

Wannier90: Tutorial

CECAM/Psi-k, Lyon

26th June 2007

1 Introduction

Welcome to the Wannier90 tutorial held at the CECAM/Psi-k workshop entitled “Maximally-Localized Wannier Functions: Concepts, Applications, and Beyond” in Lyon, 26 June 2007. The webpage for the workshop can be found at

<http://irrmawww.epfl.ch/~pasquarello/cecam2007/wannier.html>.

The tutorial is in two parts: a morning session consisting of PWSCF examples; and an afternoon session on Wannier90. Input files for both can be found in a gzipped-tar file at

http://quasiamore.mit.edu/wannier/cecam07/tutorial_files.tgz.

There are four PWSCF examples (in directories `pwscf-ex[1-4]`), plus three Wannier90 examples (in directories `w90-ex[1-3]`) that are closely linked to the first three PWSCF examples.

This document deals specifically with the three Wannier90 examples, namely silicon, barium titanate and copper. The following sections describe each of them in detail.

For further information, see the Wannier90 webpage at <http://www.wannier.org>. In particular, see http://www.wannier.org/user_guide.html for the Wannier90 user guide and complete tutorial.

We hope you enjoy using Wannier90!

The Wannier Developers

1: Silicon

Valence Bands

- Outline: *Obtain MLWF for the valence bands of Silicon*
 - Input Files
 - `silicon.scf` *Pwscf input file for ground state calculation*
 - `silicon.nscf` *Pwscf input file to obtain Bloch states on a uniform grid*
 - `silicon.pw2wan` *Input file for pw2wannier90*
 - `silicon.win` *Wannier90 input file*
1. Run pwscf to obtain the ground state of silicon
`pw.x < silicon.scf > scf.out`
 2. Run pwscf to obtain the Bloch states on a uniform kpoint grid. Note that we request the lower 4 (valence) bands
`pw.x < silicon.nscf > nscf.out`
 3. Run Wannier90 to generate a list of the required overlaps (written into the `silicon.nnkp` file).
`wannier90.x -pp silicon`
 4. Run pw2wannier to compute the overlap between Bloch states and the projections for the starting guess (written in the `silicon.mmn` and `silicon.amn` files).
`pw2wannier90.x < silicon.pw2wan > pw2wan.out`
 5. Run Wannier90 to compute the Wannier functions.
`wannier90.x silicon`

Inspect the output file `silicon.wout`. The total spread converges to its minimum value after just a few iterations. Note that the geometric centre of each Wannier function lies at the centre of the Si-Si bond. Note also that the memory requirement for the minimisation of the spread is very low as the Wannier functions are defined by just the 4x4 unitary matrices, $U_{\mathbf{k}}$. Plot the Wannier Functions by adding the following keywords to the input file `silicon.win`

```
wannier_plot = true
```

and re-running Wannier90. To visualise the Wannier functions we must explicitly represent them on a real space grid (see User Guide Eq 1.1). As a consequence plotting the Wannier functions is slower and uses more memory than the minimisation of the spread. The 4 files that are created (`silicon_00001.xsf`, etc) can be viewed using XCrySDen

```
xcrysden --xsf silicon_00001.xsf
```

[Once XCrySDen has loaded, goto **Tools**, then **Data Grid**.]

For large systems plotting the Wannier functions may be time consuming and require a lot of memory. Use the keyword `wannier_plot_list` to plot a subset of the Wannier functions. Eg to plot the 1st, 2nd and 7th Wannier functions use

```
wannier_plot_list = 1 2 7
```

The Wannier functions are plotted in a supercell of the unit cell. The size of this supercell is set through the keyword `wannier_plot_supercell`. The default value is 2 (corresponding to a supercell with 8 times the unit cell volume). We recommend not using values great than 3 as the memory and computational cost scales cubically with supercell size.

Valence + Conduction State

- Outline: *Obtain MLWF for the valence and low-lying conduction states of Si. Plot the interpolated band structure*
- Input Files
 - `silicon.scf` *Pwscf input file for ground state calculation*
 - `silicon.nscf` *Pwscf input file to obtain Bloch states on a uniform grid*
 - `silicon.pw2wan` *Input file for pw2wannier90*
 - `silicon.win` *Wannier90 input file*

The valence and lower conduction states can be represented by Wannier Functions with sp^3 -like symmetry. The lower conduction states are not separated by an energy gap from the higher states. In order to form localised Wannier function we use the disentanglement procedure introduced in SMV. The position of the inner and outer energy windows are shown in Figure 1.

1. Run `pwscf` and the Wannier90 code.
Inspect the output file `silicon.wout`. The minimisation of the spread occurs in a two step procedure. We first minimise Ω_I , this is the extraction of the optimal subspace in the disentanglement procedure. We then minimise $\Omega_O + \Omega_{OD}$
2. Plot the band structure by adding the following commands to the input file `silicon.win`

```
restart = plot
bands_plot = true
```

and re-running Wannier90. The files `silicon_band.dat` and `silicon_band.gnu` are created. To plot the band-structure using gnuplot

```
myshell> gnuplot
gnuplot> load 'silicon_band.gnu'
```

The kpoint path for the bandstructure interpolation is set in the `kpoint_path` block. Try plotting along different paths.

Further ideas

- Compare the Wannier interpolated band structure with the full pwscf bandstructure you calculated this morning. Compute Wannier functions using a finer MP grid (eg 6x6x6 or 8x8x8) and note how the accuracy of the interpolation increases (see Ref [3] for more details)
- Compute Wannier Functions for 4 conduction states. (see ref [2])

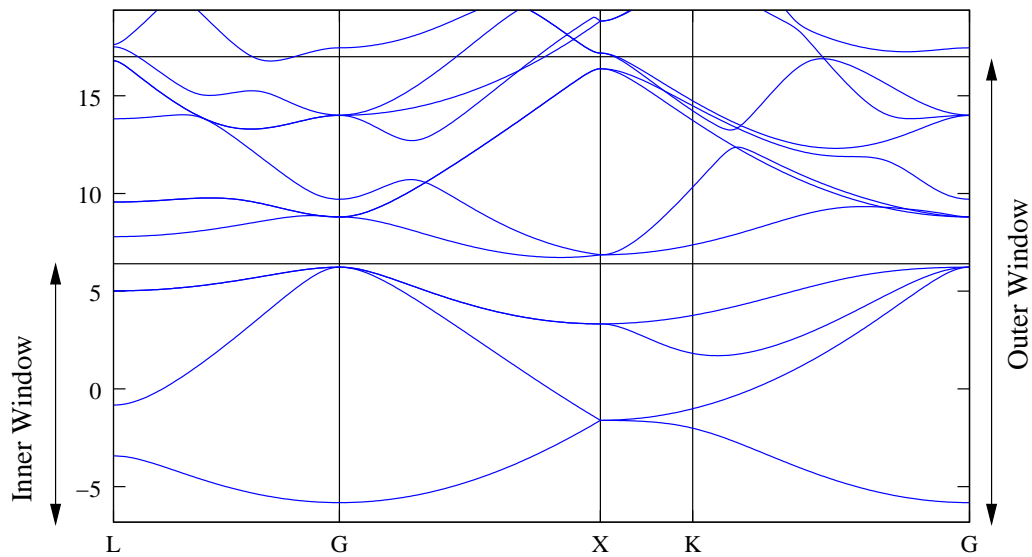


Figure 1: Band Structure of Silicon showing the position of the outer and inner energy windows.

2: Cubic BaTiO₃

- Outline: *Obtain MLWF for a Perovskite*
- Input Files
 - `batio3.scf` *Pwscf input file for ground state calculation*
 - `batio3.nscf` *Pwscf input file to obtain Bloch states on a uniform grid*
 - `batio3.pw2wan` *Input file for pw2wannier90*
 - `batio3.win` *Wannier90 input file*
- 1. To start with we are just going to obtain WF for the oxygen 2p states. From the bandstructure (see Ref. 4) these form an insolated group of bands. We use the wannier90 keyword `exclude_bands` to remove all but the 2p bands from the calculation of the overlap and projections (we don't have to do this, but it saves time).
- 2. Run `pwscf` and the Wannier90 code.
- 3. Plot the 2nd wannier function.
- 4. We can now simulate the ferroelectric phase by displacing the Ti atom. Change its position to

```
Ti 0.505 0.5 0.5
```
- 5. regenerate the WF (ie compute the ground state charge density, bloch states etc) and look at the change in the second WF.

Further ideas

- Look at WF for other groups of bands. What happens if you form WF for the whole valence manifold?
- Following Ref 4, compute the Born Effective charges from the change in Wannier centres under an atomic displacement. Compare with the values you obtained this morning.

3: Copper

- Outline: *Obtain MLWF to describe the states around the Fermi-level in Copper*
- Generation Details: *From pwscf using “ultrasoft” pseudopotentials and a 4x4x4 kpoint grid. Starting guess: 5 atom centred ‘d’ orbital and two ‘s’ orbitals centred on one of each of the two tetrahedral interstices.*
- Input Files
 - `copper.scf` *Pwscf input file for ground state calculation*
 - `copper.nscf` *Pwscf input file to obtain Bloch states on a uniform grid*
 - `copper.pw2wan` *Input file for pw2wannier90*
 - `copper.win` *Wannier90 input file*

1. Run the Wannier90 code to minimise the Wannier function spread

```
wannier90.x copper
```

Inspect the output file `copper.wout`.

2. 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. The Wannier90 code calculates the band energies, through wannier interpolation, on a dense mesh of kpoints in the Brillouin zone. The density of this grid is controlled by the keyword `fermi_surface_num_points`. The default value is 50 (ie 50^3 points). The Fermi surface file `copper.bxsf` can be viewed using XCrySDen

```
xcrysden --bxsf copper.bxsf
```

3. Plot the interpolated band structure. A suitable path in kspace 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 band structure with the full pwscf bandstructure. Obtain WF using a more dense MP grid.
- Investigate the effect of the outer and inner energy window on the interpolated bands.
- Instead of extracting a subspace of 7 states, we could extract a 9 dimension space (ie with s,p and d character). Examine this case and compare the projected bandstructures.

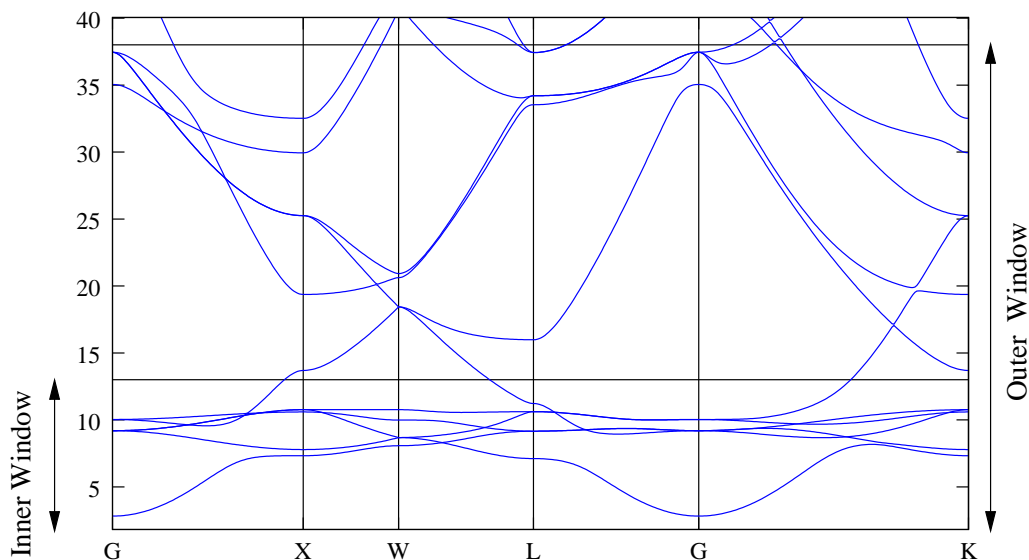


Figure 2: Band Structure of Copper showing the position of the outer and inner energy windows.

Futher ideas and references

Example of WF for various metals and graphite can be found at <http://www.tcm.phy.cam.ac.uk/~jry20/wannier/>

1. Maximally localized generalized Wannier functions for composite energy bands
N. Marzari and D. Vanderbilt, Phys. Rev. B 56, 12847 (1997).
2. Maximally localized Wannier functions for entangled energy bands
I. Souza, N. Marzari and D. Vanderbilt, Phys. Rev. B 65, 035109 (2002)
3. Spectral and Fermi surface properties from Wannier interpolation
J. R. Yates, X. Wang, D. Vanderbilt and I. Souza, Phys. Rev. B 75, 195121 (2007)
4. Maximally-localized Wannier functions in perovskites: Cubic BaTiO₃
N. Marzari and D. Vanderbilt <http://arxiv.org/abs/cond-mat/9802210>