



# **xfplo Documentation**

*Release 22.00-62*

**Klaus Koepernik**

**Jul 06, 2022**



# CONTENTS

<b>1</b>	<b>Topics</b>	<b>1</b>
1.1	More help	1
1.2	Introduction	1
1.2.1	General	1
1.2.2	The Structure viewer	2
1.2.3	Fermi surface viewer	2
1.2.4	Molecular/individual band weightsband weights	3
1.2.5	Density mapper	3
1.3	Trouble shooting	4
1.4	Hotkeys	4
1.5	The 3D View	6
1.6	Atom distances and angles	6
1.7	Symmetry Dialog	6
1.7.1	Groups	7
1.7.2	Wyckoff positions	7
1.7.3	Symmetry determination	7
1.7.4	Symmetries for non-minimal cell	8
1.8	Boundary Dialog	9
1.9	Bond Dialog	10
1.10	Bond-Poly Dialog	11
1.11	Find atoms Dialog	11
1.12	Fermi energy and bands	11
1.13	Brillouin zones: High symmetry points	11
1.13.1	Interface	11
1.13.2	Simple Orthorhombic Lattices	13
1.13.3	Facecentered/Bodycentered Orthorhombic Lattices	13
1.13.4	Base Centered Orthorhombic Lattices	13
1.13.5	Simple Monoclinic Lattices	13
1.13.6	Centered Monoclinic Lattices	14
1.13.7	Triclinic Lattices	14
1.14	Mesh Dialog	14
1.15	Fat band editor	14
1.16	Band Weight Contrib Dialog	15
1.17	Unfold Editor	15
1.18	Annotations	16
1.18.1	Axes	16
1.18.2	Display cells	16
1.19	Fog Dialog	16
1.20	Atom properties	17
1.21	Printing	17
1.22	Preferences	18
1.22.1	Graphics	18
1.22.1.1	Resolution	18
1.22.1.2	Antialiasing	18

1.22.2	External . . . . .	18
1.22.3	OpenGL . . . . .	19
1.23	Manage executables . . . . .	19
1.24	Light Themes . . . . .	19
1.25	Lighting . . . . .	20
1.26	Color Model dialog . . . . .	21
1.27	Wannier functions/Grid plot . . . . .	22
1.28	Show Symmetry Operations . . . . .	28
1.29	CIF Import Dialog . . . . .	28
1.30	Atom Label Dialog . . . . .	28

## 1.1 More help

---

**Note:** that links to external material only work under certain circumstances (namely in the pdf-files if they are opened in their intended location with a suitable pdf-viewer.

They do not work in the **xfplo** help system.

---

Please also consult [FPLO/.../DOC/MANUAL/doc.pdf](#)

## 1.2 Introduction

- *General*
- *The Structure viewer*
- *Fermi surface viewer*
- *Molecular/individual band weightsband weights*
- *Density mapper*

### 1.2.1 General

Many controls show tool tips, when the mouse hovers over them for a short time. The help screens try to tell you what's important. However, some basis knowledge about the corresponding topic is assumed. The **xfplo** tool kit is under constant development and hence not always complete or consistent.

The program has a set of command line options, which are displayed by:

```
xfplo -h
```

and printed below for reference.

```
usage: xfplo [-h] [-fs] [-str] [-bw] [-o] [-p printfilename] [-die] [file1 [file2 .  
↪...]]  
-h: print this help  
-fs: start fresh in Fermi surface mode  
-str: start fresh in structure mode  
-bw: start fresh in band weight edit mode  
-raw: show +iso_* file data as they are without filling the boundary  
-o: open output window
```

(continues on next page)

(continued from previous page)

```

-die: quit program after loading
-p printfilename: print the screen after loading (combine with -die)
-magfac factor: magnification factor (int > 0) for printing the screen
                (combine with -p and -die as in

                xfplo -die -magfac 4 -p t.jpg =.in

                )
file1...: load corresponding file. If several files are given
xfplo tries to load all files which fit together.
If a mode option and a file is given, the
first file determines the mode if the file belongs
to a mode.

Example: 'xfplo -fs =.in' ignores -fs and starts in structure mode

Example: 'xfplo t.vtk' will not know the mode
since the vtk files do not a priori belong to a certain mode.
Use a mode option or a moded file as first argument instead.

possible file names:
=.in : FPLO input                -> structure mode
=.xstr: XFPLD structure file     -> structure mode
=.xef:  XFPLD fermi surface file -> fermi surface mode
=.bwdef: FPLO/XFPLD bandweight file -> band weight mode
+coeff: FPLO coefficient file    -> band weight mode
*.cif: cif file                  -> structure mode
+iso_*: dHvA iso files          -> fermi surface mode

*.vtk: polygonal data sets      -> no mode
wfdata...: Wannier functions    -> structure mode
grid_...: gridplot (densplot and such) -> structure mode

files saved from xfplo with different names should work too.

```

## 1.2.2 The Structure viewer

Structures can be loaded via:

```
xfplo =.in
xfplo =.xstr
```

Fresh start in structure mode:

```
xfplo -str
```

Structures can be manipulated in the *symmetry dialog*. From there one can export `=.in`, which also involves a symmetry update of `=.in`. The whole picture (not just the structure) can be saved into `=.xstr`.

## 1.2.3 Fermi surface viewer

A fresh start in fermi surface mode is:

```
xfplo -fs
```

One can save all settings (but not the `+band... data [to big!]`) in `=.xef`, which can be loaded via:

```
xfplo =.xef
```

**Tip:** One can also define the path through the Brillouin zone for band structure plots in the Fermi surface mode. To achieve this first you need at least a `=.in` (use **fedit** or the symmetry dialog in the structure viewer to create this.) Then open:

```
xfplo -fs
```

Now, switch off the Fermi surface display via the Fermi-surface button and switch on the high symmetry points (Fig. *Fermi-surface and sym-points buttons*).

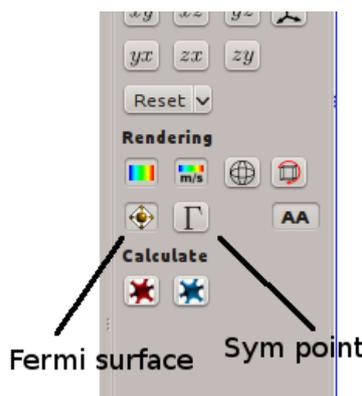


Fig. 1: Fermi-surface and sym-points buttons.

Open a dialog: Menu `plot` → `high-symmetry-points`. Switch on `user-defined`. Load default if you want. Select a point in the list. Click on the `pick` button, select `current`, click on a green point in the Brillouin zone, hit `enter` or press `accept`. You must have changed a point. Use `pick`-button to create a new point. Use `F2` in the list to edit or just start typing. Export to or import from `=.in`, if needed. For more see *Brillouin zones: High symmetry points*

## 1.2.4 Molecular/individual band weightsband weights

The default method to create an orbital projected bandstructure (fatbands/band weights) is to switch on band weights in **fedit**. The more flexible way is to use the band weight editor mode of **xfplo**. On a fresh start use:

```
xfplo -bw
```

or with an existing file:

```
xfplo =.bwdef
```

to edit the desired band weights. (see *Fat band editor*)

One can use this interface to request single weights for selected atoms, which would help to reduce the file size of the usually big file `+bweights`.

One can also create molecular orbital projectors, which are linear combinations of atomic orbitals with certain coefficients, which determine the bonding character and symmetry of the molecular orbital.

## 1.2.5 Density mapper

This is not well tested and was implemented on special request of some user.

An existing calculation (call it A) with a `=.dens` file can be a good starting point for a modified structure (call it B). However, since the structure of B determines the `=.dens`-file of B the `=.dens`-file of A cannot be used in B. You can use **xfplo** to define the file `=.densmap`.

We assume that calculation A exists and that the structure of calculation B is already set up (`=.in` exists).

1. Go into the directory of the the new calculation B.
2. Call:

```
xfplo =.in
```

to display the structure.

3. Open `tools` → `density-maper`.
4. Click button `Open old structure` and select `=.in` of directory A. This will open a new structure view, this time of A.
5. In the mapper table click on an atom of new structure B. The atom will be highlighted. Now click on the atom of structure-view A, whose density shall be copied onto the selected B-atom in the `=.dens`-file to be created.
6. You can use the `Copy this atom` button to copy this definition to all B-atoms with the same element. Proceed with all B-atoms, which have an equivalent in the A-structure. You can leave some B-atoms unmapped, in which case a default starting density for these missing atoms will be produced by **fplo**. You can also open other `old structures` and map atoms from there.
7. Save the file in the directory of calculation B (name `=.densmap`).
8. Quit **xfplo**.
9. Run **fplo** in the directory of B until it stops after creating a new `=.dens`.
10. Zip or rename or delete `=.densmap`.
11. You can now start the B calculation with the freshly mapped `=.dens`-file.

---

**Important:** Note, that only the spherical part of the local site densities are copied during mapping. Hence, the created mapped density is not necessarily a good starting point.

---

## 1.3 Trouble shooting

If you use the program remotely via ssh login and the central widget is not displaying anything (no white screen) try:

```
export LIBGL_ALWAYS_INDIRECT=yes
```

an the command line before running the program.

On some KDE systems the color modes are set wrongly after re-login. This results in a failure to display disabled widgets in a grayed-out fashion. This behavior is system wide and not **xfplo** specific. A remedy (after loggin) is to edit the file `.kde/share/config/kdeglobals` and to remove all color sections from the beginning of the file.

## 1.4 Hotkeys

In the main window several hotkeys are defined.

**Mac users:** `Ctrl` is the Apple-key on Mac OS. Alt keys do not have an equivalent on Mac. Especially, there are no hotkeys to select menu entries or controls in dialogs. That's Mac OS's restriction. On Linux all these hotkeys are working.

- *Axes*
- *Boundary/Display cells*
- *View directions*
- *Pick high symmetry points*

Table 1: Axes hotkeys :align: center

Action	Key
Show axes popup	a
toggle cartesian	a-x
toggle conentional	a-c
toggle primitive	a-p
toggle cartesian frame	a-a
toggle conventional frame	a-v
toggle primitive frame	a-r

*Hotkeys*

Table 2: Boundary/Display cell hotkeys :align: center

Action	Key
Show boundary popup	b
toggle main boundary	b-b
toggle conentional	b-c
toggle primitive	b-p

*Hotkeys*

Table 3: View direction cell hotkeys :align: center

Action	Key
Show view direction popup	v
view top (down the z axis)	v-t
view front (down -y axis)	v-f
view right (down x axis)	v-r
Many more options	...

*Hotkeys*

Table 4: Pick high symmetry points (Fermi surface module) :align: center

Action	Key
Picking	p
Pick points	p-p
Pick lines	p-l
Pick planes	p-a
Pick general points	p-g

*Hotkeys*

## 1.5 The 3D View

**Tool-buttons** are used in the GUI. They have a main part, which acts like a button and a little down-arrow part, which acts like a combobox. When choosing from the combobox menu the corresponding action will be executed and the corresponding action becomes the buttons default action. So, pushing the button will execute the recently chosen action.

### Moving in the scene

The 3D view has extensive mouse input actions.

**Trackball rotate** Hold down the left mouse button and move it around. The rotation is not commutative. So after a while one gets a pretty good feeling for orienting the scene.

**Circle rotate** `Ctrl+left` mouse button 2D-rotates the scene around the center of the scene.

**Pan/Move** middle mouse button or `Shift+left` mouse button will move the scene

**Zoom** the mouse wheel or the right mouse button or `Ctrl+Shift+left` mouse button will zoom the scene in and out.

**Default views** The view buttons and their *hotkeys* will set a default view angle/rotation around the object center. After moving the scene the center does not coincide with the object center. The object center can be attached to atoms (right mouse/context menu)

**Default pan** `Ctrl+R` will reset the pan and zoom such that the object basically fits into the viewport. This is not always perfect, though.

### Canvas size

The canvas or view has by default a fixed size. This comes in handy if several similar pictures have to be created. The canvas size can be set with a menu (`Ctrl+V`) or by unlocking the canvas by clicking the lock icon in the lower left corner of the window. Then window resizing will resize the canvas as well. One can lock the canvas at any size.

**Right mouse click** On an atom you get a context menu.

## 1.6 Atom distances and angles

**Menu: Tools** → Measure or hotkey `M` will open the measuring window.

Use the mouse to click on a visible part of an atom. The atom gets added to the list in the dock-window if it is not yet selected, otherwise it gets removed from the list. Atoms get added to the bottom of the list but can be removed arbitrarily. This will change the order of the remaining atoms and can be convenient.

Up to 3 atoms can be selected. If there are 3 atoms they form a sequence (1st, 2nd, 3rd). The distance between the 1st and the 2nd and the 2nd and the 3rd is shown in the dock-window, as well as the angle formed by the vectors (1st-2nd) and (3rd-2nd).

If `M` is pressed again the dock-window closes the measuring stops and the list gets reset.

If `ESC` is pressed the list is reset.

If the underlying data change (symmetry, boundary) the list gets reset.

## 1.7 Symmetry Dialog

- *Groups*
- *Wyckoff positions*

- *Symmetry determination*
- *Symmetries for non-minimal cell*

## 1.7.1 Groups

The groups can be specified as space groups and points groups and layer groups, depending on the structure type (crystal, molecule and slab).

The point groups are specified internally via a suitable space group, being build on the desired point group. This also means that we do not have  $C_\infty$  and icosahedral groups. For these use proper subgroups.

If several settings are defined for a certain group it can be specified. While changing settings the lattice constants, axes angles and Wyckoff positions are not changed unless the `Update` check box is checked, in which case the data are adjusted such that the same structure results.

Additionally to the group settings a global rotation matrix can be specified, which rotates the whole structure. These rotations are active when the `rotate` check box is checked.

## 1.7.2 Wyckoff positions

For crystals the Wyckoff positions are relative coordinates according to the standard crystallographic conventions. For molecules the positions are given in absolute Cartesian coordinates.

**Alt-W** to select the Wyckoff position table.

**Cursor keys** Move around with the help of the cursor keys.

**Editing** Start editing above any item by just typing.

**F2** Starts editing with the old value selected. Use cursors, `Del`, `Backspace` ...

When not editing. ...

**Ins/Ctrl-I** will insert a new row below the current.

**Del/Ctrl-D** will delete the current row.

**Ctrl+up/down** will move the selected row around

Once in editing mode use

**Esc** to leave the editor without committing changes!

**Enter** to finish editing (on Mac that is restricted)

**Ctrl+A** to select all

**Ctrl+C** to copy the value

**Ctrl+V** to paste previously copied values

**Tab** to move to the next table item

In order to enter one complicated value in more then one position use these keys!

## 1.7.3 Symmetry determination

Note that all operations start from the primitive cell, even if the boundary shown is larger than that.

Also note, that the resulting space group might not be a unique solution. This happens if the resulting cell does not have the maximum possible translational symmetry (by user request: `button conventional cell`, flag `reduce translation` being off). For further information see here: *Symmetries for non-minimal cell*.

**Conventional cell** Remove centering operations and return maximum symmetry of the non-centered cell.

**Remove generators** Keep the current lattice parameters but remove all non translational group operations by setting the subgroup generators to  $\{E\}$ . This will make all sites inequivalent. The spacegroup number will stay the same (for the lattice cell) but its pointgroup operations are switched off. Use this to pick a few atoms and make them different (set different element or type). Use `determine symmetry` afterwards to get maximum symmetry.

**Determine symmetry** The symmetry is calculated. The result is a spacegroup according to certain standard definitions. If the `reduce translations` box is checked the smallest possible unit cell is calculated. If it is not checked the cell can change in shape, but not in volume. When calculating the symmetry all atoms with the same type number (second column in the Wyckoff table) are considered to be potentially symmetry related. Hence to request two atoms to be definitely different Wyckoff positions the types have to differ. The easiest way is to change the element of the atom, since this will set the type to the elements nuclear charge. It also changes the color which makes it easier to evaluate the new cell. However, one can also just change the type.

After determining the full symmetry of a layer-derived bulk system (magnetism) the new c-axis might be different from the origin direction. On the other hand the restriction to operations, which are compatible with the q-vector, might be too tight if the resulting system is a bulk system.

The rotation matrix describes an active rotation of the unit cell which results from the symmetry settings. It is applied according to

$$A_{\text{rot}}^T = R A_{\text{def}}^T$$

where

$$A^T = (\vec{a}_1 \vec{a}_2 \vec{a}_3), \quad \vec{a}_i = \begin{pmatrix} a_{ix} \\ a_{iy} \\ a_{iz} \end{pmatrix}$$

Hence  $R$  is the matrix which transforms a vector  $v$  like

$$\begin{pmatrix} v'_x \\ v'_y \\ v'_z \end{pmatrix} = R \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

which makes the columns of  $R_i$  the new  $x, y, z$ -directions.

**Helpfull feature:** The table to enter the matrix accepts non-normalized columns. It is also possible to set a single column to zero, in which case this column will be set to be an orthogonal direction to the other two columns.

All operations of the manipulate panel will update the rotation matrix such that the orientation with respect to the global cartesian system stays unchanged. This is of course only visible if the rotation checkbox is checked. The rotation will always be updated but it only becomes active if checked.

The setting changer (update button changed) does not change the rotation, since this is the whole point about different settings.

**The enlarge function** is used to produce a commensurate multiple of the cell.

**Layered cell** is used to create a unit cell which has the first two lattice vectors perpendicular to the given axis. Check the rotate button to keep the cell orientation. If `orthog. if possible` is checked the cell will be larger to ensure an orthogonal c-axis (if possible).

**Insert vacuum** is used to create a repeated or free standing slab from the current cell. For this the a and b-axis are considered fixed. The third axis will be enlarged to insert the vacuum. Use this together with `Layered cell`.

**To Molecule** is used to convert the visible cell into a molecule. Use this to achieve more control when you prepare a structure picture.

## 1.7.4 Symmetries for non-minimal cell

Usually, the maximum symmetry is calculated. This involves finding the smallest possible translational unit. The user can however, request the symmetry with certain restrictions. Notably, the reduction of the translation group

can be skipped. In this case we have a set of symmetry operations for the lattice which is the maximum point group allowed by the required translational group and by the sites and all the sub-cell translations which were forbidden to be reduced. This leads to a group which has a larger order than the point group of the required lattice. To find a space group which describes this lattice one has to fulfill a group multiplication table of the order of the target space group within a set of operations which is parametrized by all the sub translations from the non-reduced translation group. The result is a set of operations which depends on integer parameters. Now, one has to pick a set of parameters and it turns out that the resulting space groups may not be identical. The algorithm picks one of those choices. Hence, several consecutive invocations of the algorithm might yield different groups with the same order, but different number of Wyckoff positions!

To make matters worse, in certain symmetries the expected maximal subgroup does not exist. E.g. if we have a non symmorphic body centered tetragonal cell and want to determine the symmetry of the corresponding simple tetragonal lattice (with 2 times the number of atoms) and if the bct cell has a  $4_1$  screw axis there is no simple tetragonal spacegroup with a  $4_1$  axis. Hence, we cannot find a cell with this symmetry element. Consequently, the resulting group will be of lower order than expected. In practice, we apply an additional constraint: if there is a subgroup which contains the inversion we pick this subgroup in order to be able to use the parity algorithm for topological insulators. Note, that the use cases for this symmetry determination are usually quite rare.

## 1.8 Boundary Dialog

Items are only displayed if they (there origin) fall(s) within the boundary cell defined here. (Exceptions are atoms/bonds, which have a corresponding flag set.) The boundary is either the Wigner-Seitz cell (for Fermi surfaces), a simple box spanned by three directions or a polyhedron defined by faces.

A box is defined by three vectors  $V_i, i \in [1, 3]$  and three factors  $d_i$ . These vectors are expressed in terms of either the primitive, conventional or cartesian basis vectors and span a lattice. The faces of the box are lattice planes of this lattice defined by  $r \cdot G_i = d_i$ :  $G_i$  are the reciprocal vectors of  $V_i$ :  $G_i V_j = \delta_{ij}$ . The cell's diagonal goes from  $(0, 0, 0)$  to  $\sum V_i d_i$ .

If the `center around origin` checkbox is checked, the cell's center is at  $(0, 0, 0)$ . Additionally an origin shift vector  $s$  can be defined. The shift vector defines an active shift of the boundary box by the amount specified, which again can be given in different basis systems. The basis system `Box` means that the shift is given in terms of the boxes dimensions, e.g.  $(\frac{1}{2} \frac{1}{2} \frac{1}{2})$  would be the vector from the box's corner to it's center.

**In case of a slab geometry**, the underlying  $z$ -direction/"third lattice vector" is  $(0, 0, 1)$  so that e.g.  $V_3 = (102)$  in conventional basis gives the vector  $a_1 + 2e_z$ . For conventional/primitive basis the factors  $d_{3,0} = 0$  and  $d_{3,1} = 1$  represent two lattice planes at the atoms with smallest and largest  $z$ -component respectively, such that the default box encloses all atoms in  $z$ -direction. For cartesian basis, all factors are in cartesian basis. The  $z$ -component of the origin follows the same logic. Conventional/Primitive:  $\text{map } s_z = (z_{\max} - z_{\min}) z$ , Cartesian:  $s_z$  is cartesian (absolute).

The option `faces/User defined` describes a cell formed by a polyhedron, which is spanned by all points lying inside of a set of planes defined via  $r \cdot R = |R|^2 d$ . The vector  $R$  is expressed in a certain basis  $b_i$  (see `Reference basis` combobox in the dialog):  $R = n_i b_i$ . Hence, what we really give as input are the three numbers  $n_i$ . Now, if we want to scale  $n$  we need to do understand this:

$$r \cdot R = |R|^2 d \tag{1.1}$$

$$r \cdot (n_i b_i) = |n_i b_i|^2 d \tag{1.2}$$

$$r \cdot (\lambda n_i b_i) = \lambda |n_i b_i|^2 d \tag{1.3}$$

$$r \cdot (\lambda n_i b_i) = |n_i b_i|^2 \frac{d}{\lambda} \tag{1.4}$$

Scaling works like:  $n_i \rightarrow \lambda n_i, d \rightarrow \frac{d}{\lambda}$ .

Example:  $n = (110), d = 1$  gives a plane which is perpendicular to the  $(110)$ -direction in the chosen basis. For an orthogonal basis this would be the a plane going through the point  $b_1 + b_2$ .

The `Load` button initializes the faces with certain settings depending on the currently chosen basis. Hence, a change of basis requires a repeated `load` in order to obtain the same physical cell.

**In case of a slab geometry**, and faces,  $d = 0$  or  $d = 1$  in the  $z$ -direction refer to a plane through the lowest or highest lying atom respectively.

The `save current` button can be used to save the current faces. They can be reloaded via the corresponding option under the `load` button.

When being in a line-edit

**Ctrl+A** select all

**Ctrl+C** copy selection

**Tab** jump to next line-edit

**Ctrl+V** paste selection

When being in a table

**F2** starts editing

**Any key** start editing

Real numbers can be rationals: one can enter  $3/8$  instead of  $0.375$ .

## 1.9 Bond Dialog

Hotkeys:

**Ctrl+1...Ctrl+4** Expand the bond tree view to the according level.

In the Bonds/Poly tree view

**Right/Left** Expand/collapse the current item

**Space** (un)check the current item, or open the *Bond-Poly dialog*.

**Any edit key** when on a root item open the *Bond-Poly dialog*.

**double click** when on root item open the *Bond-Poly dialog*, on other items expand/collapse the item.

**Enter** commit the data and update the scene

**Ctrl-A** select all over

**(Ctrl/Shift) click** change current selection (only the root items selection actually matters)

In the BondGroups table view

**Cursor keys** move around

**Ins/Ctrl-I** will insert a new row below the current.

**Del/Ctrl-D** will delete the current row.

**Ctrl-Up/Ctrl-Down** move column

**Space** on checkbox: toggle, on color button: open color dialog, on Name column: start editing

**Any key/F2** on name column: start editing

When clicking on an atom in the 3D view, the corresponding atom gets selected in the current BondGroup table in an open Bond Dialog. If you have multiple groups in the table check the other groups if you search for the settings of that atom. If more than one root-item is selected in the Bonds/Poly tree view and the *Bond-Poly dialog* is open clicking on an atom in the 3D view selects the properties of this atom unless the atom is not in the current selection. This allows to transfer the settings of one atom to all the atoms in the selection.

## 1.10 Bond-Poly Dialog

Here you define which atoms (bonds) are depicted. You also define polyhedra for the corresponding Wyckoff position.

The first thing is the `Atom/Bond visibility`, which has four possible settings:

**In Boundary** Show only atoms in the boundary as specified in the boundary dialog.

**Add Missing** Show atoms outside the boundary in order to complete all bonds of all atoms of this type, which are inside the boundary

**Delete Incomplete** Delete atoms for which some of the bonds point outside the boundary. The atoms are not deleted if they are needed to fulfill the `In Boundary` option for another atom.

**Delete Loners** Delete atoms, which have only bonds, which point outside the boundary.

Note, that in the case of multiple bondgroups the bond definitions of the later bond group supercede those from earlier groups. Also note that a bond has two ends and that both ends can have different `Atom/Bond visibility`.

If `Apply` is pressed the whole Bond Dialog gets committed and the scene updated. If `Close` is pressed the data are committed to the Bond Dialog but not to the application. `Cancel (Escape)` closes the Dialog. When the dialog is open you also can click at atoms in the scene to load the data of another atom. This discards all previously made non-committed changes to this dialog.

## 1.11 Find atoms Dialog

todo

## 1.12 Fermi energy and bands

Here we select the bands, which have to be plotted.

## 1.13 Brillouin zones: High symmetry points

- *Interface*
- *Simple Orthorhombic Lattices*
- *Facecentered/Bodycentered Orthorhombic Lattices*
- *Base Centered Orthorhombic Lattices*
- *Simple Monoclinic Lattices*
- *Centered Monoclinic Lattices*
- *Triclinic Lattices*

### 1.13.1 Interface

There are two modes of special symmetry points treatment, the automatic one, which adapts to the current lattice type and parameters and the user defined, which cannot adapt in all lattices, since some lower symmetry lattices have points whose coordinates depend on the lattice parameters irrespective of the choice of coordinate basis. The

default is automatic. If the user clicks on the `pick` button and the mode is automatic a warning will be issued. Only user defined points can be manipulated (label moving and point picking).

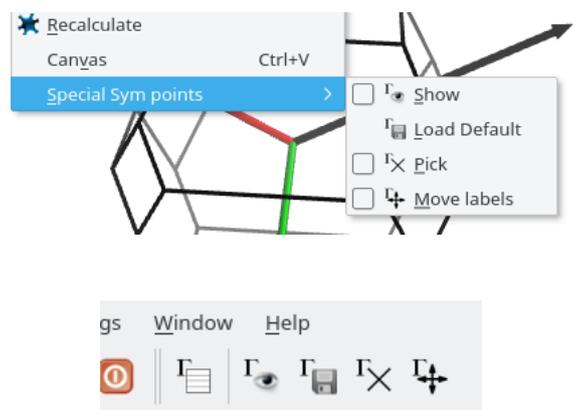


Fig. 2: Special symmetry point controls.

When picking points is active some green dots appear, which can be clicked on. The corresponding point is inserted into the table in the dialog. Usually insertion happens at the end. But if a row in the point table is selected the point is inserted after this. Use the up/down buttons (`Ctrl-Up/Ctrl-Down`) to move points around in the table. The names of the green dots are suggestions. There are cases when the naming depends on the actual relation of point to some axis. In these cases the names have different names than their translationally equivalent point, which is why they are suggestions. Also the subscripts make only sense with respect to the default path. If another path is chosen by picking, it might be reasonable to delete the subscripts or add some or rename the points altogether. E.g. in fcc the  $K$ -point is related to similar points via translations (and inversion and fourfold rotations). The same holds for the  $U$ -point. However if we apply the threefold rotation  $K$  and  $U$  become the same point class. In other words if we look at the fcc lattice from the  $z$ -axis,  $K$  and  $U$  interchange if translations are considered as equivalences. In this case we decided to discard translational equivalence and use rotational equivalence instead, when labels of non standard points are assigned (green dot labels). If a non standard cell is used to pick labels,  $K$  and  $U$  might be mixed up.

When moving labels is activated, the user can click with the mouse pointer on a label and move it.

When picking or moving is activated and the main window is the current window, the `escape` key will terminate picking/moving. So does closing the special points dialog.

The dialog allows to select the high symmetry points. You can set the point names, colors, font size and label depth. You can also pick points in the scene. Note, that all boundary cells shown in the scene will provide pickable points. Beside points, you can insert jumps, which allows to backfold a path, which goes straight through more than one first BZ into the first BZ. A jump will consist usually of three point. Let's say there is a path  $\Gamma X S$ , which forms a straight line, where  $\Gamma X$  is in the first BZ and  $X S$  in the adjacent BZ. There is an equivalent line segment  $X' S$  in the first BZ, with another  $X$  point, which is equivalent to the first  $X$  point. The path now reads  $\Gamma X$  (Jump)  $X' S$ . In the bandplot this is translated into  $\Gamma X S$ . In other words the segment  $X X'$  is cut out and the band structure will look as if you go through the two BZ. This is most useful for showing all high symmetry points in a single first BZ picture.

The `break` option under the `plus` button-menu inserts break point, at which a new path segment starts. Currently, in bandplot the segments are glued together where the two labels of the endpoint of the first segment and the starting point of the second segment are combined to something like " $X|K$ ". This glueing produces discontinuities in the bands, if the points before and after the break-point are not equivalent.

If there are equivalent points they are named according to their translation, or rotation. Some points are subscripted following arXiv:1004.2974v1 even if they are translational equivalent.

The reference basis in which the point coordinates are given are:

Table 5: :align: center

basis	$\mathbf{k}$ in absolute coordinates as function of input $k_i$
Cartesian (units $2\pi$ )	$2\pi (k_x, k_y, k_z)$
Cartesian	$(k_x, k_y, k_z)$
FPLO (units $\frac{2\pi}{a}$ )	$\frac{2\pi}{a} (k_x, k_y, k_z)$
Conventional	$(k_x, k_y, k_z)^T G_c$
Primitive	$(k_x, k_y, k_z)^T G_p$

### 1.13.2 Simple Orthorhombic Lattices

We do not enforce the order of the lattice constants with respect to length. Instead the points are named according to their relation with respect to the three axes.

### 1.13.3 Facecentered/Bodycentered Orthorhombic Lattices

We adopt the scheme that the conventional axes are sorted internally to obtain  $a \leq b \leq c$  before determining the special symmetry points. This is done in a way that one always gets the same picture (except for axes) if changes the setting while having the rotate and update check box checked (symmetry menu, empty lattice mode). What is meant by this is that two lattices which are only differing by the setting but otherwise are identical have the BZ and high symmetry points. This leads to three different possible kinds of BZs for the facecentered lattice. The points are named according to this order, which means that e.g. the Z-point will lie on the shortest reciprocal axis, whichever this is. The default path will jump in space if a critical point of the parameter is crossed.

### 1.13.4 Base Centered Orthorhombic Lattices

We adopt internally sorting of the conventional axes such that the axis opposing the centered plane is  $c$  and that the remaining two axis fulfill  $a \leq b$ . This means that similar looking BZ shapes have the same special points irregardless of the actual orientation of the cell. E.g. the Z-point is the center of the “hexagonal” lid. The default path will jump in space if the critical point  $a = b$  is crossed. The setting which looks like the pictures in the above mentioned arxiv paper if the lattice is by itself reduced is “axis -a cell 1”.

### 1.13.5 Simple Monoclinic Lattices

For discussion we assume that the  $c$ -axis is the monoclinic axis and we focus on the reciprocal lattice. A simple monoclinic lattice can always be transformed into one, which has the shortest possible  $a^*$ ,  $b^*$  reciprocal lattice constants (for the two axis perpendicular to the monoclinic axis) and a monoclinic angle  $\geq 90^\circ$ . At the same time the angle between  $a^*$  and  $-a^* - b^*$  and between  $a^*$  and  $-a^* - b^*$  can be made non-acute  $\geq 90^\circ$ . This is a Delaunay reduced lattice. If we furthermore require  $a^* \geq b^*$  we arrive at a unique lattice basis. Note, that the corresponding direct space lattice is not at the same time Delaunay reduced. To achieve this one has to flip the monoclinic axis  $c$  and one of the other two ( $c \rightarrow -c, \dots$ ). This uniquely defined lattice basis is used for the assignment of special points and their names. This means that the special points will jump around if a critical value of the lattice constants is crossed. In fact there are infinitely many such critical values for the lattice parameters. It is however most natural to use the reduced basis for point nameing. Cases with other monoclinic axis are obtained by cyclic permutation of the axis. The names (like  $X$  and  $Y$ ) have the usual meaning for instance in the following setting

Space group	System	Bravais	Std.Symbol									
3	Monoclinic	Simple	P 1 2 1									
Setting	axis c	P 1 1 2 <input type="checkbox"/> Update										
Subgroup generators full group												
Lattice const. <input type="radio"/> Angstroern <input checked="" type="radio"/> Bohr radii <input type="checkbox"/> rotate												
7	10	5	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </table>	1	0	0	0	1	0	0	0	1
1	0	0										
0	1	0										
0	0	1										
Axis angles												
90	90	70										

Fig. 3: Simple monoclinic setting where the names are intuitive.

If you now check the Update button and the rotate button and change the setting to another monoclinic axis, the same BZ and naming is obtained only with permuted axis (and rotated in space if the rotate button is not clocked, since our  $c$ -axis is always in the  $z$ -direction [not the reciprocal  $c^*$ -axis!]).

### 1.13.6 Centered Monoclinic Lattices

For base centered monoclinic lattices a similar condition as for the simple monoclinic lattice can be applied to reduce the lattice to a unique standard. Point nameing is based on that. There are in total 5 cases of different BZs in this lattice. The default path jumps around when parameters cross critical points. Otherwise we follow the setting condition explained for face-centered orthorhombic lattices.

### 1.13.7 Triclinic Lattices

We use reduction to Type I/Type II lattices (Delaunay reduction) to determine which of two distinct cases we have. In Type I we have to employ a simple case decision which decides where the  $N$  point lies. Then the points (all face mid points) are named according to their relation to the reduced axis. Again it means that the labels jump if critical parameter values are crossed. Consider the naming a suggestions. At least we have a predefined set of points now.

## 1.14 Mesh Dialog

The Fermi surface mesh is created such that it adapts best to the symmetry of the lattice. To achieve that an automatic mesh subdivision is employed. Only a mesh subdivision parameter is needed. This is usually the subdivision along the  $x$ -axis or first lattice direction. In order to reduce the calculation time for the band structure code, point group symmetry is used, If the checkbox for the irreducible part is checked. The subdivision is done such that the resulting micro-cubes and tetrahedra are as isotropic as possible. For slabs that would be a waste. Therefore, one can overwrite individual subdivisions from the automatic determination. In the slab case that will be the  $z$ -direction, where we need only 1 subdivision. If the manual subdivision line-edits stay empty, the automatic value is used. Beware, that depending on symmetry restriction for possible values apply. For instance, in tetragonal lattices the  $x$  and  $y$  values must be equal.

## 1.15 Fat band editor

**Fplo** can export band weights in the files `+bweights...`. This has limitations. **Fplo** can also export the complex coefficient matrix in `+coeff` when requested via a switch in the **fedit** bandstructure submenu. This file can be used to create custom tailored band weights by selecting certain molecular/atomic orbital projectors, which define

the band weights. The resulting data are written to a file with a user specified file name, which then can be plotted in the usual way (xfbp).

Another option is to not create the `+coeff` file and to put the file `=.bwdef` (or whatever name you gave when saving the content of the fat band editor) into the **fedit** bandplot menu. Then **fplo** will create bandweights according to the definitions contained in this file and ignores the other settings in the bandplot menu.

If the DOS is required by the standard options (bandplot on and option `NO_DOS` not set) additional files `+bdos...` are created containing the LDOS corresponding to the patterns defined in this dialog.

To edit molecular band weights there is a helpful feature: if a structure view of the corresponding `=.in`-file is open, the atoms and absolute positions of a molecular orbital projector can be selected from the structure view via mouse click, see *Band Weight Contrib Dialog*.

**Warning:** If you define let's say a molecular pattern consisting of two  $p_z$ -orbitals sitting at two sites, you have to define the relative phases (factor in the contrib dialog) in order to get bonding/anti-bonding orbitals. If you draw a little sketch it is clear that the equal phase pattern  $(1, 1)$  must be the anti-bonding state. However, if the two radial functions belong to different main quantum numbers the radial functions can have inverted signs, since the sign of the radial part is defined by the behavior close to the nucleus and the additional nodes change the sign in the valence region. Hence, then the anti-phase  $(1, -1)$  gives the antibonding pattern!!!

Hotkeys in weight-tree-view:

**F2/Enter/Space/Double-Click** open editor

**DEL** delete item

**Ctrl-Up/Ctrl-Down** move item

## 1.16 Band Weight Contrib Dialog

Edit the contribution of a single atom to the band weight definitions. If it is a molecular orbital only the name can be edited. Otherwise more information needs to be defined. If it is a single atom weight definition, the position is determined automatically. If it is a contrib to a molecular pattern, the absolute position has to be given. In order to facilitate this you can use the `getAtom` or `getPosition` buttons. In order for this to work a structure view of the corresponding compound has to be open. Note, that there is no cross check whether the structure belongs to the current data or not. If a valid `+coeff` exists site and orbital information is taken from this file and put into comboboxes. If the orbital information is absent, autocompletion in the orbital-linedit is provided.

Valid orbitals can be `all`, `allnlm`, `3d`, `"3d-1"`, `3d3/2` (relativistic) and `3d5/2/-1/2` (relativistic) if it is a single atom definition. If it is a contrib to a pattern only fully qualified orbitals ( $nlm$  or  $nlj\mu$ ) are valid. The local axis are checked for orthogonality. If they are not orthogonal it will be adjusted. If they are coplanar an error is issued.

Important: In relativistic mode pseudo non-relativistic symmetries ( $nl$  and  $nlm$ ) can be used. In this case the underlying colinear approximation of the full relativistic **fplo** implementation requires that the resulting spin axis points along the quantization axis of the exchange field. That means that although this tool allows to select the local quantization axis for harmonics of the pseudo non relativistic orbitals, the spin projection always follows the field axis. Keep this in mind when interpreting the "spin-up" and "spin-down" LDOS/weights. Furthermore, beware that the pseudo non-relativistic projections are approximate!

The `all` orbital descriptor produces  $nlj\mu$  weights and `allnlm` produces  $nlm\sigma$  weights in full relativistic mode.

## 1.17 Unfold Editor

Here the unfolding information is edited. The right side is a text editor, in which the content can be freely edited. One can load and save the file.

The unfolding is explained in detail in [FPLO/.../DOC/MANUAL/doc.pdf#Unfolding](#). The information here specifies, which super cell sites are considered translationally equivalent with respect to normal cell translations. The list contains lines. Each line contains an identifier, which best is chosen to be the site number in the normal cell, which is considered a representative of the translationally equivalent super cell sites, followed by a list of all super cell sites, which are equivalent. The list of equivalent sites can be incomplete, which leads to partial unfolding.

Additionally, there is a possibility to define the relation between the normal and super cell by defining the transformation matrix from one to the other. If the combobox "large cell in small cell units is chosen" the matrix should be integer. In the other case the matrix usually is rational. The "change on units" checkbox influences the matrix on changing the combobox setting. Either the matrix is transformed/inverted such that the new matrix and the new combo setting define the same cell relation. Or the matrix is left alone, when the combo setting is changed. The tolerance determines if super cell atoms, which are slightly displaced from their ideal normal cell position are considered translational equivalent with respect to the translation vectors given by the matrix. The Apply button calculates the site lists according to the matrix and tolerance settings.

Note: that different elements can be considered as being translationally equivalent as long as their basis is equivalent. e.g. Fe and Ni. The automatic creation explained in the last paragraph does not know about the basis and hence puts different elements into different unfolding definitions. If you know what you are doing you can reshuffle the unfolding definitions by hand. **Fplo** will abort with an error, if unfolding of different elements in the same site list is impossible.

## 1.18 Annotations

- *Axes*
- *Display cells*

There are several kinds of objects, which can be shown in the scene.

### 1.18.1 Axes

Main axes sit at the origin of the first unit cell (by default), while frames sit at a fixed screen position given in relative screen coordinates.

### 1.18.2 Display cells

Display cells are polygonal cells. The first (*Main boundary*) is special in that it defines the clipping volume for atoms and Fermi surfaces. The second and the third are the conventional and primitive cell respectively. The user can add more cells, which then have to be edited accordingly. The dialog does not allow to edit the conventional and primitive cell.

The main boundary is edited in the boundary dialog. That is so to emphasize its special purpose.

The visual properties can be set. The visibility can be toggled (not the special hotkeys for the first three cells.)

The add (CTRL+I, *Ins*) and delete (CTRL+D, *del*) buttons are active when the focus is in the cells-section of the Annotations list view.

## 1.19 Fog Dialog

Fog can be added to the scene. For perspective view linear fog is best. The other fogs can make the scene to appear totally white. The color is probably best set to white.

The fog is determined by parameter

**Exponential/Gaussian** The clearness decays with an exponential or gaussian of decay rate `density`. In perspective view smaller densities are needed to avoid complete fogging.

**Linear** The scene is clear at `near` and becomes foggier up to `far`, where the fog has the full fog color. `far` can be larger than one, which means that the fog will not reach maximum saturation at the object being furthest away. The closest object is at 0 and the furthest away one at 1;

## 1.20 Atom properties

Atoms can have colors and radii. The Tab `Themes` allows to define different themes, containing default properties for each element. The button `Reset to app default` resets the whole table of the current theme to some standard values. The `Accept/Close` buttons save the changes and make the current theme the default for all actions, which require atom properties, like adding Wyckoff positions to the Symmetry Dialog, or opening **xfplo** input files like `=.in` (which do not contain color/radius information). The button `Save` and `Apply` saves the settings and applies the current theme to the current Wyckoff positions and sites. Note, that the current (all) theme(s) is remembered persistently in the file `$HOME/.xfplo/atomprops.ini`.

In the symmetry menu the color/radii for each Wyckoff position can be Set individually. Additionally, there is the `set Default` button, which can be used to set the color/radii of all Wyckoff positions to the current theme's values. If the file `=.in` is loaded the current theme (combobox `theme`) is used to define the atom properties. The properties are saved if the file `=.xstr` is saved. So, there are two sets of values. 1: `in=.xstr` 2: in the themes. The `=.xstr` settings supersede the theme settings on loading, unless new atoms are created.

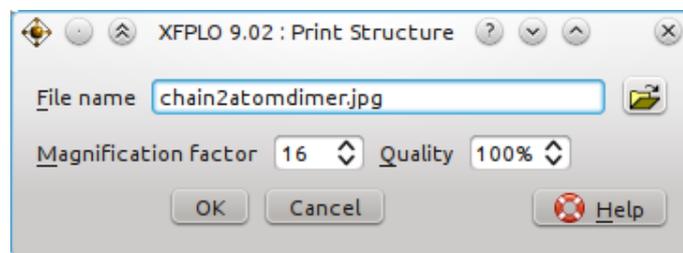
Editing: in the table there is the radius type. To edit it, either type `F2` or simply the letters (`cimvc`). The radius field is updated and can be edited. All radii are remembered, such that one can easily switch them. Radii are given in Angstrom. Note, that the displayed atom radii are half of the actual radii. The "Bond Dialog" contains a scale factor, which scales all atoms uniformly.

Specular color: the spot light color combines with the specular color of an object to create a highlighted spot.

Shine: the smaller the shine value the more diffuse the spot will be. Maximum value is 128. Imagine a metallic polished object, in this case the spot will be very focused, while for a more diffuse surface the spot is smeared. This is why shininess produces a smaller spot for larger values.

## 1.21 Printing

Currently, printing is possible to `jpg` and `bmp` files only. The scene is printed by piecing together several enlarged shots of parts of the total scene. In this way you get a higher pixel density than depicted on screen. E.g. a `magnification factor` of 4 will glue together 16 pieces, resulting in a 4 times higher pixel density than on screen. Together with *Antialiasing* this gives high quality pictures.



While printing it may happen that the intermediate enlarged shots are displayed on screen. This is ugly but not a bug.

Printing can be interrupted using the stop button close to the progress bar in the status bar.

## 1.22 Preferences

- *Graphics*
  - *Resolution*
  - *Antialiasing*
- *External*
- *OpenGL*

### 1.22.1 Graphics

#### 1.22.1.1 Resolution

Each display has different graphic capabilities. This might effect performance, especially while moving the scene. Several options are implemented to ease the pain.

**Low resolution** Atoms and bonds can be drawn with less polygonal resolution.

**Wireframe** Some graphics are faster with wireframes.

**No antialiasing** Antialiasing, when allowed, will remove some of the pixelation effects. It is however more expensive. With a jitter of 4, for example, the scene is drawn 4 times at every redraw request (moving). You can switch it of during moving. See *Antialiasing* for more control.

**Inaccurate transparency** Transparency requires sorting of the polygons, which is slow. With inaccurate transparency this sorting is switched off while moving with the mouse.

#### 1.22.1.2 Antialiasing

Antialiasing removes the pixelation effects by jittering the scene, which basically redraws the scene by superposing several single paints with sub-pixel offsets.

**Jitter** gives the number of redraws.

**For printing only** uses antialiasing only when printing the scene and not on the display. This gives faster graphics on screen. You can find a compromise using the resolution option *No antialiasing*.

### 1.22.2 External

For some tools external editing is possible. Give the editor and check the `run in terminal box`, if needed (e.g. for `vi`).

Additionally the terminal program can be specified. Default is `xterm`. You could e.g. use `konsole` (KDE). The working directory is set implicitly (at least we try).

There is the option to set command line options to the terminal program and editor. Some placeholders can be used

`%f` absolute file path (directory/filename)

`%d` directory

`%fn` relative filename (no directory)

Depending on the user input, **xfplo** will run the following commands. Note, that no `%f` is added to the editor if the user-defined editor contains `%f`. Similarly, if the user-defined terminal contains `%d` no working-directory options are added by **xfplo**. On the other hand if `%d` is missing **xfplo** adds `--workdir %d` if the terminal is `konsole`.

Editor	Terminal	run in terminal	xfplo runs	what happens
emacs			emacs %f	a new emacs gets started with the file which needs to be edited
emacs %f			emacs %f	
emacsclient -n			emacsclient -n %f	the file is sent to an open emacs
emacsclient -n %f			emacsclient -n %f	
vi	xterm	X	xterm -e vi %f	a terminal will open with an open vi
vi %f	xterm	X	xterm -e vi %f	
vi	konsole	X	konsole --workdir %d -e vi %f	
	konsole - some_option %d		konsole - some_option %d	konsole is opened with the current working directory

### 1.22.3 OpenGL

On modern hardware, things should be standardized by now. However, ...

The OpenGL settings found here, were at least one times critical on some older platform:

**Vertex arrays** Vertex arrays are unlimited per OpenGL definition. Some platforms, however have limits, which on top of all this are not queryable. Hence, the only way out is to not use vertex arrays on such platforms (mostly old graphics cards or software emulators). You will find this out, if the Fermi surface is not plotted completely, when using more *k*-points.

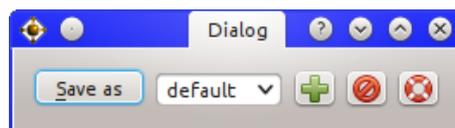
## 1.23 Manage executables

The **fplo** executables are selected by looking for filenames containing **fplo** in paths defined by environment variables (such as \$PATH) or in selected directories or explicitly specified files. This dialog allows to compile a list of such search patterns.

The actual list of executables returned by the search algorithm in the run dialog puts the executables, matching the version of the current =.in file the best, at the top of the list. If there has been a previous run, the program remembers the executable used for this =.in and selects it. Otherwise the best match is selected.

## 1.24 Light Themes

Some data are more generally useful than just for a single session. Among them are atom properties and lighting. The atom properties are explained in a separate section. The save button opens a save dialog.



If the save button is pressed, the current light settings is saved under the name given in the combobox. At the same time this theme is set to be the default light theme for each session, which does not provide previously saved light settings (loading =.in, or starting fresh) until changed again. The Combobox allows to edit the name of the theme (only when save pressed). You can add a new theme and delete any theme (but the first). The load-theme dialog



allows to select an existing theme and load it into the current light dialog. At the same time the current theme is set as the default for each session. The themes are stored in `.xfplo/lightthemes.ini`.

## 1.25 Lighting

Lighting is a subtle issue. The program comes with a standard lighting, which works for all scenes. There is an option to define and save predefined lighting, which gets re-used unless `.xstr` or `.xef` files are loaded (they contain their own light settings). Themes can be easily loaded via the theme load and save buttons  (see [Light Themes](#)).

For starters you can define the background color and the global ambient light. Ambient light is lighting the scene evenly. Use it with caution. Try setting it to 1 and see what happens.

There are three different sorts of lights. All lights have a diffuse color, which is diffusely reflected according to the angle between the surface of an object and the light position. Each light can have ambient light, which usually is better set via the global ambient light. The last is the specular light, which defines the reflection spots. The effect is defined by the specular color of the light and the specular color of the objects material.

**Directional lights** Directional lights shine from a certain direction given in the position spinboxes. The light comes from an infinitely remote point and has parallel rays. So it lights all surfaces in the same angle. It is well suited for achieving a basic lighting. Displacing it away from  $(x, y) = (0, 0)$  make the light spot (if specular color is not black) appear sideways. The  $z$ -coordinate is relative to the camera position to ensure working light for all scenes. Positive  $z$  means that the light sits at or behind the camera in outward direction from the screen.

**Positional lights** These lights sit at a definite position in space. In the program this position is defined in units of the scene dimensions. All coordinates are measured relative to the focal point, which is in the middle of the view area and usually in the middle of the physical scene. Positive  $z$  coordinates mean that the light sits closer to the viewer than the focal point. Negative  $z$ -coordinates put the light deeper into the scene (lighting from behind). Positional lights shine in radial rays from the position. The positional light at 000 usually sits in the center of the scene at the focus point and shines in all directions. The light diminishes the further away the object is from the light, which is controlled by an attenuation factor, given by falloff  $\propto \frac{1}{c_0 + c_1 d + c_2 d^2}$ , where  $d$  is the distance. The most important coefficient is the constant coefficient, which scales the light strength. Positional lights have another parameter  $w$ , which controls the allover distance of the light. The real position is given by  $\vec{p} = \frac{1}{w}(x, y, z)$ . Hence, the larger  $w$  the closer the light will be to the scene. Default is  $w = 1$ . You can get the same effect, by changing  $x$ ,  $y$  and  $z$  by the same factor, but  $w$  is more convenient.

**Spot lights** Spot lights are like positional lights, except for an added characteristic, of being confined to a cone going out of the position into the direction specified in the dialog. The cone opening is controlled by a cutoff angle, which must be in the interval  $[0, 90]$ . It defines the half-angle of the cone opening. Outside the cone there will be no light. Additionally the light is dimmed from the center axis of the cone towards higher angles via the exponent.

**Options:** (this happens internally, it seems)

**Phong Lighting** can be done per vertex (Gouraud shading) or per pixel/fragment (Phong shading). The latter is prettier but more costly.

**Local Viewer** In the modern version the reflection angles are calculated correctly from the locally visible light direction. In the old version these angles are calculated by assuming lights at infinity. The latter leads to more regular light spots (could be desirable).

**Warning:**

1. Do not use too much diffuse/specular color. Often good light settings have quite gray diffuse colors and more white-ish specular colors. The reason is that all lights together should not overexpose the scene. Ambient color is best used from the global ambient settings at the top of the dialog.
2. Too bright settings (small exponent, and cutoffs (opening angles)) give ripple effects on approximated surfaces. So use wider spot lights, with larger exponents and correct with constant attenuation.
3. Don't underestimate the power of the constant attenuation in connection with positional/spot lights. Going below the default value 1 can give results in combination with the ripple avoidance, discussed above.

**Tip:** In order to achieve good lighting the following tips are helpful.

1. Use the default.
2. There are light helpers, which can be switched on, which are not perfect, but help orient.
3. Use perspective viewing for better orientation, when setting up lights.
4. Switch off all lights but one, in order to get somewhere. Start from position 000 and reasonable direction (spot lights) (the default direction 00 – 1 points into the screen.)
5. Directional lights with too negative  $z$  are most likely not much visible.
6. Try small changes of all parameters to understand the effects and the light helpers.
7. Note, that the bright spots get reflected by the surface in an angle and hence appear at some angle between the light direction and the eye. Don't get confused by the direction the light helper points to being different from the spot position on the object.

## 1.26 Color Model dialog

The *color map editor* has a button from which one can choose pre-made color maps.

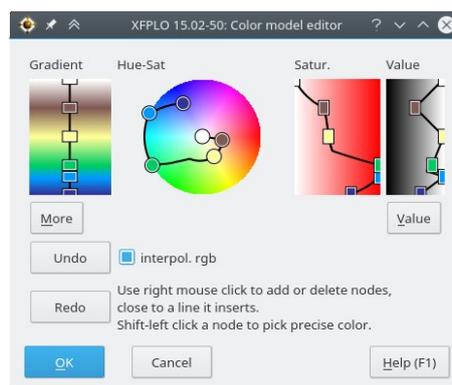


Fig. 4: The color map editor

A map consists of a chain of nodes, each with a certain hue-saturation combination and a certain value (darkness). These nodes are interpolated to obtain all colors. The interpolation happens either on straight lines in the Hue-Sat space or in *rgb* space. The latter will happen if the `interpol. rgb` checkbox is on. The difference is most visible if two colors are chosen such that their connection line crosses the white center of the circle.

Right click somewhere and a node gets added.

- If there is no node yet, a single node appears.
- If there is one node a second appears and will be connected to the first node by a line.
- If there are more than two nodes, the new node will be connected to the closest end node, or it will be inserted if the mouse click was close to a connection line. (Currently, a straight line between neighboring nodes is considered and not the curved lines which appear in `rgb` interpolation scheme. If you click on the visible line close to a node it should work in almost all cases)

Immediately after the `right click` the new node can be dragged around.

`Right click` on a node and it gets deleted.

`Left click` on a node to move it around.

`Shift-left click` a node to set a precise color.

Any of the 4 color widgets can be clicked on.

If you prefer to role your own color model start with a two color map. Choose `more->clear` to remove all prior nodes. `Right click` into the round Hue-Sat widget and drag the new node to your color of choice. `Right click` somewhere else in the same widget and drag to the desired color. This is a colormap.

Now, optionally, switch on/off the `interpol. rgb` checkbox. This will interpolate between the two end nodes either in RGB or Hue-Saturation colorspace. These two-end-color maps are usually rather useful, especially with RGB interpolation.

You can invert the node sequence (`more->invert`)

The `more->set-equal-distance` function will make the gradient nodes equidistant.

## 1.27 Wannier functions/Grid plot

**Xfplo** can display iso surfaces of 3d data. Currently data created via the **fedit** grid output submenu and Wannier function real space representations are supported. Originally, these data were written together with `.general` and `.net` files for the use in `opendx`. Since **fplo-18** this has changed. If you want to use these old-style files a special option in `=.wandef` must be given (explained in the WF documentation [FPLO/.../DOC/MANUAL/doc.pdf#WFs](#) or an option in the grid output **fedit** submenu must be set (this option is now off by default).

The data files (`wfdata...`, `grid_...`) now contain headers, which describe the data structure. The user can write it's own files for loading into **xfplo** following these examples.

A way to load these files is:

```
xfplo =.in wfdata001 wfdata005
```

or if the `use-data-directories` switch was set in **fedit**:

```
xfplo =.in +wfdata/wfdata001 +wfdata/wfdata005
```

which displays the iso surfaces with some default settings and opens the corresponding dialog. The default iso value is not always a good choice. Hence, sometimes nothing can be seen after loading. Change the iso value to make the data visible.

---

**Note:** that **xfplo** can save it's settings in `=.xstr`, in which case it also saves, which data-files are in use. However, you can also do:

```
xfplo =.xstr wfdata001 wfdata005
```

in which case `wfdata001 wfdata005` are loaded additionally no matter if they are already indicated in `=.xstr` or not!

---

We try to walk through all options. Some of them are only useful for certain situations.

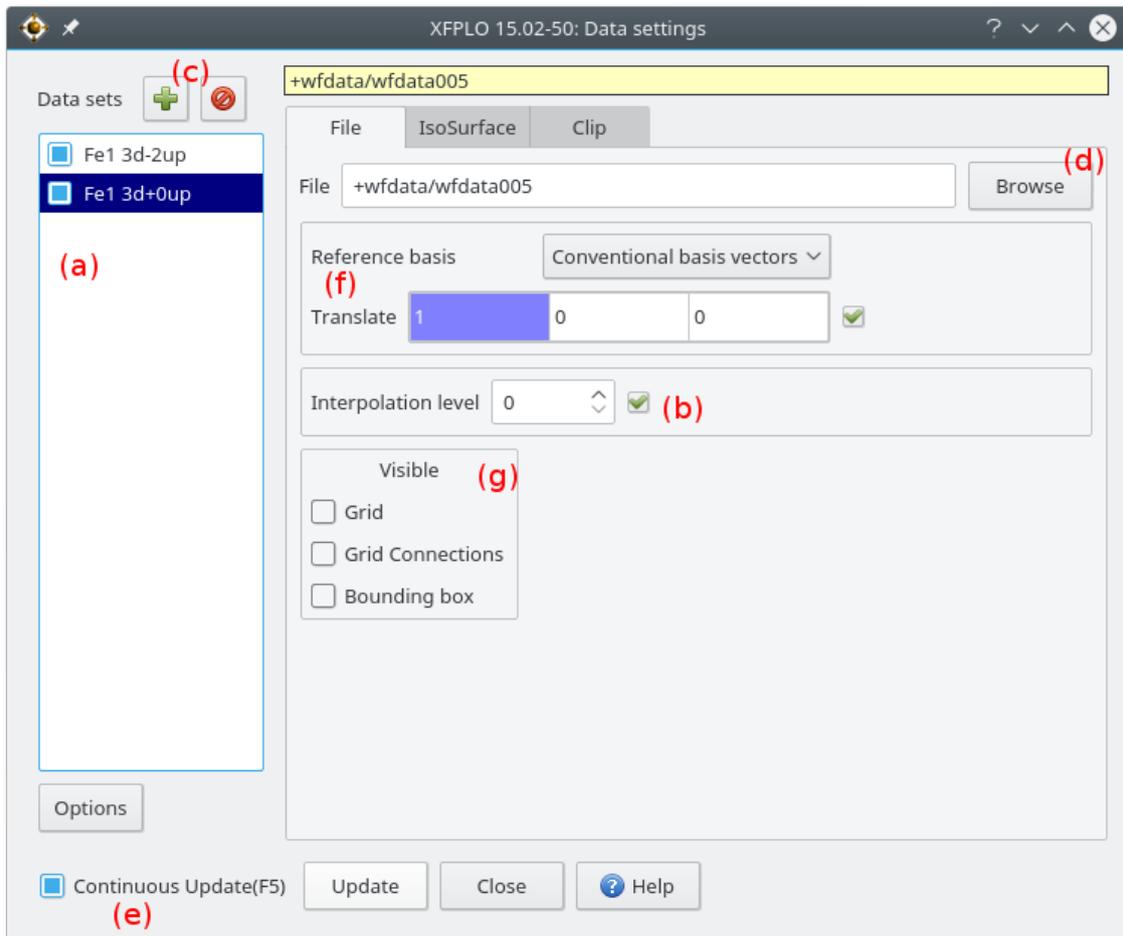


Fig. 5: The Data dialog: file tab.

After loading a data file the Data Settings dialog opens in the `IsoSurface` tab. The dialog can also be opened from the Main window via the menu `Plot->WF/Gridplot` (in structure view mode).

The dialog works differently from other **xfplo** dialogs in that there is no cancel button. All changes are registered internally and applied immediately or when the `Update` or `Close` button are pressed. This has to do with the internal organization of the tool chain.

If you save the current state in `=.xstr` (or another name) all settings of this dialog (including the loaded data-file names) will be saved but not the data files themselves. So, to reload you need `=.xstr` and the data files, but you call:

```
xfplo =.xstr
```

Also see note *loading WF-data*.

The bottom of the dialog contains the following controls:

- If the `Continuous Update` check box (hotkey F5) (see Fig. *Data dialog: file tab (e)*) is checked, all changes to the dialog will be immediately applied.
- The `Update` button right of the `Continuous Update` check box applies all changes which were not yet applied.
- The `Close` button right of the `Update` button will apply all pending changes and close the dialog.

At the left side of the dialog you will find the list of data files:

- The `Data Sets` list Fig. *Data dialog: file tab (a)* contains one item for each loaded file. You can select any number of items via `Shift+left-mouse-button + drag` and `Ctrl+left-mouse-button + click`. If several items are selected the data widgets (in the tabs to the right) are only enabled, if they are consistent among all selected items. The data widgets show the values belonging to the current item (usually the last put into a selection). `Right-mouse + click` can be used to change the current item in a selection. The yellow label above the tabs shows the current item's file name. In order to set a value for all items in a selection use the little green check-buttons *Data dialog: file tab (b)*. The items can be (un-)checked, which (hides) shows the corresponding iso surface. `Double click` or `F2` on the current item allows to change the name of the data set.
- The `Options` button below the `Data sets` list allows to reset the names of all selected items to their default.
- The `Plus` button *Data dialog: file tab (c)* or `Ctrl-I` on the current item will insert a new item with invalid file name.
- The `Delete` button *Data dialog: file tab (c)* or the `delete` key on the current item will delete the current item.

Remember the little green check buttons. It is often faster to first find good settings for one data set (e.g. a Wannier function) and then to select all sets in the `Data sets` list, make the one with good settings current item (`right click` in the `Data sets` list) and then to use the green check buttons to copy the settings to all selected items.

Furthermore Fig. *Data dialog: file tab* contains the following items

- The `Browse` button Fig. *Data dialog: file tab (d)* allows to load a file into the current item.
- Each data set is positioned in space as it's file's data indicate. It is possible to shift the data around via the `Translate` tool *Data dialog: file tab (f)*. Chose an appropriate `Reference basis` for the translation vector in the combo box. Click on a cell of the `Translate` vector and start typing or use `F2` to edit. Use the `tab` key to edit the next cell (cycles around when in the last cell).
- The data files are usually expensive to create. The resulting iso surface might look jittery. Use the `Interpolation` tool *Data dialog: file tab (b)* to increase the size of the data set by interpolation. The maximum interpolation level is 5 to avoid crazy input. Usually a level of 1 or 2 should be enough. At higher levels some interpolation artifacts may occur. Interpolation will slow things down, hence it is a good idea to set interpolation last.
- There are 3 check boxes (Fig. *Data dialog: file tab (g)*) which toggle the visibility of the data set's grid points, grid connections and it's bounding box. This is useful for orientation. Grid points and connections relate to the interpolated data not the input data.

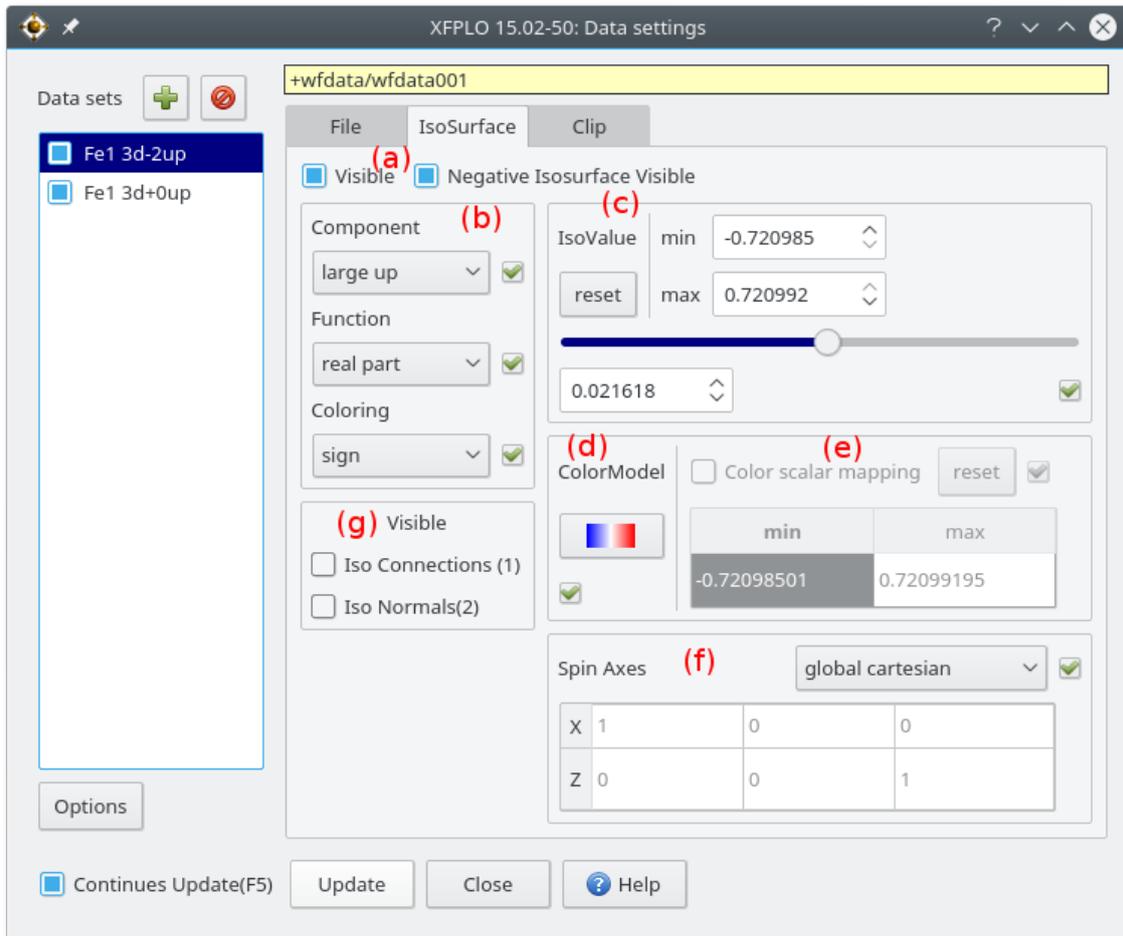


Fig. 6: The Data dialog: iso surface tab.

Figure *Data dialog: iso surface tab* shows the iso surface tab of the dialog. Here all settings which determine the iso surface of a data set are defined.

- The `Visible` check box switches the whole iso surface on or off, while the `Negative Isosurface Visible` check box hides or shows the iso surface for the negative iso value. The negative iso surface is useful for Wannier functions and wave functions (`fedit` grid output data) and not so useful for others.
- Depending on the data file you get a number of data components. E.g. for full relativistic Wannier functions there are the real and imaginary part of the large and small component of the Wannier function in the `wfdata...` files. From this additionally the up, down, spin and total density of this WF is calculated and available. The up, down and spin densities of course depend on the chosen spin quantization axes *Data dialog: iso surface tab* (f). *Data dialog: iso surface tab* (b) allows to select the Component for which the iso surface should be shown. After a component is selected a function can be applied as for instance the real or imaginary part or the absolute value. Not all combinations are useful. E.g. for the density-component the abs-function and the real-function are identical and the imag-function is zero. Both the choice of the component and the function will change the valid iso value range. To reset the min/max iso value use the iso value reset button in *Data dialog: iso surface tab* (c). After Component and Function are specified a Coloring mode needs to be selected (Fig. *Data dialog: iso surface tab* (b)). Available options depend on the data in the data files. The coloring mode can be

**sign** positive parts of the iso surface get one color and negative parts another. The two colors are the end colors of the color model *Data dialog: iso surface tab* (d).

**phase** for complex data (e.g. the large component of the WF) it is a good choice to use the abs-function, which combines real and imaginary part into one real data set and to use the `phase-coloring` to depict the phase of the complex data as color. Note, that this option is only different from other

combinations if the data set is in a non trivial way complex. In this `Coloring` mode a particular circular color map is used, which cannot be changed at present.

**spin polarization** this is usefull in combination with the density-component. It colors according to the spin polarization  $\zeta = \frac{n^\uparrow - n^\downarrow}{n}$  with respect to the selected spin quantization axes Fig. *Data dialog: iso surface tab (f)*. This of course is interesting only if the spin is not a good quantum number of the data set as e.g. for mixed spin full relativistic Wannier functions or relativistic wave functions.

**gradient** this uses the legnth of the gradient for coloring. Probably not very useful.

**self** This uses the values of the data set itself, as defined by `component` and `function`, to color the iso surface. The result is a single color iso surface (and another color for the negative part). Currently, multiple iso-values are not yet implemented, which makes this coloring option less usefull.

- After the choice of a `component` and `function` (Fig. *Data dialog: iso surface tab (b)*) it is probably good to use the `reset` button for the iso value range Fig. *Data dialog: iso surface tab (c)*. This will update the `min/max` values of the possible iso value range and will put a heuristic default value into the iso value spinbox itself. If multiple data sets are selected in the `Data sets` list the range will be set to the largest range covering all data sets. This range determines the step width in the iso value slider and spinbox. It can be useful to put user defined `min/max` values in the range control to make the slider/spinbox more sensitive in a particular iso value interval. The slider under the `reset` button and the iso value spinbox under the slider actually change the iso value of the iso surface. The little green button will set the current range and iso value for all selected data sets as explained in general above.
- The Color Model button Fig. *Data dialog: iso surface tab (d)* allows to pick the colors.
- If the Coloring-mode does not have a defined data range (as e.g. `gradient` and `self`) the Color scalar mapping can be activated and be set to a particular data range (Fig. *Data dialog: iso surface tab (e)*). This is particularly useful if multiple related data sets are displayed with the same iso value in which case one wants equal iso values to be displayed with equal colors for all data sets. One cannot assume that the data ranges of all the different sets are equal. Hence, we need to map them to a common data interval. Think of a case where several Wannier functions are displayed in the same structure.
- If data sets contain spin information the choice of the spin quantization axes matters. For instance a full relativistic WF for a spin polarized calculation is a four-spinor. Now, if we use WF projectors onto orbitals of quasi non-relativistic symmetry (e.g. `3d-2up` and `3d-2dn`) and have a global spin  $z$ -axis (in the **fedit** main menu) which is not `(001)`, the spin character of the WF is determined by the thusly defined global spin coordinate system. This has to be taken into account when plotting the iso surfaces. Since **xfplo-18** one can explicitly define the spin axes of the WF projectors. This axis information will be contained in the `wfdata...` files and will be offered in the `Spin Axes` combobox (Fig. *Data dialog: iso surface tab (f)*). For convenience there is also an option to set the spin quantization axes by hand.
- For debugging and orientation the iso surface connections and normals can be switched on (Fig. *Data dialog: iso surface tab (g)*).

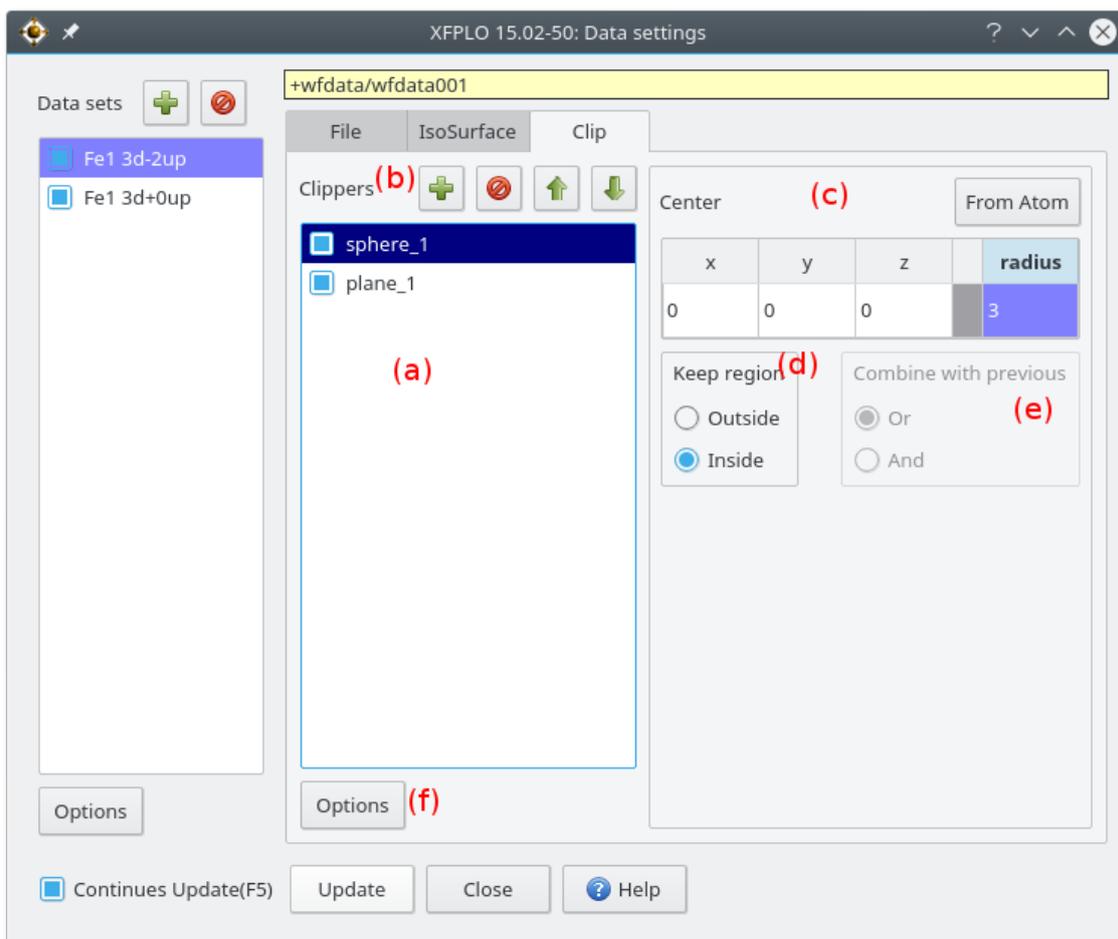


Fig. 7: The Data dialog: clipper tab.

In Figure *Data dialog: clipper tab* the last leg of the tool chain is shown, the clippers tab. Here, one can define how the iso surface shall be clipped, which is useful to remove e.g. small ugly parts of the hybridization tails of Wannier functions. There are two kind of clippers, a sphere and a plane. Both define a 2d surface which has an inside and outside region, either of which can be chosen to be the region being kept after clipping. More than one clipper can be defined. The various clippers are combined in order of their definition with an or-function or an and-function. This means that first the first clipper is applied to the iso surface. Then the region which is kept by the second clipper is determined and applied to the left-over part after the first clipping procedure via one of the two logical functions. This gives quite some freedom to construct various clipping functions, although it does not have the maximally possible flexibility.

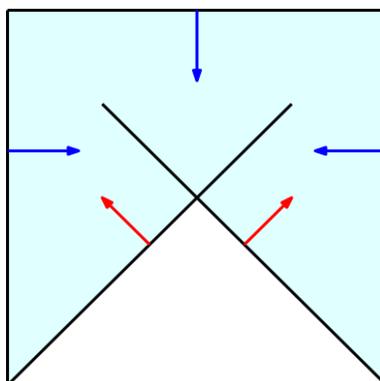


Fig. 8: A plane clipper combination.

Figure *A plane clipper combination* shows a combination of plane clippers which define a non-convex region. First we define the two planes with red arrows, where we keep the outside part of the planes and combine them with the or-function. Then we define the three planes with blue arrows, where we keep the outside part of the plane and combine each of them via the and-function to the previous plane.

Now, let's explain the details:

- Fig. *Data dialog: clipper tab (a)* is the list of `clippers`. Only one clipper can be selected at a time. The clippers can be switched on and off, which is helpful when constructing complicated clippers. The clippers can be renamed by double clicking at a clipper item in this list or using F2.
- Clippers are added, deleted or moved (the order matters as explained above) by the four buttons *Data dialog: clipper tab (b)*. If you hover with the mouse over the buttons, tooltips will be shown, which explain the hotkeys. The add-button will give the option to add a sphere or a plane.
- If the clipper is a sphere the `From Atom` button will be visible, Fig. *Data dialog: clipper tab (c)*. Click it to start picking-mode. Click it again to cancel picking-mode. In picking-mode click on an atom in the structure view to copy its location into the `Center` vector. After picking an atom the picking-mode is canceled. Next, put an appropriate radius into the `radius` cell.
- If it is a plane clipper just enter the plane `direction` in the first three cells and the plane distance in the `distance` cell (not shown in the figure). A plane is defined via  $nr = d$ . The direction vector  $n$  can be given without being normalized. Normalization will be performed internally.
- Next, chose at which side of the clipping object (sphere or plane) the iso surface will be kept (Fig. *Data dialog: clipper tab (d)*).
- If more than one clipper is defined in the list Fig. *Data dialog: clipper tab (a)* you must chose how the current clipper's keep-region is combined with the clipping result of the clippers before the current clipper (Fig. *Data dialog: clipper tab (e)*).
- Finally, the `Options` button Fig. *Data dialog: clipper tab (f)* allows to check/uncheck all clippers or to set the current data set's clippers for all selected data sets (the equivalent of the little green check button discussed above).

## 1.28 Show Symmetry Operations

For orientation the symmetry operations of the chosen group can be visualized with the help of this dialog.

---

**Note:** that in spin-polarized full-relativistic mode a symmetry reduction may be performed while **fplo** runs. This effect depends on the chosen group and the global quantization axis and is not taken into account here (yet).

---

## 1.29 CIF Import Dialog

todo

## 1.30 Atom Label Dialog

todo