Fermisurfer Documentation

Release 2.0.0

kawamura

Jun 27, 2019

Contents

CHAPTER 1

Introduction

This document is a manual for the Fermi surface drawing program "FermiSurfer". FermiSurfer has been developed since 2012 by Mitsuaki Kawamura (ISSP, The University of Tokyo); it is opened on web at November, 2014. It draws Fermi surfaces, and plot k-depend matrix elements such as the superconducting gap and orbital character with colors.

CHAPTER 2

Directories and important files

- **`doc/`** [Directory for manuals]
  - `doc/index.html` : Index page
- `examples/` : Directory for samples
- `src/` : Directory for source code
- `configure` : Configuration script for build

CHAPTER 3

Install

3.1 Installation in Linux

1. Install the required package
   - For Debian/Ubuntu

```
$ sudo aptitude install libwxgtk3.0-dev
```

   - For Red Hat Enterprise Linux/CentOS

```
$ sudo yum install wxGTK3-devel.x86_64
```

2. Install

```
$ ./configure
$ make
$ sudo make install
```

Then a binary file `src/fermisurfer` is generated and copied into `/usr/local/bin/`.

3.2 Installation in Mac and Windows

Download the binary file for each OS.

Alternatively, we can build FermiSurfer by ourselves after we install wxWidgets library. For Mac, we can beuild by using configuare script as in Linux. For Windows, we can use `fermisurfer.vcxproj` together with VisualStudio.

CHAPTER 4

Input file

4.1 input-file format

You have to prepare following data:

- The number of k grid (three direction)
- Reciprocal lattice vectors
- The number of bands
- The orbital energy at each band and k (We call it "energy") .
- Variables that you want to plot with color (We call it "matrix elements").

The input file is as follows (`mgb2_vfz.fs`):

```
40          40          36        (1)
0                                 (2)
3                                 (3)
1.0000000      0.57735026      -0.0000000 (4)
0.0000000       1.1547005       0.0000000 (5)
0.0000000      -0.0000000      0.87206507 (6)
2.91340202E-02                            (7)
2.93242838E-02
2.98905596E-02
3.08193434E-02
:
:
0.14393796
0.12800488
0.0000000                                 (8)
0.36269817
0.71675694
1.0535113
1.3644149
:
```

```
:
-26.409407
-19.318560
-10.315671
```

1. The number of k in each direction
2. Switch to specify type of k grid (Choose from `0, 1, 2`)

   k grid is represented as follows:

   begin{align} {boldsymbol k}_{i,j,k} = x_i {boldsymbol b}_1 + y_j {boldsymbol b}_2 + z_k {boldsymbol b}_3, end{align}

   where $i, j, k = 1 \cdots N_1, 1 \cdots N_2, 1 \cdots N_3$, and N_1, N_2, N_3 are the number of k in each direction.

   x_i, y_j, z_k can be chosen from below:

   - `0` (Monkhorst-Pack grid) : $x_i = \frac{2i-1-N_1}{2N_1}$
   - `1` : $x_i = \frac{i-1}{N_1}$
   - `2` : $x_i = \frac{2i-1}{2N_1}$
3. The number of bands
4. Reciprocal lattice vector 1 (arbitrary unit)
5. Reciprocal lattice vector 2
6. Reciprocal lattice vector 3
7. Energy (The order of component is written in *How to produce the input file in C and fortran programs*)

   `fermisurfer` assume that the Fermi energy is `0.0` in the default. You can shift the Fermi energy by using `Shift Fermi Energy` menu described at the section 6.5.
8. Matrix elements (The order of component is written in *How to produce the input file in C and fortran programs*)

   Same as the energy, but in this case we can write 0 to 3 blocks for this quantity, i.e. we can omit to write this.

4.2 BXSF format

The BXSF format also can be treated by FermiSurfer. In this case this program behaves as "Matrix elements" are omitted.

4.3 How to produce the input file in C and fortran programs

fortran

```
real(4) :: bvec1(3), bvec2(3), bvec3(3) ! Resiplocal lattice vector
integer :: nk1, nk2, nk3 ! k-grid of each direction
integer :: ishift ! 1 for shifted grid, 0 for unshifted grid.
integer :: nbnd ! The number of bands
real(4) :: eig(nk3,nk2,nk1,nbnd) ! energy
real(4) :: x(nk3,nk2,nk1,nbnd) ! matrix element

integer :: ik1, ik2, ik3, ibnd, fo
```

```
open(fo, file = "sample.fs")
write(fo,*) nk1, nk2, nk3
write(fo,*) ishift
write(fo,*) nbnd
write(fo,*) real(bvec1(1:3))
write(fo,*) real(bvec2(1:3))
write(fo,*) real(bvec3(1:3))
do ibnd = 1, nbnd
   do ik1 = 1, nk1
      do ik2 = 1, nk2
         do ik3 = 1, nk3
            write(fo,*) real(eig(ik3,ik2,ik1,ibnd))
         end do
      end do
   end do
end do
do ibnd = 1, nbnd
   do ik1 = 1, nk1
      do ik2 = 1, nk2
         do ik3 = 1, nk3
            write(fo,*) real(x(ik3,ik2,ik1,ibnd))
         end do
      end do
   end do
end do
close(fo)
```

C

```
float bvec1[3], bvec2[3], bvec3[3]; /*Resiplocal lattice vector*/
int nk1, nk2, nk3; /*k-grid of each direction*/
int ishift; /*1 for shifted grid, 0 for unshifted grid.*/
int nbnd; /*The number of bands*/
float eig[nbnd][nk1][nk2][nk3]; /*Energy*/
float x[nbnd][nk1][nk2][nk3]; /*Matrix element*/

FILE* fo;
int ibnd, ik1, ik2, ik3;

fo = fopen("sample.frmsf", "w");
ierr = fprintf(fo, "%d %d %d\n", nk1, nk2, nk3);
ierr = fprintf(fo, "%d\n", iswitch);
ierr = fprintf(fo, "%d\n", nbnd);
ierr = fprintf(fo, "%e %e %e\n", bvec1[0], bvec1[1], bvec1[2]);
ierr = fprintf(fo, "%e %e %e\n", bvec2[0], bvec2[1], bvec2[2]);
ierr = fprintf(fo, "%e %e %e\n", bvec3[0], bvec3[1], bvec3[2]);
for (ibnd = 0; ibnd < nbnd; ++ibnd) {
   for (ik1 = 0; ik1 < nk1; ++ik1) {
      for (ik2 = 0; ik2 < nk2; ++ik2) {
         for (ik3 = 0; ik3 < nk3; ++ik3) {
            ierr = fprintf(fo, "%e\n", eig[ibnd][ik1][ik2][ik3]);
         }
      }
   }
}
for (ibnd = 0; ibnd < nbnd; ++ibnd) {
   for (ik1 = 0; ik1 < nk1; ++ik1) {
```

```
      for (ik2 = 0; ik2 < nk2; ++ik2) {
         for (ik3 = 0; ik3 < nk3; ++ik3) {
            ierr = fprintf(fo, "%e\n", x[ibnd][ik1][ik2][ik3]);
         }
      }
   }
}
fclose(fo);
```

4.4 For the 2D color plot (See srvo3_t2g.frmsf in examples)

fortran

```
real(4) :: bvec1(3), bvec2(3), bvec3(3) !Resiplocal lattice vector
INTEGER :: nk1, nk2, nk3 !k-grid of each direction
integer :: ishift !1 for shifted grid, 0 for unshifted grid.
integer :: nbnd !The number of bands
real(4) :: eig(nk3,nk2,nk1,nbnd) !energy
real(4) :: x(nk3,nk2,nk1,nbnd,2) !matrix element (2D or complex)

integer :: ik1, ik2, ik3, ibnd, fo, ii

open(fo, file = "sample.frmsf")
write(fo,*) nk1, nk2, nk3
write(fo,*) ishift
write(fo,*) nbnd
write(fo,*) real(bvec1(1:3))
write(fo,*) real(bvec2(1:3))
write(fo,*) real(bvec3(1:3))
do ibnd = 1, nbnd
   do ik1 = 1, nk1
      do ik2 = 1, nk2
         do ik3 = 1, nk3
            write(fo,*) real(eig(ik3,ik2,ik1,ibnd))
         end do
      end do
   end do
end do
do ii = 1, 2
   do ibnd = 1, nbnd
      do ik1 = 1, nk1
         do ik2 = 1, nk2
            do ik3 = 1, nk3
               write(fo,*) real(x(ik3,ik2,ik1,ibnd,ii))
            end do
         end do
      end do
   end do
close(fo)
```

4.5 Omit the quantity for the color plot

fortran

```
real(4) :: bvec1(3), bvec2(3), bvec3(3) ! Resiplocal lattice vector
INTEGER :: nk1, nk2, nk3 ! k-grid of each direction
integer :: ishift ! 1 for shifted grid, 0 for unshifted grid.
integer :: nbnd ! The number of bands
real(4) :: eig(nk3,nk2,nk1,nbnd) ! energy

integer :: ik1, ik2, ik3, ibnd, fo, ii

open(fo, file = "sample.frmsf")
write(fo,*) nk1, nk2, nk3
write(fo,*) ishift
write(fo,*) nbnd
write(fo,*) real(bvec1(1:3))
write(fo,*) real(bvec2(1:3))
write(fo,*) real(bvec3(1:3))
do ibnd = 1, nbnd
   do ik1 = 1, nk1
      do ik2 = 1, nk2
         do ik3 = 1, nk3
            write(fo,*) real(eig(ik3,ik2,ik1,ibnd))
         end do
      end do
   end do
end do
```

CHAPTER 5

Control FermiSurfer

5.1 Launch

5.1.1 For Linux, Unix, Mac

You can launch generated executable as follows:

```
$ fermisurfer mgb2_vfz.fs
```

You need a space between the command and input-file name. (The sample input file `mgb2_vfz.fs` contains z element of the Fermi velocity in MgB_2.)

5.1.2 For Windows

Click mouse right button on the input file. Choose "Open With ..." menu, then choose `fermisurfer.exe`.

Then, Operations are printed, and Fermi surfaces are drawn (Fig. 5.1).

The following operations are available:

- Rotation of objects with mouse drag
- Expand and shrink with mouse wheel
- Window re-sizing
- Moving objects with cursor keys (wasd for Windows)
- Opeerate by using the panel

Here, I will explain all menus.

Note: Some operations are not applied immidiately, and after th "Update" button is pushed they are applied. Such operations are refered as "Update required".

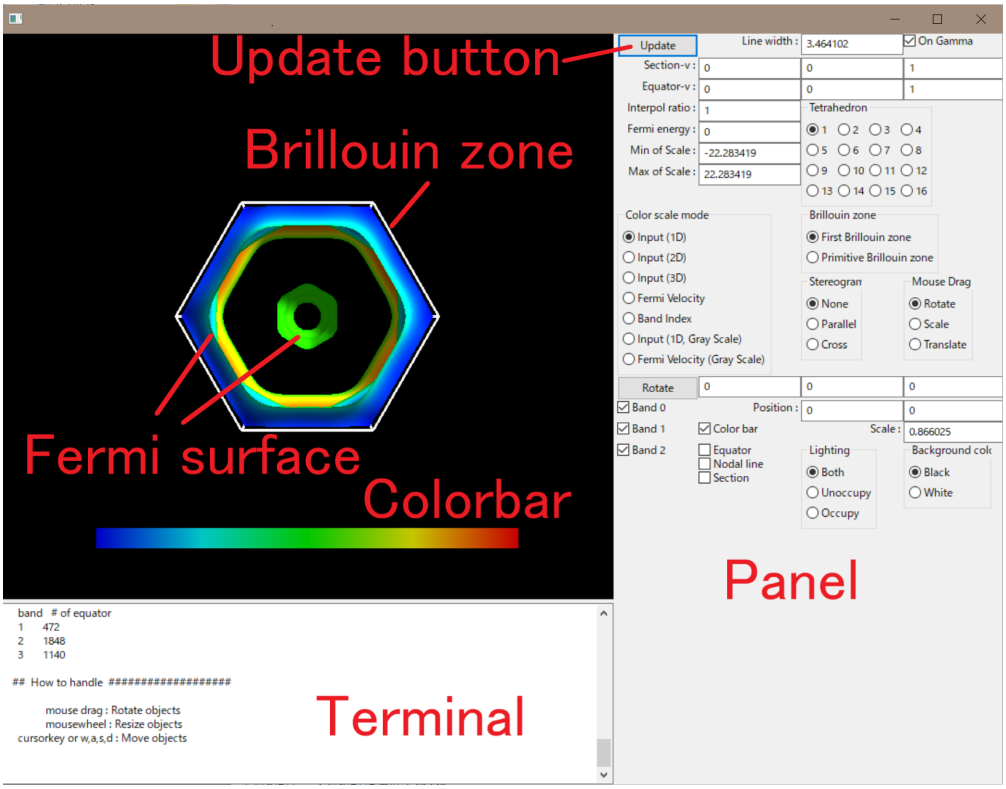


Fig. 5.1: Main view.

5.2 Background color

The background color is toggled between black and white; the edge of the Brillouin Zone is also toggled between white and black (Fig. 5.2).

5.3 Band

It makes each band enable/disable (Fig. 5.3).

5.4 Brillouin zone (Update required)

You choose Brillouin-zone type as follows (Fig. 5.3):

First Brillouin Zone The region surrounded by Bragg's planes the nearest to Γ point.

Primitive Brillouin Zone A hexahedron whose corner is the reciprocal lattice point.

5.5 Color bar

The color bar becomes enable/disable (Fig. 5.4).

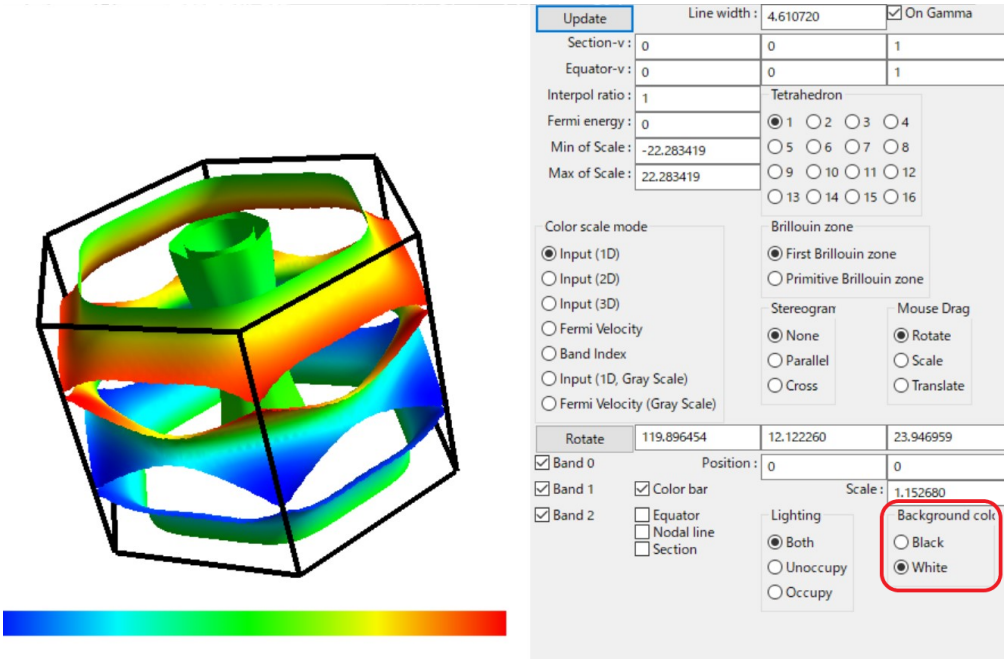
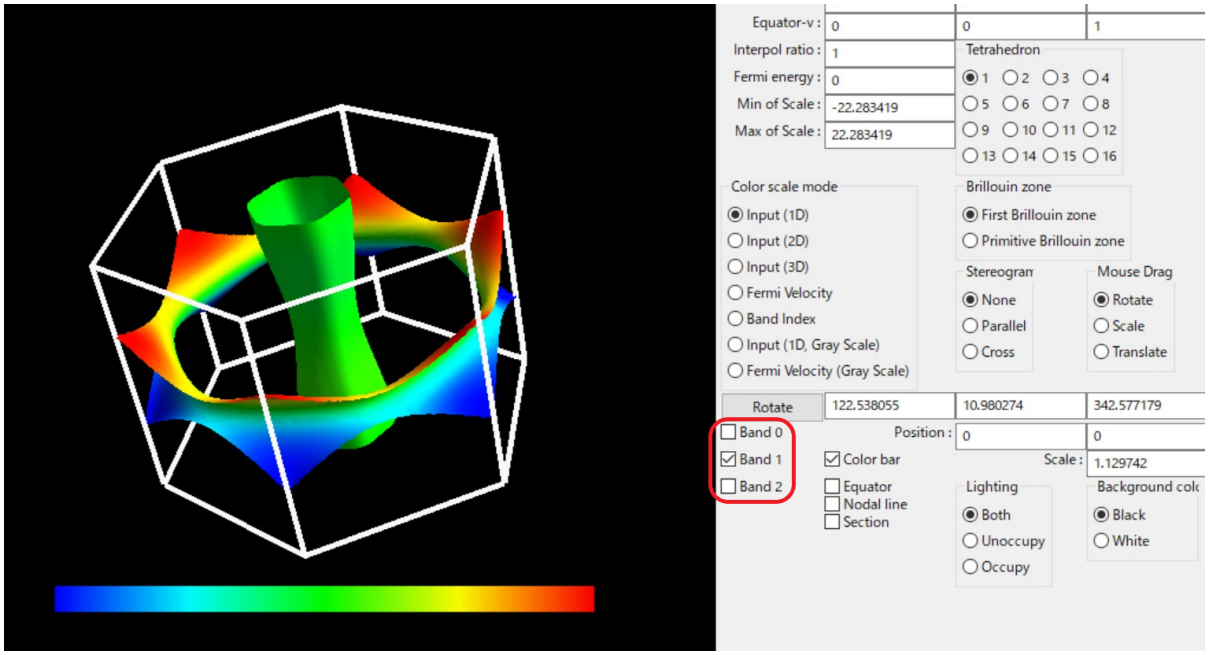


Fig. 5.2: The background color is toggled.



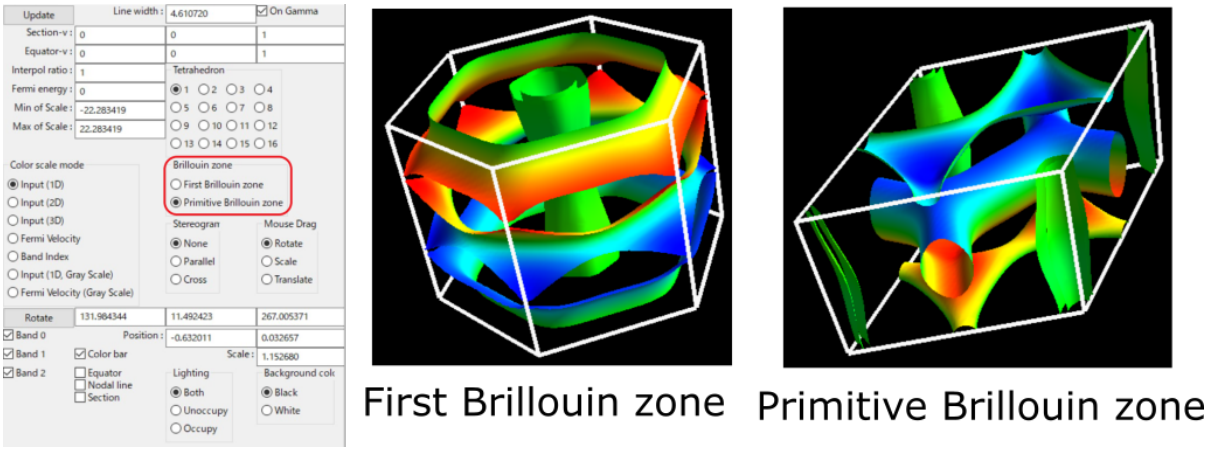


Fig. 5.3: You can change the type of the Brillouin zone with “Brillouin zone” menu.

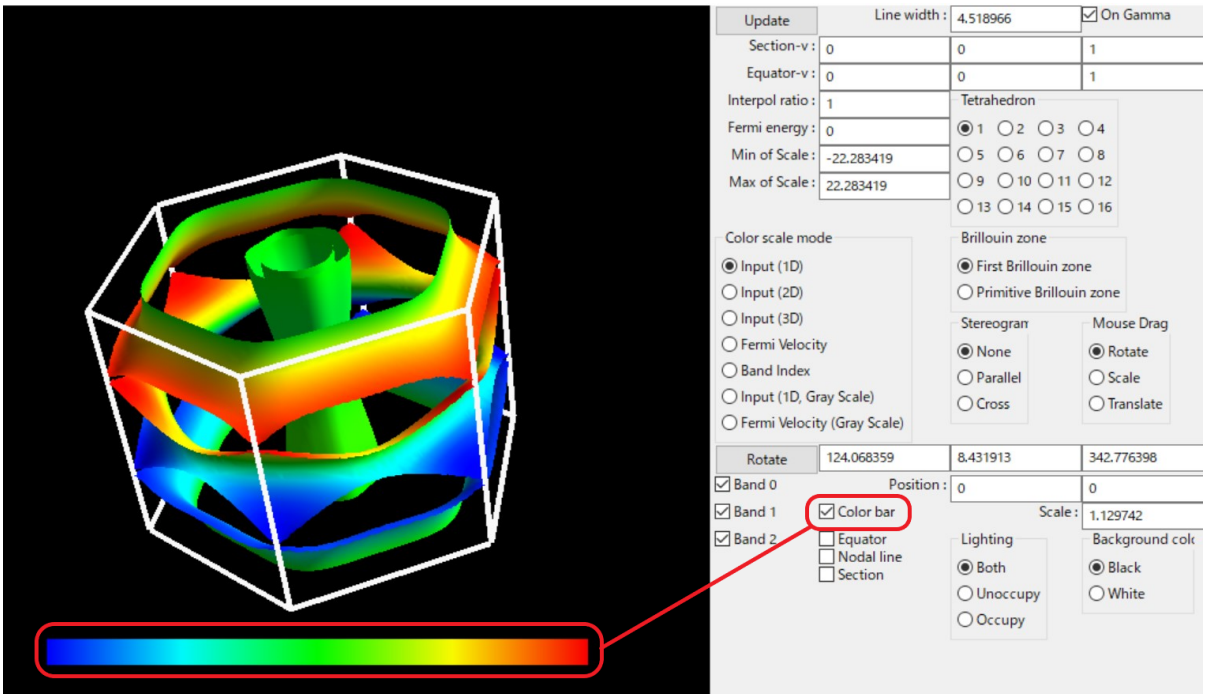


Fig. 5.4: Toggling the color bar with “Color bar On/Off” menu.

5.6 Color scale mode (Update required)

It turns color pattern on Fermi surfaces (Fig. 5.5).

Input (1D) (default for the single input quatity) : It makes blue as the minimum on Fermi surfaces and red as the maximum on them.

Input (2D) (default for the double input quatity) : The color plot is shown with the color circle (see the figure).

Input (3D) (default for the triple input quatity) : The input value is shown as arrows on the Fermi surfaces. The color of the Fermi surfaces are the same sa "Band Index" case.

Fermi velocity (default for no input quantity) Compute the Fermi velocity $\mathbf{v}_\mathrm{F} = \nabla_k \varepsilon_k$ with the numerical differentiation of the energy, and plot the absolute value of that.

Band Index : Fermi surfaces of each band are depicted with uni-color without relation to the matrix element.

Input (1D, Gray), Fermi Velocity (Gray) : Plot with gray scale.

We can change the range of the color plot or the length of arrows for 3D line plot by inputting into the text boxes at "Min of Scale" and "Max of Scale", respectively.

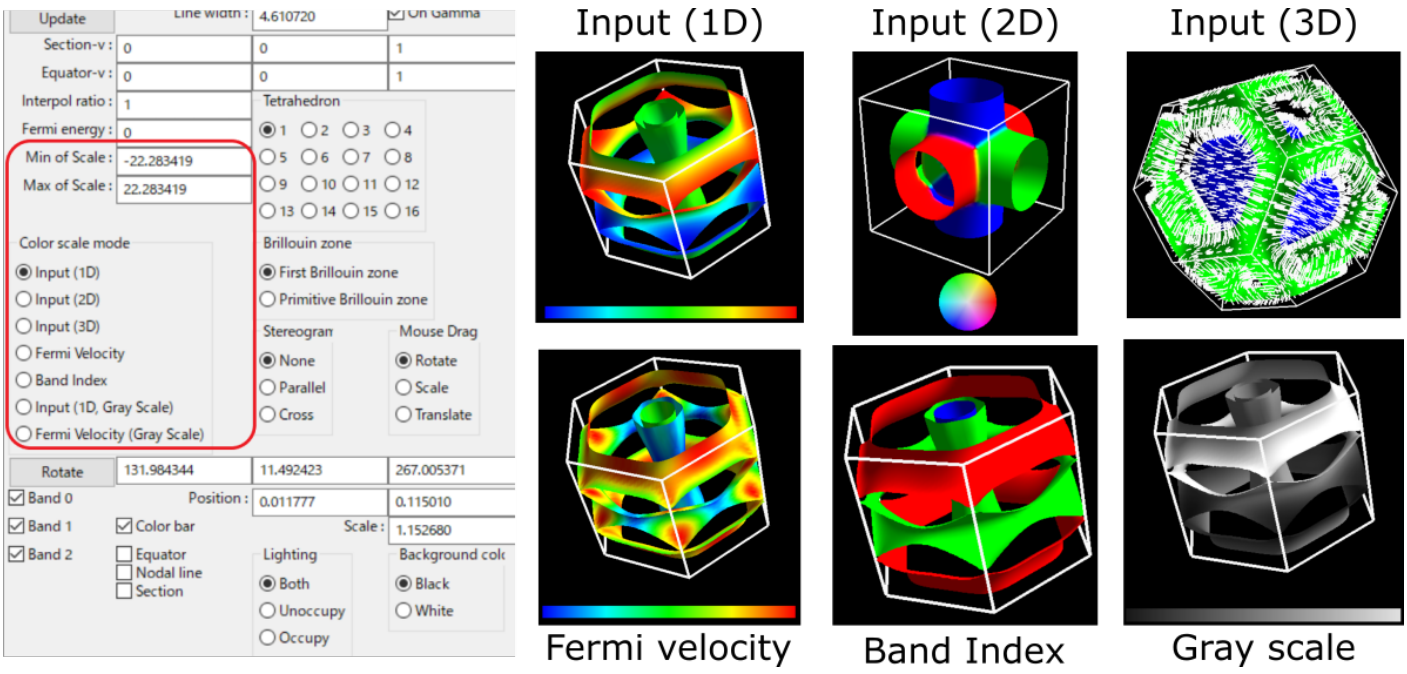


Fig. 5.5: "Color scale mode" menu.

5.7 Equator (Update required)

We can draw the line where $\mathbf{v}_\mathrm{F} \cdot \mathbf{k} = 0$ for a vector $\mathbf{k}$ (equator or extremal orbit). See fig. 5.6. We can toggle equator with the checkbox "Equator" (this operation doed not require the update, and modify the direction of the tangent vector $\mathbf{k}$ by using the textbox at "Equator-v :" (**fractional coordinate**).

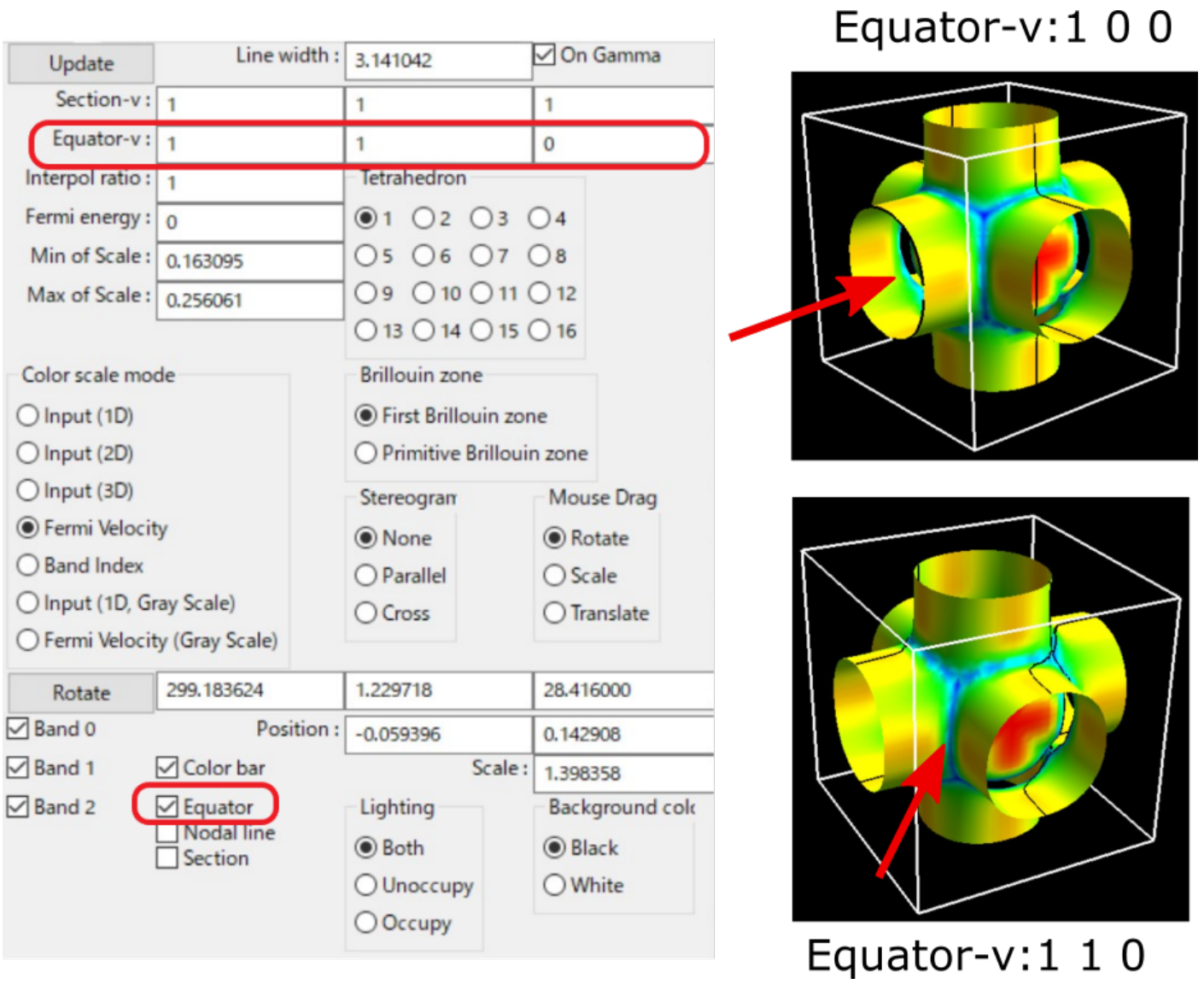


Fig. 5.6: Display the equator with the “Equator” menu.

5.8 Interpolation (Update required)

Smooth the Fermi surface with the interpolation (Fig. 5.7). The time for the plot increases with the interpolation ratio.

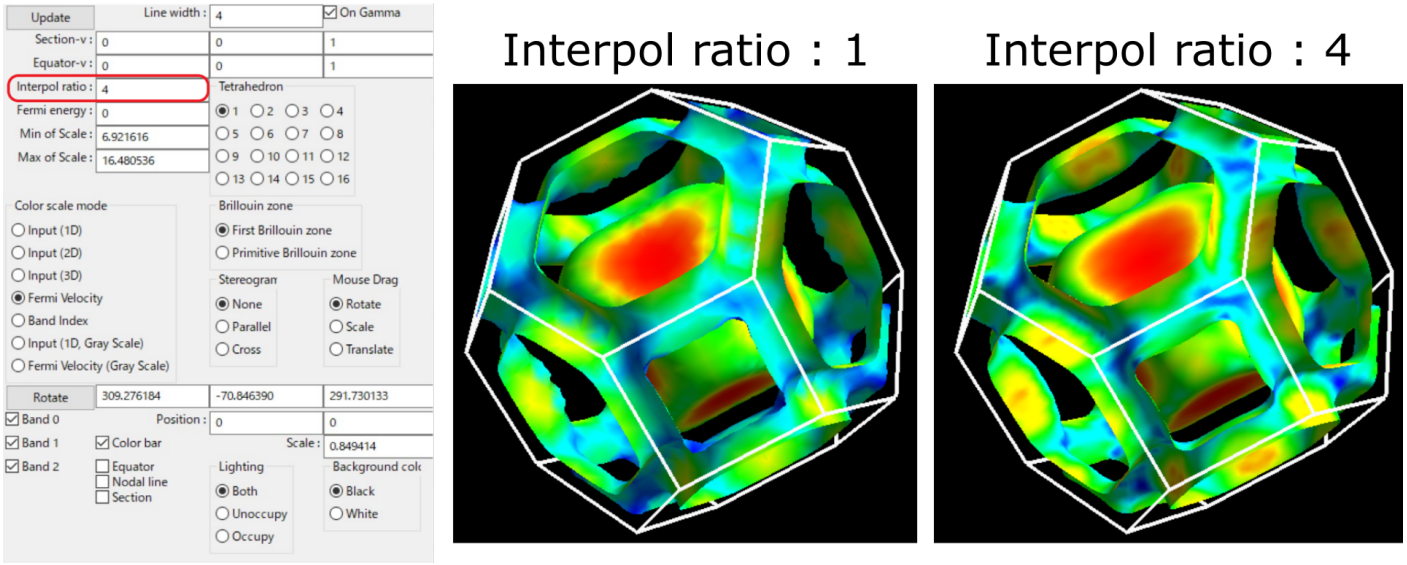


Fig. 5.7: Modify the number of interpolation points from 1 to 4 with “Interpolate” menu.

5.9 Which (or both) side of Fermi surface is illuminated

We can choose the illuminatedside of the Fermi surface (Fig. 5.8).

Both : Light both sides.

Unoccupy : Light unoccupied side.

Occupy : Light the occupied side.

5.10 Line width

Modify the width of the Brillouin-zone boundary, the nodal line, etc.

5.11 Mouse Drag

It turns the event of the mouse-left-drag.

Rotate(default) Rotate the figure along the mouse drag.

Scale Expand/shrink the figure in upward/downward drag.

Translate Translate the figure along the mouse drag.

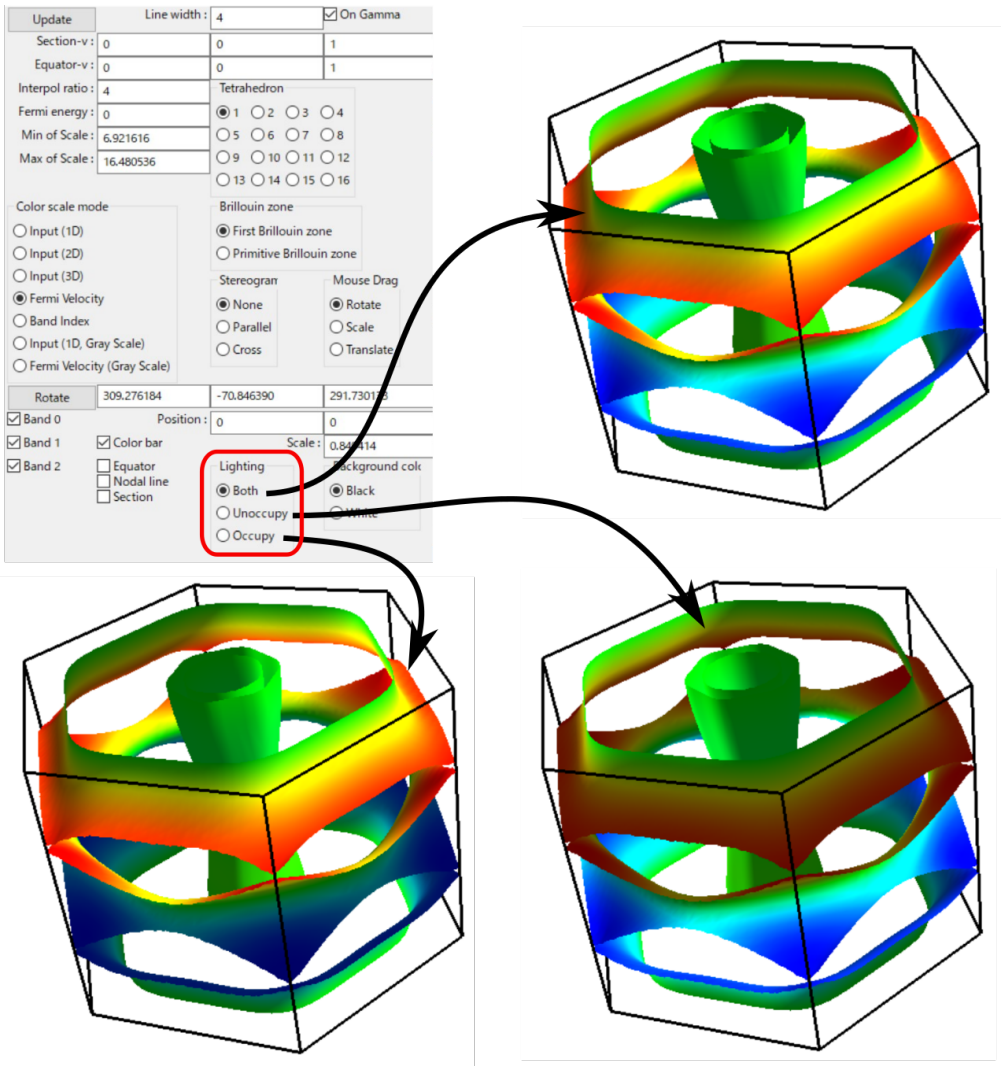
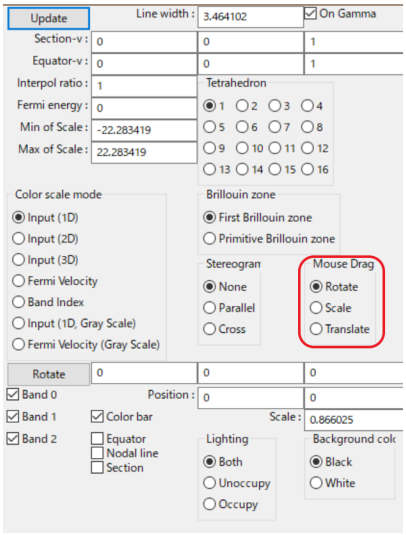


Fig. 5.8: Change the lighted side by using the “Lighting” menu.



5.12 Nodal line

The line on which the matrix element becomes 0 (we call it nodal line) becomes enable/disable (Fig. 5.9).

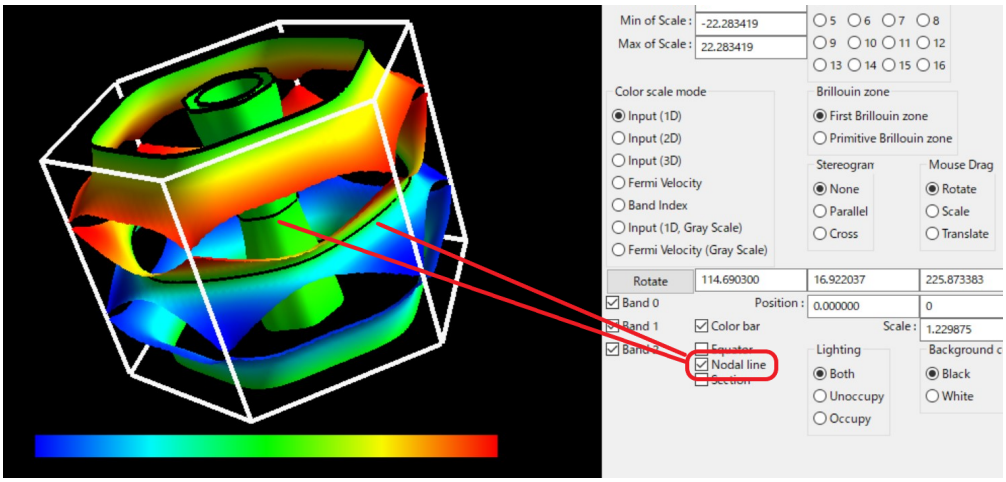


Fig. 5.9: Toggling the node line with “Nodal line” menu.

5.13 Section of the Brillouine zone (Update required)

Display a 2D plot of the Fermi surface (line) on an arbitrary section of the Brillouin zone (Fig. 5.10).

We can toggle it with the checkbox “Section” (this operation does not require update), and can change the normal vector with the textbox at “Section-v :” (**fractional coordinate**).

If the checkbox “On Gamma” is turned on, the section crosses Γ point.

5.14 Shift Fermi energy (Update required)

It shifts the Fermi energy (= 0 in default) to arbitrary value (Fig. 5.11).

5.15 Stereogram

The stereogram (parallel eyes and cross eyes) becomes enabled/disabled (Fig. 5.12).

None (Default)

Parallel Parallel-eyes stereogram

Cross Cross-eyes stereogram

5.16 Tetrahedron (Update required)

You change the scheme to divide into tetrahedra (`tetra # 1` as default). It is experimental.

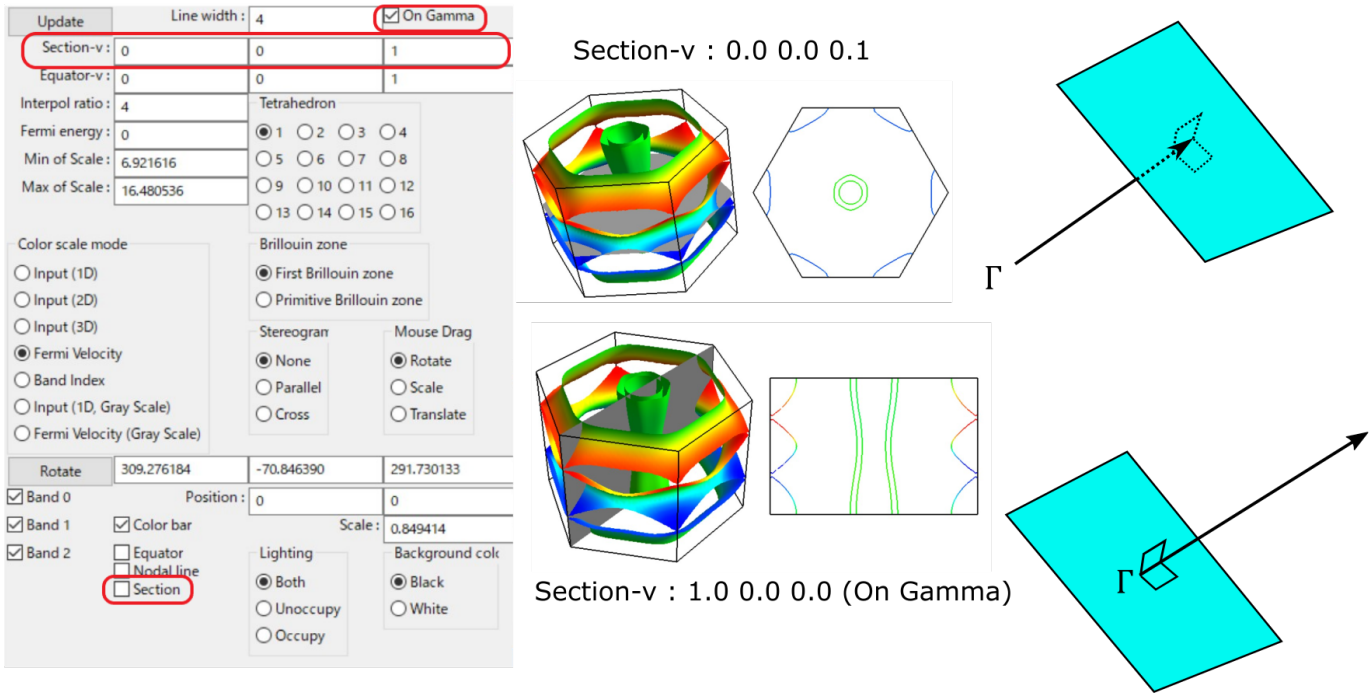


Fig. 5.10: Display 2D plot of the Fermi surface (line) with “Section” menu.

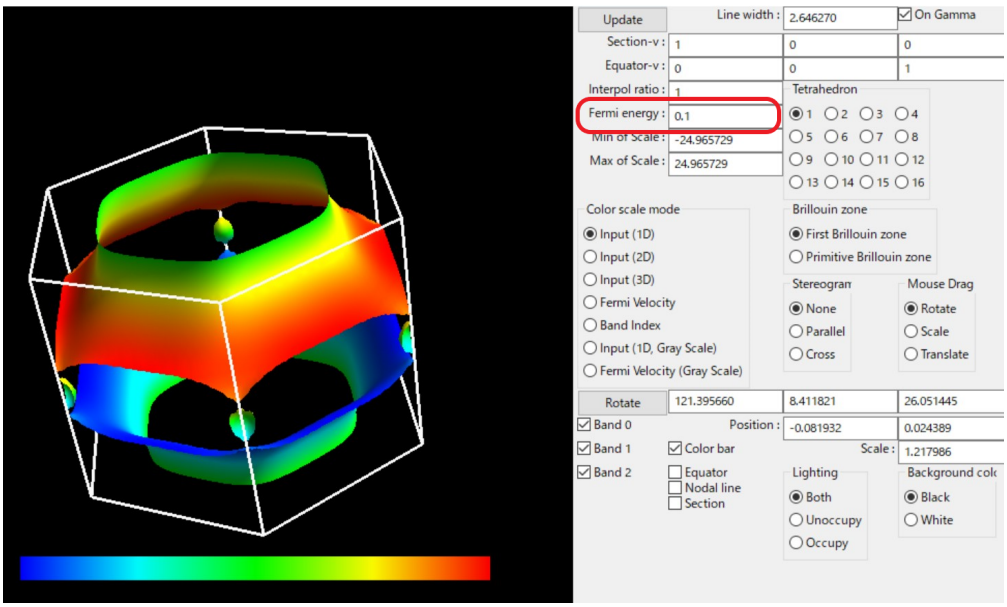


Fig. 5.11: The Fermi energy is set from 0 Ry to 0.1 Ry with “Shift Fermi energy” menu

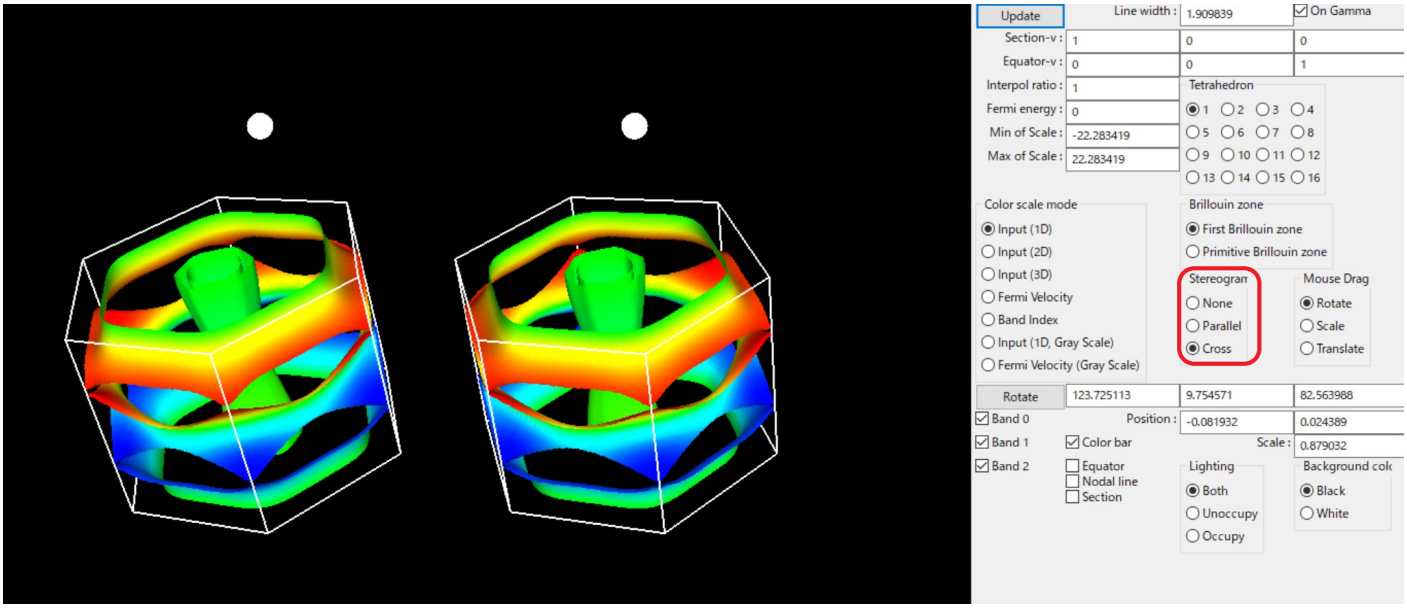
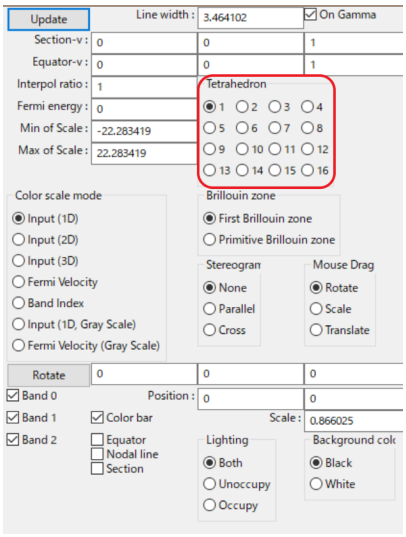


Fig. 5.12: The stereogram becomes enabled/disabled with “Stereogram” menu.



5.17 View point

Changing the view point.

Scale Change the size of the figure.

Position Change the xy position of the figure.

Rotate Change angles at x-, y-, z- axis. Rotations are performed as z-y-x axis if the "Roate" buttone is pushed.

In each menu, first the current value is printed. then a prompt to input the new value appears (Fig. 5.13).

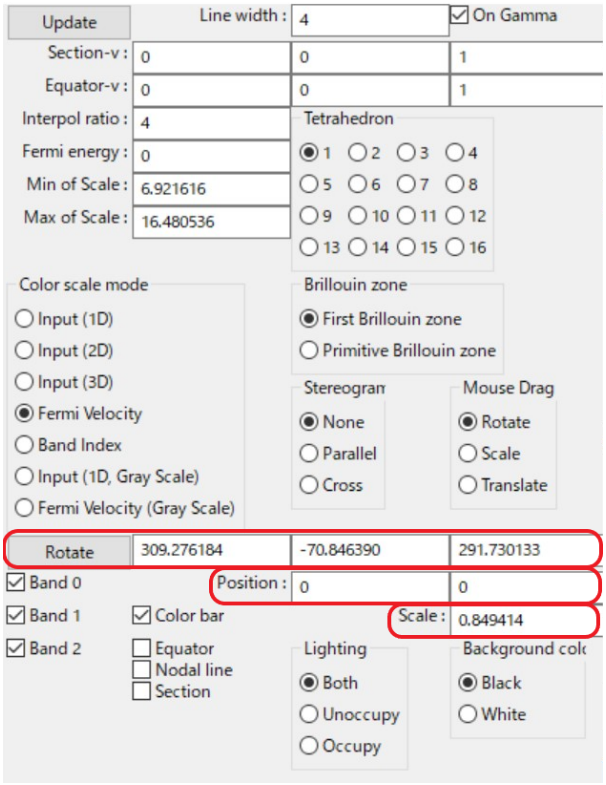


Fig. 5.13: Modify the view point by using "View point" menu

5.18 Saving images

`fermisurfer` does not have any functions to save images to a file. Please use the screenshot on your PC.

CHAPTER 6

Batch mode

By using "Batch mode", we can generate an image (PNG) file drawn by FermiSurfer only with the command-line operation. By using this batch mode, we can easily make this kind of many figures.

For example, in `example/` directory, when we execute

```
$ fermisurfer mgb2_vfz.frmsf frmsf.in 500 500
```

we will obtain an image file `frmsf.in.png`. The last two numbers are the width and the height of the window. `frmsf.in` is a configuration file for the batch mode; its contents are as follows:

```
   background  black
         band  0 0 1
#brillouinzone  primitive
     colorbar  1
   colorscale  fermivelocity
       minmax  -22 22
#      equator  1.0 0.0 0.0
     interpol  4
    linewidth  3.0
     lighting  both
    nodalline  0
#      section  1.0 0.0 0.0
  acrossgamma  1
     position  0.0 0.0 0.0
        scale  1.0
     rotation  120.0 40.0 0.0
  fermienergy  0.0
   stereogram  none
  tetrahedron  1
```

They are corresponding to the operations in the panel written in the previos section, and the available keywords are as follows (for the ignored keyword, each default value is used) :

Keyword	Available parameter	Default value	Description
background	black, white	black	Background color
band	1 or 0 for each band	1 1 1 1 . . .	Show(1) or hide(0) each band
brillouinzone	first, primitive	first	Kind of the Brillouin zone
colorbar	0, 1	1	Show(1) or hide (0) the color bar
colorscale	input1d, input2d, input3d, fermivelocity, bandindex, inputgray, fermivelocitygray	input1d	Kind of the color plot
minmax	float float	The min. and max. through Fermi surfaces	the range of the color scale
equator	float float float	If it is not specified, equator is not shown	Tangent vector for equator (fractional coordinate)
interpol	int	1	Degree of the interpolation
linewidth	float	3.0	Line width
lighting	both, unoccupied, occupied	both	Which side is illuminated
nodalline	0, 1	0	Show(1) or hide(0) the nodal line
section	float float float	Section is not shown	Normal vector for the section (fractional coordinate)
acrossgamma	0, 1	1	Whether math:*{rm Gamma}* is included (1) or not (0) in the section.
position	float float	0.0, 0.0	The position of the figure
scale	float	1.0	The scale of the figure
rotation	float float float	0.0, 0.0, 0.0	Rotation arround x-, y-, and z-axis
fermienergy	float	0.0	Fermi energy
stereogram	none, parallel, cross	none	Stereogram
tetrahedron	int from 0 to 15	0	Direction to cut tetrahedra

Note: This function uses "import" command to get the screen-shot in ImageMagic. Therefore ImageMagic have to be installed to use this function.

CHAPTER 7

Tutorial with Quantum ESPRESSO

Since the version 6.2, Quantum ESPRESSO can generate data-files for FermiSurfer. The following quantities can be displayed through FermiSurfer.

- The absolute value of the Fermi velocity $|\mathbf{v}_\mathrm{F}|$ (`fermi_velocity.x`).
- The projection onto each atomic orbital $|\langle\phi_{nlm}|\psi_{nk}\rangle|^2$ (`fermi_proj.x`)

7.1 Building `PostProcess` tool

For displaying the above quantities with FermiSurfer, we have to build `PostProcess` tools (tools for plotting the band structure, the charge density, etc.) in QuantumESPRESSO as follows:

```
$ make pp
```

7.2 SCF calculation

Now we will move on the tutorial. First, we perform the electronic-structure calculation with `pw.x`. We will treat MgB_2 in this tutorial. The input file is as follows.

`scf.in`

```
&CONTROL
 calculation = 'scf',
  pseudo_dir = './',
      prefix = 'mgb2' ,
      outdir = './'
/
&SYSTEM
       ibrav = 4,
   celldm(1) = 5.808563789,
   celldm(3) = 1.145173082,
```

```
         nat  = 3,
        ntyp  = 2,
     ecutwfc  = 50.0 ,
     ecutrho  = 500.0 ,
 occupations  = 'tetrahedra_opt',
/
&ELECTRONS
/
ATOMIC_SPECIES
 Mg    24.3050   Mg.pbe-n-kjpaw_psl.0.3.0.upf
  B    10.811    B.pbe-n-kjpaw_psl.0.1.upf
ATOMIC_POSITIONS crystal
 Mg    0.000000000    0.000000000    0.000000000
  B    0.333333333    0.666666667    0.500000000
  B    0.666666667    0.333333333    0.500000000
K_POINTS automatic
16 16 12 0 0 0
```

Pseudopotentials used in this example are included in PS Library, and they can be downloaded from the following address:

- http://theossrv1.epfl.ch/uploads/Main/NoBackup/Mg.pbe-n-kjpaw_psl.0.3.0.upf
- http://theossrv1.epfl.ch/uploads/Main/NoBackup/B.pbe-n-kjpaw_psl.0.1.upf

We put the input file and the pseudopotential in the same directory, and run `pw.x` at that directory.

```
$ mpiexec -np 4 pw.x -npool 4 -in scf.in
```

the number of processes and the number of blocks for *k*-paralleliztion (`npool`) can be arbitlary numbers. We also can perform additional non-scf calcultion with a different *k*-grid.

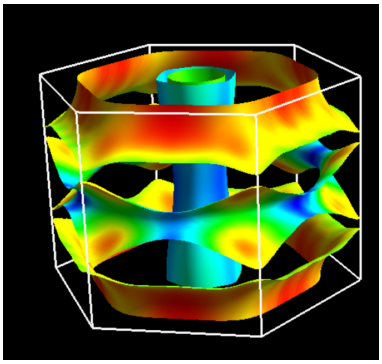
7.3 Compute and display Fermi velocity

We run `fermi_velocity.x` program with the same input file as `pw.x`.

```
$ mpiexec -np 1 fermi_velocity.x -npool 1 -in scf.in
```

For this calculation, the number of blocks for *k*-paralleliztion (`npool`) should be 1 (or not specified). Then, the file for the Fermi velocity, `vfermi.frmsf`, is generated; this file can be read from FermiSurfer as

```
$ fermisurfer vfermi.frmsf
```



For the case of the collinear spin calculation, two files, `vfermi1.frmsf` and `vfermi2.frmsf` associated to each spin are generated.

7.4 Compute and display projection onto the atomic orbital

Then we will computeb the projection onto the atomic orbital. First we run `projwfc.x` with the following input file:

`proj.in`

```
&PROJWFC
   outdir = './'
    prefix='mgb2'
    Emin=-0.3422,
    Emax=10.0578,
    DeltaE=0.1
/
2
6 10
```

The input dates after the end of the name-list `PROJWFC` (`/`) is not used by `projwfc.x`. The number of processes and the number of blocks for the *k*-parallelization (`npool`) must to be the same as those for the execution of `pw.x`.

```
$ mpiexec -np 4 projwfc.x -npool 4 -in proj.in
```

excepting `wf_collect=.true.` in the input of `pw.x`.

the following description can be found in the beginning of the standard output of `projwfc.x`.

```
Atomic states used for projection
(read from pseudopotential files):

state #   1: atom   1 (Mg ), wfc  1 (l=0 m= 1)
state #   2: atom   1 (Mg ), wfc  2 (l=1 m= 1)
state #   3: atom   1 (Mg ), wfc  2 (l=1 m= 2)
state #   4: atom   1 (Mg ), wfc  2 (l=1 m= 3)
state #   5: atom   2 (B  ), wfc  1 (l=0 m= 1)
state #   6: atom   2 (B  ), wfc  2 (l=1 m= 1)
state #   7: atom   2 (B  ), wfc  2 (l=1 m= 2)
state #   8: atom   2 (B  ), wfc  2 (l=1 m= 3)
state #   9: atom   3 (B  ), wfc  1 (l=0 m= 1)
state #  10: atom   3 (B  ), wfc  2 (l=1 m= 1)
state #  11: atom   3 (B  ), wfc  2 (l=1 m= 2)
state #  12: atom   3 (B  ), wfc  2 (l=1 m= 3)
```

This indicates the relationship between the index of the atomic orbital (`state #`) and its character (for more details, please see `INPUT_PROJWFC.html` in QE). When we choose the projection onto the atomic orbital plotted on the Fermi surface, we use this index. For example, we run `fermi_proj.x` with above `proj.in` as an input file,

```
$ mpiexec -np 1 fermi_proj.x -npool 1 -in proj.in
```

and we obtain the data-file for FermiSurfer, `proj.frmsf`. In this case, after `/` in `proj.in`

```
2
6 10
```

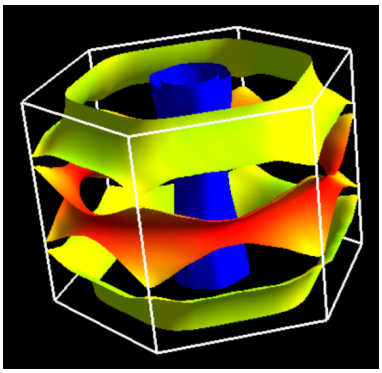
we specify the total number of the displayed projection onto the atomic orbital as the first value (`2`) and projections to be summed as following indices. In this input, the sum of the 2pz of the first B atom (`6`) and the 2pz of the first B

atom (10),

$$|\langle\phi_{B_1 2pz}|\psi_{nk}\rangle|^2 + |\langle\phi_{B_2 2pz}|\psi_{nk}\rangle|^2$$

is specified. We can display the Fermi surface as

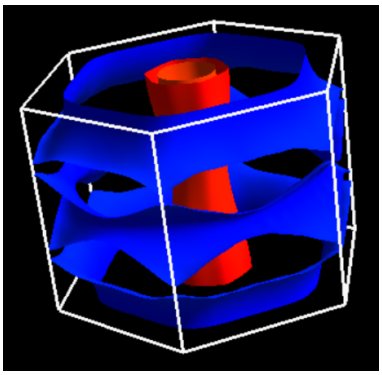
```
$ fermisurfer proj.frmsf
```



If we want to plot the projections onto 2px and 2py orbitals of all B atoms, the input file for `fermi_proj.x` becomes

```
&PROJWFC
outdir = './'
prefix='mgb2'
Emin=-0.3422,
Emax=10.0578,
DeltaE=0.1
/
4
7 8 11 12
```

We do not have to run `projwfc.x` again.



CHAPTER 8

Acknowledgment

I thank Dr. Yusuke Konishi in ISSP; he performed a test in Mac OSX, and proposed Makefiles and a patch.

CHAPTER 9

Re-distribution of this program

9.1 Contain Fermisurfer in your program

FermiSurfer is distributed with the *MIT License*. To summarize this, you can freely modify, copy and paste FermiSurfer to any program such as a private program (in the research group, co-workers, etc.), open-source, free, and commercial software. Also, you can freely choose the license to distribute your program.

9.2 MIT License

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 10

Contact

Please post bag reports and questions to the forum

http://sourceforge.jp/projects/fermisurfer/forums/

When you want to join us, please contact me as follows.

The Institute of Solid State Physics

Mitsuaki Kawamura

mkawamura__at__issp.u-tokyo.ac.jp