

INSTALLING QUANTUM ESPRESSO

1. Download the quantum espresso from <https://github.com/QEF/q-e/releases>
2. On the terminal run: `tar xzf q-e-qe-6.4.tar.gz`,
3. `cd q-e-qe-6.4.tar.gz`
4. `./configure`
5. `make all`
6. `make epw`
7. `vi .bashrc`, add `export PATH = $PATH:/home/username/ q-e-qe-6.4/bin`

PSEUDOPOTENTIAL GENERATION/ VIRTUAL CRYSTAL APPROXIMATION

Ti and Se norm-conserving pseudopotentials were downloaded from the website www.pseudo-dojo.org. They were produced using the code Optimized Norm-Conserving Vanderbilt Pseudopotential (ONCVPSP) scalar relativistic version [D. R. Hamann, Phys. Rev. B 88, 0857117 (2013)]. And their electronic configurations are Ti: [Ne] 3s2 3p6 4s2 3d1 and Se: [Ar] 4s2 3d10 4p4

The Li (1s2, 2s1) and He (1s2) norm-conserving pseudopotentials, which were mixed together based on the virtual crystal approximation scheme, were generated using Atomic (whose executable is `ld1.x`), a code available in Quantum Espresso. The user manual for this code can be found here <https://www.quantum-espresso.org/Doc/pseudo-gen.pdf> while the input file description can be found here http://www.quantum-espresso.org/Doc/INPUT_LD1.html

Using `ld1.x`

We will generate a norm-conserving PP for Li atom

1. Go to the directory **Re_mtd/PP_gen/Li/NC/**
2. Confirm that this directory contains the input file *Li.atomic.in*, right click in this directory and select the option *Open in Terminal*
3. On the terminal type `ld1.x < Li.atomic.in > Li.atomic.out`

The PP file is written in *Li.1.pw-n-tm-nc.UPF* which is the file name specified in `file_pseudopw` option of the *Li.atomic.in* input file.

In generating a norm-conserving PP for He atom, go to the directory **Re_mtd/PP_gen/He/NC/** and follow the same steps above.

Now we will mix the Li and He atoms based on the virtual crystal approximation (VCA). VCA involves the linear combination of the pseudopotentials of two atoms. Please for the theory of the VCA see these references **S. de Gironcoli et al. Phys. Rev. Lett. 66, 2116 (1991)** and **L Bellaiche and D. Vanderbilt, Phys. Rev. B 61, 7877 (2000)**. In our case, Li with atomic number 3 was mixed with He with atomic number 2, with the new atom having an atomic number, y given by:

$$y = 3x + 2(1-x)$$

where x is the doping concentration of Li.

Technically, the VCA is implemented by the executable `virtual_v2.x` in Quantum Espresso. As far as I know, there is no user manual available for this code. However, its execution is straight forward as we will see below:

Using `virtual_v2.x`

1. Go to the directory `Re_mtd/VCA/` containing the PP files of Li and He atoms `Li.1.pw-n-tm-nc.UPF` and `He.1.pw-n-tm-nc.UPF` respectively.
2. On this directory right click and select the option *Open in Terminal*
2. On the terminal make sure `virtual_v2.x` is in your path by typing *which virtual_v2.x*. If absent, get it from your quantum espresso directory; `q-e-qe-6.4/upftools/virtual_v2.x`, and then add to your path (see HOW TO ADD EXECUTABLES TO YOUR PATH on page)
3. Now on your terminal type `virtual_v2.x`, and something like this will appear:

Program VIRTUAL_V2.X v.6.4 starts on 17Feb2020 at 0:30:11

This program is part of the open-source Quantum ESPRESSO suite for quantum simulation of materials; please cite

"P. Giannozzi et al., J. Phys.:Condens. Matter 21 395502 (2009);

"P. Giannozzi et al., J. Phys.:Condens. Matter 29 465901 (2017);

URL <http://www.quantum-espresso.org>",

in publications or presentations arising from this work. More details at <http://www.quantum-espresso.org/quote>

Parallel version (MPI), running on 1 processors

MPI processes distributed on 1 nodes

Generate the UPF pseudopotential for a virtual atom combining two pseudopotentials in UPF format

Input PP file # 1 in UPF format >

4. Type the file name of the first PP, in our case `Le.1.pw-n-tm-nc.UPF` and this will appear:

Input PP file # 1 in UPF format > `Le.1.pw-n-tm-nc.UPF`

Input PP file # 2 in UPF format >

5. Now type the file name of the second PP, in our case `He.1.pw-n-tm-nc.UPF` and this will appear:

Input PP file # 2 in UPF format > `He.1.pw-n-tm-nc.UPF`

CAUTION !!! You are mixing pseudos with different number of electrons in valence

CAUTION !!! You are mixing pseudos with a different number of projectors

New Pseudo = x `Le.1.pw-n-tm-nc.UPF` + $(1-x)$ `He.1.pw-n-tm-nc.UPF`

mixing parameter x [$0 < x < 1$] =

6. Notice the CAUTIONS, but for our specific calculation we confirmed that they were harmless. The program now requires that you input the mixing parameter x , let's choose 0.11.

7. The program then produces a new PP with the file name `NewPseudo.UPF` that is automatically stored in the same directory containing the Li and He PP files.

To understand the Unified Pseudopotential Format (UPF) of the PP files used in Quantum Espresso, please check out this link: <http://www.quantum-espresso.org/pseudopotentials/unified-pseudopotential-format>.

You can also access some ready-to-use pseudopotentials here <http://www.quantum-espresso.org/pseudopotentials> and specifically, norm-conserving ones here <http://www.pseudo-dojo.org/>

DENSITY FUNCTIONAL THEORY

After generating/getting PP file, the next thing to do is to go on to:

1. Relax the system/structure/material
2. Calculate the electronic band structure
3. Calculate the density of states and the projected density of states
4. Get the fermi surface plot
5. Charge density plot

1. Structure relaxation:

Relaxation is necessary to obtain a stable system for the phonon calculation. Relaxation loosely means to allow the system's cell parameters and atomic positions to change such that the total energy of the system is minimized. The phonon eigen modes of a system as implemented in Density Functional Perturbation Theory DFPT (DFPT) is usually computed effectively if the system is stable, in other words relaxed.

In a broad sense, there are two ways to relax a system in Quantum Espresso. They are full relaxation and ionic relaxation. Full relaxation consists of relaxing both the cell parameters and the atomic/ionic positions. Whereas the ionic relaxation only relaxes the ionic positions. The executable/program for structure relaxation calculation in QE is *pw.x*

We will now proceed to do a relaxation calculation for these two methods using TiSe₂ system.

Full relaxation:

1. Set up the input file named *vc_relax.in* as shown in [Re_mtd/Relax/TiSe2_full_relax/](#)
Pay attention to the namelists &CONTROL, &IONS, and &CELL. In order to run a full relaxation, the input option *calculation* must be set to "vc_relax" in the &CONTROL namelist. Here also the values of *tstress* and *tpnfor* are set to true. Finally, both the &IONS and &CELL namelists must be present in the input file.
2. Go on to run the calculation in the terminal by typing *pw.x < vc_relax.in > vc_relax.out*
3. Examine the output file and note that relaxation is complete when this phrase appears towards the end: *Begin final coordinates* with the final coordinates of the atomic positions and crystal coordinates printed afterwards.

Ionic relaxation:

For the ionic relaxation, go to the directory [Re_mtd/Relax/TiSe2_relax/](#). As with the variable-cell relaxation, the whole input file structure (see below) is the same except that in the &CONTROL namelist, the input option *calculation* must be set to “relax”. Also, while the namelist &IONS is still present, the namelist &CELL is entirely deleted.

Notice that in the relaxation calculations there are two important namelists. The &IONS and &CELL namelists. To get more insights about their options and functions, please check out pw.x table of contents in Quantum Espresso: http://www.quantum-espresso.org/Doc/INPUT_PW.html

NOTE: The above relaxation calculation is for the pristine TiSe₂ system. Nevertheless, the doped TiSe₂ systems have equivalent input files and they follow the same procedures as can be seen in [Re_mtd/Relax_doped/](#)

2. Band Structure Calculation:

After relaxation, there are four steps in making an electronic band structure plot.

- a. Self-Consistent Field Calculation (scf), to obtain the wave functions
- b. Non-Self-Consistent Field Calculation (nscf), to obtain the energy eigen values along a k-path within the Irreducible Brillouin Zone (IBZ).
- c. Post-processing to obtain the data for electronic band structure
- d. Post-processing to plot the band structure

a. Self-Consistent Field Calculation (scf):

- i. Set-up the input file as shown in: [Re_mtd/E_structure_pristine/Band_structure/TiSe2.scf.in](#)
- ii. Then run the calculation *pw.x < TiSe2.scf.in > TiSe2.scf.out*
- iii. Examine the output file, *TiSe2.scf.out*. Try to locate the Fermi energy and the total energy, they are calculated and written by the program. They both appear towards the end of the output file.

b. Non-Self-Consistent Field Calculation (nscf):

- i. Set-up the input file as shown in:

[Re_mtd/E_structure_pristine/Band_structure/TiSe2.band.in](#) This input file can be constructed by copying and pasting the *TiSe2.scf.in* file, followed by making some important modifications. Firstly, in the &control namelist change *calculation* = ‘scf’ to *calculation* = ‘bands’. Secondly, in the &system namelist add *nbnd* and set it equal to the number of bands to calculate. In my case, *nbnd* = 40. See https://www.quantum-espresso.org/Doc/INPUT_PW.html#idm253 for the description of *nbnd* parameter.

Thirdly, the card *K_POINTS* is changed for automatic to crystal. The grids dimension is no longer specified, rather we specify individual k-points along some specific special points within the BZ. In order to generate these k-points, the special points names and positions within the BZ must be known.

- The special points and direction can be known by looking up literature for the band structure plots for the system in consideration. In my case, G, M, K, G, A, L, H, A (most papers on TiSe₂ use these points). The direction is also along these consecutive points.
- Next, we need to know the coordinates of these points in crystal unit. Follow this link

<https://cryst.ehu.es/> Click space-group symmetry, and then KVEC. Choose the space group of the system (TiSe2 is space group number 164, a P-3m1 space group symbol). On the webpage, click optimized listing of k-vector types using ITA description. Now, read the special points/label accordingly. Below, I have attached screenshots of the windows for each step:

bilbao crystallographic server

Contact us

About us

Publications

How to cite the server

Space-group symmetry

KVEC

The k-vector types and Brillouin zones of Space Groups

Please, enter the sequential number of the space group as given in *International Tables for Crystallography*, Vol. A, or [choose it](#):

164

Comparative listing of k-vector types

Optimized listing of k-vector types using ITA description

k-vector identification

k-vector description	
k-vector label	Conventional basis
GM	0,0,0
A	0,0,1/2
K	1/3,1/3,0
H	1/3,1/3,1/2
DT	0,0,u
M	1/2,0,0
L	1/2,0,1/2

- Lastly, we use Xcrysden, a crystal visualization software, to produce the k-vectors along these special points we have obtained. Please see this link <http://www.xcrysden.org/doc/kpath.html> on how to do this. It is straight forward and not complicated. (Hint: First of all, on the terminal, execute the command `xcrysden --pwi TiSe2.scf.in` to launch the crystal structure in Xcrysden. Then, follow the instruction presented in the link above.)

Now, the generated k-points by Xcrysden is stored in the file *kpoints.pwscf*. They can be copied into the *TiSe2.band.in* file.

ii. After constructing the *TiSe2.band.in* file. We run the calculation `pw.x < TiSe2.band.in > TiSe2.band.out`.

c. Post-processing to obtain the data for electronic band structure

i. Set up the input file **`Re_mtd/E_structure_prisistine/Band_structure/bands.in`**

ii. Run the command `bands.x < bands.in > bands.out`

iii. Examine the output file, bands.out. Make sure you pay attention to this opening part of the output file:

```
High-symmetry point: 0.0000 0.0000 0.0000 x coordinate 0.0000
high-symmetry point: 0.5574 0.1490 0.0031 x coordinate 0.5769
high-symmetry point: 0.4712 0.4712 0.0000 x coordinate 0.9105
high-symmetry point: 0.0000 0.0000 0.0000 x coordinate 1.5770
high-symmetry point: -0.0019 0.0019 0.2946 x coordinate 1.8716
high-symmetry point: 0.5552 0.1509 0.2972 x coordinate 2.4482
high-symmetry point: 0.4692 0.4729 0.2946 x coordinate 2.7815
high-symmetry point: -0.0019 0.0019 0.2946 x coordinate 3.4476
```

We will use the x coordinates of these 8 High-symmetry points for our band structure plot. In fact, these 8 x coordinates correspond respectively to the G, M, K, G, A, L, H, A high symmetry points.

d. Post-processing to plot the band structure:

Quantum Espresso has an in-built program for plotting band structure. However, for more flexibility, I recommend using external plotting programs like GNUPLOT, XMGRACE, etc. So far, I have experience in using only GNUPLOT, and so I will give the steps on how to use GNUPLOT in visualizing band structure plots right away.

i. Make a gnuplot input file with the name *TiSe2_band2.gnu* contained in

[Re_mtd/E_structure_pristine/Band_structure/](#)

- When you open this file, the first line, **set terminal** establishes the plot environment and the figure format. In my case, it is **png**. Other terminal types are **eps**, **pdf**, **pngcairo**, etc.
- **set output** is the name of the output file with an extension that is appropriate with the set terminal type.
- **set arrow** are used to draw the lines of the high-symmetry points. I also, used it to draw the Fermi energy line.
- Finally, the last line **plot** takes the input file (**bands.dat.gnu**) containing the data for rendering the band structure. This input file contains two columns. The first is the k-points values. The second is the energy eigen values. Thus, **u 1 : (\$2 - 9.717)** instructs gnuplot to use the first column as the x-axis (k-point axis) and the second column as the y-axis (energy axis). Because we want the fermi energy to be our reference point, the zero mark, we subtract the Fermi energy (9.717) from the eigen values of the second column. Remember that the Fermi energy is written in the **TiSe2.scf.out** output file. The next instruction, **w l lw 2 lc 'black'** means **with line linewidth of 2** and a **line color of black**

ii. After making the gnuplot input file, we go to the terminal by right clicking in the directory [Re_mtd/E_structure_pristine/Band_structure/](#) and then clicking from the options **Open in Terminal**.

iii. On the terminal, type the command *gnuplot TiSe2_band2.gnu* If everything works fine, the band structure figure file would appear in the directory.

3. Calculate the density of states and the projected density of states

There are five steps in making the density of states (DOS) and projected density of states (PDOS) plots.

- a. SCF calculation
- b. Non-SCF calculation
- c. Post-processing to obtain the DOS
- d. Plot DOS
- e. Post-processing to obtain the PDOS

a. SCF calculation

- i. Open the directory **Re_mtd/E_structure_pristine/DOS/**
- ii. Using the same TiSe2.scf.in input file used in the Band structure calculation run
pw.x < TiSe2.scf.in > TiSe2.scf.out

b. Non-SCF calculation

- i. Open the TiSe2.nscf.in file. The content is the same with TiSe2.scf.in except that under &control calculation = 'nscf', and under &system we add the parameter nbnd = 40 (Note: See the Quantum Espresso pw.x input descriptions to understand the nbnd – **number of bands** – parameter)
- ii. Run *pw.x < TiSe2.nscf.in > TiSe2.nscf.out*

c. Post-processing to obtain the DOS

- i. Open the input file TiSe2.dos.in.
 - The prefix should be consistently set to the same file name.
 - DeltaE is the energy increment
 - fildos is the file where the DOS data is stored
 - outdir, like the prefix, should consistently point to the same directory.
 - To see other options of the input file structure, please visit this link https://www.quantum-espresso.org/Doc/INPUT_DOS.html
- ii. Run *dos.x < TiSe2.dos.in > TiSe2.dos.out*. The TiSe2.dos.dat file required for plotting will be produced.

d. Plot DOS.

Like the band structure calculation, DOS is visualized by utilizing GNUPLOT.

- i. Make the gnuplot input file, this is named TiSe2_dos.gnu in the working directory **Re_mtd/E_structure_pristine/DOS/**. The details of constructing this file has been described above for band structure plot.
- ii. Run *gnuplot TiSe2_dos.gnu*.

e. Post-processing to obtain the PDOS

- i. Construct the input file TiSe2.pdos.in.
 - Notice that this file is the same as the TiSe2.dos.in file only with the substitution of fildos by filproj.
 - Please to see the detailed input description of the PDOS, check out this link: https://www.quantum-espresso.org/Doc/INPUT_PROJWFC.html
- ii. Now, run *projwfc.x < TiSe2.pdos.in > TiSe2.pdos.out*. This will produce some output files.
- iii. There are two output files of interest to us:
 - The TiSe2.pdos.dat.lowdin, this gives the lowdin charges of the ions in the crystal. Please visit the following links to learn about Lowdin charges and how to interpret them: https://www.researchgate.net/post/How_to_interpret_Lowdin_Charges_in_Quantum_Espresso
 - To learn about Lowdin charge and spilling parameter: <https://spiral.imperial.ac.uk:8443/bitstream/10044/1/68409/1/Nemytov-V-2019-PhD-Thesis.pdf> (as this is a large thesis file, I would suggest you use specific key terms to search)
 - The other important output file is TiSe2.pdos_atm#1(Ti)_wfc#1(s)
 - This file gives the PDOS for each orbital of each atom in the system/crystal

4. Get the fermi surface plot

- i. Still in the folder **Re_mtd/E_structure_pristine/DOS/** construct the input file TiSe2.FS.in
- ii. Run *fs.x < TiSe2.FS.in > TiSe2.FS.out*. This will produce the TiSe2.FS.out and TiSe2_fs.bxsx output files.

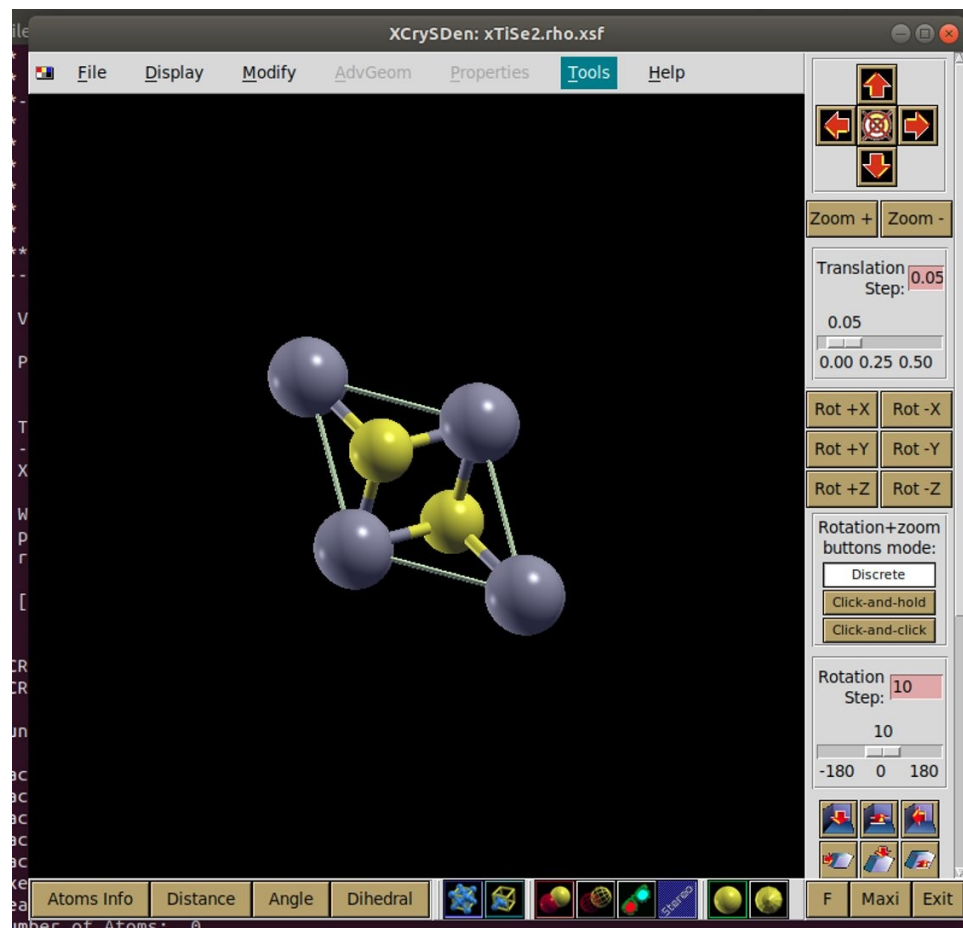
- iii. We will then use Xcrysden to render the plot of the fermi surface. In the terminal, run `xcrysden --bxsf TiSe2_fs.bxsf`
- iv. Check the fermi energy Xcrysden finds from your input file and click ok. Play around with the various menu and options presented by Xcrysden to visualize you Fermi surface plot

5. Charge density plot

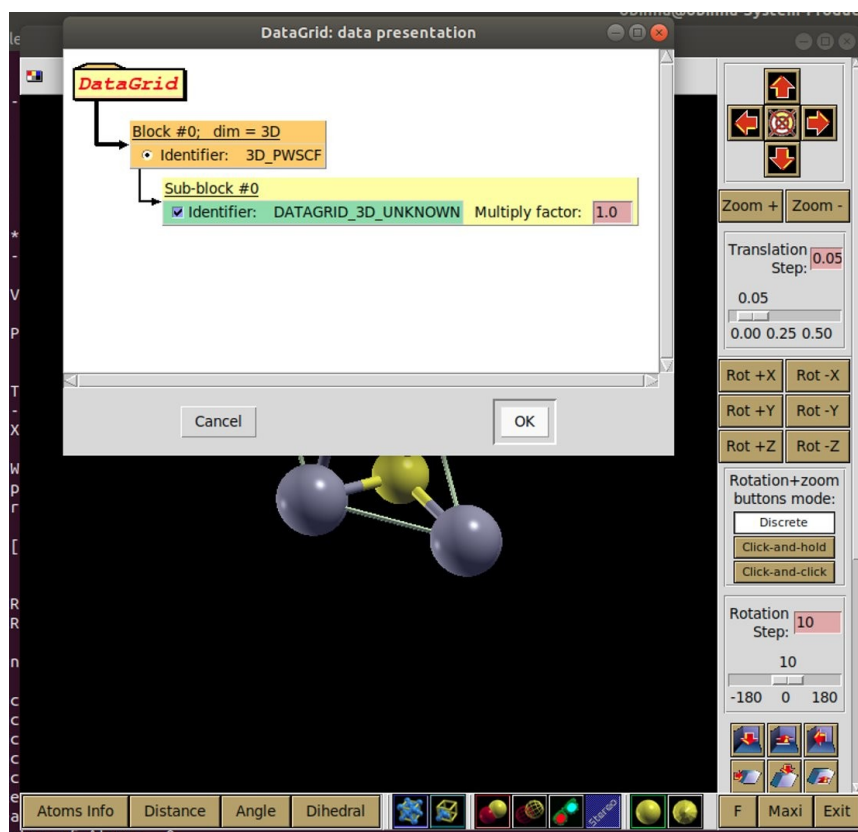
- i. Go to the directory [Re_mtd/E_structure_pristine/Charge_density/](#)
- ii. Construct the input file `pp_rho.in`.
 - Please visit the link https://www.quantum-espresso.org/Doc/INPUT_PP.html to understand the input file description
- iii. Run `pp.x < pp_rho.in > pp_rho.out`
- iv. Use Xcrysden to visualize charge density plot by running on the terminal
 - `xcrysden --xsf xTiSe2.rho.xsf` where `xTiSe2.rho.xsf` is an output file produced from the previous run.
 - By now the xcrysden window should appear, please take a look at this tutorial video on how to use xcrysden to visualize the charge density plot <https://www.youtube.com/watch?v=nbIA1fAJoj4>
 - You can also see the step-to-step procedure by the aid of screenshots.

Charge density plot: Step-to-step process using screenshots.

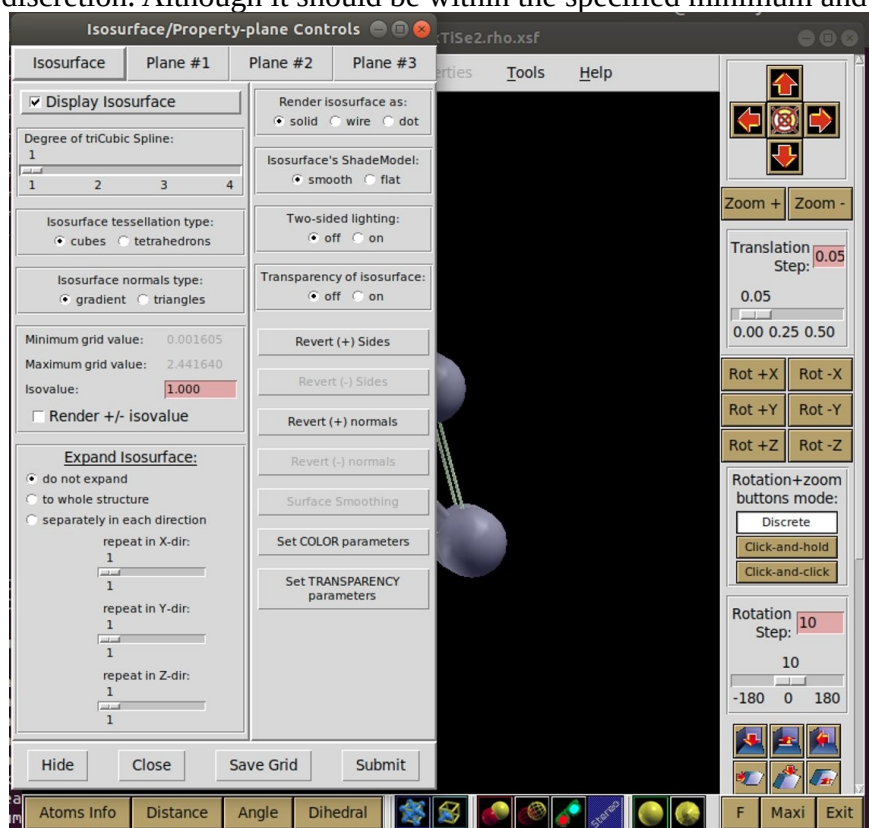
1. On the terminal type `xcrysden --xsf xTiSe2.rho.xsf`. This window will appear.



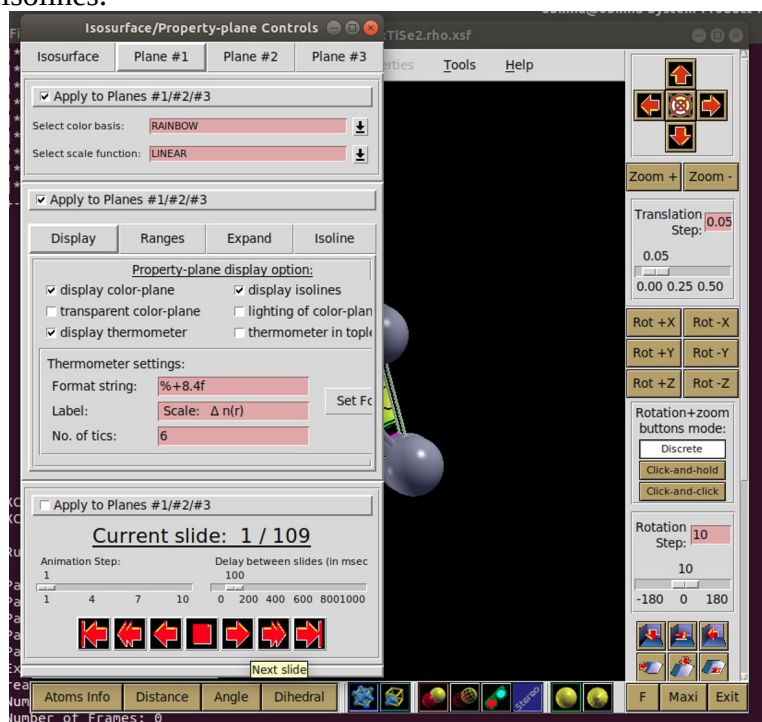
2. The Xcrsyden Taskbar, click Tools and select Data Grid, this window will appear



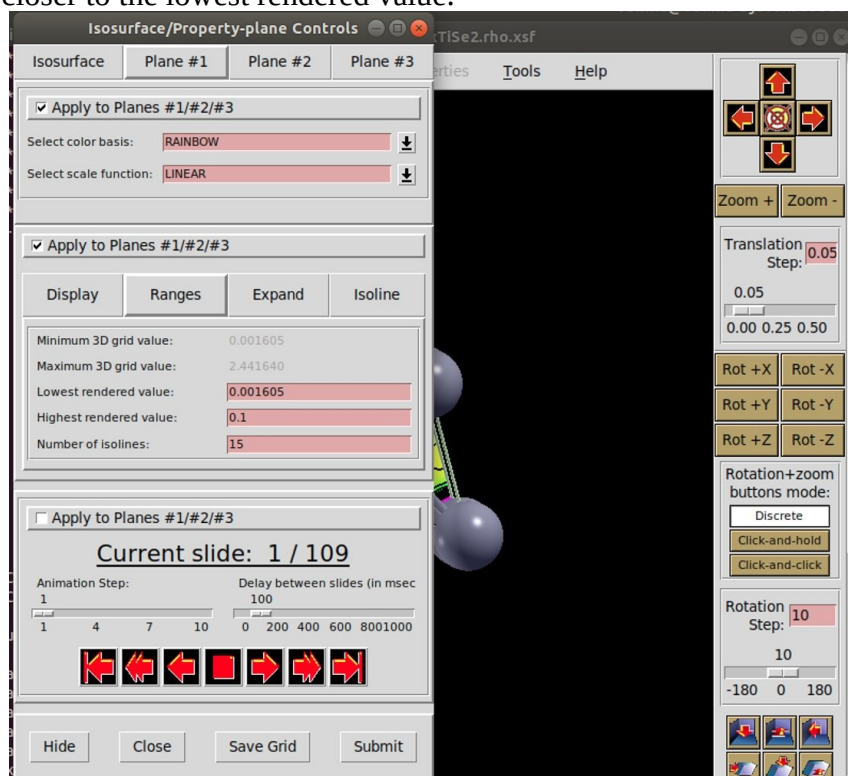
3. Press OK. Next, this window will appear. On this Isosurface panel, I usually stick to the default values, except that one has to input a number for isovalue (I chose 1.00) at one's discretion. Although it should be within the specified minimum and maximum values.



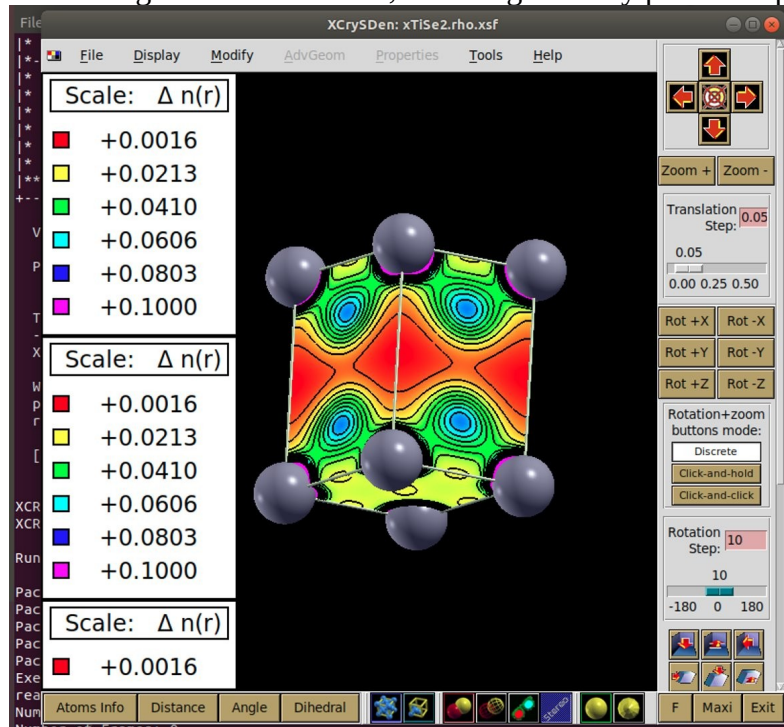
4. We are getting close! Next thing to do is to click Plane #1 on the taskbar, and the window below will appear.
 - Click on the first two Apply to Planes.
 - On the first Apply to Planes, change the color basis from Monochrome to Rainbow.
 - On the second Apply to Planes, tick display color-plane, display thermometer, display isolines.



5. Now, click Ranges and this window appears.
 - The Highest rendered value can sometimes be very large. I usually set this value somewhat closer to the lowest rendered value.



- On clicking the Submit button, the charge density plot will appear.



- Finally, The figure file can simply be exported by: File --> Print Crystal --> Save

DENSITY FUNCTIONAL PERTURBATION THEORY (DFPT)

After the electronic structure calculations, the next thing to do is the phonon calculations and electron-phonon calculations using DFPT. This calculation consists of the following steps:

- SCF calculation for electron-phonon calculation
- SCF calculation for phonon calculation
- Calculation of the dynamical matrices
- Calculation of the Inter-atomic Force Constants and electron-phonon coefficient in real space
- Calculation of the phonon frequencies and plotting
- Calculation of the phonon density of states, the electron-phonon coupling strength, and the Eliashberg spectral function.

Open the directory [Re_mtd/DFPT_pristine/](#)

- SCF calculation for the electron-phonon calculation:
 - Construct the `TiSe2.scf.fit.in` input file. This is the usual `scf.in` input file with the addition of the option `la2F = .true.` in the `&system` namelist. This option is necessary to produce `TiSe2.a2Fsave` file which is later read during the electron-phonon calculation.
 - Now run `pw.x < TiSe2.scf.fit.in > TiSe2.scf.fit.out`
- SCF calculation for phonon calculation
 - Construct the `TiSe2.scf.in` input file. This should be the same with the `TiSe2.scf.fit.in` file without the option `la2F = .true.`
 - Now run `pw.x < TiSe2.scf.in > TiSe2.scf.out`

3. Calculation of the dynamical matrices

- Make the TiSe2.elph.in input file. Example is given below.

#Electron-phonon coefficients for TiSe2

&inputph

```
tr2_ph = 1.0d-14,  
prefix = 'TiSe2',  
fildvscf = 'dvscf',  
amass(1) = 47.867  
amass(2) = 78.971  
outdir = './tmp',  
fildyn = 'TiSe2.dyn',  
electron_phonon='interpolated',  
el_ph_sigma = 0.005,  
el_ph_nsigma = 10,  
trans = .true.,  
ldisp = .true.,  
nq1=2, nq2=2, nq3=2  
/  

```

- The option tr2_ph should be very small, of the order of 10^{-14} or 10^{-18}
- fildvscf must be set equal to dvscf. This is later used by EPW.
- The nqs should be uniform, and be an integer multiple of the nks in the TiSe2.scf.in and TiSe2.scf.fit.in input files
- Run `ph.x < TiSe2.elph.in > TiSe2.elph.out`

4. Calculation of the Inter-atomic Force Constants and electron-phonon coefficient in real space

- Make the file q2r.in
- Run `q2r.x < q2r.in > q2r.out`

5. Calculation of the phonon frequencies and plotting

- Construct the file matdyn.in.freq
- Use the same method in obtaining the k-points in the unit of crystal to obtain the q-points
- Since the q-points are in the unit of crystal is very vital to specify the option `q_in_band_form = .true.`
- Run `matdyn.x < matdyn.in.freq > matdyn.out.freq`
- To plot the phonon frequencies, on the terminal type `plotband.x` and the press enter
- Type `TiSe2_222.freq` for the Input file
- Emin and Emax should be written as it appears on the terminal
- The high-symmetry points and their corresponding x-coordinates is displayed. Make sure the x-coordinates are copied and saved. It will be needed when making the gnuplot input file.
- Next, write the output file name of your choice. This is where the phonon frequencies and q-points would be stored.
- Finally, appropriately edit the gnuplot input file `TiSe2_bands.gnu`. Note that you will need the x-coordinates of the high-symmetry points produced by `plotband.x`
- After which you run `gnuplot TiSe2_bands.gnu`

6. Calculation of the phonon density of states, the electron-phonon coupling strength, and the Eliashberg spectral function

- Make the file `matdyn.in.dos`
- Make the nks large as well as the option `ndos`.
- Run `matdyn.x < matdyn.in.dos > matdyn.out.dos`
- To plot the Phonon DOS from DFPT, you can appropriately edit the `TiSe2_dos2.gnu` file, after which you can run it.

- To plot the Eliashberg spectral function from DFPT, similarly you can edit the `TiSe2_spectra.gnu` file, after which you can run it.

To learn more of the input file options for `ph.x` see

https://www.quantum-espresso.org/Doc/INPUT_PH.html

ELECTRON PHONON WANNIER (EPW) version 5.1.0:

In this section, I will describe how to obtain a more robust superconducting properties of a system using the electron phonon wannier program. The calculation will consist of the following steps.

1. Make a **save** directory where you can gather the `.dyn` and `.dvscf` files created in the DFPT section. You can easily do this by using a python script, the `pp.py`, developed by EPW.

To do this:

- I. Make back-up copy of the `dfpt tmp` directory, name this back-up copy as `tmp2`; this can be done by executing the code:
`cp -rf tmp/ tmp2/`
- II. In the `tmp` directory Copy the `pp.py` into the `tmp` directory of the `dfpt` calculation
- III. Copy the `xxx.dyn0`, `xxx.dyn1`,.....`xxx.dynN` files into the `tmp` directory of the **dfpt** calculation. (xxxx is your prefix name, for my case is it `TiSe2`)
- IV. Run `python pp.py`. You will be asked to enter the prefix of your calculation followed by the number of irreducible q-points. (Note: the number of irreducible q-points is already printed for you in the `TiSe2.elph.out` output file contained in the `dfpt` directory.
- V. Finally, after sometime, the code will produce a **save** directory. Copy and paste this directory to your `epw` directory (in my case `EPW_pristine`).
- VI. If everything goes well, delete the back-up `tmp2` file.

Open the **Re_mtd/EPW_pristine** directory

2. Run a self-consistent calculation

- `pw.x < TiSe2.scf.in > TiSe2.scf.out`

3. Run a non-self-consistent calculation on a 12x12x12 k grid, this file is already present. But to create it by yourself;

- `cp TiSe2.scf.in TiSe2.nscf.in`
- Modify using `vi` or another text editor: set number of bands to `nbnd = 40`, and delete the last two lines specifying the k-point grid.
- Then you can use a simple script provided in the `wannier90` distribution to generate the homogenous k grid, `kmesh.pl`, run this on your terminal:
`yourpath/to/q-e/wannier90-version/utility/kmesh.pl 12 12 12 >> TiSe2.nscf.in`
- You can now run `pw.x < TiSe2.nscf.in > TiSe2.nscf.out`

4. Run `epw` calculation to obtain superconducting properties: `epw.x < epw_a2f_in > epw_a2f_out`

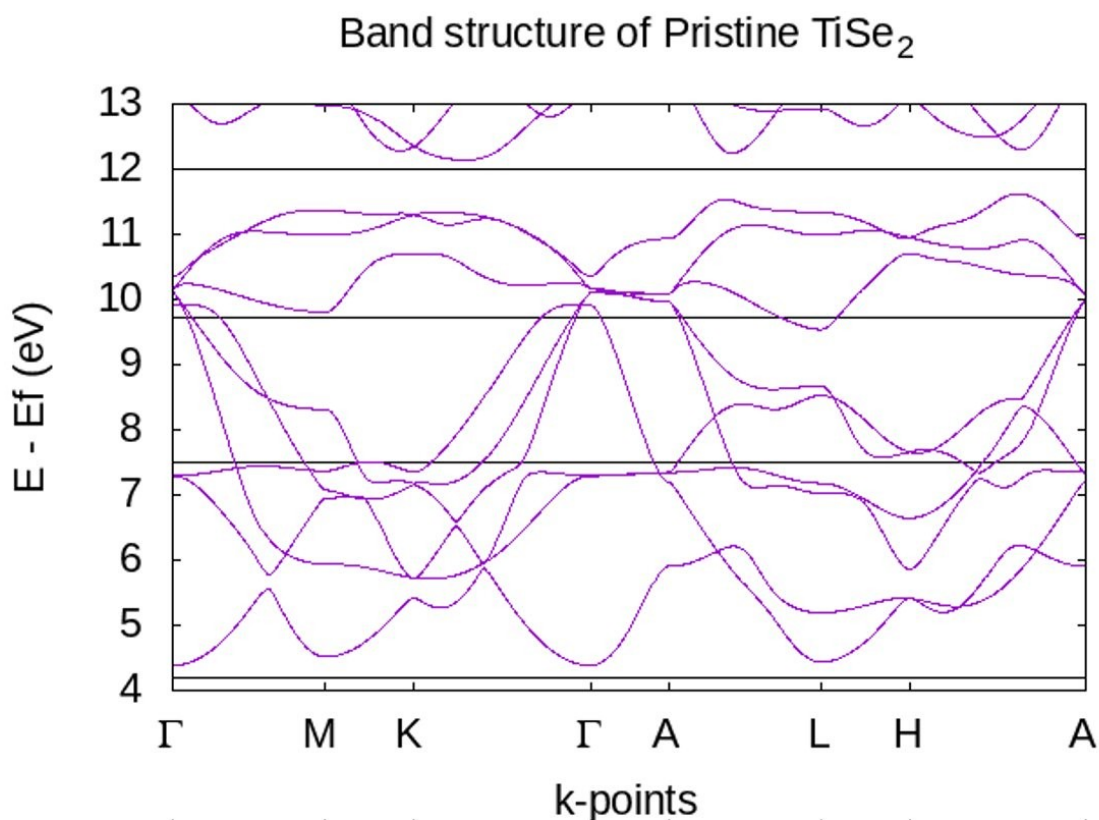
5. Run `epw` to obtain the phonon plots: `epw.x < epw_phfreq.in > epw_phfreq.out`

6. The output files together with plots of `a2f`, phonon DOS, phonon dispersion, wannier plots, are contained in the folder **Re_mtd/EPW_pristine**.

HOW TO CONSTRUCT THE EPW INPUT FILES:

- The input file for my `epw` calculation is written in `epw_a2f_in`
- I recommend you do the exercise on lead to properly understand the parameters in the input file. The exercise for the lead calculation is contained in the directory `lead`. The pdf file that is contained in the `lead` directory titled “Electron-phonon coupling with EPW”, would guide you in doing the lead calculation.

- You would need to always refer to the meaning of input parameters by looking them up on the EPW webpage: <https://docs.epw-code.org/doc/Inputs.html>
- For my own TiSe₂ calculation. I will give a brief summary of some input parameters:
 - I. nbndsub is the number of wannier functions to utilize, and this is exactly equal to the number of projecting wavefunctions specified as proj(1), proj(2)....proj(6) as seen in the input file.
 - II. nbndskip represents the number of bands lying below the disentanglement window. You can determine this number from your plot of electronic band structure. Make a plot of the band structure to see low lying bands down to the ground state. You may not need to specify this number if you are using EPW v5.3
 - III. How to choose the disentanglement window:
 - dis_win_min and dis_win_max represent the minimum and maximum energies of the outer window. It contains the states for which we want to exactly construct and other states that are mixed (entangled) with those states we want to exactly construct. Therefore, it contains the inner window.
 - dis_froz_min and dis_froz_max represent the minimum and maximum energies of the inner window. It contains the states for which we want to exactly construct. Therefore, it is contained in the outer window.
 - Tuning these parameters are relevant in obtaining a nice construction of the DFT electronic structure.



STEPS:

- From your DFT electronic structure (about 9.7 eV is the Fermi energy level), select the states you wish to construct.
- For my pristine TiSe₂ electronic structure, I wish to construct 6 states between the energy window 7.5 eV and 12.0 eV.
- In the DFT TiSe₂ E-structure, notice that there is no clear distinction between the states that lie above 7.5 eV and states that lie below it. Therefore, there is a mixing of states of our desired states and some states below 7.5 eV down to about 4.2 eV.
- Hence, set `dis_froz_min` = 7.5 and `dis_froz_max` = 12.0 as the inner window, since it contains the states you wish to exactly reproduce.
- Then set `dis_win_min` = 4.2 and `dis_win_max` = 12.0 as the outer window, since it contains states we wish to construct and some states we wish to disentangle from our states of choice.

IV. Next we need to specify the states/orbitals used for wannierization. In my TiSe₂ calculation, I used 6 states. They are three d-orbitals positioned at Ti, and three p-orbitals positioned at one of the Se atoms. This specified as proj(1), proj(2),..., proj(6).

V. To write data needed to obtain the wannier electronic band structure, see `wdata(1)` to `wdata(11)`, we specify `bands_plot = true` and state the `kpoint_path` of interest. Note: This has already been obtained initially with the help of Xcrysden.

VI. Lastly, since we are interested in superconductivity, we set `phononselfen = .true.` and `a2f = .true.`

```
&inputepw  
    prefix      = 'TiSe2',  
    amass(1)    = 47.867,  
    amass(2)    = 78.971,  
    outdir      = './'  
  
    elph        = .true.  
    kmaps       = .false.  
    epbwrite    = .true.  
    epbread     = .false.  
    epwwrite    = .true.  
    epwread     = .false.  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!RESTART!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!elph          = .true.  
!kmaps         = .true.  
!epbwrite      = .false.  
!epbread       = .false.  
!epwwrite     = .false.  
!epwread      = .true.  
!wannierize   = .false.  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
etf_mem        = 1  
  
nbnbsub        = 6,  
nbndskip       = 20,
```



```

wannierize = .true.
num_iter   = 500
dis_win_min = 4.2
dis_win_max = 12
dis_froz_min = 7.5
dis_froz_max = 12
proj(1)    = 'Ti:dxy'
proj(2)    = 'Ti:dxz'
proj(3)    = 'Ti:dyz'
proj(4)    = 'f=0.333899797,0.666100202,0.250208591:px'
proj(5)    = 'f=0.333899797,0.666100202,0.250208591:py'
proj(6)    = 'f=0.333899797,0.666100202,0.250208591:pz'
wdata(1)   = 'bands_plot = .true.'
wdata(2)   = 'begin kpoint_path'
wdata(3)   = 'G 0.0000 0.0000 0.0000 M 0.5000 0.0000 0.0000'
wdata(4)   = 'M 0.5000 0.0000 0.0000 K 0.3333 0.3333 0.0000'
wdata(5)   = 'K 0.3333 0.3333 0.0000 G 0.0000 0.0000 0.0000'
wdata(6)   = 'G 0.0000 0.0000 0.0000 A 0.0000 0.0000 0.5000'
wdata(7)   = 'A 0.0000 0.0000 0.5000 L 0.5000 0.0000 0.5000'
wdata(8)   = 'L 0.5000 0.0000 0.5000 H 0.3333 0.3333 0.5000'
wdata(9)   = 'H 0.3333 0.3333 0.5000 A 0.0000 0.0000 0.5000'
wdata(10)  = 'end kpoint_path'
wdata(11)  = 'bands_plot_format = gnuplot'
wdata(12)  = 'use_ws_distance = .true.'
wdata(13)  = 'write_hr = .true.'

```

```

wdata(13) = 'write_hr = .true.'
wdata(14) = 'write_hr_diag = .true.'

```

```

iverbosity = 0
elecselfen = .false.
phonselfen = .true.
delta_approx = .true.

```

```

max_memlt = 100 ! Maximum memory that can be allocated per pool
!eps_acustic = 0.0 ! Lowest boundary for the phonon frequency in el-ph and a2f calculations in [cm-1]
!ephwrite = .true. ! Writes .ephmat files used when eliasberg = .true.

```

```

fsthick = 4 ! eV
eptemp = 300 ! K
degaussw = 0.1 ! Smearing in the energy-conserving delta functions in [eV]

```

```

degaussq = 0.05 ! meV Smearing for sum over q in the e-ph coupling
delta_qsmear = 0.05 ! meV Change in the energy for each additional smearing in the a2f in [meV]
!nqstep = 500 ! Number of steps used to calculate the a2f

```

```

a2f = .true.

```

```

dvscf_dir = './save/'

```

BAND UNFOLDING:

In this section, I will give you the steps in unfolding the electronic band structure of a system in its supercell k-space to its primitive cell k-space.

The theory can be found in these papers:

<https://journals.aps.org/prb/abstract/10.1103/PhysRevB.89.041407>

<https://journals.aps.org/prb/abstract/10.1103/PhysRevB.91.041116>

The band unfolding package called BandUP can be downloaded and installed from:

<https://github.com/band-unfolding/bandup>

I suggest that after installing the BandUp package, do not rename the files or folders in the Quantum Espresso tutorial folder of BandUp (see the directory bandup-master in Band_Unfolding folder).

You may copy and paste the contents of your specific work into the input files but do not rename.

I would suggest you do the example for bulk Si and reproduce the results. Follow the instructions in:

https://github.com/band-unfolding/bandup/tree/master/tutorial/Quantum_ESPRESSO/example_2_bulk_Si

After you are done with this, you can do the band unfolding for a pristine 2x2x2 TiSe2 supercell.

Band unfolding for a pristine 2x2x2 TiSe2 supercell.

Open **Re_mtd/Band_Unfolding/example_2_bulk_Si_TiSe2_2x2x2** directory

Although my calculations are for a 2x2x2 TiSe2 supercell, the labelling of the files and prefixes have Silicon in them. Don't worry about this, this calculation is not for Silicon.

NOTE: In step1 and step3, please ensure that the job.sh is correctly modified to suite your environment. For example, set:

1. `pwscf=/home/username/q-e-qe-5.4/bin/pw.x`, username means your machine's user ID
2. `ESPRESSO_TMPDIR = "/outdir"`
3. `ESPRESSO_PSEUDO='pwd`/'path_to_your_pseudopotentialfolder'`
4. `mpirun -np 4 $pwscf -input bulk_Si_pwscf.in > pwscf.out`
You can change the number of processors you want mpirun to use

Step 1: Wave function calculation/getting converged charge density

- Inspect the input file "bulk_Si_pwscf.in": Although the title and prefix have Silicon in them, this calculation is not for Silicon
- Set outdir = "/outdir"
- Set pseudo = ../../pseudo
- Inspect the script file "job.sh": here set outdir and pseudo as in "bulk_Si_pwscf.in", also modify the path to Quantum Espresso pw.x
- Run the "job.sh" script file, simply by typing: `./job.sh`
- You should obtain the output file, pwscf.out.

NOTE: Make another copy of the completed step 1 folder.

Step 2: Create k-points

- Use a dense grid k-points: To do this modify the "KPOINTS_prim_cell.in" file by increasing the k grid in the second line of this file.
- "KPOINTS_prim_cell.in", "prim_cell_lattice.in" and "supercell_lattice.in" file demand some careful generation, see table below:

KPOINTS_prim_cell_in	prim_cell_lattice.in	Sup_cell_lattice.in
2 nd line, use dense grid (300 300 300 is great)	2 nd line gives lattice const. In Angstrom. Can be found by inspecting the first few lines of “pwsf.out” in step1 (only unit conversion from bohr to Angstrom is required)	2 nd line is the same value as prim_cell_lattice.in
The rest lines below are the k-path for the prim. Cell. This was gotten from xcrysden (see how to do this in the DFT section above)	3 rd to 5 th lines give the cell dimension. To get this, open the “pwsf.out” in step 1 and divide by the supercell factor.	3 rd to 5 th lines give the cell dimension. Exactly the same values with those found in the first few lines of “pwsf.out”

- Run the script file: `./run_BandUP_get_sckpts_to_be_unfolded.sh`
- Copy the kpoints generated in the KPOINT_supercell.out file to be used in step 3

Step 3: Band calculation

- Inspect the input file “bulk_Si_pwsf_bands.in”
- Paste the kpoints obtained from Step 2 in the “bulk_Si_pwsf_bands.in” file
- Inspect the “job.sh” file and make sure that the outdir and pseudo is set to same path as in step 1
- Run the script: `./job.sh`

Step 4: Unfolding and plotting

- Inspect the script file
 - Is emin, emax, and dE values appropriate for your calculation?
- From the insight of already done DFT calculation for prisitne TiSe2, I use emin = -6 and emax = 3. efermi = 7.5542 was obtained from the pwsf.out file of Step 1. For clear plot, I recommend dE = 0.05 (you may want to play around with this parameter to get a better plot).
- Run the script file: `./run_BandUP_unfold_and_plot_tasks.sh`
 - If you are using the cluster, you will get an error message relating to the inability to visualize the plot.
 - Don’t worry about this, the important file “unfolded_EBS_symmetry-averaged.dat” has already been reproduced.
 - Copy all of the contents of Step 4 in the cluster and paste them on your computer/workstation for visualization, into a folder Postprocess_Plot. This Postprocess_Plot should be empty and should only contain the script, run_plot_tasks.sh
 - Now run the script: `./run_plot_tasks.sh`.
 - The unfolded plot is stored in the folder named plot

NOTE: dE is very important for the sharpness of the plot
cmap and vmax should be controlled adequately to give a fine figure.
For cmap options see the link:

https://matplotlib.org/examples/color/colormaps_reference.html