Density of States and Optical properties

Davide Grassano

December 13, 2017

Executing

The Density of states(DOS) and optical properties are quantities, as shown in Chapter 2, that requires the integration over the whole Brillouin Zone(BZ). In order to produce reliable data, a convergency test over the size of the kpt mesh needs to be run. For the optical properties that depends on both occupied and unoccupied states, another convergency test on the number of empty bands (or total bands) needs to be run in order to understant the uppermost energy to which our result is reliable.

1.1 Density of states - DOS and Optical properties

Run the self consistent calculation in order to obtain a good charge density stored in the tmp.

```
pw.x < Si.scf.in > Si.scf.out
```

Save the tmp in order to not have to repeat the scf calculation everytime.

```
cp -r tmp tmp_scf
```

Run the non self consistent calculation.

```
pw.x < Si.kpt_??.nscf.in > Si.kpt_??.nscf.out (??=5/10/15/20)
```

Create a new folder and move the outputs into it.

```
mkdir kpt_??
mv -t kpt_?? Si.kpt_??.nscf.out tmp
```

Go inside the folder and run the dos.x and pw2gw.x to obtain the DOS and Optical properties.

```
cd kpt_??
dos.x < ../Si.dos.in
pw2gw.x < ../Si.pw2gw.in
```

Go back to the previous folder, copy the backup tmp and repeat for all kpt mesh.

```
cd ..
cp -r tmp_scf tmp
```

Repeat the same procedure for the band convergence ('kpt' in file and folder names becomes 'bnd').

Compile the broad.c program to apply a gaussian broadening to the file containing the dielectric function

```
gcc -o broad.x broad.c -lm -std=c11 -Wall -O2 -g
```

where $-o$ indicates the output file, $-lm$ links the math.h library, $-std = c11$ tells the compiler to use the 2011 standard, $-Wall\ -O2\ -g$ are debug and optimization options (can be omitted).
To use the broad.x executable, copy it in the folder where you want to use it (or use the relative/absolute path when launching it) and use the following syntax:

```
./broad.x "input file" "broadening value in eV" > "output file"
e.g.: ./broad.x   kpt_5/epsTOT.dat   0.1   >   kpt_5_0.1_epsTOT.dat
```

Use the gnuplot script to plot the data.

```
gnuplot dos.gnu
gnuplot eps_kpt.gnu
gnuplot eps_bnd.gnu
```

Formulas and Theory

2.1 Plane wave expansion

$$H(\rho(\mathbf{r}))\psi_{n,\mathbf{k}}(\mathbf{r}) = \varepsilon_{n,\mathbf{k}}\psi_{n,\mathbf{k}}(\mathbf{r}), \quad n = \mathit{band\ index} \tag{2.1.1}$$

$$\psi_{n,\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{n,\mathbf{k}}(\mathbf{G})e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \tag{2.1.2}$$

$$|\mathbf{k}+\mathbf{G}|^2 < E_{cut} \tag{2.1.3}$$

2.1.1 Gradient of the PW expansion

$$\nabla\psi_{n,\mathbf{k}}(\mathbf{r}) = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \sum_{\mathbf{G}} c_{n,\mathbf{k}}(\mathbf{G})e^{i[x(k+G)_x+y(k+G)_y+z(k+G)_z]} =$$
$$\sum_{\mathbf{G}} c_{n,\mathbf{k}}(\mathbf{G}) \ \begin{bmatrix} i(k+G)_x \ e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \\ i(k+G)_y \ e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \\ i(k+G)_z \ e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \end{bmatrix} = i\sum_{\mathbf{G}} c_{n,\mathbf{k}}(\mathbf{G}) \ (\mathbf{k}+\mathbf{G}) \ e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \tag{2.1.4}$$

2.1.2 Fourier Transfor of a Dirac's Delta

$$F(\mathbf{q}) = F.T.(f(\mathbf{r})) = \frac{1}{(2\pi)^{3/2}}\int d^3re^{-i2\pi \ \mathbf{r}\cdot\mathbf{q}}f(\mathbf{r}) \tag{2.1.5}$$

$$f(\mathbf{r}) = F.T.^{-1}(F(\mathbf{q})) = \frac{1}{(2\pi)^{3/2}}\int d^3qe^{i2\pi \ \mathbf{r}\cdot\mathbf{q}}F(\mathbf{q}) \tag{2.1.6}$$

It can be shown that:

$$\delta(\mathbf{k}-\mathbf{k}_0) = \frac{1}{(2\pi)}\int d^3re^{i2\pi \ \mathbf{r}\cdot(\mathbf{k}-\mathbf{k}_0)} \tag{2.1.7}$$

2.2 BZ integrations

In order to calculate quantities such as:

$$G = \sum_n \frac{1}{\Omega_{B.Z}} \int_{\Omega_{B.Z}} d^3k f(\varepsilon_{n,\mathbf{k}})\ g_n(\mathbf{k}) \tag{2.2.1}$$

Where f is the Fermi-Dirac distribution.
In order to implement such integral into a code, a discretization of the integral is needed. Most codes generates a Monkhorst-Pack mesh and then reduce it using the symmetry properties of the system, by assigning a weight to each kpt. For example is 6 kpt fold int k_1 and only 3 kpt fold into k_2, the weight of k_1 will be double that of k_2

$$G = \sum_n \sum_{\mathbf{k}} w(\mathbf{k})\ f(\varepsilon_{n,\mathbf{k}})\ g_n(\mathbf{k}), \quad \sum_n^{N_{kpt}} w(\mathbf{k}) = 1 \tag{2.2.2}$$

In the case of a semi-conductor/insulator f is a step-function and can be included in the summation by running it only on occupied states:

$$G = \sum_n^{occ} \sum_{\mathbf{k}} w(\mathbf{k})\ g_n(\mathbf{k}) \tag{2.2.3}$$

thus a semi-conductor/insulator require a moderate amount of kpt to achieve convergence. In the case of a metal this cannot be done, and we need enough kpts in order to sample the fermi surface of the material.

$$example \quad \rho = \sum_n^{occ} \sum_{\mathbf{k}} w(\mathbf{k})\ |\psi_{n,\mathbf{k}}|^2, \quad g_n(\mathbf{k}) = |\psi_{n,\mathbf{k}}|^2 \tag{2.2.4}$$

2.3 DOS

$$DOS(E) = \sum_{n,\mathbf{k}} \delta_{E,\varepsilon_{n,\mathbf{k}}} \cdot w(\mathbf{k}) \tag{2.3.1}$$

A gaussian broadening is applied in order to obtain a spectra akin to the experimental ones.

$$f_{broad}(x) = \int_{-\infty}^{\infty} dx' f(x') \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x-x'}{2\sigma^2}} \tag{2.3.2}$$

2.4 Optical properties

$$Im(\tilde{\varepsilon}) \propto \sum_{v,c,\mathbf{k}} \frac{| < \psi_{v,\mathbf{k}}(\mathbf{r})|\mathbf{e}\cdot\mathbf{p}|\psi_{c,\mathbf{k}}(\mathbf{r}) > |^2}{(\varepsilon_{c,\mathbf{k}} - \varepsilon_{v,\mathbf{k}})^2} \tag{2.4.1}$$

Calculation of the matrix elements:

$$\begin{aligned} M_{c,v,\mathbf{k}} &= < \psi_{v,\mathbf{k}}(\mathbf{r})|\mathbf{e}\cdot\mathbf{p}|\psi_{c,\mathbf{k}}(\mathbf{r}) > \propto < \psi_{v,\mathbf{k}}(\mathbf{r})|\nabla|\psi_{c,\mathbf{k}}(\mathbf{r}) > = \\ &\int d^3r \sum_{\mathbf{G}} \sum_{\mathbf{G}'} c_{v,\mathbf{k}}(\mathbf{G})^* \; c_{c,\mathbf{k}}(\mathbf{G}') \; e^{-i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} \nabla e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}} = \\ &i \sum_{\mathbf{G}} \sum_{\mathbf{G}'} c_{v,\mathbf{k}}(\mathbf{G})^* \; c_{c,\mathbf{k}}(\mathbf{G}')(\mathbf{k}+\mathbf{G}) \int d^3r e^{i(\mathbf{G}'-\mathbf{G})\cdot\mathbf{r}} = \\ &i \sum_{\mathbf{G}} \sum_{\mathbf{G}'} c_{v,\mathbf{k}}(\mathbf{G})^* \; c_{c,\mathbf{k}}(\mathbf{G}')(\mathbf{k}+\mathbf{G}) \; \delta(\mathbf{G}'-\mathbf{G}) = \\ &i \sum_{\mathbf{G}} c_{v,\mathbf{k}}(\mathbf{G})^* \; c_{c,\mathbf{k}}(\mathbf{G})(\mathbf{k}+\mathbf{G}) \end{aligned} \tag{2.4.2}$$