

Unfolding band structure/ bcc-Fe versus DO₃ FeCo/FeAl

Klaus Koepernik

July 4, 2022

Contents

1	Introduction	1
2	bcc Iron	2
3	DO₃ Fe₃Co	6
4	Interlude	11
4.1	Fast argument	11
4.2	Slow argument	12
4.3	Could have been so much easier	13
5	Back on track: the BZ mapping	14
5.1	The smart way	14
5.2	The hard, more general way	15
6	The supercell bandstructure on the normal cell path.	17
7	The unfolding mapping	18
8	To wrap it up: DO₃ Fe₃Al	20

1 Introduction

In this tutorial you will learn to use the band unfolding tool. This task is a bit involved due to the nature of the topic. There is a rather detailed documentation of the background of band unfolding in `FPL0.../DOC/MANUAL/unfolddoc.pdf`. Have fun.

The solution will be found in the `solution` directory. You will work in the `task` directory.

We will setup a bcc iron calculation and from it create a DO₃ structure, in which we can replace one of four iron by either Co or Al. The resulting supercell band structure will be unfolded onto the bcc Brillouin zone (BZ) and compared to the original bcc-Fe calculation. You will see that although Al presents a rather strong perturbation of the iron lattice, similarities are still to be

found. This is a rather extreme and artificial example and was chosen for its speed. Yet, unfolding is often rather useful in real-world applications.

In principle unfolding consists of finding a mapping of all sites (or as much as possible) of a supercell (SC) onto corresponding sites in a normal (target) cell (NC). The mapping is used to create band weights, which in turn are fat for bands that survive unfolding and are small for the extinguished bands. It is assumed that the SC can be obtained by approximately filling the space with f normal cells, followed by some atom substitutions and possibly some atom movements. However, roughly the two lattices should be commensurate.

The normal cell shall have n sites s_i , $i \in [1, n]$ and the SC $N = fn$ sites S_j , $j \in [1, N]$. Now, the requirement $N = fn$ can be relaxed if some atoms are missing (see `unfolddoc.pdf`), in which case we just imagine the missing atoms among the set of S_j . We assumed that the lattice vectors r of the NC lattice can be approximately mapped on the lattice vectors R of the SC lattice in such a way that the sets of all sites in both lattices fulfill $\{r + s_i\} \approx \{R + S_j\}$. This means that for each SC S_j we find a NC site s_i and lattice vector r_j such that $S_j = s_i + r_j$, which maps f SC sites S_{s_j} onto one single s . Repeat that for all s and you get a listing like e.g.:

```
#NC-site SC-sites
s1      S2 S4 S7 S8
s2      S1 S3 S5
s3      S6
```

where 8 SC sites are identified with 3 NC sites according to the translational symmetry of the NC lattice (can be approximately translationally connected). Fig. 1 shows a realization of such a mapping. In this case $f = 4$. However, we have an atom substitution on the pink sites. Hence only three black SC sites can be mapped onto the one black NC site. The pink sites is left as a single site and it's unfolding weights will be identical to it's SC band weights (so no effect at all).

That is all what is needed. The next step is that one usually wants to plot the band structure along a path in the NC BZ. However, the SC calculation is setup in the SC geometry. The SC can be related to the NC by rather complex transformations. So, the second ingredient is to find the special points of the NC BZ in the SC BZ.

Once you understood the principle the main task is to map the two BZs and to determine the equivalences of all sites in both unit cells. Let's start.

2 bcc Iron

- Go to the `task` directory and create a directory called `bcc` and change into it

```
mkdir bcc
cd bcc
```

- Execute

```
fedit22.00-62-x86_64
```

- Go to the `symmetry-menu` and set the `spacegroup` to 229, the `lattice constants` to 5.4,, and the first `Wyckoff position` to Fe. Call `update` (hotkey `+`) and go back to the `main-menu` (hotkey `x`). Now, `Quit/Save`.
- Execute

```
fplo22.00-62-x86_64 | tee out
```

- Go back into `fedit` and switch on `bandplot` \triangleright `bandstructure plot` and `bandplot` \triangleright `weights`. Back to `main-menu` and `Quit/Save`.

- Re-run `fplo`.

- Execute

```
faddwei22.00-62-x86_64
```

and answer `y`.

- Edit `.addwei` using your favorite editor (`vi .addwei`, `emacs -nw .addwei`, ...) to obtain:

```
weightinfile +bweights
weightoutfile +bsum

name Fe
  atom Fe sites 1 orbitals 4s 4p 3d
```

- Execute

```
faddwei22.00-62-x86_64
```

```
xfbp +bsum
```

You got:

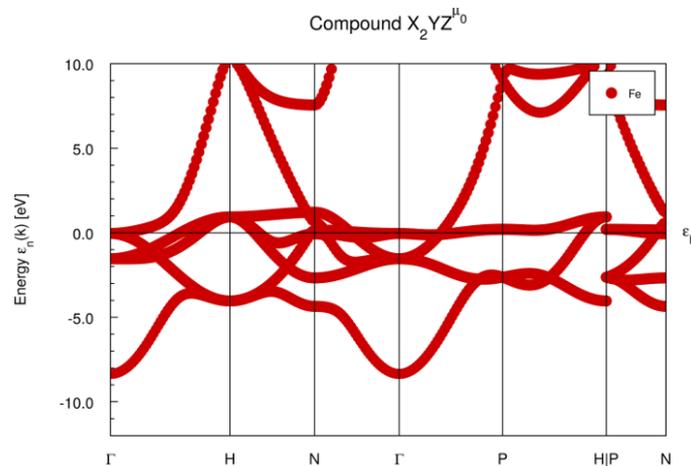
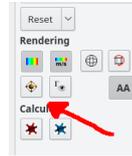


Figure 2: bcc fatbands

For training we will now look at the BZ and remind the user of the Fermi surface module of `xfplo`. The main purpose of this module is to visualize the Fermi surface. But you can also use it to setup the path through high symmetry points. There are several ways of invoking this mode. For reference we list them in the following (don't do anything yet).

1. `xfplo -fs`: opens the Fermi surface window (FSW) with `show/hide Fermi surface` button being switched off



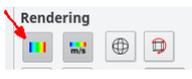
If the screen is blank try this: Hit `Ctrl-C` or click the  button or select the menu item `Plot > Calculate` → an empty BZ, the conventional frame and the legend will appear.

2. `xfplo =.in`, followed by `Activity > Fermi Surface` will also open the FSW.
3. `xfplo =.xef`: loads and executes what ever was saved previously into `=.xef`. This will switch the `show/hide Fermi surface` button according to the settings in the file.

Now back to the tutorial:

- Execute

```
xfplo -fs
```

- Hit `Ctrl-C`. Switch on the cartesian frame  (hotkey: `a a`). Switch of the legend button  and zoom properly (`mouse wheel` or `right mouse click+mouse move`).
- Now, select `Plot > High symemtry points` (`Ctrl-H`) to open the high symmetry points dialog (HSPD) Fig. 3

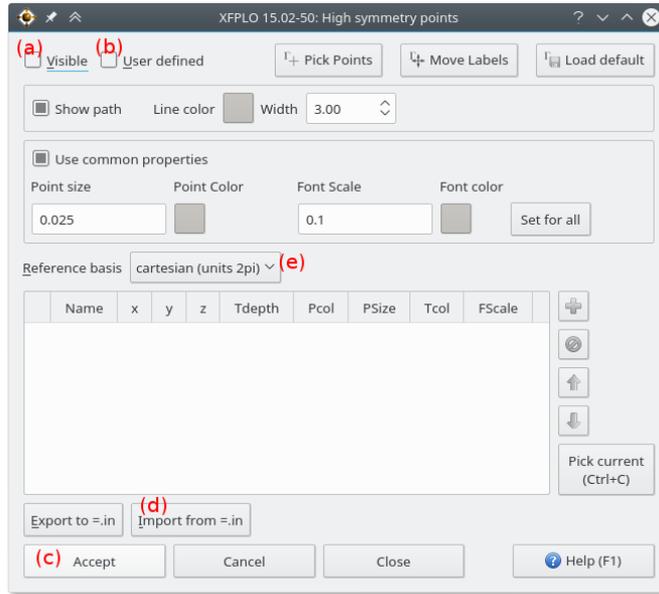


Figure 3: High symmetry points dialog

- Click the **Import from =.in** button (Fig. 3(d)). Press **enter**.
- Check the **Visible** checkbox (Fig. 3(a)) and the **User defined** checkbox (Fig. 3(b)).
- Again press **enter** or click the **accept** button (Fig. 3(c)). You should see something like Fig. 4. You will notice that the point coordinates are displayed in the HSPD according to FPLO convention $\mathbf{k} = \frac{2\pi}{a}(k_x, k_y, k_z)$. Other reference bases are available via the **Reference basis** combobox (Fig. 3(e)).

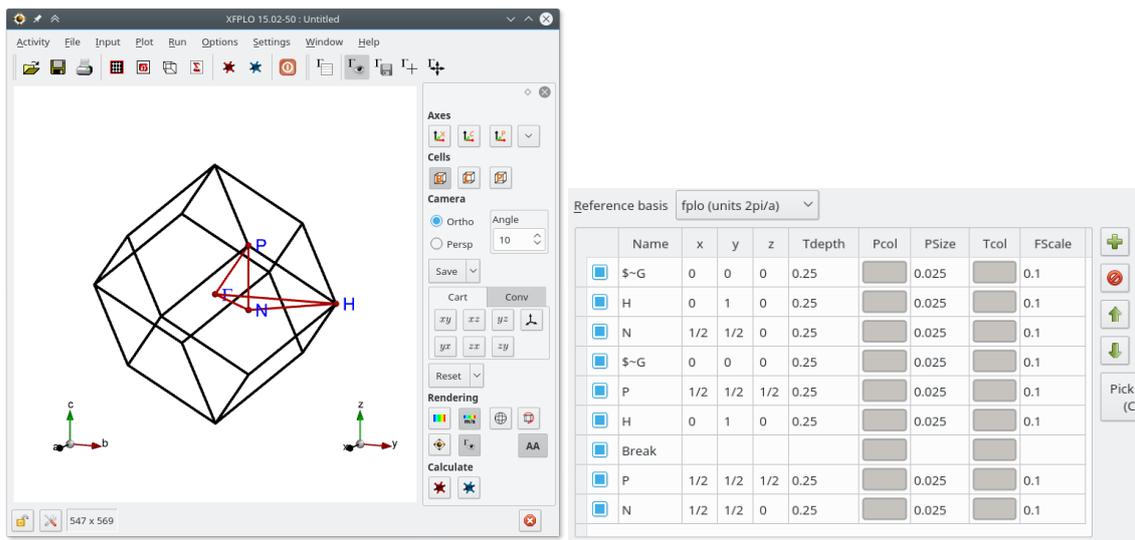


Figure 4: bcc BZ

- Optionally, you can change the point/font colors, pick new points, export the points to =.in or move the labels.

The moving works like this (picking is similar)

- hit the `move labels` button once
- drag the labels in the main view
- hit the same button a second time or press `escape` in the main view to stop the move-mode

The label position/color etc. would only be saved in `=.xef`.

Note, that each label is printed on the screen. So, if you want to move them and produce a picture just change the name of the superfluous labels in the table to an empty string or delete the whole point, if it is not needed for the path (if you want the path, that is).

Also, occasionally, read the help screen in the HSPD.

- Finally, just for fun close the HSPD and in the FSW save the current state: `Ctrl-S` or `File > Save`, accept the default filename `=.xef`. Exit the FSW (`Ctrl-Q`, `Activities > Quit all`)
- For reference, please execute

```
xfplo =.xef
```

- and close it again.

A note on activities. As you might have noticed, several activities can be open at the same time. This is used by some tools to communicate between them (e.g. Fatband editor and structure view). Each activity has a `File > Close` (`Ctrl-W`) which closes this activity. The menu entry `Activities > Quit all` (`Ctrl-Q`) closes all connected activities.

3 DO₃ Fe₃Co

Now, we will derive a supercell from the `bcc` setup.

- Go up into the `task` directory and make a directory called `D03Co` and change into it. Copy the files `=.in` and `=.addwei` from `bcc` to the current directory (`D03Co`):

```
cd ..
# now we are in task
mkdir D03Co
cd D03Co
# we are in D03Co
cp ../bcc/= .in .
cp ../bcc/= .addwei .
```

- Execute

```
xfplo =.in
```

- Open `Input > Symmetry` and in the symmetry dialog click on the `Manipulate` button (Fig. 5(h)).

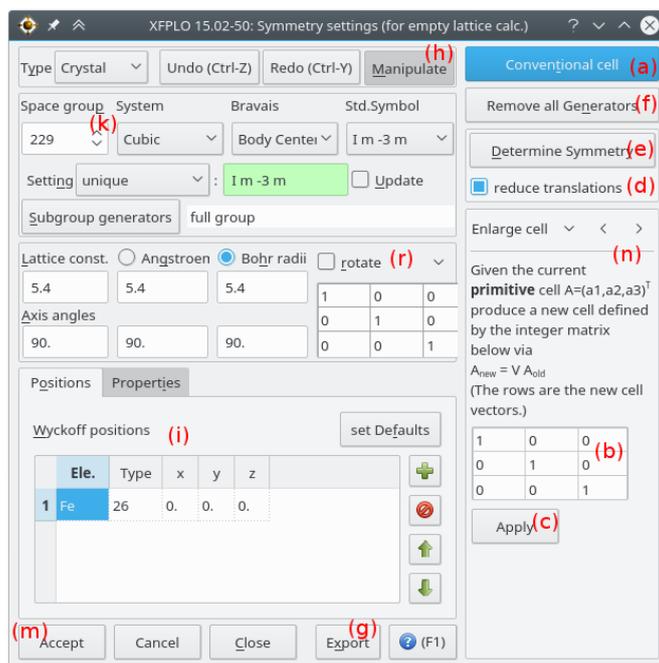


Figure 5: Symmetry dialog, with open manipulate extension.

- Have a look at the list of **Wyckoff positions** (Fig. 5(i)) and on the **Spacegroup** settings (Fig. 5(k)) .
- Now click the **Conventional cell** button (Fig. 5(a)), it changed the list of **Wyckoff positions** and the **Spacegroup** . We, now have a simple cubic setup.
- In the **enlarge-cell** table (Fig. 5(b)) fill the diagonal with 2 and click the **apply** button (Fig. 5(c)). We now have 16 sites and spacegroup 1.
- Uncheck the **reduce translations** checkbox (Fig. 5(d)) and hit the **Determine Symmetry** button (Fig. 5(e)). We now have simple cubic (spacegroup 221) and 5 atoms.
- Hit the **reset** button in the structure view (SV) (Fig. 6, arrow) and rotate the image with **left mouse click+mouse move** such that it looks something like Fig. 6.

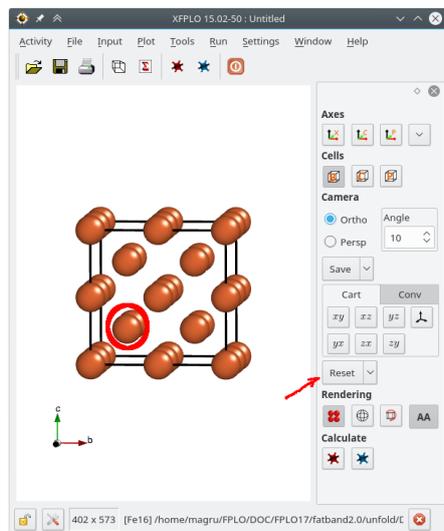


Figure 6: big iron cell

- In the SV click on the atom marked in the Fig 6. This will select the corresponding **Wyckoff position** (5) in the symmetry dialog. Change the **Wyckoff position name**: in the element column of the **Wyckoff positions** table click on the selected row (basically at Fe)

Wyckoff positions						set Defaults
Ele.	Type	x	y	z		
3	Fe	26	0	1/2	1/2	
4	Fe	26	1/2	1/2	1/2	
5	Fe	26	1/4	1/4	-1/4	

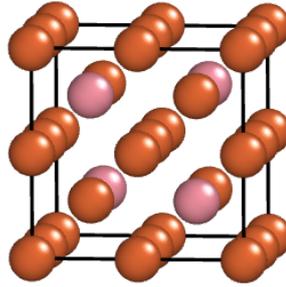
to give the keyboard focus to the table and type `co` followed by **enter**. Hit the **Accept** button (Fig. 5(m)) or hit **enter** again.

Now, you have 8 pinkish Co atoms.

- Hit the **Remove all generators** button (Fig. 5(f)) → back to 16 Wyckoff positions (8 of them Co).

Note, that we have space group 221, but only the identity in **Subgroup generators**. This way you can produce a lattice as defined by spacegroup/lattice-parameter/angles without using the group generators to create the sites from the Wyckoff positions. In other words the **Subgroup generators** can be used to reduce site symmetry without changing the lattice setup.

- Go back to the structure view and click on one Co atom. This will select the Wyckoff position in the symmetry dialog **Wyckoff positions** table. Change the element back to Fe and hit **accept**. Continue to change another three cobalt into iron to obtain the following tetrahedrally coordinated four Co atoms:



- In the symmetry dialog check the **reduce translations** checkbox (Fig. 5(d)) and hit the **Determine Symmetry** button (Fig. 5(e)). The result is spacegroup 225 (fcc) with three Wyckoff positions, two Fe and one cobalt. In the status bar of the SV it says Fe3Co1. That is our final structure:

Ele.	Type	x	y	z
1 Fe	26	-1/4	-1/4	1/4
2 Fe	26	0	0	0
3 Co	27	0	0	-1/2

- In the symmetry dialog click the **export** button (Fig. 5(g)). Chose **=.in** hit **enter**, read the message.
- Now, in the SV save the state to **=.xstr** (**Ctrl-S** or **File > Save**).
- Press **Ctrl-Q** and ignore warnings. We have input.
- As a confirmation reload the saved **=.xstr**.

```
xfplo =.xstr
```

- Close **xfplo**.

One note on the symmetry manipulation. The transformations which can be performed will produce new cells in some internally defined default orientation. This has to do with the fact that one has to somehow decide where to put the **a**, **b** and **c**-axes. If you want to keep the absolute orientation of the resulting crystal the same way as the original crystal, please check the **rotate** (Fig. 5(r)) checkbox **before** any manipulations. This will record all necessary rotations of the resulting crystal in the table below **rotate**. If you follow this recipe the resulting crystal's orientation is unaltered by axis conventions. (This helps a lot with unfolding.) We however, did not use this, since in our case it does not really matter. You could repeat the structure creation process above using the rotate feature and at most you would get a simple axis permutation. Afterall, it is a cubic example. (Actually, I got the identity matrix). Also there are more options than **enlarge**, which you get when you use these arrows (Fig. 5(n)).

We will now perform a calculation in the new setup.

- In `fedit` switch off `bandplot > bandstructure plot`.
- Execute

```
fplo22.00-62-x86_64 | tee out
```

This will take about 1 min.

- It will not be converged. So re-run `fplo`! Now, it should have worked.
- Switch on `bandplot > bandstructure plot` in `fedit` and re-run `fplo`. When done have a look at the population analysis or the “Atom sites” table in the beginning of the output file (`out`). We have four sites, Fe 1—3 and Co 4.
- Edit `=.addwei` to look like this:

```
weightinfile +bweights
weightoutfile +bwsum

name Fe
  atom Fe sites 1..3 orbitals 4s 4p 3d
name Co
  atom Co sites 4 orbitals 4s 4p 3d
```

and

- run

```
faddwei22.00-62-x86_64
xftp +bwsum
```

You got Fig. 7 and should compare to Fig. 2.

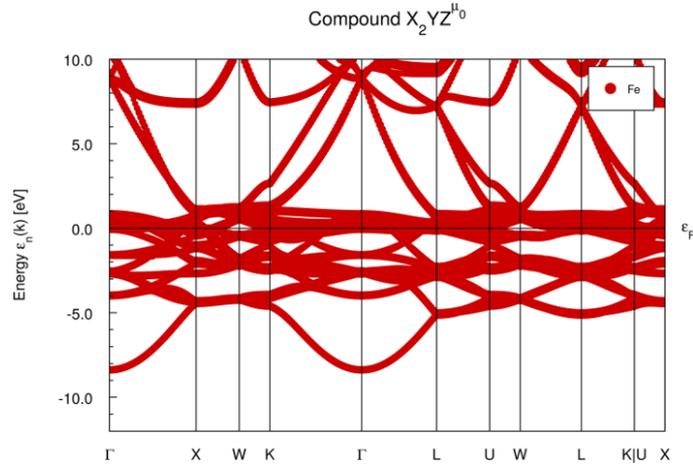


Figure 7: DO_3 Fe_3Co , fatbands

4 Interlude

4.1 Fast argument

We should take a break from clicking and think about the BZ mapping. We started with a bcc normal unit cell. This makes an fcc reciprocal cell. Remember the steps we took. We went from fcc to its conventional unit cell (simple cubic) by removing the centering. Then we doubled the conventional cell in all directions and reduced the symmetry to obtain an fcc cell with this doubled-lattice-constant conventional cell. Its reciprocal cell is bcc in a reciprocal conventional cell which now must be 2-times smaller in all directions. Hence, in the supercell setup, we need to define an fcc Brillouin zone (as the original reciprocal cell) in a conventional reciprocal cell of twice the reciprocal lattice constant of the SC setup. Have a look at Fig. 8.

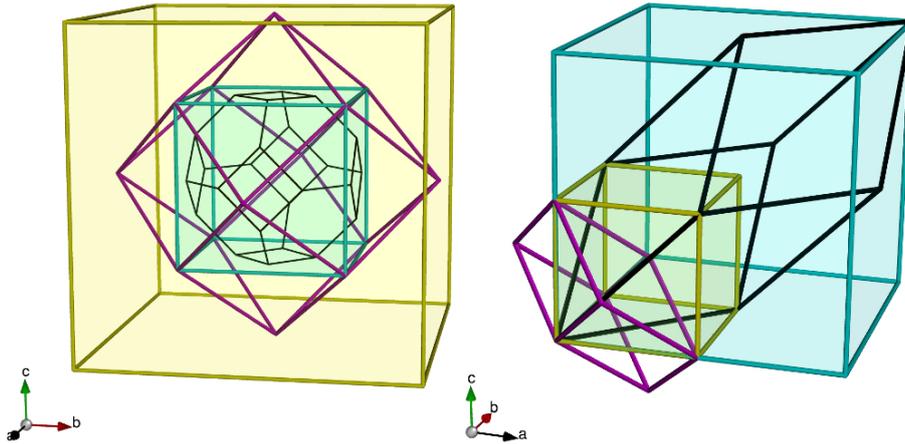


Figure 8: All cells in one picture. Left: the yellow cube is the NC conventional reciprocal cell. The magenta Wigner-Seitz cell is the NC BZ (reciprocal fcc). The blue-ish cube is the SC conventional reciprocal cell and the black Wigner-Seitz cell is the SC BZ (reciprocal bcc). Right: the same colors are used to depict the NC and SC primitive and conventional cells.

4.2 Slow argument

If this was too much here we do the math. The conventional cubic cell is

$$A_c = \begin{pmatrix} a & & \\ & a & \\ & & a \end{pmatrix}$$

where each row is a lattice vector. The primitive cell is

$$A_p = B_{bc}A_c$$

where

$$B_{bc} = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

is the body-centering matrix. The primitive reciprocal lattice is obtained by the inverse-transpose of the direct lattice:

$$\begin{aligned} G_p &= A_p^{-T} \\ &= B_{bc}^{-T} A_c^{-T} \\ &= B_{bc}^{-T} A_c^{-1} \end{aligned}$$

Note, that $A_c^T = A_c$ and that we do not use the 2π factor usually used in the definition of the reciprocal lattice vectors. It just hurts the eye. Furthermore, both inversion and transposition would reverse the order of matrices in a product: $(AB)^T = B^T A^T$. But we have a combined inverse-transpose, which restores the original matrix order. Lastly, you might have learned that the reciprocal of the bcc is an fcc lattice, but not quite. In fact

$$B_{bc}^{-T} = B_{bc}^{-1} = 2B_{fc} \tag{1}$$

$$B_{fc}^{-T} = B_{fc}^{-1} = 2B_{bc} \tag{2}$$

where

$$B_{fc} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

is the face-centring matrix. This factor of two is important. So, finally our normal cell reciprocal space is spanned by

$$G_p = B_{fc} (2A_c^{-1})$$

which is why we call two-times the reciprocal of the direct conventional lattice the reciprocal conventional lattice:

$$G_c = 2A_c^{-1} \tag{3}$$

Only then the centering relation $G_p = B_{fc}G_c$ holds. (Note that this factor of two also appears in the reciprocal of base-centered lattices, but only in the two centering directions. The axis opposite to the centered plane does not have such a factor.)

To build the BZ we need to find the planes which are perpendicular to lattice vectors and closest to the Γ -point, which lie half-way to the next lattice vector. A plane is defined by a plane equation with a normal \mathbf{n} and a distance d :

$$\mathbf{n}\mathbf{k} = d, \quad |\mathbf{n}|^2 = 1$$

A half-way plane perpendicular to a \mathbf{G} is then

$$\frac{\mathbf{G}}{|\mathbf{G}|}\mathbf{k} = \frac{1}{2}|\mathbf{G}|$$

One of the shortest reciprocal vectors in our case is $\mathbf{G} = \left(\frac{1}{2}\frac{1}{2}0\right)^T G_c$ (a centering vector in the fcc reciprocal lattice). When defining faces for a polyhedron in xfplo we can chose a basis. Typically we chose the conventional basis G_c . Then by definition this particular face is defined by the input $n = \left(\frac{1}{2}\frac{1}{2}0\right)$, $d = \frac{1}{2}$ or if you want to use another scale $\mathbf{G}' = \lambda \left(\frac{1}{2}\frac{1}{2}0\right)^T G_c = \lambda \mathbf{G}$ the following is equivalent

$$\begin{aligned}\mathbf{G}\mathbf{k} &= \frac{1}{2} |\mathbf{G}|^2 \\ \lambda \mathbf{G}\mathbf{k} &= \frac{1}{2\lambda} |\lambda \mathbf{G}|^2 \\ \mathbf{G}'\mathbf{k} &= \frac{1}{2\lambda} |\mathbf{G}'|^2\end{aligned}$$

which results in input $n = \lambda \left(\frac{1}{2}\frac{1}{2}0\right)$, $d = \frac{1}{2\lambda}$. We come back to this later.

Let's turn to the supercell. First, we transform the primitive normal cell to conventional

$$B_{bc}^{-1} A_p = A_c$$

then we double the size

$$2B_{bc}^{-1} A_p = 2A_c$$

and add the face-centering to obtain the new primitive lattice basis:

$$A'_p = B_{fc} 2B_{bc}^{-1} A_p = B_{fc} 2A_c$$

It's primitive reciprocal lattice is

$$\begin{aligned}G'_p &= A_p'^{-T} = B_{fc}^{-T} \frac{1}{2} A_c^{-T} \\ &= 2B_{bc} \frac{1}{2} A_c^{-1} \\ &= B_{bc} A_c^{-1}\end{aligned}$$

and hence it's reciprocal conventional must be $G'_c = A_c^{-1}$. If you compare to Eq. 3 you see that indeed the SC reciprocal conventional cell is half as big in all directions (missing factor 2). This gives $2G'_c = 2A_c^{-T} = G_c$ or $G_c = 2G'_c$. The original reciprocal fcc lattice can now be expressed in SC reciprocal conventional basis:

$$g_{fcc}^{NC} = B_{fc} G_c = 2B_{fc} G'_c$$

Our faces in SC conventional basis are hence spanned by all permutations of $n = (\pm 1 \pm 1 0)$, $d = \frac{1}{2}$ (or alternatively $n = (\pm \frac{1}{2} \pm \frac{1}{2} 0)$, $d = 1$).

4.3 Could have been so much easier

Last, we have a look at the list of high symmetry points found in the fedit bandplot menu in the original bcc calculation. The k-points are always given in units of $\frac{2\pi}{a}$. In the SC calculation we need to define them in units of $\frac{2\pi}{a_{SC}}$, with $a_{SC} = 2a$. No other transformations need to be taken into account, since the centering does not enter the definition of the k-points in the menu. Hence, an original k-point transforms as

$$\begin{aligned}k_{NC} &= \frac{2\pi}{a} (k_x k_y k_z) \\ &= \frac{2\pi}{a_{SC}} \frac{a_{SC}}{a} (k_x k_y k_z) \\ &= \frac{2\pi}{a_{SC}} (2k_x 2k_y 2k_z)\end{aligned}$$

All we had to do is to take the original points and multiply them by a factor of two. We will however not do this in order to teach more features of the programs. The path of transformations eluded to above might be useful in more involved cases. There might be additional rotations involved in the most general case, which we did not take into account in our little detour.

5 Back on track: the BZ mapping

We will show you two ways of creating the normal cell BZ inside the supercell setup. The first uses smart tools. The second is taking a stroll through the GUI and is more general. Use the first method if time is of the essence and come back for the second method when time allows it. It is teaching more GUI elements.

5.1 The smart way

We will use the fact that in our case the normal cell BZ in the supercell setup is identical to the normal cell BZ in the normal setup, since we did not have any rotations in the transformation between the two setups. If the supercell is derived by the symmetry manipulation tool of xfplo, rotations can be avoided by using the `rotate` (Fig. 5(r)) feature in the symmetry dialog during structure manipulation.

- So, go to the `bcc` directory and launch

```
xfplo -fs
```

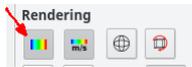
- open the Boundary dialog `Input > boundary (Ctrl-B)`.
- In the `Boundary Type` selectbox choose `User defined`.
- In the `reference basis` combobox (at the top of the dialog) select `cartesian basis vectors`
- Click on the arrow of the `Load` button to open a menu. Choose `Wigner Seitz Faces`.
- Click the `Save current` button.
- Close the dialog and the Fermi surface window (FSW).

This saves the normal cell (bcc) Wigner Seitz cell in the file `~/xfplo/userfaces.ini`. We are going to load it now.

- Change back into the `D03Co` directory.
- Execute

```
xfplo -fs
```

- Hit `Ctrl-C` if you have a blank screen.
- In the Fermi surface window (FSW) open the `Plot > Annotations` dialog and click on last item (“Primitive”) in left panel.
- click the `plus` button above the left panel or hit `Ctrl-I` or the `ins`-key.
- Hit `enter`. You should see a magenta box in the FSW.

- In the **Boundary type** selectbox choose **user defined**.
- Click on the arrow of the **Load** button to open a menu. Choose **User saved faces**.
- Hit **enter**, close the dialog, switch off the legend  and switch on the cartesian frame (a a). Finally, in the FSW save the state via **Ctrl-S** or **File > Save**.
You should see a magenta normal cell BZ together with the super cell Brillouin zone, Fig. 9.

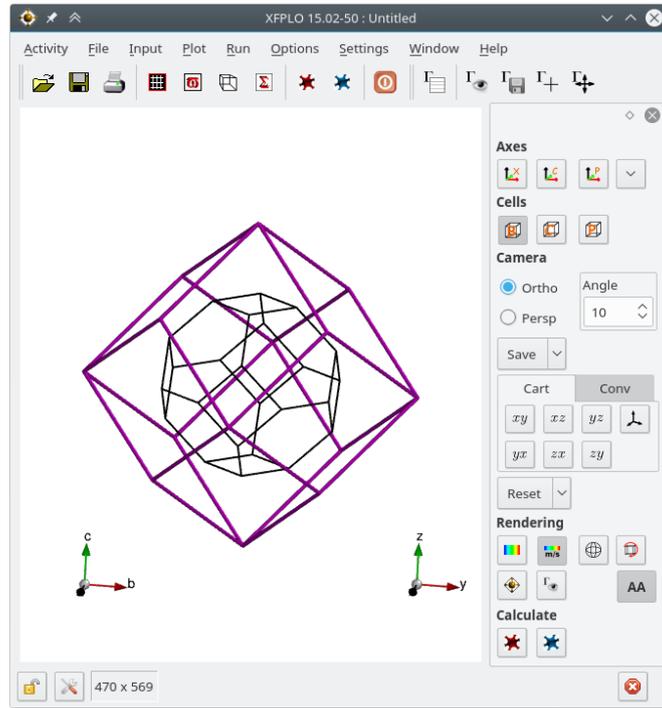


Figure 9: The BZ of the normal cell together with the BZ of the supercell.

- Open the high symmetry points dialog (HSPD) via **Plot > High symemtry points (Ctrl-H)**
- Check the **Visible** checkbox and the **User defined** checkbox.
- Hit the **import from =.in** button, change into the **bcc** directory and click on **=.in**.
- Hit **enter**.

Now, we have loaded the bcc high symmetry points of the bcc calculation into the super cell setup. This works, as we already explained, because in absolute coordinates the bcc sym-points are the same in both case. It was not necessary to load the magenta BZ. The point import works without this. But it helps orienting yourself.

- In the HSPD **export to =.in**: make sure you export into **D03Co/= .in!!**
- Close the dialog and in the FSW save the state.

5.2 The hard, more general way

- Execute

```
xfplo =.in
```

- Select the menu item **Activities** \triangleright **Fermi surface**.
- In the Fermi surface window (FSW) open the **Plot** \triangleright **Annotations** dialog and click on last item (“Primitive”) in left panel.
- click the **plus** button above the left panel or hit **Ctrl-I** or the **ins**-key.
- Hit **enter**. You should see a magenta box in the FSW.
- In the **Boundary type** selectbox choose **user defined**.
- Now convince yourself that we have a the conventional basis as the **reference basis** (should be default).
- Remove all items from the **Faces** table: either click into the table and use the **delete**-key or use the **delete button**  .
- Insert the following table. To insert a new line use the **plus** button right of the table or click into the table (to get the keyboard focus there) and hit the **ins** or **Ctrl-I**-key. If a cell is highlighted in the table, insertion will copy this line. Use the cursor keys to move around and just start typing. Use enter to finish editing a cell. You can use the **tab**-key to jump to the next cell without hitting enter. You can enter fractionals like “1/2”.

x	y	z	d0
1	1	0	$\frac{1}{2}$
-1	1	0	$\frac{1}{2}$
1	-1	0	$\frac{1}{2}$
-1	-1	0	$\frac{1}{2}$
1	0	1	$\frac{1}{2}$
-1	0	1	$\frac{1}{2}$
1	0	-1	$\frac{1}{2}$
-1	0	-1	$\frac{1}{2}$
0	1	1	$\frac{1}{2}$
0	-1	1	$\frac{1}{2}$
0	1	-1	$\frac{1}{2}$
0	-1	-1	$\frac{1}{2}$

- When done with the table hit the **accept**-button or hit **enter**. You should see Fig. 9. This is the normal cell BZ which we derived in Sec. 4.
- Close the Annotations dialog.
- In the FSW save the state via **Ctrl-S** or **File** \triangleright **Save**.
- Open the high symmetry points dialog (HSPD) via **Plot** \triangleright **High symemtry points** (**Ctrl-H**)
- Check the **Visible** checkbox and the **User defined** checkbox.
- Click on the **Pick points** button, go to the FSW and type **p 1** (as in **P**-key followed by the **L**-key) to switch off the high-symmetry lines and pick points equivalent to the Γ , H, N and P points in the magenta box according to the geometry of Fig. 4. Use zoom (**mouse wheel** or **right-click+move**) and rotation (**left-click+move**) to get a clear shot at the points. When done clicking the four points, hit **escape** in the FSW window or click the **stop picking** button in the HSPD. Now, the table in the HSPD should contain four points Γ , Γ , X_2 , Γ . The names are according to the supercell symmetry. We now rename the last three points into H,N and P and hit **enter**. Compare to 4 and correct names if needed.

- Click into the HSPD table. Go to the first line (should be the real Γ point) and hit **Ctrl-I** followed by the **p**-key and then 2 times **Ctrl-down**. You should have Γ HNTFP.
- Go to the H. Hit **Ctrl-I p Ctrl-down Ctrl-down Ctrl-down** to get Γ HNTPH.
- Now, go to the bottom of the table and hit **Ctrl-I b** to get Γ HNTPH(Break).
- Go to the lowest P and **Ctrl-I p Ctrl-down Ctrl-down** to get Γ HNTPH(Break)P.
- Go to the N and **Ctrl-I p 5 \times Ctrl-down** to get the final Γ HNTPH(Break)PN.
- Hit enter (**accept** button).
Your FSW should have a magenta BZ which looks like 4.
- In **Reference basis** choose **fplo (units $2\pi/a$)**. According to our theory in Sec. 4 we have the table as in Fig. 4 but each coordinate doubled.
- Click on the **Export to =.in** button and hit **enter**. (We do not need to chose the FPLO basis for export. It will be exported just fine, whatever basis is chosen.)
- Leave the dialog and in the FSW save the state via **Ctrl-S** or **File \triangleright Save**.

6 The supercell bandstructure on the normal cell path.

To show the difference to Fig. 7 we will re-calculate the band structure on the newly defined path.

- Have a look at the fedit **bandstructure submenu** of fedit to convince yourself that the new high symmetry points are indeed there and run

```
fplo22.00-62-x86_64 | tee out
```

- followed by

```
faddwei22.00-62-x86_64
xfbp +bwsum
```

Compare the result (Fig. 10) to Fig. 7 and Fig. 2. The latter comparison demonstrates the band backfolding due to the larger unit cell.

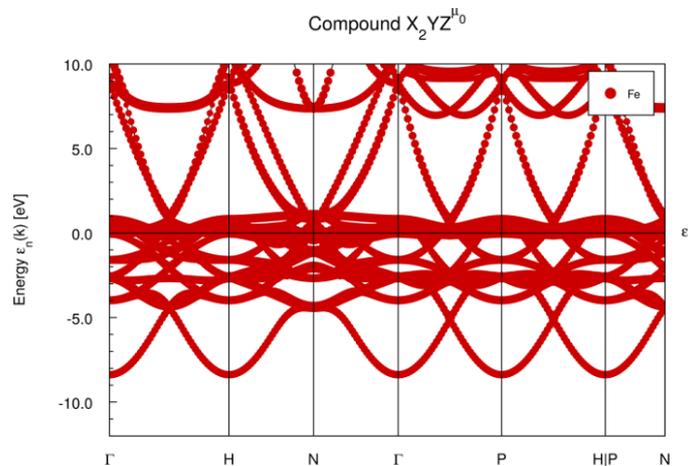


Figure 10: The SC fatbands along the NC path.

7 The unfolding mapping

We will now map the SC sites onto the NC sites. For this we could collect the site mapping table by hand, or we could use a tool. The tool requests the matrix which transforms the NC direct lattice unit cell into the SC direct lattice unit cell. So, let's do the math (also see Sec. 4). We have to map the primitive cells, since these determine the number of sites!

In the NC we have the primitive basis

$$A_p = B_{bc}A_c$$

We un-center

$$B_{bc}^{-1}A_p = A_c$$

multiply by 2 in all directions

$$2B_{bc}^{-1}A_p = 2A_c$$

and re-center as fcc to get the SC primitive unit cell

$$A'_p = B_{fc}2B_{bc}^{-1}A_p = B_{fc}2A_c$$

Use the relations for the inverse of the centering matrix Eq. (1)

$$\begin{aligned} A'_p &= 4B_{fc}^2A_p \\ &= \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} A_p \end{aligned} \tag{4}$$

This expresses the SC primitive lattice in terms of the NC lattice. Now, for input:

- Execute

```
xfplo =.in
```

- Open the unfold editor via **Tools** ▸ **Unfold**.
- Chose **large cell in small cell units** in the **units** combobox.
- Enter the matrix Eq. (4) into the **basis** table and hit the **apply** button.

You will note that the **unfold site list** now looks much like the example in the introduction, only that we only have one type in the NC.

- Hit the **save** button and save it to **=.unfold**.
- Leave the dialog and close **xfplo**.
- Execute

```
fplo22.00-62-x86_64 | tee out
```

Now we have a new file **+bweights_unfold**. We need a new **=.addwei** to add all the weights. The weights in **+bweights_unfold** will have atom number as given in **=.unfold** in the **NCsite** column (first column).

```

#NCsite SCsites
# Fe
1 1 2 3
# Co
2 4

```

Try to understand why the new `=.addwei_unfold` must look like

```

weightinfile +bweights_unfold
weightoutfile +bwsum_unfold

name Fe
  atom Fe sites 1 orbitals 4s 4p 3d

name Co
  atom Co sites 2 orbitals 4s 4p 3d

```

- Especially, note the first two lines and the sites!
- Ok, we assume you have created `=.addwei_unfold`. Then execute

```

faddwei22.00-62-x86_64 -s =.addwei_unfold
xfbp +bwsum_unfold

```

Voila, you probably got Fig. 11, which looks a lot like Fig. 2.

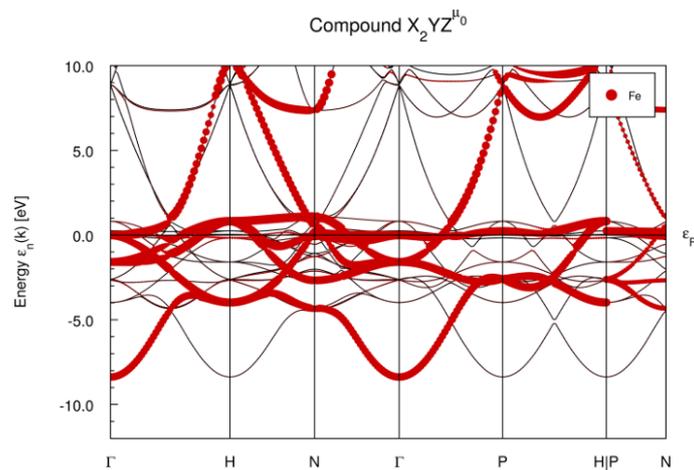


Figure 11: The unfolded SC Fe fatbands

- You should be able to switch on the Co-weight in `xfbp` to get this

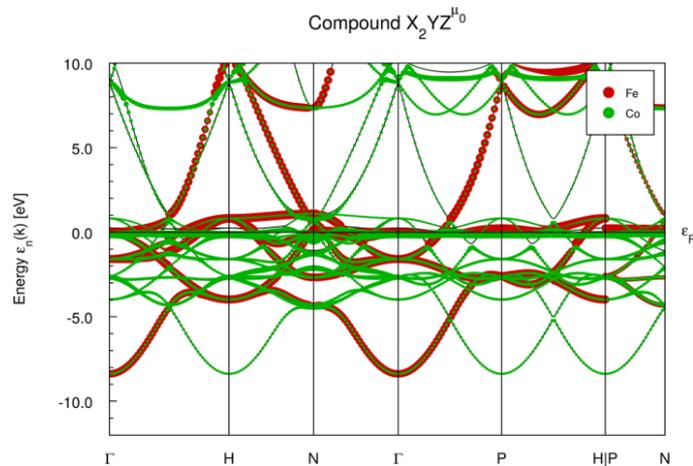


Figure 12: All unfolded SC fatbands

8 To wrap it up: $\text{DO}_3 \text{Fe}_3\text{Al}$

If you want to see what happens if we take Al instead of Co, please repeat everything with Al in place of Co.

- Go back into the `task` directory.
- Create a directory called `DO3Al` and copy files

```
mkdir DO3Al
cd DO3Al
cp ../D03Co/= .in .
cp ../D03Co/= .addwei .
cp ../D03Co/= .addwei_unfold .
```

- Use `fedit` to change Co into Al in the `symmetry` menu. Don't forget the `symmetry update`.
- Switch off `bandplot` \triangleright `bandstructure plot`.
- Run `fplo` to convergence
- Switch on `bandplot` \triangleright `bandstructure plot`.
- Re-run `fplo`.
- edit `=.addwei` to look like this (`element names and orbitals!`)

```
weightinfile +bweights
weightoutfile +bwsun

name Fe
  atom Fe sites 1..3 orbitals 4s 4p 3d
name Al
  atom Al sites 4 orbitals 3s 3p
```

- run

```
faddwei22.00-62-x86_64
xfbp +bwsun
```

to produce Fig. 13.

- Copy =.unfold

```
cp ../D03Co/= .unfold .
```

- and edit it. (Replace Co by Al)
- run fplo.
- edit =.addwei_unfold to look like this (ORBITALS!)

```
weightinfile +bweights_unfold
weightoutfile +bwsun_unfold

name Fe
  atom Fe sites 1 orbitals 4s 4p 3d
name Al
  atom Al sites 2 orbitals 3s 3p
```

- run

```
faddwei22.00-62-x86_64 -s =.addwei_unfold
xfbp +bwsun_unfold
```

to get Fig. 14 (and/or Fig. 15)

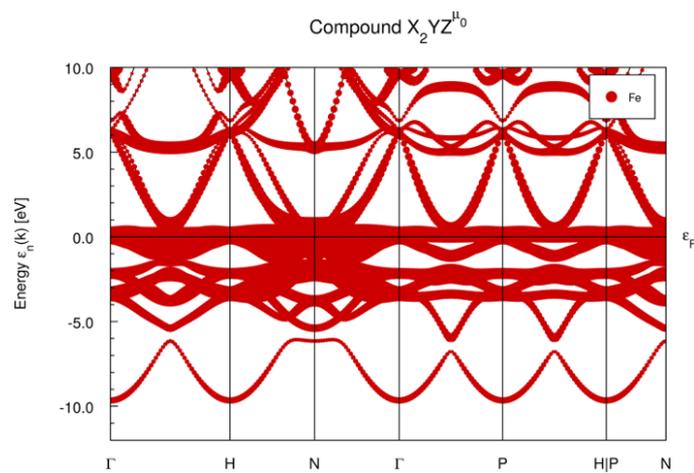


Figure 13: DO₃ Fe₃Al fatbands along the NC path

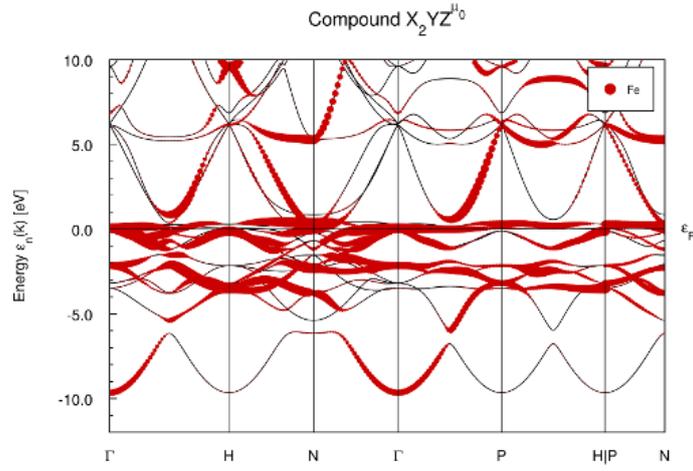


Figure 14: DO_3 Fe_3Al unfolded

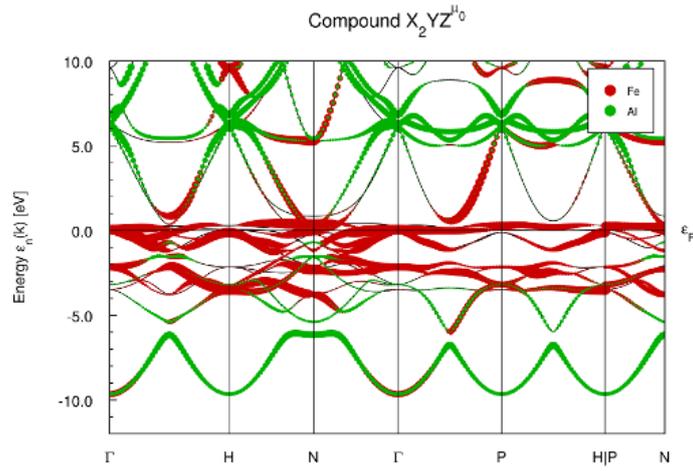


Figure 15: DO_3 Fe_3Al unfolded, all weights

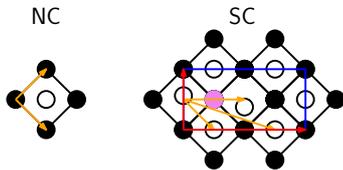


Figure 1: a 1 : 4 mapping of NC to SC. Four white SC sites map onto one white NC site, But only three black SC sites map onto the black NC site (partial unfolding). The orange arrows are NC lattice vectors. The red arrows are the SC lattice vectors.