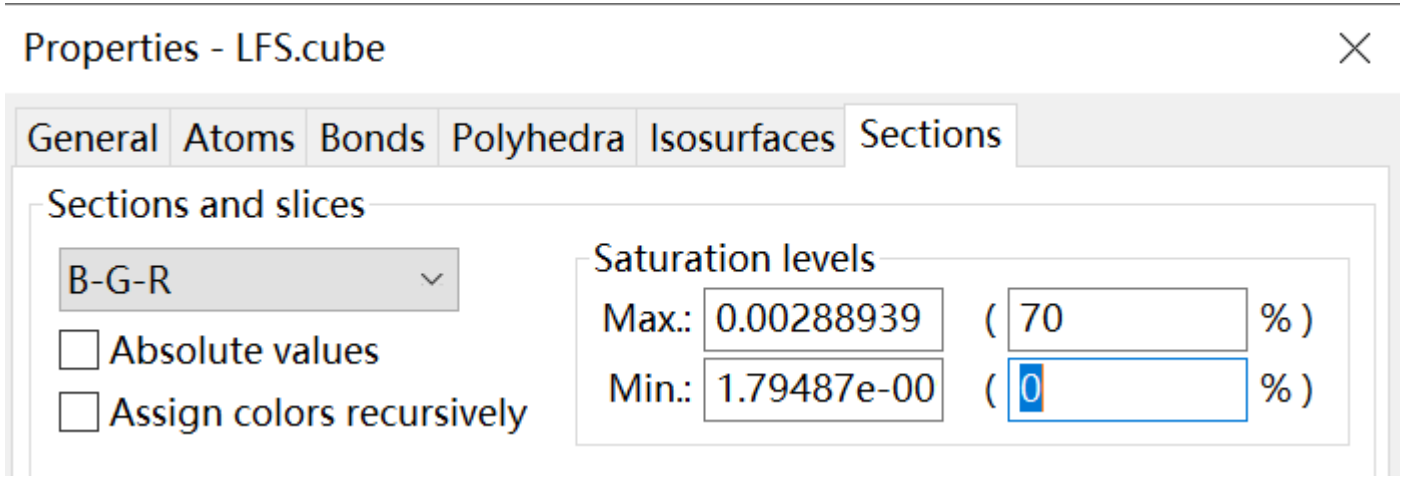


4. Visualize Local Fermi-Softness

2. Contour plante

Objects->Properties-> Sections

- Set appropriate Saturation level

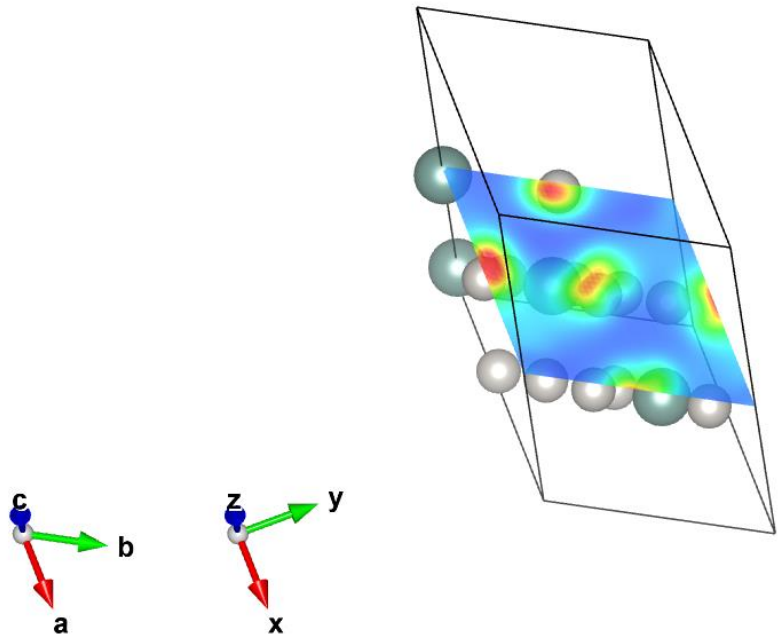


4. Visualize Local Fermi-Softness

2. Contour plante

Local Fermi-Softness mainly distributes on

Pt atoms in $Pt_3Y(111)$ surface.

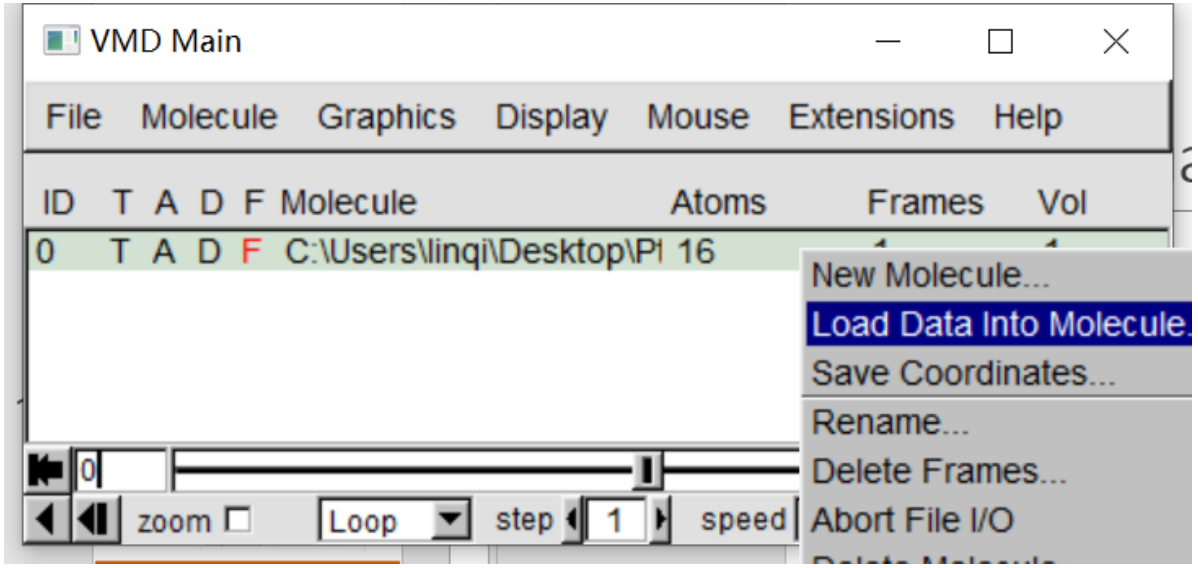


4. Visualize Local Fermi-Softness

3. Projected isosurface

Load *charge density files* by VMD

Load Data Into Molecule: *LFS.cube*



4. Visualize Local Fermi-Softness

3. Projected isosurface

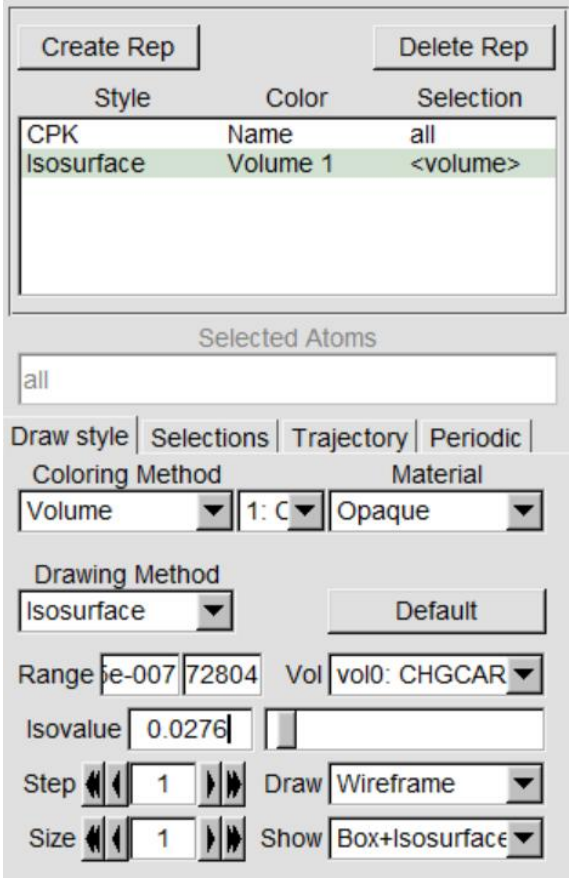
Graphics->Colors->Color Scale->Method: BWR

Graphics->Representation

- Make first Rep: CPK Name all
- Make second Rep: Isosurface Volume 1

Graphics->Representation->Trajectory

- Set appropriate Color Scale Data Range



4. Visualize Local Fermi-Softness

3. Projected isosurface

Local Fermi-Softness mainly distributes on

Pt atoms in $Pt_3Y(111)$ surface.