

Hands-On Session Day 1: Quantum Espresso and Wannier90

Arash Mostofi and Jonathan Yates

May 15, 2022

Foreword

This document contains exercises to help you learn **wannier90** (W90) - the computer program that calculates maximally-localised Wannier functions (MLWFs).

This document is based on the Quantum-Espresso and Wannier90 Tutorial by Junfeng Qiao, Giovanni Pizzi and Nicola Marzari for the 2021 Virtual School on Electron-Phonon Physics and the EPW code, which can be found at https://github.com/wannier-developers/wannier-tutorials/tree/master/2021_06_EPW.

Contents

Exercise 1 Silicon valence bands	3
Exercise 2 Silicon valence and conduction bands	8
Exercise 2.1 Further explorations for the curious	9
Exercise 3 Lead Fermi surface and band structure	10
Exercise 4 BaTiO3 ferroelectricity	14
Exercise 5 bcc Fe: spin polarised	17
Exercise 6 bcc Fe: iron – Spin-orbit-coupled bands	18
Exercise 7 FeSe: A final Challenge	19

Code path and tutorial files

All codes (including `wannier90.x`, `pw.x`, `bands.x`, etc.) are in the `/media/ictpuser/AiiDA/bin` directory. Your `$PATH` environment variable should already include this directory so these codes should be available to you in your terminal without specifying the full path.

The tutorial files are available on github at

https://github.com/wannier-developers/wannier-tutorials/tree/master/2022_05_Trieste

and also at

https://github.com/wannier-developers/wannier-tutorials/tree/master/2022_05_Trieste.

Please copy the tutorial files into your own directory.

Tutorial Videos

Walk-through videos for some of these tutorials can be found at

<https://youtube.com/playlist?list=PLHxLj3bAKS6ybfzwxHrgxhig1mFC09otT>

Software for visualization

Visualisation and data analysis software (`gnuplot`, `xmgrace`, `VESTA`, `xcrysden`, `FermiSurfer` etc.) are also available in `/media/ictpuser/AiiDA/bin`. We recommend that you run these through the graphical user interface (GUI) of the virtual machine.

Links to these software packages are given below:

- To plot band structures:
 - `gnuplot` <http://www.gnuplot.info/index.html>
 - `xmgrace` <https://plasma-gate.weizmann.ac.il/Grace/>
- To plot the Wannier functions, Fermi surfaces:
 - `xcrysden` <http://www.xcrysden.org/>
 - `VESTA` <https://jp-minerals.org/vesta/en/>
 - `FermiSurfer` <https://fermisurfer.osdn.jp/>

Exercise 1 Silicon valence bands

In this exercise you will learn how to obtain maximally-localised Wannier functions (MLWFs) for the valence bands of silicon.

- Go to the `ex1` folder and inspect the input file `01_scf.in`. The first step is to perform a ground-state calculation for a silicon crystal (FCC) with two atoms per unit cell. Check if you understand all parameters (you can use the web page https://www.quantum-espresso.org/Doc/INPUT_PW.html for keywords that you do not know or ask us). To visualize the crystal structure, there are many options, for example,
 - Quantum ESPRESSO input generator and structure visualizer: open the following link in a browser, <https://www.materialscloud.org/work/tools/qeinputgenerator>, click Choose File button, select the file `01_scf.in`, and click "Generate the PWscf input file" button. In the new webpage you find a 3D visualization of the structure.
 - `xcrysden` open the program, select from the menu `File`→`Open PWscf...`→`Open PWscf Input File` and select the input file `01_scf.in`.

```
&control
  calculation      = 'scf'
  restart_mode     = 'from_scratch'
  prefix           = 'si'
  pseudo_dir       = 'pseudo/'
  outdir           = 'out/'
/
&system
  ibrav            = 0
  nat              = 2
  ntyp             = 1
  ecutwfc          = 25.0
  ecutrho          = 200.0
/
&electrons
  conv_thr         = 1.0d-10
/
ATOMIC_SPECIES
Si 28. Si.pbe-n-van.UPF
ATOMIC_POSITIONS crystal
Si -0.25 0.75 -0.25
Si 0.00 0.00 0.00
K_POINTS automatic
10 10 10 0 0 0
CELL_PARAMETERS bohr
-5.1 0.0 5.1
0.0 5.1 5.1
-5.1 5.1 0.0
```

- The Si pseudopotential that we will use for the calculation has four valence electrons ($Z_{\text{val}} = 4$) (this information can be obtained reading the first lines of the pseudopotential file, for instance with the command `less pseudo/Si.pbe-n-van.UPF`). Using the information given above, and knowing that FCC Si is a semiconductor, how many occupied valence bands do you expect (and why)? _____
- Run the ground state calculation using the `pw.x` code of the Quantum Espresso suite, which you can find in the following directory: `/media/ictpuser/AiiDA/bin`. In this directory you will also find all the other executables required for this tutorial unless otherwise specified. The syntax for codes in the Quantum Espresso suite is: `command < inputfile > outputfile` (i.e. `pw.x < 01_scf.in > scf.out`). You may want to use parallelization to run the simulation faster (up to 4 processors permitted on the virtual machine), e.g., for 4 processors: `mpirun -np 4 pw.x < 01_scf.in > scf.out`
- After the calculation finishes, inspect the output file `scf.out` to check if there are any errors/warnings. Check that the number of electrons and valence bands reported is what you expect.

- Now we want to plot the band structure of silicon from quantum-espresso (we will use this plot also for the next exercise, where we need also the conduction bands; therefore, we plot also a few of the lowest conduction bands). Copy the file `01_scf.in` to the file `02_bands.in`. Make the following modifications to the file `02_bands.in` (use the INPUT_PW documentation of Quantum Espresso mentioned above for an explanation of the meaning of the input parameters, if needed):

- In the **CONTROL** section (also known as a “namelist”), change the **calculation** keyword from ‘**scf**’ to ‘**bands**’ to perform a band structure calculation starting from the ground state density obtained from the ground state calculation you performed earlier.
- Ask the code to compute 12 bands (add the input parameter **nbnd=12** to the **SYSTEM** namelist).
- Set **diago_full_acc = .true.** in the **ELECTRONS** namelist (see documentation for the meaning of this flag).
- Change the sampling of the Brillouin zone (BZ) to compute the eigenstates and eigenenergies of the Hamiltonian (which determine the electronic bandstructure) along the following path in the BZ, where the coordinates are given in the basis of the reciprocal lattice vectors:
 - * $L(0.5, 0.5, 0.5) \rightarrow \Gamma(0, 0, 0)$
 - * $\Gamma(0, 0, 0) \rightarrow X(0.5, 0, 0.5)$

You can do this using by replacing the **K_POINTS** section of the input file with:

```
K_POINTS crystal_b
3
0.5 0.5 0.5 50
0.0 0.0 0.0 50
0.5 0.0 0.5 50
```

where the number 3 denotes the number of vertices in our path (L, Γ , and X), and the number 50 denotes the number of k-points the code should calculate along each segment of the path.

- Run the calculation using the **pw.x** code, as explained before.
- When the calculation has finished, run the **bands.x** executable on the file `03_bandsx.in` (be sure to read and understand the input file):
`bands.x < 03_bandsx.in > bandsx.out`

```
&bands
prefix = 'si'
outdir = 'out/'
filband = 'bands.dat'
lsym = .false.
/
```

This will produce the file **bands.dat**.

- Finally, execute the **plotband.x** code (interactively) and answer the questions it prompts on the command line. In particular, the input file is the **bands.dat** file created in the previous step; call the xmgrace file **qebands.agr**. When prompted, call the postscript (ps) file **qebands.ps**. You will be asked to provide the value of the Fermi level, which in this case can be set equal to the highest occupied energy level (which you can find from the QE output file **scf.out**). When asked for the deltaE and reference E for the energy axis, type 2 and the value of the Fermi level (use a space to separate the two numbers), you can also tweak these two numbers to adjust the visual output of the figure. At the end, open the xmgrace file (or directly the postscript file with, eg, an application such as **evince**) and inspect the band structure, identifying the valence and conduction bands.
- Now we are ready to calculate the wavefunctions on a complete grid of k-points in the BZ. We need these to compute the $\mathbf{M}^{(\mathbf{k}, \mathbf{b})}$ and $\mathbf{A}^{(\mathbf{k})}$ matrices later. We will do this by conducting a non-self-consistent (nscf) calculation using the electronic ground state density obtained in the earlier self-consistent (scf) calculation in the first step of this example. Copy the file `02_bands.in` that you created before to `05_nscf.in`, and modify the following:

- Change the calculation type from 'bands' to 'nscf'.
 - Change the number of bands from 12 back to the number of valence bands that you expect (see your answer a few lines above: the answer should be 4), since for this exercise we need only the valence bands.
 - Change the k-point list to a full $4 \times 4 \times 4$ Monkhorst-Pack mesh, that will be used to calculate the overlap matrices needed to obtain Wannier functions. To obtain the list of k-points, use the `kmesh.pl` utility in the utility folder of the Wannier90 code, that you can find at:
`/media/ictpuser/AiiDA/codes/wannier90/utility/` using the following command for a $4 \times 4 \times 4$ mesh:
`/media/ictpuser/AiiDA/codes/wannier90/utility/kmesh.pl 4 4 4`
 (use the command without parameters to get an explanation of its usage).
 - Run the nscf calculation using the `pw.x` code:
`pw.x < 05_nscf.in > nscf.out`
- Now we have to prepare the input file for Wannier90. Open the file `ex1.win`, which is a template of the Wannier90 input file (note that Wannier90 input file must have the extension `.win`). Change the values marked with XXX, inserting the correct values. In particular:
 - Insert the value of `num_bands` (this must be equal to the value of `nbnd` set in the nscf calculation).
 - Insert the value of `num_wann` (this is the number of Wannier functions required. In this case, for the valence manifold of silicon, it should be equal to the value of `num_bands`).
 - Set the value of `mp_grid` to `4 4 4` (since we are using a $4 \times 4 \times 4$ k-mesh).
 - Insert, between the `begin kpoints` and `end kpoints` lines, the list of the 64 k-points, one per line. Note that while `pw.x` requires four numbers per line (the three coordinates of each k-point, and the weight), Wannier90 needs only three numbers (the three coordinates). To obtain these lines, use again the `kmesh.pl` utility, but this time specifying a fourth parameter to get the list in the format required for the Wannier90 input file:
`/media/ictpuser/AiiDA/codes/wannier90/utility/kmesh.pl 4 4 4 wan`
Note Using the `kmesh.pl` utility, ensures that we provide enough significant digits, and that the list of k-points given to `pw.x` and to Wannier90 is the same.
 - Inspect the remaining part of the input file, using the Wannier90 user guide (that can be found at https://github.com/wannier-developers/wannier90/raw/v3.1.0/doc/compiled_docs/user_guide.pdf) for a description of any input parameters that you do not understand. Try to understand, in particular, the `projections` section. Can you say where the four s-like projection orbitals are located with respect to the Si atoms? _____

```

use_ws_distance = .true.

num_bands      =   XXX
num_wann       =   XXX
num_iter       =  100

iprint         =    2
num_dump_cycles =  10
num_print_cycles = 10

!! To plot the WFs
! restart              = plot
wannier_plot          = true
wannier_plot_supercell =  3
! wannier_plot_list    = 1,5

!! To plot the WF interpolated bandstructure
bands_plot           = true
begin kpoint_path
L 0.50000 0.50000 0.50000 G 0.00000 0.00000 0.00000
G 0.00000 0.00000 0.00000 X 0.50000 0.00000 0.50000
end kpoint_path

```

```

!! !! Bond-centred s-orbitals
begin projections
f=-0.125,-0.125, 0.375:s
f= 0.375,-0.125,-0.125:s
f=-0.125, 0.375,-0.125:s
f=-0.125,-0.125,-0.125:s
end projections

begin atoms_frac
Si -0.25 0.75 -0.25
Si 0.00 0.00 0.00
end atoms_frac

begin unit_cell_cart
bohr
-5.10 0.00 5.10
0.00 5.10 5.10
-5.10 5.10 0.00
end unit_cell_cart

mp_grid = XXX XXX XXX
begin kpoints
XXX
end kpoints

```

- Finally, we are ready to perform a Wannier90 calculation. This is done in three steps:

1. We first run a pre-processing (“pp”) step using the command: `wannier90.x -pp ex1` which produces an output file `ex1.wout` and another file `ex1.nnkp`, which contains relevant information from Wannier90 in a format suitable for being read by QE in the next step for computing various quantities required by Wannier90.
2. Then we run the `pw2wannier90.x` code (of the Quantum ESPRESSO distribution). The input file for `pw2wannier90.x` is provided (file `06_pw2wan.in`). Here we ask the code to calculate the overlap matrices $\mathbf{M}^{(\mathbf{k},\mathbf{b})}$ (that will be written in the `ex1.mmn` file) and the $\mathbf{A}^{(\mathbf{k})}$ matrices (file `ex1.amn`). Since we want to plot the Wannier functions in real space, we need also the $u_{nk}(r)$ – the periodic parts of the Bloch wavefunctions on a real-space grid. We thus also set the `write_unk` flag in `06_pw2wan.in`, that will produce a set of files with names `UNK00001.1`, `UNK00002.1`, ... Finally, the code will also produce a file `ex1.eig`, with the eigenvalues on the initial $4 \times 4 \times 4$ k-grid (**Note**: this is not needed to obtain the MLWFs of an insulator, but only for the interpolation and band plotting routines). Note that `pw2wannier90.x` expects to find the `ex1.nnkp` file produced in the previous step. Run the code using


```
pw2wannier90.x < 06_pw2wan.in > pw2wan.out
```

```

&inputpp
outdir      = 'out/'
prefix      = 'si'
seedname    = 'ex1'
write_amn   = .true.
write_mmn   = .true.
write_unk   = .true.
/

```

3. Finally we can run Wannier90 to obtain MLWFs. Execute


```
wannier90.x ex1
```

 and, when it finishes, inspect the output file, called `ex1.wout`.
 - Check lines containing `<-- DLTA` to check for the convergence of the spread during the iterations.
 - Check the lines after the string `Final state`: you find the centers and spreads of the maximally-localised Wannier functions.

- To check the quality of the obtained MLWFs, you typically need to:
 - * *Compare the Wannier-interpolated band structure with the ab initio one.* The provided **Wannier90** input file computes the interpolated band structure (files **ex1_band.dat**, and **ex1_band.gnu**) and you can carry out the comparison with the ab initio one obtained in the earlier steps of this tutorial (**qebands.agr**).
 - To plot with **gnuplot**: run **gnuplot** in terminal, and in **gnuplot**, type

```
set xtics nomirr
set x2tics
set xrange [:*] noextend
set x2range [:*] noextend
plot 'qebands.agr' w l, 'ex1_band.dat' axes x2y1 w l
```

Note you can reuse the script in following exercises when comparing band structures, by replacing the file names **qebands.agr** and **ex1_band.dat**.

- Or plot with **xmgrace**: in terminal, type


```
xmgrace qebands.agr ex1_band.dat
```

Note that you may need to rescale the x axis.

Note that the **Wannier90** code also outputs in the **ex1_band.kpt** file a list of the k-points used for the interpolation, that could be used to plot the band structure on the same grid.

- * *Plot the real-space Wannier functions and check if they are real:* if you ask **Wannier90** to plot the Wannier functions, it will print also the ratio of the imaginary and real part of each of them at the end of the **ex1.wout** file: check that the value is small to verify that the MLWFs are indeed real.
- Plot one of the Wannier functions, which are output in files **ex1_00001.xsf**, To visualize the Wannier functions, you can use either **VESTA** or **xcrysden** (which are both installed on the virtual machine):
 - * Open the **xcrysden** application; open the **xsf** file; then choose **Tools**→**Data Grid**→**OK**, and choose a reasonable **isovalue**, activate the **Render +/- isovalue** flag, and press **Submit**.
 - * Open the **VESTA** application; open the **xsf** file; **VESTA** automatically selects an **isovalue**.

Exercise 2 Silicon valence and conduction bands

While in Exercise 1 we obtained a Wannier-interpolated band structure for the occupied bands only, now we will see how to obtain an interpolated band structure for the occupied and first few conduction bands.

- Copy the `ex1` folder to a new folder called `ex2`. The first step, i.e., the self-consistent electronic ground state calculation (`01_scf.in`), is identical to Exercise 1. Hence, if you copied also the `out` directory, you don't need to rerun it.
- In the `05_nscf.in` file, change the value of `nbnd` to 12 to calculate the eigenenergies and wavefunctions for 12 bands, and run the `nscf` calculation (using `pw.x` as in Exercise 1).
- Rename `ex1.win` to `ex2.win`, and modify the following parameters:
 - Change `num_bands` to 12 to be consistent with the new `nscf` calculation.
 - Change the projections to 4 sp^3 orbitals for each Si atom in the unit cell. To do this, the projections section should read:


```
begin projections
Si:sp3
end projections
```
 - Change the `num_wann` parameter to the correct number of Wannier functions: how many do we want, according to the number of projections given above? _____
 - Set the maximum energy for the frozen window (by adding the input parameter `dis_froz_max`) to sit inside the energy gap (use the band structure plot obtained in Exercise 1 to choose a suitable value).
 - Set the maximum for the energy window to be used in the disentanglement procedure (by adding the input parameter `dis_win_max`) to an energy large enough so as to contain enough bands for each k-point; 17.0 eV should be a reasonable value (check where this value lies in the band structure plot of Exercise 1).
- Inside the file `06_pw2wan.in`, change `seedname` to '`ex2`' to reflect the new name of the `.win` file.
- Run `wannier90.x -pp ex2`
- Run `pw2wannier90.x` using `06_pw2wan.in` as the input file.
- Run `wannier90.x ex2`
- Check the output (`ex2.wout`):
 - Before the start of the Wannierization iterations, there is a new section (containing the string `<--DIS`) with the iterations of the disentanglement procedure. It is important that at the end of this section the convergence is achieved (with a string `<<< Disentanglement convergence criteria satisfied >>>`).
 - A practical note: Especially when using disentanglement, it is possible that the disentanglement convergence is not achieved, and/or that the obtained Wannier functions are not real, and/or that the interpolated band structure differs significantly from the ab initio one within the frozen window. Then, you need to change/tune the number of Wannier functions, the projections you chose and/or the energy values for the frozen and disentanglement windows, until you get “good” Wannier functions.
- Check the final WF centers and verify that WFs are real
- Plot the Wannier functions
- Compare the Wannier-interpolated band structure with the ab initio one (obtained in Exercise 1).

Exercise 2.1 Further explorations for the curious

- Do the symmetry and the centres of the Wannier functions agree with your intuition? (We would like 4 sp^3 -like WFs centered on each Si atom, with similar spreads). Try to rerun everything with a $6 \times 6 \times 6$ k-point grid for the `nscf` and `Wannier90` steps to check whether the results improve, and how the spreads change with respect to the grid density.
- Investigate the effect of changing the orientation of the sp^3 orbitals used for the initial projections. Consult the Wannier90 user guide for how to do this. You should be able to find two different solutions for the MLWFs, one in which the lobes of the atom-centred sp^3 -like WFs lie along the direction of the Si-Si bonds, and one in which they lie towards the interstitial region in the opposite direction to the nearest-neighbour Si-Si bonds. Which has the smaller spread?
- Each time you change the initial projections in the `.win` file, you must go through the process of generating a new $\mathbf{A}^{(\mathbf{k})}$ matrix. There is an input parameter (`select_projections`) that allows you to specify more initial projections than you need and, hence, create an $\mathbf{A}^{(\mathbf{k})}$ matrix that is larger than required, and to then select a subset of these for the Wannier90 calculation. See the Wannier90 user guide for a description of this parameter and try to use it to be able to do Exercises 1 and 2 by just generating a single $\mathbf{A}^{(\mathbf{k})}$ matrix.

Exercise 3 Lead Fermi surface and band structure

In this exercise we will see how to interpolate the band structure of lead, in particular around the Fermi energy. The first goal is to obtain the Fermi surface from Wannier interpolation. This will clearly show some of the advantages of using Wannier-interpolation schemes with respect to full k-points direct calculations. In order to build a MLWFs model that describes the band structure around the Fermi energy, we need to have an idea of the orbital character of the bands we are interested in. The crystal structure of pure lead has one atom per primitive cell, and since we also want to describe some states above the Fermi Energy, we include five d orbitals (d_{xy} , d_{xz} , d_{yx} , d_{z^2} and $d_{x^2-y^2}$), and four sp^3 orbitals. Hence our guess for the projections is

- 5 d orbitals centered on the lead atom
- 4 sp^3 orbitals centred on the lead atom

Now we are ready to obtain MLWFs and describe the states of lead around the Fermi-level.

- Directory: `ex3`
- Input Files
 - `01_sc.f.in` *The PWSCF input file for the ground state calculation*

```
&control
calculation='scf'
restart_mode='from_scratch',
pseudo_dir = './pseudo/',
outdir='./'
prefix='pb'
/
&system
ibrav = 2, celldm(1) = 9.3555, nat= 1, ntyp= 1,
ecutwfc = 47.0, ecutrho = 189,
occupations='smearing', smearing='cold', degauss=0.02
/
&electrons
conv_thr = 1.0e-9
mixing_beta = 0.7
/
ATOMIC_SPECIES
Pb 207.2 Pb.pbe-dn-kjpaw_psl.0.2.2.UPF
ATOMIC_POSITIONS
Pb 0.0 0.0 0.0
K_POINTS (automatic)
8 8 8 0 0 0
```

- `04_nscf.in` *The PWSCF input file to obtain Bloch states on a uniform grid*

```
&control
calculation='nscf'
pseudo_dir = './pseudo/',
outdir='./'
prefix='pb'
/
&system
ibrav = 2, celldm(1) = 9.3555, nat= 1, ntyp= 1,
ecutwfc = 47.0, ecutrho = 189,
occupations='smearing', smearing='cold', degauss=0.02
nosym=.true., nbnd=13
/
&electrons
conv_thr = 1.0e-9
```

```

/
ATOMIC_SPECIES
Pb 207.2 Pb.pbe-dn-kjpaw-psl.0.2.2.UPF
ATOMIC_POSITIONS
Pb 0.0 0.0 0.0
K_POINTS crystal
512
0.00000000 0.00000000 0.00000000 1.953125e-03
0.00000000 0.00000000 0.12500000 1.953125e-03
...

```

– 05_pw2wan.in *Input file for pw2wannier90*

```

&inputpp
outdir = './'
prefix = 'pb'
seedname = 'ex3'
write_amn = .true.
write_mmn = .true.
/

```

– ex3.win *The wannier90 input file*

```

use_ws_distance = .true.

num_bands      = 13
num_wann       = 9
num_iter       = 200

dis_win_max    = 38.0
dis_froz_max   = 16.0
dis_num_iter   = 50
dis_mix_ratio  = 1.0

Begin Kpoint_Path
G 0.00 0.00 0.00    X 0.50 0.50 0.00
X 0.50 0.50 0.00    W 0.50 0.75 0.25
W 0.50 0.75 0.25    L 0.00 0.50 0.00
L 0.00 0.50 0.00    G 0.00 0.00 0.00
G 0.00 0.00 0.00    K 0.00 0.50 -0.50
End Kpoint_Path

! SYSTEM

begin unit_cell_cart
bohr
-4.67775 0.00000 4.67775
0.00000 4.67775 4.67775
-4.67775 4.67775 0.00000
end unit_cell_cart

begin atoms_frac
Pb 0.00 0.00 0.00
end atoms_frac

begin projections
Pb:d;sp3
end projections

```

```
! KPOINTS

mp_grid : 8 8 8

begin kpoints
0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.12500000
...
```

1. Run PWSCF to obtain the ground state of lead
`pw.x < 01_scf.in > scf.out`
2. Run PWSCF to obtain the Bloch states on a uniform k-point grid
`pw.x < 04_nscf.in > nscf.out`
3. Run `wannier90` to generate a list of the required overlaps (written into the `lead.nnkp` file).
`wannier90.x -pp ex3`
4. Run `pw2wannier90` to compute the overlap between Bloch states and the projections for the starting guess (written in the `lead.mmn` and `lead.amn` files).
`pw2wannier90.x < 05_pw2wan.in > pw2wan.out`
5. Run `wannier90` to compute the MLWFs.
`wannier90.x ex3`

Inspect the output file `ex3.wout`.

1. Use Wannier interpolation to obtain the Fermi surface of lead. Rather than re-running the whole calculation we can use the unitary transformations obtained in the first calculation and restart from the plotting routine. Add the following keywords to the `lead.win` file:

```
restart = plot
fermi_energy = [insert your value here]
fermi_surface_plot = true
```

and re-run `wannier90`. The value of the Fermi energy can be obtained from the output of the initial first principles calculation. `wannier90` calculates the band energies, through wannier interpolation, on a dense mesh of k-points in the Brillouin zone. The density of this grid is controlled by the keyword `fermi_surface_num_points`. The default value is 50 (i.e., 50^3 points). The Fermi surface file `lead.bxsf` can be viewed using `XCrySDen`, e.g.,

```
xcrysdn --bxsf ex3.bxsf
```

2. Plot the interpolated band structure. A suitable path in k-space is

```
begin kpoint_path
G 0.00 0.00 0.00 X 0.50 0.50 0.00
X 0.50 0.50 0.00 W 0.50 0.75 0.25
W 0.50 0.75 0.25 L 0.00 0.50 0.00
L 0.00 0.50 0.00 G 0.00 0.00 0.00
G 0.00 0.00 0.00 K 0.00 0.50 -0.50
end kpoint_path
```

Further ideas (if you have time)

- Compare the Wannier interpolated band structure with the full PWSCF band structure. Obtain MLWFs using a denser k-point grid. To plot the band structure you can use the PWSCF tool `bands.x`.
- Investigate the effects of the outer and inner energy windows on the interpolated bands.

- We might just need a Wannier basis to describe the states around the Fermi level. We can extract an basis of 4 sp^3 orbitals. As the 4 bands around the Fermi level form an isolated manifold we can do this without disentanglement. Repeat the nscf calculation setting `nbnd=9`. Set `exclude_bands : 1:5` in the `.win` to remove the d states from the Wannierisation. We can then set `num_bands=4` and `num_wann=4`. Compare the spreads and the interpolated band-structure to the case above.
- Lead is a heavy element, and we naturally think about relativistic effects such as spin-orbit coupling. We can easily compute Wannier functions which come from a spinor basis and include spin-orbit coupling.
 1. Edit `01_scf.in` to add the following lines to the system namelist

```
lspinorb=.true., noncolin=.true.,
```

 Also change the pseudopotential to `Pb.relpednrrkjus_psl.1.0.0.UPF`
 2. Make the same changes to `04_nscf.in`. You will also need to double the number of bands `nbnd=18` (as each spinor state will be two-fold degenerate in the absence of magnetisation).
 3. For `ex3.win` set `spinors : T` and double both the number of bands and the number of wannier functions.
 4. You can now run the whole sequence of calculations starting with the SCF. The final Wannier hamiltonian will contain spin-orbit coupling. The effect will be clear in the interpolated band-structure (less so in the Fermi-surface).

Lead - d;sp3

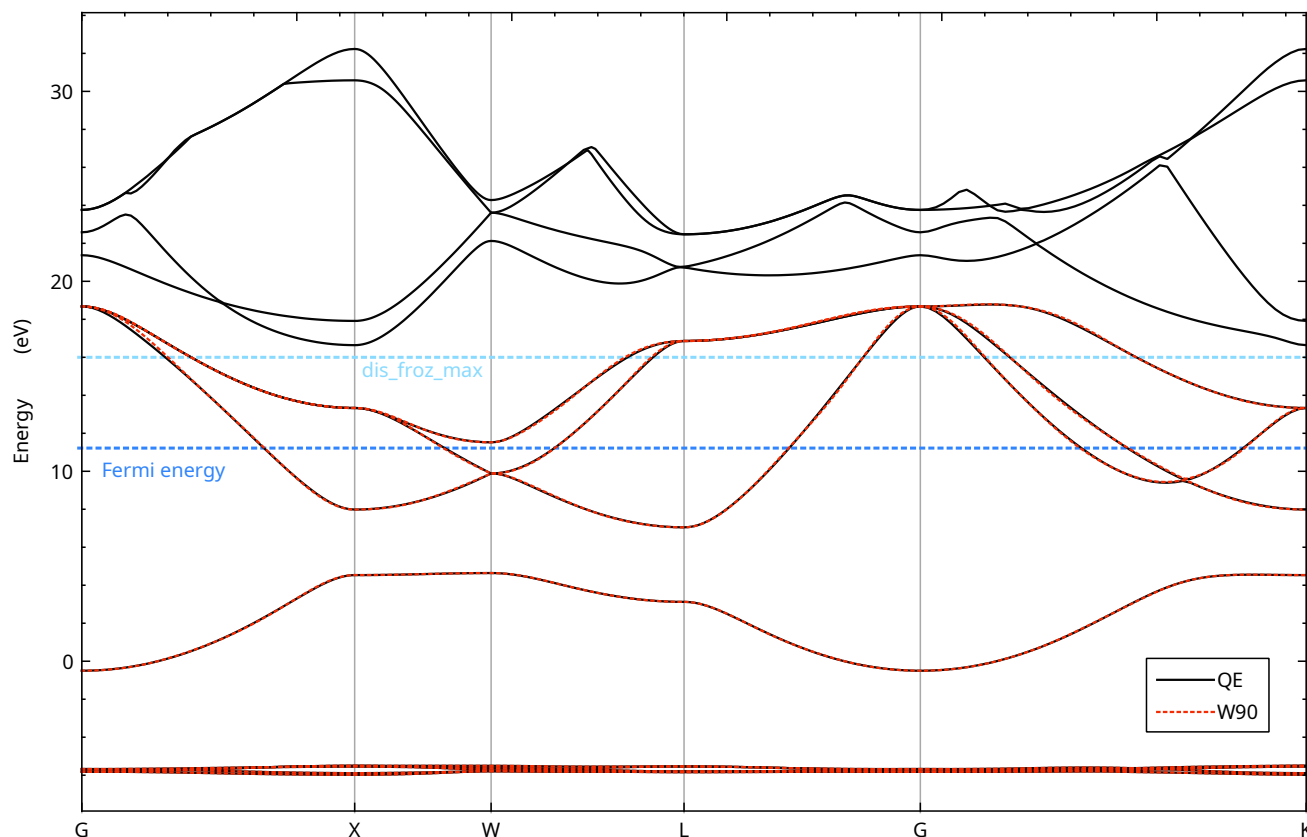


Figure 1: Band Structure of lead showing the position of the Fermi energy and inner energy windows.

Exercise 4 BaTiO₃ ferroelectricity

- Outline: *Obtain MLWFs for a perovskite*
- Directory: ex4/
- Input Files
 - 01_scf.in *The PWSCF input file for ground state calculation*

```
&control
calculation = 'scf'
restart_mode='from_scratch',
prefix='batio3',
pseudo_dir = './pseudo/',
outdir='./'
/
&system
ibrav= 1, celldm(1) =7.44266, nat= 5, ntyp= 3,
ecutwfc =40.0,
/
&electrons
conv_thr = 1.0d-10
/
ATOMIC_SPECIES
O 8 O.pbe-van_ak.UPF
Ti 22 Ti.pbe-sp-van_ak.UPF
Ba 56 Ba-vdb-usc-pbe.UPF
ATOMIC_POSITIONS {crystal}
Ba 0.0 0.0 0.0
Ti 0.5 0.5 0.5
O 0.0 0.5 0.5
O 0.5 0.5 0.0
O 0.5 0.0 0.5
K_POINTS {automatic}
8 8 8 0 0 0
```

- 02_nscf.in *The PWSCF input file to obtain Bloch states on a uniform grid*

```
&control
calculation = 'nscf'
prefix='batio3',
pseudo_dir = './pseudo/',
outdir='./'
/
&system
ibrav= 1, celldm(1) =7.44266, nat= 5, ntyp= 3,
ecutwfc =40.0,
/
&electrons
conv_thr = 1.0d-10
/
ATOMIC_SPECIES
O 8 O.pbe-van_ak.UPF
Ti 22 Ti.pbe-sp-van_ak.UPF
Ba 56 Ba-vdb-usc-pbe.UPF
ATOMIC_POSITIONS {crystal}
Ba 0.0 0.0 0.0
Ti 0.5 0.5 0.5
O 0.0 0.5 0.5
O 0.5 0.5 0.0
```

```

0 0.5 0.0 0.5
K_POINTS crystal
64
0.00000000 0.00000000 0.00000000 1.562500e-02
0.00000000 0.00000000 0.25000000 1.562500e-02
...
```

– 03_pw2wan.in *Input file for pw2wannier90*

```

&inputpp
outdir = './'
prefix = 'batio3'
seedname = 'ex4'
spin_component = 'none'
write_mmn = .true.
write_amn = .true.
write_unk = .true.
/
```

– ex4.win *The wannier90 input file*

```

num_wann      = 9
num_iter      = 50

!here we exclude all bands except the O2p bands
exclude_bands : 1,2,3,4,5,6,7,8,9,10,11

guiding_centres = T

begin atoms_frac
Ba 0.0 0.0 0.0
Ti 0.5 0.5 0.5
O 0.0 0.5 0.5
O 0.5 0.5 0.0
O 0.5 0.0 0.5
end atoms_frac

begin kpoint_path
L 0.50000 0.50000 0.5000 G 0.00000 0.00000 0.00000
G 0.00000 0.00000 0.0000 X 0.50000 0.00000 0.50000
X 0.50000 -0.50000 0.0000 K 0.37500 -0.37500 0.00000
K 0.37500 -0.37500 0.0000 G 0.00000 0.00000 0.00000
end kpoint_path
bands_plot = T

!!To plot the WF
!wannier_plot = T
!wannier_plot_supercell = 2
!wannier_plot_list = 2
!restart = plot

begin projections
O:p
end projections
```

```

begin unit_cell_cart
bohr
7.44266 0.0 0.0
0.0 7.44266 0.0
0.0 0.0 7.44266
end unit_cell_cart

mp_grid : 4 4 4

begin kpoints
0.00000000 0.00000000 0.00000000
0.00000000 0.00000000 0.25000000
...
```

To start with, we are going to obtain MLWFs for the oxygen 2p states. From the band structure [1], these form an isolated group of bands. We use the `wannier90` input parameter `exclude_bands` to remove all but the 2p bands from the calculation of the overlap and projection matrices (we don't have to do this, but it saves time).

1. Run PWSCF to obtain the ground state of BaTiO₃
`mpirun -np 4 pw.x < 01_scf.in > scf.out`
2. Run PWSCF to obtain the Bloch states on a uniform k-point grid
`mpirun -np 4 pw.x < 02_nscf.in > nscf.out`
3. Run `wannier90` to generate a list of the required overlaps (written into the `ex4.nnkp` file).
`wannier90.x -pp ex4`
4. Run `pw2wannier90` to compute the overlap between Bloch states and the projections for the starting guess (written in the `ex4.mmn` and `ex4.amn` files).
`mpirun -np 4 pw2wannier90.x < 03_pw2wan.in > pw2wan.out`
5. Run `wannier90` to compute the MLWFs.
`wannier90.x ex4`

Inspect the output file `ex4.wout`. Plot the three MLWFs centred on one of the oxygen atoms (e.g., the first three MLWFs) by adding the following keywords to the input file `ex4.win`

```

wannier_plot = true
restart = plot
wannier_plot_list = 1, 2, 3
wannier_plot_supercell = 3
```

and re-running `wannier90`. Visualise them using XCrySDen or VESTA.

Simulating the ferroelectric phase

Simulate the ferroelectric phase by displacing the Ti atom. Make a copy of the `ex4` directory (e.g., call it `ex4-fe`) and re-run all of the steps above (i.e., compute the ground-state charge density and Bloch states using `pw.x`, etc.), but this time change the position of the Ti atom (in crystal/fractional coordinates) to

```
Ti 0.505 0.5 0.5
```

and add the input parameter `nosym=.true.` to the `system` namelist of `02_nscf.in`. Run Wannier90, regenerate the MLWFs and consider what changes compared to the previous run.

Further explorations for the curious

- Look at MLWFs for other groups of bands. What happens if you form MLWFs for the whole valence manifold?
- Following Ref. [1], compute the Born effective charges from the change in Wannier centres under an atomic displacement.

Exercise 5 bcc Fe: spin polarised

- Outline: *Generate both maximally-localized and projected Wannier functions for ferromagnetic bcc Fe.*
 - Directory: 05/
 - Input Files
 - `iron.scf` *The PWSCF input file for the spin-polarized ground state calculation*
 - `iron.nscf` *The PWSCF input file to obtain Bloch states on a uniform grid*
 - `iron_{up,down}.pw2wan` *Input files for pw2wannier90*
 - `iron_{up,down}.win` *Input files for wannier90 and postw90*
 - Note that in a spin-polarized calculation the spin-up and spin-down MLWFs are computed separately. (The more general case of spinor WFs will be treated in Example 17.)
1. Run PWSCF to obtain the ferromagnetic ground state of bcc Fe
`pw.x < iron.scf > scf.out`
 2. Run PWSCF to obtain the Bloch states on a uniform k-point grid
`pw.x < iron.nscf > nscf.out`
 3. Run `wannier90` to generate a list of the required overlaps (written into the `.nnkp` files).
`wannier90.x -pp iron_up`
`wannier90.x -pp iron_dn`
 4. Run `pw2wannier90` to compute the overlap between Bloch states and the projections for the starting guess (written in the `.mmn` and `.amn` files).
`pw2wannier90.x < iron_up.pw2wan > pw2wan_up.out`
`pw2wannier90.x < iron_dn.pw2wan > pw2wan_dn.out`
 5. Run `wannier90` to compute the MLWFs.
`wannier90.x iron_up`
`wannier90.x iron_dn`

Further explorations for the curious

- See example 8 from the Wannier90 tutorial for additional ideas

Exercise 6 bcc Fe: iron – Spin-orbit-coupled bands

- Outline: *Plot the spin-orbit-coupled bands of ferromagnetic bcc Fe.*
- Directory: `ex06/`
- Input files
 - `Fe.scf` *The PWSCF input file for ground state calculation*
 - `Fe.nscf` *The PWSCF input file to obtain Bloch states on a uniform grid*
 - `Fe.pw2wan` *The input file for pw2wannier90*
 - `Fe.win` *The wannier90 and postw90 input file*

Note that `num_wann = 18` in `Fe.win`, but only nine trial orbitals are provided. The line

```
spinors = true
```

tells `wannier90` to use in step 3 below the specified trial orbitals on both the up- and down-spin channels, effectively doubling their number.

1. Run PWSCF to obtain the ferromagnetic ground state of iron¹
`pw.x < Fe.scf > scf.out`
2. Run PWSCF to obtain the Bloch states on a uniform k-point grid
`pw.x < Fe.nscf > nscf.out`
3. Run `wannier90` to generate a list of the required overlaps (written into the `Fe.nnkp` file)
`wannier90.x -pp Fe`
4. Run `pw2wannier90` to compute:
 - The overlaps $\langle u_{n\mathbf{k}} | u_{m\mathbf{k}+\mathbf{b}} \rangle$ between *spinor* Bloch states (written in the `Fe.mmn` file)
 - The projections for the starting guess (written in the `Fe.amn` file)
 - The spin matrix elements $\langle \psi_{n\mathbf{k}} | \sigma_i | \psi_{m\mathbf{k}} \rangle$, $i = x, y, z$ (written in the `Fe.spn` file)`pw2wannier90.x < Fe.pw2wan > pw2wan.out`
5. Run `wannier90` to compute the MLWFs.
`wannier90.x Fe`

Further explorations for the curious

- See example 17 from the Wannier90 tutorial for additional ideas

¹Please note the following counterintuitive feature in `pwscf`: in order to obtain a ground state with magnetization along the *positive* z -axis, one should use a *negative* value for the variable `starting_magnetization`.

Exercise 7 FeSe: A final Challenge

- Outline: *Can you obtain WF to describe the states around the Fermi energy*
- Directory: `ex07/`
- Obtain Wannier functions suitable for describing the states around the Fermi energy. We've given you the input structure in the `scf` input file. (this might be quite time consuming, and it might be something to try after the tutorial).

References

- [1] Nicola Marzari and David Vanderbilt. Maximally-localized wannier functions in perovskites: Cubic BaTiO₃. volume 436, pages 146–155. AIP, 1998. doi:10.1063/1.56269.