

Maximally-localised Wannier functions and Wannier interpolation

Antimo Marrazzo, Valerio Vitale and Giovanni Pizzi

March 20th, 2018

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 http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html for keywords that you do not know or ask us). Visualize the crystal structure 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`). 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: `/home/nfs3/smr3191/q-e/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 the parallelization to run the simulation faster, using (for 2 processors)

```
mpirun -np 2 /home/nfs3/smr3191/q-e/bin/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. Compare your answer to the previous point (number of electrons and occupied valence bands) with the information provided in the output file.
- Now 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` keyword 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=12` 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 crystal units), using 50 points per segment:
 - * `L(0.5, 0.5, 0.5) → Γ(0, 0, 0)`
 - * `Γ(0, 0, 0) → 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, as explained before.
 - 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 2 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 ps file `abinitiobands.ps`. You will be asked to provide the value of the Fermi level, which in this case can be put equal to the highest occupied energy level (see the output file `scf.out`). 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 $4 \times 4 \times 4$ 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:


```
/home/nfs3/smr3191/q-e/wannier90-2.1.0/utility/
```

 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 2 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 coordinates of each kpoint, 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 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 where the four s-like orbitals are located with respect to the Si atoms? _____

- Finally, we are ready to perform a Wannier90 calculation. This is done in three steps:
 1. We first run a preprocessing step using the command (it **must** be run in serial)


```
wannier90.x -pp ex1
```

 which produce a `ex1.wout` file and `ex1.nnkp` file, that contains the relevant information from the Wannier90 input file in a format to be used in the next step.
 2. Then we run the `pw2wannier90.x` code (of the Quantum ESPRESSO distribution). 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 in 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 to obtain the MLWFs of an insulator, 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 2 pw2wannier90.x < 05_pw2wan.in > pw2wan.out
```
 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 if the obtained MLWFs 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 each of them at 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

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 conduction bands.

- Copy all the `ex1` folder to a new folder named `ex2`. The first step, i.e. the SCF calculation (`01_scf.in`), is identical. Hence, 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 in the previous exercise).
- 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` keyword to the correct 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 a reasonable value (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 `wannier90.x -pp ex2`.
 - Run `pw2wannier90.x` using `05_pw2wan.in` as input file.
 - Run `wannier90.x 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 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 final WF centers and verify that WFs 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 (Do it only if you have enough time):** Do the symmetry and the centers of the Wannier functions agree with your intuition? (We would like 4 sp^3 -like orbitals centered on each Si 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 - Copper Fermi surface and band structure

In this exercise we will see how to interpolate the band structure of copper, 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 copper has one atom per primitive cell, and we consider as valence states the five double filled $3d$ orbitals (d_{xy} , d_{xz} , d_{yx} , d_{z^2} and $d_{x^2-y^2}$) and a single filled $4s$ orbital. Since we also want to describe some states above the Fermi Energy we need to consider the whole $4s$ manifold. Hence our guess for the projections is

- $5d$ orbitals centered on the copper atom
- $2s$ orbitals centred on one of each of the two tetrahedral interstitial sites

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

- Directory: `ex3`

- Input Files

- `copper.scf` *The PWSCF input file for the ground state calculation*
- `copper.nscf` *The PWSCF input file to obtain Bloch states on a uniform grid*
- `copper.pw2wan` *Input file for pw2wannier90*
- `copper.win` *The wannier90 input file*

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

Inspect the output file `copper.wout`.

1. Use Wannier interpolation to obtain the Fermi surface of copper. 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 `copper.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 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 `copper.bxsf` can be viewed using `XCrySDen`, e.g.,

```
xcrysdn --bxsf copper.bxsf
```

2. Plot the interpolated bandstructure. 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

- Compare the Wannier interpolated bandstructure with the full PWSCF bandstructure. Obtain MLWFs using a denser k-point grid. To plot the bandstructure you can use the PWSCF tool `bands.x` or the small fortran program available at <http://www.tcm.phy.cam.ac.uk/~jry20/bands.html>.
- Investigate the effects of the outer and inner energy windows on the interpolated bands.
- Instead of extracting a subspace of seven states, we could extract a nine dimensional space (i.e., with s , p and d character). Examine this case and compare the interpolated band structures.

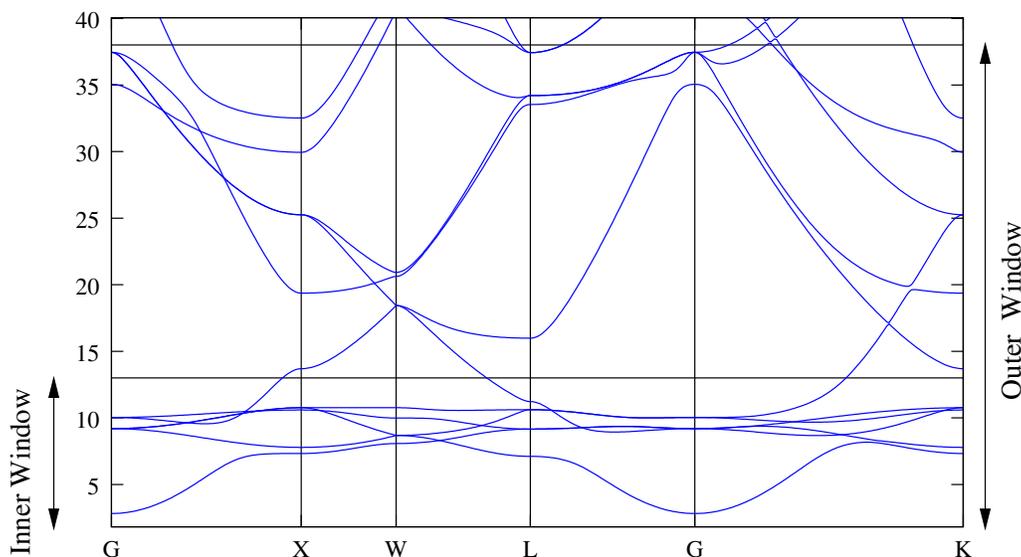


Figure 1: Band Structure of Copper showing the position of the outer and inner energy windows.

OPTIONAL

Do the following exercises only if you have enough time, i.e. you completed all the previous exercises.

Exercise 4 - G_0W_0 bands structure interpolation of silicon

Note: This example requires a recent version of the `ypp` post-processing code of `yambo`. You might need to install Yambo on your computer as this might not be preinstalled.

- Outline: *Interpolate the bands structure of silicon obtained from many-body perturbation theory at the G_0W_0 level. Using the `yambo` code, the quasi-particle corrections (QP) are summed to Kohn-Sham eigenvalues, while the wavefunctions remain the same.*
- Directory: `ex4`
- Input Files
 - `silicon.scf` *The PWSCF input file for the ground state calculation*
 - `silicon.nscf` *The PWSCF input file to obtain Bloch states on a uniform grid*
 - `silicon.gw.nscf` *The PWSCF input file to obtain Bloch states on a reduced grid with many empty bands*
 - `silicon.pw2wan` *The input file for `pw2wannier90`*
 - `silicon.win` *The `wannier90` input file*
 - `silicon.gw.win` *The `wannier90` input file (for the G_0W_0 step)*
 - `yambo.in` *The `yambo` input file*
 - `ypp.in` *The `ypp` input file*

1. Copy the input files from the `INPUT` directory into a working directory (e.g. `WORK`)
2. Run PWSCF to obtain the ground state charge of silicon


```
pw.x < silicon.scf > scf.out
```

3. Run PWSCF to obtain the Bloch states reduced grid. We use a 8x8x8 with many bands (many empty bands are needed to perform a G_0W_0 with yambo)


```
pw.x < silicon.gw.nscf > nscf.gw.out
```
4. Use the `k_mapper.py` utility to find the indices of a 4x4x4 uniform grid into the 8x8x8 reduced grid


```
./k_mapper.py 4 4 4 "./nscf.gw.out"
```

 Use the output to complete the `yambo.in` input file (you also need to specify the on how many bands you want to compute the QP corrections, here you can use all the bands from 1 to 14). Then, you should have obtained something like:


```
1| 1| 1|14|
3| 3| 1|14|
5| 5| 1|14|
13| 13| 1|14|
...
```
5. Enter the `si.save` directory and run `p2y`. A `SAVE` folder is created, you can move it up in the `/WORK/` directory.
6. Run a G_0W_0 calculation from the `/WORK/` directory (remember, we are using a 8x8x8 grid but computing QP corrections only on a 4x4x4 grid)


```
yambo
```
7. Run PWSCF to obtain the Bloch states on a uniform k-point grid


```
pw.x < silicon.nscf > nscf.out
```
8. Run `wannier90` to generate a list of the required overlaps (written into the `silicon.nnkp` file).


```
wannier90.x -pp silicon
```
9. Run `pw2wannier90` to compute the overlap between Bloch states, the projections for the starting guess (written in the `silicon.mmn` and `silicon.amn` respectively).


```
pw2wannier90.x < silicon.pw2wan > pw2wan.out
```
10. Run `wannier90` to compute the MLWFs.


```
wannier90.x silicon
```

 At this point, you should have obtained the interpolated valence bands for silicon at the DFT level.
11. Run a `ypp` calculation (just type `ypp`)
 You should obtain a file `silicon.gw.unsorted.eig` which contains the QP corrections on a uniform 4x4x4 grid.
12. Run the `gw2wannier90.py` script to reorder, align and correct all matrices and files using the QP corrections


```
./utility/gw2wannier90.py silicon mmn amn
```
13. Run `wannier90` to compute the MLWFs.


```
wannier90.x silicon.gw
```

 At this point, you should have obtained the interpolated valence bands for silicon at the G_0W_0 level.

After you completed the tutorial for the valence bands only, you can repeat the final steps to interpolate also some conduction bands using disentanglement (the code is already present as comments in the input files).

Exercise 5 - Linear C chain

We will study now ballistic transport in a linear chain of equispaced carbon atoms. Go to the `ex5` 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 `ex5.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 ex5` (check the output).
 - Run `pw2wannier90.x` using `05_pw2wan.in` as input (check the output).
 - Run `wannier90.x ex5`.
 - Check the output of the wannierization (if the disentanglement convergence was achieved, if the Wannierization 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: `ex5_dos.dat` (the Density Of States) and `ex5_qc.dat` (the Quantum Conductance). Plot them together using the command
`xmgrace ex5_dos.dat ex5_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 6 - Dimerized C chain

- Copy the `ex5` folder to `ex6`.
- 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 `ex5.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 5 and 6, and also visually inspect them to understand if there is a significant change or if they remain similar.