Hands-on Tutorial on Electronic Structure Computations: Transport with Wannier functions

Giovanni Pizzi

January 15th, 2013

This PDF document and the files for this tutorial can be found in the folder
`/afs/ictp.it/public/shared/smr2522/WANNIER_TUTORIAL/`

Before starting the tutorials, copy the content of this folder to your scratch folder (either using a graphical file browser, or using the command

```
cp -r /afs/ictp.it/public/shared/smr2522/WANNIER_TUTORIAL/ /scratch/WANNIER_TUTORIAL
```

The codes (`Quantum Espresso`, `Wannier90`) are in subfolders of `/afs/ictp.it/public/shared/smr2522/`, but the executables should be already in your path (that means that for instance, to run the `pw.x` code of the `Quantum Espresso` suite, you just need to type `pw.x`)

Exercise 1 - Silicon valence bands

In this exercise we will learn to obtain the Wannier functions for the valence bands of silicon.

- Go to the `ex1` folder (`cd /scratch/WANNIER_TUTORIAL/ex1`). Inspect the input file `01_scf.in`. It is a ground-state calculation for a silicon crystal with two atoms per unit cell, with a FCC cell. Check if you understand all parameters (you can use the web page
  http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html
  for keywords that you do not know). Visualize the unit cell with `xcrysden` (either opening the program and selecting from the menu `File`→`Open PWscf...`→`Open PWscf Input File` and selecting the input file, or directly from the command line with the command `xcrysden --pwi 01_scf.in`
- The Si pseudopotential that we will use for the calculation has $Z_{val} = 4$ (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`). With 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 (`pw.x < 01_scf.in > scf.out`). You may want to use the parallelization to run the simulation faster, using (to use 8 processors)

  ```
  mpirun -np 8 pw.x < 01_scf.in > scf.out
  ```

- When the calculation finishes, inspect the output file to check if there were any errors/warnings. Check your answer to the previous point (number of electrons and occupied valence bands).
- We want to plot the band structure of silicon (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_band.in`. Do the following modifications to the file `02_band.in` (use the INPUT_PW documentation from the link above for an explanation of the meaning of the flags, if needed):
  - In the `CONTROL` namelist, change the `calculation` from `'scf'` to `'bands'` to perform a band structure calculation starting from the ground state density obtained from the scf run.
  - Ask the code to print 12 bands (flag `nbnd` in the `SYSTEM` namelist)

- set `diago_full_acc = .true.` in the `ELECTRONS` namelist (see documentation for the meaning of this flag)
- Change the k-point list to plot the band structure along the following path (coordinates are given in crystalline units), using 50 points per segment:
  * $\mathrm{L}(0.5, 0.5, 0.5) \to \Gamma(0, 0, 0)$
  * $\Gamma(0, 0, 0) \to \mathrm{X}(0.5, 0, 0.5)$

  You can do this simply using the following `K_POINTS` card:

```
K_POINTS crystal_b
3
0.5 0.5 0.5 50
0.  0.  0.  50
0.5 0.  0.5 50
```

- Run the calculation using the `pw.x` code.
- When the calculation has finished, run the file `03_bandplot.in` through the `bands.x` executable (be sure to read and understand the input file):

```
mpirun -np 8 bands.x < 03_bandplot.in > bandplot.out
```

  This will produce the `bands.dat` file.
- Finally, execute the `plotband.x` code (interactively) and answer to its questions. In particular, the input file is the `bands.dat` file created in the previous step; call the xmgrace file `abinitiobands.agr`. When asked, call the At the end, open the xmgrace file (or directly the postscript PS file) 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. Copy the `02_band.in` file that you created before to `04_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 $4 \times 4 \times 4$ mesh, that will be used to calculate the discrete derivatives needed for the calculation of the Wannier functions. To obtain the list, use the `kmesh.pl` utility in the utility folder of the `Wannier90` code, using the following command for a $4 \times 4 \times 4$ mesh:

```
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:

```
mpirun -np 8 pw.x < 04_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 `.win` extension). Change the values marked with `XXX` inserting the correct values. in particular:
  - insert the `num_bands` value (this must be equal to the `nbnd` value set in the nscf calculation)
  - insert the `num_wann` value (this is the number of requested Wannier functions: in this case without disentanglement, this is equal to the `num_bands` value)
  - set the `mp_grid` value 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 kpoints, one per line. Note that while `pw.x` requires four numbers per line (the three k coordinates, and the weight), `Wannier90` wants only three numbers (the three coordinates). To obtain the lines, use again the `kmesh.pl` utility, but this time specifying a fourth parameter to get the list in the Wannier90 format:

```
kmesh.pl 4 4 4 wan
```

    Note Using the `kmesh.pl` utility, we are sure 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 on the http://www.wannier.org/user_guide.html page) for the input flags that you do not understand. Try to understand, in particular, the `projections` section. Can you say which is the location of the four s-like orbitals sitting with respect to the Si atoms? _____

- Now, we are ready to perform the Wannier calculation. This is done in three steps:

  1. We first run a preprocess step using the command (it **must** be run in serial)
     `wannier90.x -pp ex1`
     to produce a `ex1.nnkp` file, that contains the relevant information from the Wannier90 input file in a format to be used in the next step.
  2. We will now run the `pw2wannier90.x` code (of the Quantum Espresso suite).
     The input file for `pw2wannier90.x` is provided (file `05_pw2wan.in`). We are asking the code to calculate the overlap matrices M_{mn} (that will be written on the `ex1.mmn` file) and the A_{mn} matrices (file `ex1.amn`). Since we want to plot the Wannier functions in real space, we need also the $u_{nk}(r)$ wavefunctions on a real-space grid. We thus also set the `write_unk` flag in `05_pw2wan.in`, that will produce a set of files with names `UNK00001.1`, `UNK00002.1`, ... Finally, the code will also produce a `ex1.eig` file, with the eigenvalues on the initial $4 \times 4 \times 4$ k-grid (**Note**: this is not needed for the Wannierization, but only for the interpolation and band plotting routines). Note that the `pw2wannier90.x` expects to find the `ex1.nnkp` file produced in the previous step. Run the code using
     `mpirun -np 8 pw2wannier90.x < 05_pw2wan.in > pw2wan.out`
  3. Now we can finally run the Wannierization. Execute
     `wannier90.x ex1`
     (it **must** be run in serial) 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 Wannier functions.
     - To check if the obtained Wannier functions are correct, it is typically needed to:
       * *compare the Wannier-interpolated band structure with the ab-initio one*: the provided `Wannier90` input file computes the interpolated band plot; you can try to compare the ab-initio bandplot obtained in the steps before with the interpolated band structure (files `ex1_band.dat`, and `ex1_band.gnu` to plot it with `gnuplot`; otherwise, to use `xmgrace`, you can run
         `xmgrace abinitiobands.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 kpoints 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 them toward the end of the `ex1.wout` file: check that the value is small
     - Plot one of the Wannier functions, which are output in files `ex1_00001.xsf, ...` using `xcrysden`: open the `xsf` file, then choose `Tools`→`Data Grid`→`OK`, and then choose a reasonable `isovalue`, activate the `Render +/- isovalue` flag, and press `Submit`.

Exercise 2 - Silicon valence and conduction bands

- Copy all the `ex1` folder to a new folder named `ex2`. The first step (`01_scf.in`) is identical: if you copied also the out directory, you don't need to rerun it.

- In the `04_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 above).

- Rename `ex1`.win to `ex2`.win, and modify the following flags:

  - Change `num_bands` to 12 to be consistent with the new nscf run
  - 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` flag to the right number of Wannier functions: how many do we want, according to the projections list given above? _____
  - Set the maximum energy for the frozen window (flag `dis_froz_max`) inside the energy gap (use the band plot obtained in exercise 1 to get a value for this flag)
  - Set the maximum energy for the disentanglement (flag `dis_win_max`) to an energy large enough so as to contain enough bands for each k point; 17.0 eV should be OK (check where this value lies in the band plot).

- Inside the file `05_pw2wan.in`, change the seedname to '`ex2`' to reflect the new name of the .win file.
- Run `wannier -pp ex2`
- Run `pw2wannier90` using `05_pw2wan.in` as input file
- Run `wannier ex2`
- Check the output:
  - Before the start of the Wannierization iterations, there is a new section (containing the string `<--DIS`) with the iterations of the disentanglement part. 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 final WF centers and verify that WF are real; you may also want to plot the Wannier functions, or compare the interpolated band structure with the ab-initio one (obtained in exercise 1).
- **Optional part, to be done only at the end, if you have enough time**: Are the Wannier functions as expected? (We would like 4 sp^3-like orbitals centered on each atom, with similar spreads). Try to rerun everything with a $6 \times 6 \times 6$ kgrid for the `nscf` and `Wannier90` step to check if the results improve.

Exercise 3 - Linear C chain

We will study now ballistic transport in a linear chain of equispaced carbon atoms. Go to the `ex3` folder.

- Inspect the scf input file `01_scf.in`. It is a linear chain of C atoms along the x axis, with C-C distance of 1.3 Å. Run the scf calculation with `pw.x`. How many electrons are there in the system? _____
  Note: there may be a bug in old versions of `xcrysden` when the input cell is given in angstrom in the `pw.x` input file. As a workaround, run first `pw.x`, and then open the *output* file with `xcrysden --pwo scf.out`
- Copy the `01_scf.in` file to `02_band.in`, and modify the following (similarly to what was done in exercise 1):
  - change the calculation type from `'scf'` to `'bands'`
  - set the number of bands (`nbnd`) to 16
  - set the `diago_full_acc` flag to `.true.`
  - Set the k-point list to a line in k space (using `K_POINTS crystal_b`) from $\Gamma = (0, 0, 0)$ to X$= (0.5, 0, 0)$ [in crystal coordinates] with 50 points inbetween.
- Run `pw.x` using `02_band.in` as input file. Check the output to verify that the calculation was successful.

- Use the provided `03_bandplot.in` file as input for `bands.x`, and then run `bandplot.x` as described in exercise 1 to produce a band plot.
- Copy `02_band.in` to `04_nscf.in`, and modify the following:
  - change the calculation type from `'bands'` to `'nscf'`
  - change the kpoint list to a 6x1x1 mesh (use `kmesh.pl` to generate the k-point list), that we will use for the Wannierization.
- Run `pw.x` on the `04_nscf.in` input file. Check the output. Get the value of the Fermi energy from the output: _____ eV. Using this value and the band plot obtained before: Has this chain a metallic or insulating behavior? _____
- Open the `ex3.win` file, which is a template of the `Wannier90` input, try to understand the meaning of all input flags, and replace the values marked with `XXX`. In particular:
  - `num_bands` (use the value set in the nscf input file)
  - `num_wann` (understand which and how many projections were used by inspecting the `projections` section of the input file, and then set the correct value here)
  - Set the `dis_froz_max` to the value you got for the Fermi energy, and explain why this is a reasonable choice:_____
  - Using the band plot, set the window range for transport (`tran_win_max` and `tran_win_min`) large enough to include the whole band plot.
  - Set the `mp_grid` value to the correct grid size ($6 \times 1 \times 1$), and fill the kpoints section with the 6 kpoints (again, use the `kmesh.pl` script)
- Check the `05_pw2wan.in` input file for `pw2wannier90.x`, then:
  - Run `wannier90.x -pp ex3` (check the output)
  - Run `pw2wannier90.x` using `05_pw2wan.in` as input (check the output)
  - Run `wannier90.x ex3`
- Check the output of the wannierization (if the disentanglement convergence was achieved, if the Wannerization cycles reached convergence, if the Wannier functions are real). If everything went smoothly, you should get 4 atom-centered p-like orbitals, and 2 mid-bond s-like orbitals, with smaller spread. Try to plot with `xcrysden` one WF for each type, to see how they look like.
- Compare the interpolated band structure with the ab-initio one.
  - Using the knowledge of the symmetry of the system and the degeneracy of the bands, can you say which are the "p" bands and which are the "s" bands? _____
  - Do the interpolated band structure and the ab-initio band structure agree within the frozen window? And outside? Were you expecting this? _____
  - Using the obtained band structures, can you give an estimate of the lower bound for `dis_win_max` that you expect to be needed to get converged Wannier functions? _____
- In the `ex3.wout` file, there is a new section called "`Transport`". Inspect the results and try to understand its output.
- The activation of the transport flag also produces two new files: `ex3_dos.dat` (the Density Of States) and `ex3_qc.dat` (the Quantum Conductance). Plot them together using the command
  `xmgrace ex3_dos.dat ex3_qc.dat`
  (if you use this command, the black curve will be the DOS and the red one will be the QC).
  - Compare the DOS (black curve) with the band structure you plotted before. Were you expecting divergences at the band edges? Why? _____ Is the DOS behavior compatible with the band structure? _____
  - Can you explain the values of the quantum conductance? _____

Exercise 3 bis - Dimerized C chain

(Do this exercise only if you have enough time, otherwise go to exercise 4)

- Copy the `ex3` folder to `ex3bis`.
- Do all as you did for exercise 3, but changing the atomic coordinates of the C atoms in the files `01_scf.in`, `02_band.in`, `04_nscf.in` and `ex3.win` such that the x coordinates are the following: 0.6775 and 1.9225 (we are changing the positions of the C atoms such that now the interatomic C-C distances alternate with a shorter and a longer distance (a single and a triple bond).
- What do you expect that will happen to the band structure? Do you expect that you can use similar parameters for the Wannierization? _____
- Can you explain the difference in the DOS and in the quantum conductance with respect to exercise 3? _____
- You can compare the centers and the spreads of the Wannier functions of exercise 3 and 3 bis, and also visually inspect them to understand if there is a significant change or if they remain similar.

Exercise 4 - Linear C chain with a defect

- Inspect the input file `01_scf.in`, understanding the input flags; use also `xcrysden` to visualize the structure (as already discussed before, due to a bug in `xcrysden`, you should open the *output* pw.x file, after running the scf step). It is a C chain with 22 atoms, where the distance of C atoms is the same of exercise 3, except for the two central atoms which are nearer. How many k-points are being used? Is it a resonable approximation? _____ (In any case, note, that for the `lcr` transport in `Wannier90`, Γ-sampling only is *required*).
- How many electrons are there in the system? _____
- Inspect the `02_nscf.in` file: the number of bands (`nbnd`) value is missing: put the correct value (you should deduce this value from the one used in exercise 3 and considering the number of atoms in the unit cell).
- Before opening the `ex4.win` file: which initial projections would you use, knowing the results of exercise 3 (and 3 bis)? _____
- Open the `ex4.win` file, inspect it, and replace the values marked with `XXX`. In particular:
  - `num_bands`
  - `num_wann` (deduce it with the same method you used for `num_bands`, and then verify your result using the list of projections provided in the input file)
  - `dis_froz_max` and `dis_win_max` (hint: how are they related to the values used in example 3?)
  - `tran_win_max` and `tran_win_min` (hint: how are they related to the values used in example 3?)
- Inspect the `03_pw2wan.in` file. Note that for `lcr` transport calculations, we need a new flag: `write_unkg = .true.` (the Wannier90 documentation explains why)
- run the `scf` calculation, followed by the `nscf` calculation, the `wannier90` preprocess run, the `pw2wannier90.x` calculation, and finally the Wannierization.
- Check the output of the Wannierization, verifying the convergence and the results (WF centers, spreads). You may want to plot (some of) the Wannier functions: to do this modify `03_pw2wan.in` (adding the flag `write_unk = .true.`) and `ex4.win` to plot also (some of the) Wannier functions (add the `wannier_plot` and `wannier_plot_list` flags; see the `Wannier90` manual for the documentation; take also into account the possibility of using the `restart=plot` flag.)
- Plot the DOS and the QC; compare the QC with the QC obtained in exercise 3.
- Check the decay of matrix elements inspecting the `ex4_hr.dat` file (produced using to the `plot_hr` flag) that contains the Hamiltonian in real space between Wannier functions (read the `Wannier90` documentation for the explanation of the file format). Is the value `dist_cutoff`=5Å that we used in the `Wannier90` input flag sufficient, or too small? _____