

# Wannier functions with FPLO

Klaus Koepernik and Marc Höppner

November 4, 2014

## 1 Intro

Wannier functions can be defined in many ways since there is a gauge freedom of choosing a phase transformation. The ideal way of fixing the gauge is the requirement of maximum localization. This is a tedious algorithm and we do not do this in FPLO. However, it turns out that our definition leads to highly localized Wannier functions (WFs). Should the Wannier functions turn out not to be localized, it in most cases means that the WF is badly chosen. The main drawback of our approach is, that the user has to decide, where the WFs shall sit and which symmetry they shall have. However, this is at the same time intended, since for modeling it is exactly what one wants to do.

The WF sitting in cell  $R$  and being of type  $\mu$  (which denotes the WF center and its symmetry) is defined as

$$W_{R\mu} = \int e^{-ikR} \sum_n \Psi_n^k U_{n\mu}^k \quad (1)$$

where  $\Psi$  denotes the Kohns-Sham (KS) functions and  $U$  is a unitary matrix. If we define less WFs than KS functions  $U$  is a column unitary projector ( $U^\dagger U = 1$ ,  $UU^\dagger = P$ ). It maps all the KS bands on some few WFs. The choice of  $U$  is the choice of the gauge. In FPLO we use a chemically motivated local orbital basis  $\Phi$  to construct

$$\Psi_n^k = \frac{1}{\sqrt{N}} \sum_{R\nu} \Phi_{R\nu} e^{ik(R+s)} C_{\nu,n}^k$$

( $s$  is the atom position and  $\nu$  some quantum numbers specifying the orbital). These orbitals although non-orthogonal are in a way 'optimally' localized by their construction. Hence, it is clear that a WF centered at an atom and having a certain orbital symmetry has the corresponding orbital as its main contribution. This allows the following choice of  $U$ . We project the KS functions on a test function  $\chi$ , which is an FPLO orbital in the simplest case. The resulting number is the square root of the orbital character of the KS bands, as plotted in the FAT bands. If we now select only the KS functions with a large such orbital character in Eq. (1), we will end up with a WF resembling  $\chi$  the most and this is where the localization of our WFs comes from. If we want WFs corresponding to sub bands of a band complex with a character  $\chi$  we have to project onto a particular energy window as well. This happens if there are bonding (B) and antibonding (AB) bands of character  $\chi$  and if one wants, say, only WFs of the AB bands. So, in total we define a test function or WF projector  $\chi$  and an energy window for each WF. The WF projector  $\chi$  can in principle be a linear combination of FPLO orbitals. Example: suppose we have a cuprate plane. The bands are formed of linear combinations of Cu  $3d$  and ligand O- $2p$  orbitals. Around certain  $k$ -points the upper bands are clearly AB and hence correspond to a certain molecular orbital with a certain phase relation between the central  $3d$  orbital and the O- $2p$  orbitals. The lower (B) bands are clearly formed of the same orbitals, but with a different phase relation. If this is the case then defining WF-projectors using the AB molecular orbital as  $\chi$  will automatically project out the antibonding bands, given that the band topology is dominated by the clear character separation around the considered  $k$ -point. This would make the energy window obsolete.

The Wannier transformation can be described in two steps. In the first step the unitary projector is build from the users definitions of  $\chi$  and the energy windows.  $U$  is applied to the KS functions yielding the Bloch sums of the WFs

$$\begin{aligned} W_\mu^k(r) &= \sum_n \Psi_n^k(r) U_{n\mu}^k \\ &= \frac{1}{\sqrt{N}} \sum_R e^{ikR} W_{R\mu}(r) \end{aligned} \quad (2)$$

$$W_{R\mu}(r) = \int e^{-ikR} W_{\mu}^k(r) dk \quad (3)$$

(Note, that the  $k$ -integrations contain appropriate normalization factors, left out in the formulas or being hidden in the integral measure.) The orbital Bloch functions are

$$\Phi_{s\mu}^k = \frac{1}{\sqrt{N}} \sum_R \Phi_{Rs\mu} e^{ik(R+s)}$$

where the additional phase factor  $e^{iks}$  makes live easier by removing the dependence on the coordinate origin. From the formulas we get the representation of the Hamiltonian in orbital Bloch sums

$$\begin{aligned} \underline{H}_{s's}^k &= \langle \Phi_{s'}^k \hat{H} \Phi_s^k \rangle \\ &= \frac{1}{N} \sum_{R'R} \langle \Phi \hat{H} \Phi \rangle_{R's'R_s} e^{ik(R+s-R'-s')} \\ &= \sum_R \langle \Phi \hat{H} \Phi \rangle_{0s',Rs} e^{ik(R+s-s')} \end{aligned}$$

from which we get the KS eigenvalues

$$\varepsilon_n^k = \left( C^{k+} \underline{H}^k C^k \right)_n \quad (4)$$

This gives the WF Bloch representation

$$\begin{aligned} \langle W_{\mu'}^q \hat{H} W_{\mu}^k \rangle &= \sum_n U_{n\mu'}^{q*} \varepsilon_n^k U_{n\mu}^k \delta_{qk} \\ &= \frac{1}{N} \sum_{RP} \langle W_{P\mu'} \hat{H} W_{R\mu} \rangle e^{ik(R-P)} \delta_{qk} \\ &= \sum_R \langle W_{0\mu'} \hat{H} W_{R\mu} \rangle e^{ikR} \delta_{qk} \end{aligned} \quad (5)$$

with

$$\begin{aligned} \varepsilon_{0\mu',R\mu} &= \langle W_{0\mu'} \hat{H} W_{R\mu} \rangle \\ &= \int e^{-ikR} \langle W_{\mu'}^k \hat{H} W_{\mu}^k \rangle dk \end{aligned} \quad (6)$$

which is the WF Hamiltonian in real space representation, which usually contains the model we are interested in. Its  $k$ -representation (Bloch sums) Eq. (5) can be diagonalized and will give the bandstructure corresponding to the model. If the WFs represent the whole Hilbert space spanned by all  $\Psi_n^k$  of course the resulting WF-bandstructure coincides with the original bandstructure Eq. (4). There is a modification one can do, which consists of restricting the matrix elements Eq. (6) by removing hoppings with distances above a certain cut off or hoppings, which are smaller than a certain threshold. The resulting restricted hopping matrix can be Bloch summed and diagonalized again. Note, however, that this modified Hamiltonian does not strictly correspond to the WFs calculated above.

The whole transformation can be summed up, by defining a transformation from the FPLO basis into the WF basis

$$\begin{aligned} W_{R\mu}(r) &= W_{0\mu}(r - R) \\ W_{0\mu}(r) &= \sum_{R's\nu} \Phi_{R's\nu} D_{R's\nu,\mu} \end{aligned} \quad (7)$$

## 2 The FPLO WF module

The Wannier function module in FPLO is currently a postprocessing tool. First, one needs a converged calculation. Then, the relevant information must be written to the hard disk in order to access it conveniently later. This might take plenty of disk space! (See Sec. 2.5) This information is then read and used in a subsequent FPLO run to calculate the desired Wannier functions. In order to use the module a particular file (`=.wandef`) must be created by the user. If this file is found by a running FPLO process the Wannier module is activated, if the keyword `doit` is found on a single line in the file. To disable the module change the keyword into something like e.g. `xdoit`. If FPLO finds the file `=.wandef` with the keyword it will start dumping data

after every Kohn-Sham diagonalization process. If the FPLO process comes to convergence (best to start with a converged calculation to begin with) it stops like in normal mode. The file created after the initial data-dumping is `+wancoeff` (besides all the usual files). Note: that the band structure plot in the `fedit` submenu should be switched on.

On restart of FPLO (provided that `=.wandef` exist and the `doit`-keyword is set therein) all the data are used and the actual WF-module is executed. It reads WF definitions from `=.wandef` and constructs WFs accordingly. The WFs can be produced in real space for visualization and the WF Hamiltonian in Bloch representation Eq. (5) can be written to the file `+hamongrid` on a  $k$ -space grid. The WF hopping integrals Eq. (6) are produced in the output and can be used to extract models. Beware that the WF hamiltonian on the  $k$ -space grid is restricted by cutting of the real-space Hamiltonian Eq. (6) according to user input.

We will walk through an example step by step. The input-files are in the example directory. The example is  $\text{CaCuO}_2$  in a ferro-magnetic calculation (just to have that complication in). We want to create a WF for the antibonding  $3d_{x^2-y^2}$  band. In Figure 1 we show the spin-polarized

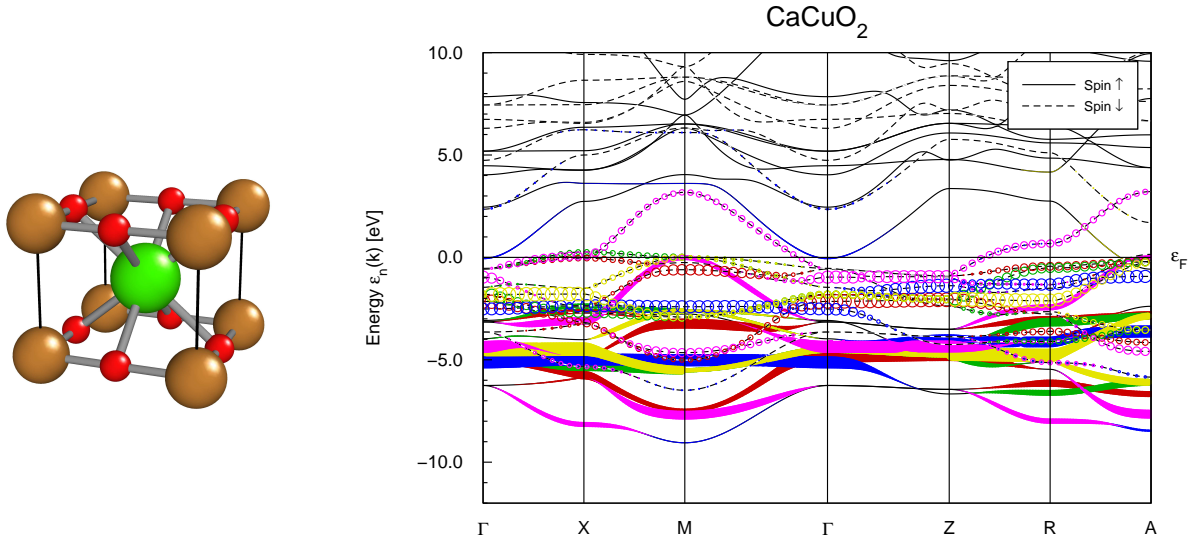


Figure 1: FPLO fat-bands  $\text{CaCuO}_2$

fat-bands as given by FPLO. The filled symbols denote the 3d majority fat-bands and the open symbols the 3d minority bands. The  $3d_{x^2-y^2}$  bands are called  $3d + 2$  according to the standard convention and orientation of the global coordinate system and have magenta color. Focus on the majority bands (filled symbols), you see a bonding and antibonding band (best seen at the M-point). We are interested in a description of the antibonding band. There is only one Cu atom in the unit cell and according to the list of sites in the output that is site number 2. We now guess that the Cu, site 2,  $3d_{x^2-y^2}$  orbital will have the largest contribution to a WF describing the anti-bonding part of the filled magenta band(s). Thus, we chose this orbital as a WF-projector  $\chi$ . We assume that the file `=.wandef` exists and that the keyword `doit` is set and that the file `+wancoeff` got already created.

We have to define, which linear combinations of local orbitals are used as WF-projectors. Note, that we have two basic choices: the WF-projector and the energy window. In our case of  $\text{CaCuO}_2$  we already identified the projector. The simplest projector consists of one single orbital centered at on site. A WF-projector is defined using the keyword `wandef`. A `wandef` can have many contributions `contrib`, each denoting an FPLO orbital with a certain weight factor. Here comes the example

```
...
----- Cu -----
wandef
on
```

```

name Cu x2-y2
emin -4 -1
emax -1 3
de 1 1
contrib
    site 2
    difvec 0 0 0
    xaxis 1 0 0
    zaxis 0 0 1
    orb 3d+2
    fac 1

```

As can be seen there can be arbitrary lines in the file, and they are considered comments as long as they do not start with a keyword. In our case '——— Cu ———' is such a comment. Then comes the keyword **wandef** starting of a WF-projector. The next thing is **on** or **off**. This allows to have several **wandefs** in one file, where only some are used (playing around). The **wandef** has a name, which can be anything after the keyword **name** (actually it should not be more than 17 characters). The energy window is discussed later. in our example the **wandef** is made of one **contrib**. The **contrib** is an orbital sitting at **site** 2, being **difvec** 0 0 0 away from the (imagined) centerpoint of the WF, which in this case is the position of site 2. The orbital is defined with respect to the local coordinate system, whose **xaxis** is 1 0 0 and **zaxis** is 0 0 1. In this local system the **orb** is 3d+2 and the **contrib** enters the whole **wandef** with relative weight **fac** 1. If there are more than one **contribs** the **fac** of each **contrib** can be given. The whole thing is normalized later on. If the **wandef** is made of several **contribs** at different sites it is important to correctly define the difference vectors **difvec**. Choose a particular point in space as WF center and express the sites of the **contribs** by vectors pointing from the WF center to the sites ... these are the **difvecs**.

If there is more than one site belonging to a crystallographic orbital (all sites generated by the same Wyckoff position), the user has to give every WF-projector separately in the moment. In this case it is important that the user takes care that all the different WF-projectors belonging to the cryst. orbit are symmetry related. This means setting up proper **difvec**, **x/zaxis**, **orb**, **fac** and so on values.

In our example the orbital used as WF-projector forms a bonding and an anti-bonding part. There is only one Cu  $3d_{x^2-y^2}$  orbital in the simple unit cell and hence it can only form one band. However, this orbital forms sigma-bonding hybrids with the plaquette oxygen  $2p$ -orbitals and this allows for and creates “two” bands. Moreover, it is clear from Figure 1 that the anti-bonding (AB) band is not clearly isolated due to other hybridizations. Hence, a well defined Wannier function cannot exist. In order to have some approximation for modeling we create a single WF of AB type. According to the rules of how WFs are created it is clear that we need to project away the part of the KS spectrum, which contains the bonding part of the corresponding band. This is done by specifying an energy window. All bands outside this window will be ignored. For the majority bands we have the AB band between -5 and 0 eV and the B band between -8 and -5 eV. For the minority band it is shifted upwards accordingly. The energy window could be defined as a sharp cutoff. This can lead to weird effects, if the band character specified through the WF-projector occurs ONLY outside this window for a particular k-point (such things can actually happen). This would mean that for one k-point the weight of the WF-projector in the considered Hilbert-subspace is zero and this leads to an indefinite problem, when calculating the WFs. So, it is better to make the window smooth in order to have the interesting subspace in the main energy window but allowing to sample outside of it, in case we have a character-run-away scenario. The energy window is defined by a function being 1 between **emin** and **emax** and falling off as a gaussian, with a width **de** outside this window. Of course, the energy window applies to all **contribs** of one **wandef**. In spin polarized cases these three keywords take two values one for the majority and one for the minority bands. The user should also pay attention that the energy window is the same for all **wandefs** belonging to the same cryst. orbit!

Suppose that we defined the above described WF projector onto one orbital, defining one WF per unit cell and that we did not specify the energy window. This would lead to a WF whose corresponding band will be pretty dispersionless situated at an energy between the B and AB bands in the original band structure. This happens because the projector  $\chi$  makes one single band out of the B and AB part of the KS-Hilbert space, which essentially forms two bands. In mathematical terms this will lead to an average of the B and AB KS wavefunctions, which will have non-bonding character. That should also make clear what happens, if we extend the energy window more and more towards the B energies. This will mix in more and more of the bonding KS functions, which leads to a WF band whose energies get pulled down more and more. The user can try to play with this to get a feeling.

Now, we have defined all the stuff in the file **=.wandef** and we can run FPLO.

1. The first thing in the WF module will be that the content of `=.wandef` is copied to the output, followed by a more condensed printout of the wannier parameters. The later output shows all parameters, also the ones not explicitly set in `=.wandef` (which then will have their default values). Then follows a section, in which the symmetry of the WF-projectors  $\chi$  is checked. If this check does not run through properly there is a mistake in the symmetry relation between **wandefs** of  $\chi$ -s belonging to the same cryst. orbit. Check all keywords. In the moment the energy window is not checked for proper symmetry setting. So, if the code runs through, but the output seems weird, check the energy windows!
2. Now, if the file `+wancoeff` is found it will be loaded. If not, the normal FPLO execution will continue. Reading the data is not the fastest that is why there is an internal loop (see below). See also Sec. 2.5.
3. Now, the Wannier bands get processed. As a result the hopping matrix elements between WFs are printed to the output. This printout is controlled by user defined restrictions to avoid enormous amounts of data. Example:

```
spin 1: WF(Cu x2-y2) -> WF(Cu x2-y2) at relative
T= 0.00000 0.00000 0.00000 hop= -2.093458893797475
T= 0.00000 0.00000 6.04712 hop= -0.061341124468540
T= 0.00000 0.00000 -6.04712 hop= -0.061341124468540
T= -7.29434 0.00000 0.00000 hop= -0.463606088774015
T= 0.00000 -7.29434 0.00000 hop= -0.463606088774014
T= 0.00000 7.29434 0.00000 hop= -0.463606088774014
T= 7.29434 0.00000 0.00000 hop= -0.463606088774015
```

The WFs are given by their name, and the symbol ' $\rightarrow$ ' means that the vectors, which follow ('T= ...') point from the WF left of ' $\rightarrow$ ' to the one right of ' $\rightarrow$ '. After each vector 'T=' the hopping element in eV is printed. If T=0 the hopping element is the onsite matrix element. The information written to the output is restricted such that only hoppings with  $|t| > \mathbf{WF\_ham\_threshold}$  are considered and only for  $|T| \leq \mathbf{ham\_cutoff}$ .

4. After this the output on the reciprocal grid is performed. Here it is important to realize that the Hamiltonian in  $k$ -space is written, which corresponds to the Bloch sums of the hoppings Eq. (6) **after** the cut off of hoppings for which  $|T| > \mathbf{ham\_cutoff}$  and  $|t| < \mathbf{WF\_ham\_threshold}$ . The resulting Hamiltonian is written to `+hamongrid`. See the `wannier.f90` source code for the order of the data.
5. Now, `+WF_coefficients` is written, which contains information about the contributions of the FPLO orbitals to the WFs Eq. (7). This file shows only contributions, which are larger than **WF\_coeff\_threshold** in `=.wandef`.
6. At the end of the module the WF are written to disk on a real space grid. There will be data files `opendxWF.dx` and `wfdata...` and control files `WF.net|cfg`. The later can be use to visualize the data using `opendx`:

```
dx -image WF.net
```

Beware that some standard installations of `opendx` are a bit buggy. For instance, it might work better if num-lock is turned off.

7. After this FPLO pauses with the message

```
CTRL_C for abort, enter for next trial
```

Here, one can type CTRL-C to stop, or one modifies the **wandefs** (in another window/editor) and hits enter to re-run the module, with re-scanning `=.wandef` but without re-reading `+wancoeff` (which is a slow read, since large file).

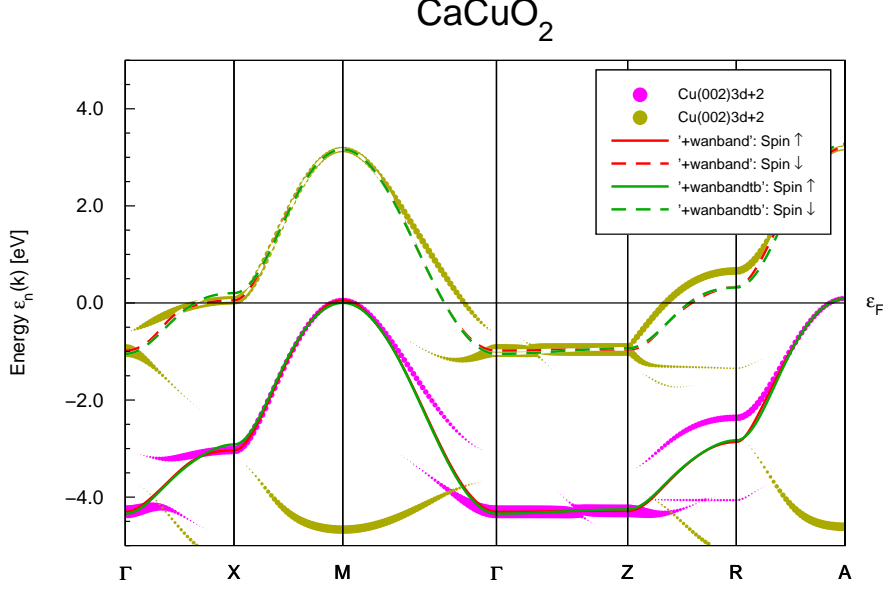


Figure 2: Wannier fit for  $\text{CaCuO}_2$

Figure 2 shows the result of the  $\text{CaCuO}_3$  example. Dark yellow and magenta are the  $3d_{x^2-y^2}$  fat bands as produced by FPLO. Red shows the full Wannier function transformation (without cutoffs) from the file `+wanband` and green shows the cutoff model WF bands including cutoffs. One can nicely observe how the WF bands interpolate the hybridization gap between  $\Gamma$  and X, and M and  $\Gamma$ .

## 2.1 Files

There are three band structure files produced.

**+wanband** The Bloch-sums of the WF Eq. (2) are obtained by doing the unitary transformation in  $k$ -space ( $U$ ). The Hamiltonian in WF-Bloch basis in  $k$ -space Eq. (5) is directly related to the WF-Bloch sums and hence is obtained straight from the  $U$ -transformed KS functions and Hamiltonian. The resulting  $k$ -dependent Hamiltonian has the dimension of the WF-basis defined by the user. It can be diagonalized to get the band structure belonging to the WF model. The result of this is written to `+wanband`. If the WFs describe an isolated band complex (one needs as many WFs as there are KS bands in the band complex) then the `+wanband` band structure must coincide with the bandstructure of this band complex in the full FPLO band structure plot. Deviations are possible, if there are not enough  $k$ -points in the FPLO-SCF calculation. This comes about since a discrete approximation to the  $k$ -integral in Eq. (1) defines WFs, which have periodic images with a period of the Born von Karman torus.

**+wanbweights** This are the fat-bands corresponding to `+wanband`. The band weights are determined with respect to the WF character in the corresponding bands. Let's make that clearer. In a local orbital basis like FPLO the overlap of the orbitals with the KS functions determines the fat-bands or how much of a certain orbital a band is made of. This concept can be applied to the WFs as well thinking of the WFs as a basis. So, how much of a particular WF makes up a band? This is given by the information in this file. Note, that the FPLO fat-bands and WF fat-bands do not coincide. That is the main reason why we do a WF analysis, see Figure 3

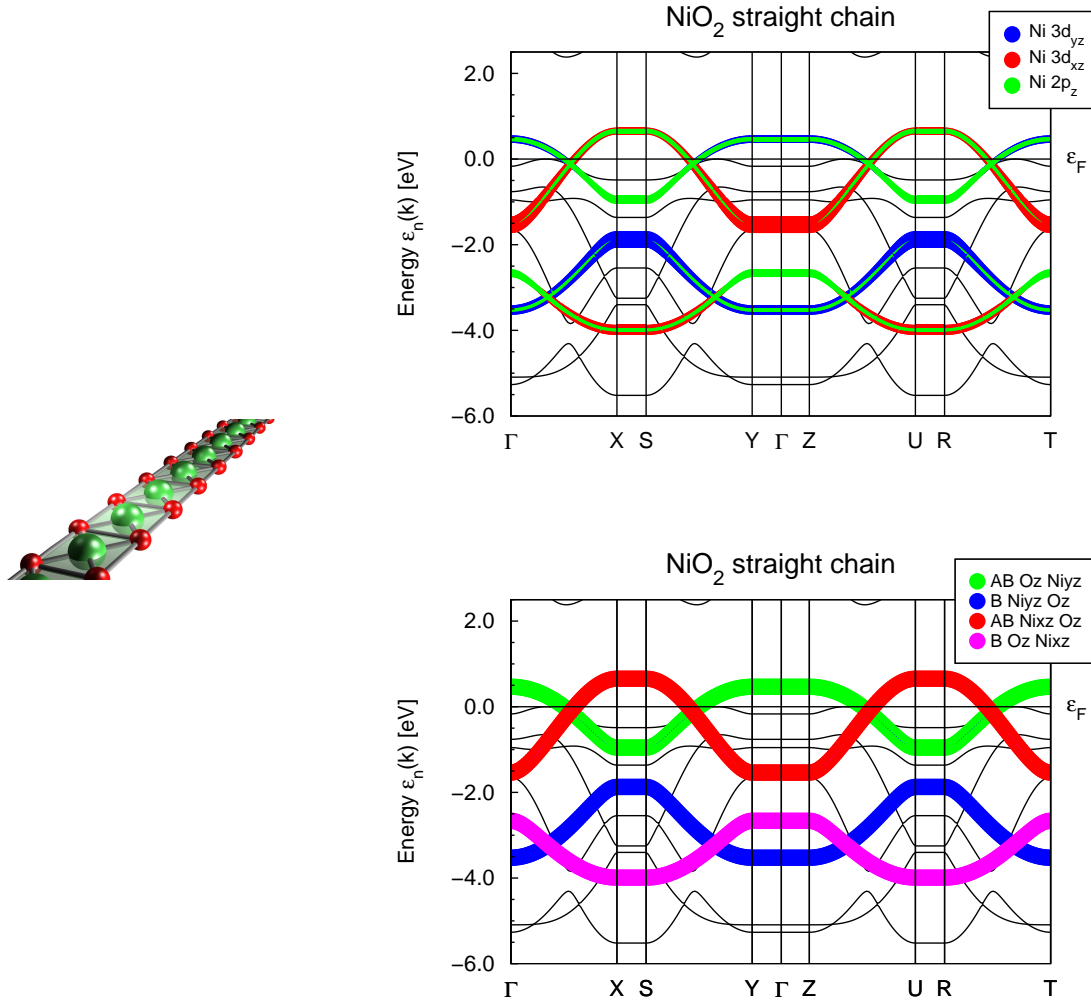


Figure 3: Comparison of FPLO fat-bands (upper panel) with WF fat-bands (lower panel) for a  $\text{NiO}_2$ -chain. Some WFs are linear combination of several orbitals, such that the WFs are centered between two orbitals. One can clearly see that in the WF basis each band has a pure character, i.e. the bands are completely decoupled by choosing a symmetry adapted basis (WFs).

**+wanbandtb** Now, we can go a step further and Fourier transform the WF-Bloch sums Eq. (2) into real space Eq. (3). This gives the actual WFs. In real space these WFs overlap and form hopping integrals Eq. (6). These hopping integrals can be cut off at some distance or if they are smaller than a threshold. This defines a modified Hamiltonian in real space. We transform this modified (model) Hamiltonian back into  $k$ -space, diagonalize it and get the band structure in this file. If the cut off and threshold is moderate the result should equal **+wanband** otherwise it can differ. Especially, for non isolated bands we can get non-analyticities in **+wanband**, which are unavoidable in some cases. It turns out that the real-space cutoff reduces these unwanted effects.

The band structure data files can be visualized using the bandplot-tools of fplo. Besides these files there are a few more:

**+WFstat...** contains the absolute values of the coefficients in Eq. (7) as a function of the distance of the corresponding orbitals from the Wannier function center. Plot this file (e.g. `xfbp ./+WFstat...`) to have an idea of the localization of the WFs.

**+T...** contains the hoppings from a given WF to the neighbouring WFs corresponding to Eq. (6) as a function of the distance between the WF centers.



## 2.2 Keywords

The keywords in `=.wandef` are explained in the following.

**doit** switch the WF module on.

**wandef** start a definition of a single Wannier function.

**on/off** make the corresponding **wandef** active/inactive.

**emin** the lower bound of the energy window (two numbers if spin-polarized).

**emax** the upper bound of the energy window (two numbers if spin-polarized).

**de** the width of the Gaussian tail above and below **emax/emin** defining a smooth energy window (two numbers if spin-polarized).

**delower/deupper** like **de**, but for the lower/upper end of the energy interval. Note, that **de** will overwrites **delower/deupper** if specified after the latter!

**ubands/lbands** band indices, which specify the maximum and minimum band to be contained in the projector (energy window). (two numbers if spin-polarized)

**contrib** add an FPLO orbital to the **wandef**. There can be several **contribs** in one **wandef**, one after the other.

**site** the site number of the orbital according to FPLO output.

**name** an arbitrary name to identify the WF.

**difvec** the distance of the contributing orbital from the WF center. The **difvec** of the first **contrib** to a **wandef** together with the **site** of the first **wandef** implicitly define the WF center. The choice of the WF centers decide whether the WF is real or not. WF must be defined such that they are real. E.g. a single orbital/contrib WF should always have **difvec** 0 0 0.

**xaxis** the local x-axis expressed in global coordinates

**zaxis** the local z-axis expressed in global coordinates

**orb** the orbital (e.g. 2s+0 or 3d-1). In full-relativistic mode the orbitals can either be pseudo non-relativistic projections denoted by “3d-1 up” or “3d-1 dn” or spherical spinors denoted by 3d3/2-1/2 or 3d5/2+3/2.

**fac** a weight factor, determining the relative weight of the orbital/contrib in cases of multiple-contrib wandeefs. The weights need not be normalized, this is taken care of.

**coefficients\_format** can be **bin** or something else. Since version 14.00, the file **+wancoeff** can be converted into binary format for faster loading. After the first FPLO run with a valid **=.wandef** present, **+wancoeff** will have been created. A rerun will start the WF creation process. If this option is set to **bin** the data file will be converted into binary format, if not already done. On any further run the binary file **+wancoeffbin** will be read instead of **+wancoeff**. This is faster. The user has to take care of deleting **+wancoeffbin** whenever **+wancoeff** got changed due to settings changes by the user.

**ham\_cutoff** restricts output of the real-space WF Hamiltonian in standard output and **+T...** Also restricts the matrix elements used in creating **+hamongrid**.

**WF\_coeff\_threshold** restricts the output of coefficients in **+WF\_coefficients**.

**WF\_ham\_threshold** restricts the output of real-space Hamiltonian in standard output and **+T...** and restricts the hoppings used in creating **+wanbandtb** and **+hamongrid**.

**WF\_write\_coeff\_stats** can be **on/off** and triggers the output of the files **+WFstat...**

**ham\_write\_t\_stats** can be **on/off** and triggers the output of the files **+T...**

For the output of the WFs on the real space grid (visualization) we have to define a grid

**WF\_grid\_basis** can be **conv/prim**. This defines the basis  $\underline{B} = \begin{pmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{b}_3 \end{pmatrix}$  of the box-like grid. We can use the conventional or the primitive basis vectors.



**WF\_grid\_directions** in terms of the basis we can define three vectors forming the rows of  $\underline{V} = \underline{DB}$ . These three directions span the grid-box. The input here is the matrix  $\underline{D}$ .

**WF\_grid\_subdivision** subdivide the box along the directions  $\underline{V}$  accordingly.

**WF\_grid\_origin** put the origin of the box here. If this keyword is commented out (e.g. **xWF\_grid\_origin**) the box will be centered around the WF center.

For the output of the WF Hamiltonian on the  $k$ -space grid (file **+hamongrid**) we have to define a grid

**k\_grid\_basis** the basis  $\underline{B}$  in reciprocal space. (Note the reciprocal relations between bcc and fcc and the like.)

**k\_grid\_directions** the directions  $\underline{V} = \underline{DB}$  defined here by giving  $\underline{D}$ .

**k\_grid\_subdivision** subdivisions, see above.

**k\_grid\_incl\_periodic\_points** can be **on/off**. The Hamiltonian in  $k$ -space is periodic. We can include or exclude the periodically equivalent points at the boundary of the box.

## 2.3 Definition of real spherical harmonics

We define real spherical harmonics  $Y_{lm}$  with magnetic qn. numbers  $m = -l, \dots, l$  as

$$Y_{lm}(x, y, z) \propto P_l^{|m|}\left(\frac{z}{r}\right) \begin{cases} \sin(|m|\varphi) & m < 0 \\ \cos(|m|\varphi) & m \geq 0 \end{cases}$$

The sin/cos can be expanded according to the addition theorems, e.g.  $\sin(2\varphi) = 2 \sin \varphi \cos \varphi$ . Using  $\sin \varphi \propto y$  and  $\cos \varphi \propto x$  we get  $\sin(2\varphi) \propto xy$ . Additionally we need  $P_l^{|m|}\left(\frac{z}{r}\right) \propto$  polynomial of degree  $l - \underline{m}$  in  $z$ . Thus,

$$Y_{2,-2} \propto P_2^2\left(\frac{z}{r}\right) \sin(2\varphi) \propto xy$$

$$Y_{2,-1} \propto P_2^1\left(\frac{z}{r}\right) \sin(\varphi) \propto zy$$

$$Y_{2,0} \propto P_2^0\left(\frac{z}{r}\right) \cos(0\varphi) \propto z^2$$

$$Y_{2,1} \propto P_2^1\left(\frac{z}{r}\right) \cos(\varphi) \propto zx$$

$$Y_{2,2} \propto P_2^2\left(\frac{z}{r}\right) \cos(2\varphi) \propto x^2 - y^2$$

For more specific information on the polynomial in  $z$  one has to look up the associated Legendre polynomials  $P_l^{|m|}(z)$ .

## 2.4 Problems

The number of **k**-points used in the SCF calculation influences the Wannier function quality. If the **k**-mesh is not fine enough, the band structure determined by diagonalizing the WF Bloch Hamiltonian Eq. (5) (**+wanband**) will not coincide with the corresponding bands of the full FPLO band structure (**+band**). (Coincidence can of course only happen anyway, if the WFs describe an isolated band complex.)

If the real space Hamiltonian cutoff is smaller than the extend of the WFs the fitted Wannier bandstructure (**+wanbandtb**) will not coincide with the full WF band structure. This may also be due to a bad Wannier definition, which is not localizing the WFs sufficiently (check energy window, use molecular orbitals (several **contribs** per **wandef**) instead of simple orbitals).

There might be unnatural spikes at certain  $k$ -points in the WF band structure. This usually means that the energy window is too narrow and that the character of  $\chi$  is only large outside the window at the corresponding  $k$ -points.

If slabs or chains are calculated, there is an artificial periodicity in the irrelevant directions. If not enough SCF **k**-points are used in these directions, we get influences of the artificial periodic replica. Hence, either the vacuum spacing is very large, which is detrimental for other parts of the code, or one has to use sufficiently many points in the irrelevant directions. (Only some experimenting can tell.)

## 2.5 Saving memory (and time)

In order to reduce the size of `+wancoeff` the user can set the FEDIT option “restrict bands to window” in the bandplot submenu. Then lower and upper energy bounds in the same submenu defines the bands, which will be written to `+wancoeff` for all data contained in this file.

Use the **coefficients\_format** option in `=.wandef` in order to convert `+wancoeff` to binary format.

## 3 Examples

### 3.1 Hexagonal: $sp^2$ , Graphene, $MgB_2$

Here we analyse the Wannier function choices for a hexagonal lattice with essentially two atoms per unit cell. The main issue is to understand the symmetry considerations.

Have a look at the accompanying example directories for  $MgB_2$ .

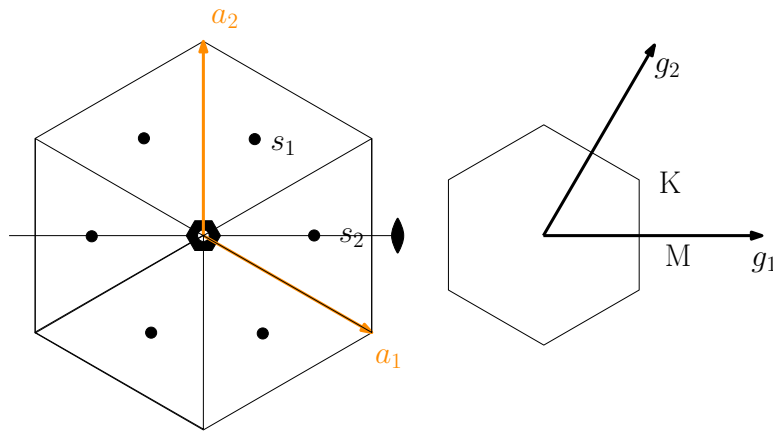


Figure 4: Real and reciprocal cell of graphene,  $MgB_2$

Figure 4 shows the basic lattice structure. We have a unit cell given by

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{pmatrix} = \underline{\mathbf{A}} \mathbf{E}, \quad \mathbf{E} = \begin{pmatrix} \mathbf{e}_x^T \\ \mathbf{e}_y^T \\ \mathbf{e}_z^T \end{pmatrix}, \quad \underline{\mathbf{A}} = \begin{pmatrix} a_H & 0 & 0 \\ 0 & a_H & 0 \\ 0 & 0 & c_H \end{pmatrix} \begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and two sites  $\mathbf{s} = \underline{\mathbf{s}}^T \underline{\mathbf{A}}$ ,  $\underline{\mathbf{s}}_1 = (\frac{1}{3} \frac{2}{3} 0)$ ,  $\underline{\mathbf{s}}_2 = (\frac{2}{3} \frac{1}{3} 0)$ . The symmorphic space group is 191 with a maximal point group  $D_{6h}$  at (000) with generators  $C_6(z)$ ,  $C_2(0)$  and  $I$  (indicated in Figure 4). Clearly  $C_6 \mathbf{s}_1 = \mathbf{s}_2$ . The local pointgroup at the two sites is  $D_{3h}$  with generators  $C_3$ ,  $C_2(0)$  and  $\bar{6} = IC_6$ . For the sake of definiteness let's define operations. We understand the operation  $g$  to actively transform the basis vectors  $\mathbf{e}_{x,y,z}$  like

$$g \mathbf{E}^T = \mathbf{E}^T R(g)$$

where  $R(g)$  is the 3D representation matrix. For example

$$g = C_6 = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The coordinates  $\underline{\mathbf{r}}$  of a vector  $\mathbf{r} = \mathbf{E}^T \underline{\mathbf{r}}$  then transform with the transposed matrix:  $\underline{\mathbf{r}}'^T = \underline{\mathbf{r}}^T R(g)^T$  (proof:  $\mathbf{E}^T \underline{\mathbf{r}}' = g \mathbf{r} = \mathbf{E}^T R(g) \underline{\mathbf{r}}$ ). A set of functions of the coordinates (orbitals) transforms as

$$g f_m(\underline{\mathbf{r}}) = f_m(g^{-1} \underline{\mathbf{r}}) = f_m(\underline{\mathbf{r}}^T R(g)) = \sum_n f_n(\underline{\mathbf{r}}) D_{nm}^f(g)$$

where  $D_{nm}^f(g)$  are the representation matrices of the group in the space spanned by  $f_m$ . In other words the orbitals  $gf_m(\underline{r})$  are also actively transforms like the cartesian basis vectors  $e_{x,y,z}$ .

In our examples we have a light  $p$  element sitting at the two sites of lower than maximal symmetry. The essential basis atoms are one  $s$  and three  $p$  orbitals. Figure 5 shows the band structure of  $\text{MgB}_2$ , where the boron bands are highlighted. (The Mg bands do not play a big role in this energy window.)

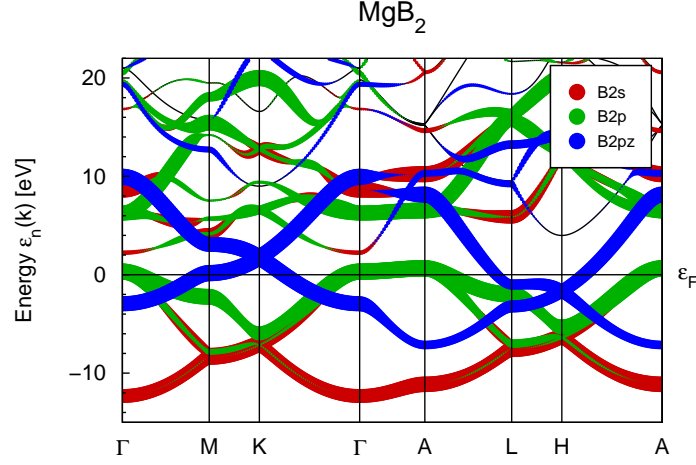


Figure 5: Bandweights of boron in  $\text{MgB}_2$

The irreducible representations of the site symmetry  $D_{3h}$  make  $p_{x,y}$  an  $E''$  doublet and  $p_z$  a  $A_2''$  singlet, while the  $s$ -orbital represents an  $A_1'$  singlet. In total we have 8 orbitals and hence bands. Due to the symmetry the  $p_z$  bands largely decouple from the other orbitals along high symmetry lines in the  $k_z = 0$  and  $k_z = \frac{\pi}{c_H}$  planes. At general points they off course couple. Furthermore, these bands cross the Fermi level, while the  $s$  and  $p_{x,y}$  bands form bonding and anti-bonding band complexes of three bands each separated by a gap. The bonding band complex has exactly three bands, while the AB complex shows a high degree of band entangling with other bands. The two  $p_z$  bands cannot be separated into two distinct bands because of their Fermi surface and symmetry. However the planar-orbital bands can be represented by different kinds of Wannier functions. The simplest way of defining WFs is to use the atomic  $s$ ,  $p_{x,y}$  orbitals sitting at the two sites. This more or less reproduces the FPLO band characters. Another alternative is to try to find WFs for the bonding bands (anti-bonding) bands only). There are essentially three bands in each band complex and it is immediately clear that atom centered functions cannot fulfill the crystall symmetry due to the number of sites (two). The bands must be combinations of orbitals from both sites and hence an odd (3) number of orbitals from an even number of sites (2) cannot be symmetric. We discussed the irreducible representations of the orbitals in the site symmetry above. We have a singlet ( $s$ ) and a doublet ( $p_{x,y}$ ) (as is reflected in the degeneracies at the  $\Gamma$ -point) but all three orbitals are mixed in the bonding bands (no further decoupling).

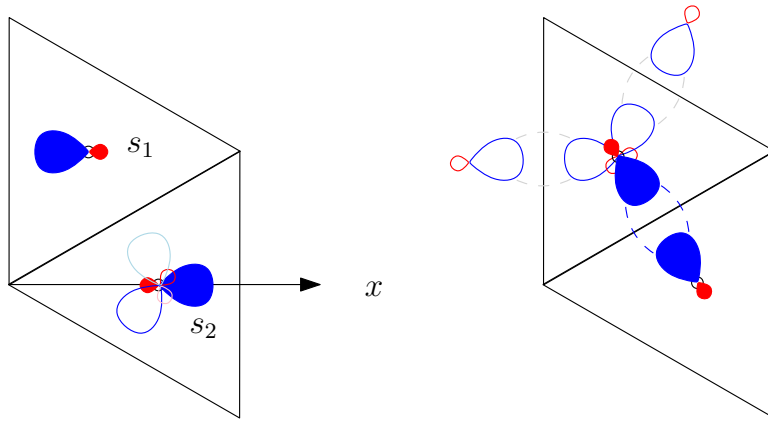


Figure 6:  $sp^2$ -hybrid basis. The left panel shows one  $sp^2$ -orbital at both sites (filled) and the remaining two  $120^\circ/240^\circ$ -rotated orbitals at  $s_1$  (open). The right panel show a bonding bond-centered combination (full) and two symmetry related combinations (open).

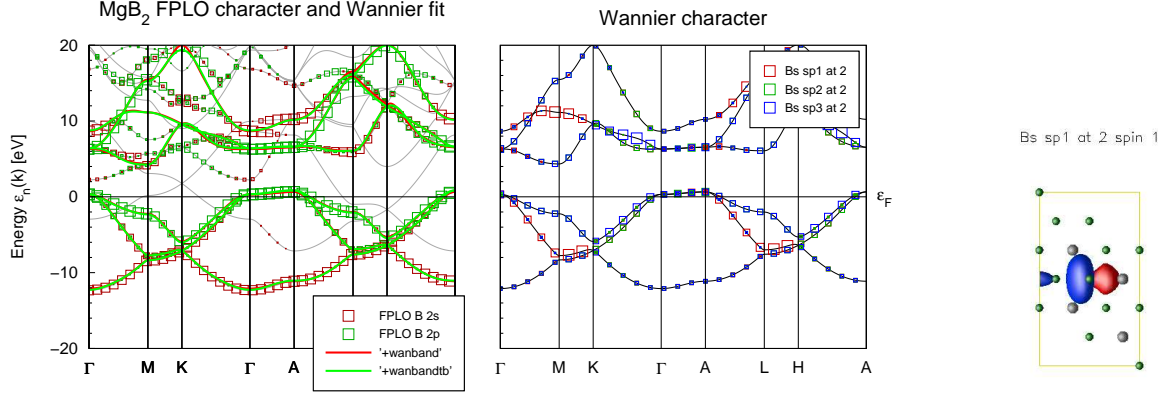


Figure 7: Wannier function fit with 3  $sp^2$ -orbital per site.

Unfortunately, there is hardly a simple linear combination of say the  $p_{x,y}$  orbitals from both sites, which will fulfill the full crystal symmetry. Hence, we have to resort to reducible representations aiming at constructing as highly symmetric orbitals as possible. The well known answer is of course  $sp^2$  hybrids. We introduce the following orbitals at each site

$$\begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{pmatrix} = \frac{1}{\sqrt{3}} \begin{pmatrix} s + \sqrt{2}p_x \\ s + \sqrt{2}C_3p_x \\ s + \sqrt{2}C_3^2p_x \end{pmatrix}$$

which are depicted in Figure 6.

Obviously,  $C_3p_x$  is not parallel to  $p_x$  and hence contains some  $p_y$ . On the other hand  $(1 + C_3 + C_3^2) = 3P_{||}$  where  $P_{||}$  is a projector onto the rotation axis. For in-plane objects  $P_{||}p_{x,y} = 0$ . This consideration shows that the  $p$ -parts of  $\Phi_{1,2,3}$  are linearly dependent, however, with the  $s$ -admixture they span the same space as  $s$  and  $p_{x,y}$ . The factors are chosen such that  $|\Phi_i|^2 = 1$  and that  $\sum_i |\Phi_i|^2$  contains one  $|s|^2$  density and two  $|p|^2$  densities.

Now, we can put three  $sp^2$  hybrids onto each site (rotated by  $180^\circ$  at the second site, which is not really necessary) and clearly they possess maximum crystal symmetry. (Figure 6, left panel). The symmetry relations are

$$\begin{aligned} C_6 \Phi_{s_2,1} &= \Phi_{s_1,3}, \dots \\ C_3 \Phi_{s,1} &= \Phi_{s,2}, \dots \\ C_2(0) \Phi_{s_2,1} &= \Phi_{s_2,1}, \quad C_2(0) \Phi_{s_2,2} = \Phi_{s_2,3}, \dots \\ I \Phi_{s_2,1} &= \Phi_{s_1,2}, \dots \end{aligned}$$

The resulting Wannier function fat bands are shown in Figure 7.

The advantage of this high symmetry is that we now can create linear combinations of one  $sp^2$  from each site at the bond centers and by construction this generates symmetry related additional two similar combinations (Figure 6, right panel). This were not possible if we used the irreducible  $s$  and  $p_{xy}$  orbitals. In that way one can separate the bonding from the anti-bonding orbitals. This is an examples for the fact that in many covalent  $p$ -electron systems the bonding/anti-bonding bands must be described by bond-centered Wannier functions. (Carbon chain:  $sp^1$ , diamond  $sp^3$ ). Once we have separated the bonding Wannier functions, we can of course produce either a separate fit for the bonding or anti-bonding bands respectively (Figure 8) or a fit for bonding and anti-bonding bands together (however with clear energy separation of the orbital).

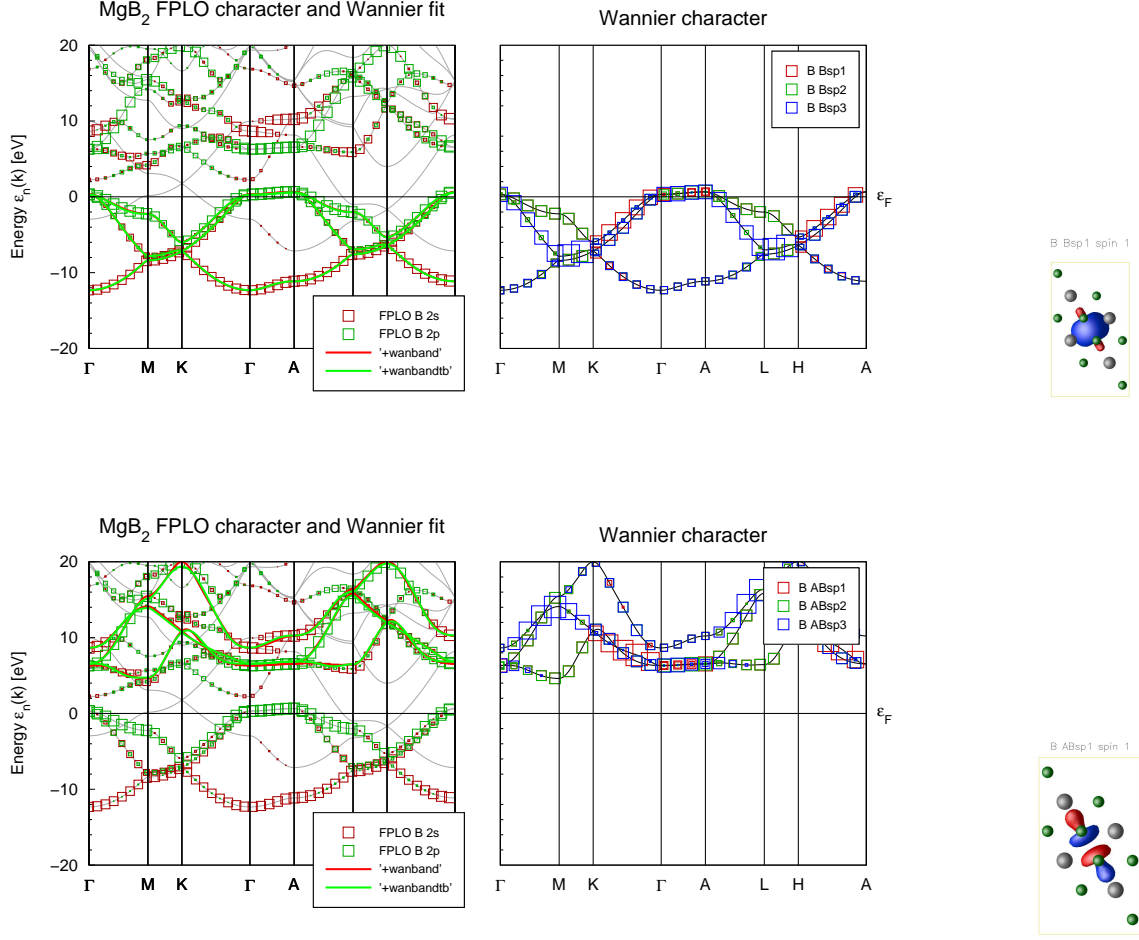


Figure 8: Bonding/anti-bonding bond centered Wannier function fits.

One last consideration concerns the choice of energy windows. For the entangled anti-bonding bands to be fitted better we put a narrow energy window onto the occupied bonding bands but allow for a large Gaussian tail at the upper side of the energy window. This pulls the majority weight from the well separated bonding bands and magically adds the missing anti-bonding weights. If we extend the main energy window to encompass the anti-bonding bands it would pull in weights from higher lying bands, which also show  $s$  and  $p_{x,y}$  character, which in turn pulls the fitted bands up in energy.