Fermisurfer Documentation

Release 1.10.1

kawamura

March 08, 2018

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

- **`bin/`** [Directory for executable for Windows]
  - `bin/fermisurfer.exe` : Executable for Windows (Main program)
  - `bin/bxsf2frmsf.exe` : Executable for Windows (Utility to convert file)
- **`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

Installation in Linux

1. Install the required package
   - For Debian/Ubuntu

```
$ sudo aptitude install freeglut3-dev
```

   - For Red Hat Enterprise Linux/CentOS

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

2. Install

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

Then binary files `src/fermisurfer` and `src/bxsf2frmsf` are generated and copied into `/usr/local/bin/`.

For Mac OSX

1. Install Xcode.
2. Install

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

Then binary files `src/fermisurfer` and `src/bxsf2frmsf` are generated and copied into `/usr/local/bin/`.

Installation in Windows

Use an executable `bin/fermisurfer.exe` . You can also compile `fermisurfer` manually by using freeglut library.

CHAPTER 4

Input file

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*)

   If you have no quantity to plot on Fermi surfaces, please use `Unicolor` switch in `Color scale mode` menu described at section 6.7. In that case, although this "Matrix elements" is not used by `fermisurfer`, please fill them with arbitrary numbers.

Converting bxsf file

You can generate input files for `fermisurfer` from an input file for XCrysDen (the bxsf format) by using the utility program `bxsf2frmsf`.

The usage is as follow (we use `examples/pb.bxsf` as an example.):

For Linux

You can launch generated executable as follows:

```
$ bxsf2frmsf pb.bxsf
```

You need a space between the command and input-file name. After it finishes, the following files are generated:

- `pb_vf.frmsf`: Absolute value of the Fermi velocity
- `pb_vfx.frmsf`: x component of the Fermi velocity
- `pb_vfy.frmsf`: y component of the Fermi velocity
- `pb_vfz.frmsf`: z component of the Fermi velocity
- `pb_vfa1.frmsf`: Component of the Fermi velocity (along $\mathbf{a}_1$)
- `pb_vfa2.frmsf`: Component of the Fermi velocity (along $\mathbf{a}_2$)

- `pb_vfa3.frmsf`: Component of the Fermi velocity (along $\mathbf{a}_3$)

For Windows

Click mouse right button on the input file. Choose "Open With ..." menu, then choose `bxsf2frmsf.exe`.

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);
```

CHAPTER 5

Control FermiSurfer

Launch

For Linux

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.)

For Windows

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

After that, `fermisurfer` runs as the same whether you use Linux or Windows. The information from the input file is printed.

```
########################################
##                                    ##
##  Welocome to FermiSurfer ver. 1.8  ##
##                                    ##
########################################

  Number of threads : 4

  Initialize variables ...


  ##  Brillouin zone informations  ###########

    k point grid : 40 40 36
    k point grid starts from Gamma.
    # of bands : 3
    bvec 1 : 1.000000 0.577350 -0.000000
    bvec 2 : 0.000000 1.154701 0.000000
    bvec 3 : 0.000000 -0.000000 0.872065

  ##  Max. and Min. of each bands  ######################
```

```
   Band   Eig_Min.       Eig_Max      Mat_Min      Mat_Max
   1     -0.428153     0.056262     -24.048639     24.048639 (1)
   2     -0.289572     0.121181     -23.320309     23.320309 (1)
   3     -0.133566     0.497620     -43.651634     43.651634 (1)

 ##  First Brillouin zone mode  #######

   band   # of patchs
   1       8824  (2)
   2       29354 (2)
   3       28293 (2)


 ##  Full color scale mode #############

   Max. value : 22.283419  (3)
   Min. value : -22.251053 (3)

   band   # of nodeline
   1       632  (4)
   2       1524 (4)
   3       2268 (4)
   band   # of Fermi-line
   1       100
   2       736
   3       0

 ##  How to handle  ###################

             mouse drag : Rotate objects
             mousewheel : Resize objects
              cursorkey : Move objects
     mouse right button : Menu
```

1. The maximum/minimum value of energies and matrix elements in each bands.
2. The number of patches (planes that makes Fermi surfaces) in each bands.
3. The maximum and the minimum of matrix elements on Fermi surface. These correspond to the red and the blue; in this case, the matrix element is -22.283419 in the blue region, and that is 22.283419 in the red region. [(1) is Max./Min. in whole Brillouin zone.]
4. The number of node lines in each band.

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
- Opening the menu with mouse right button

Here, I will explain all menus.

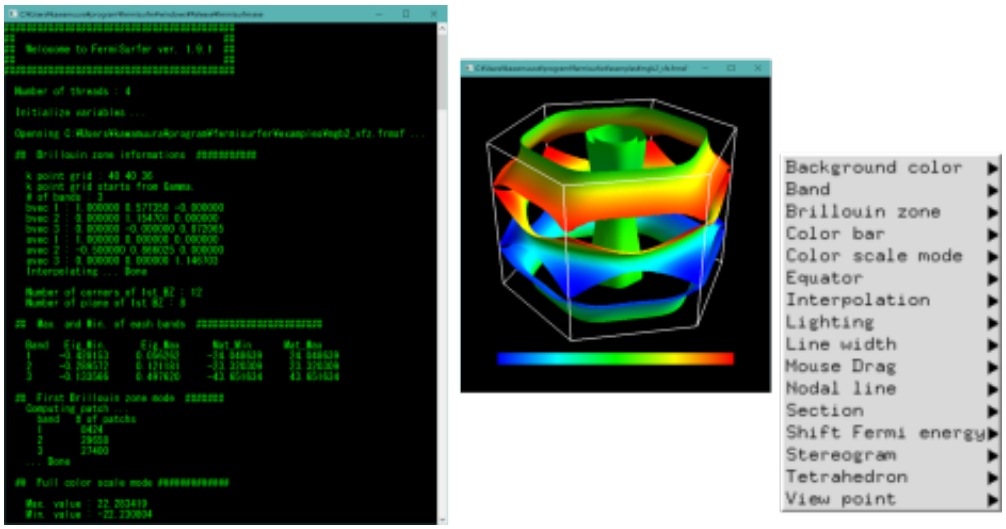


Fig. 5.1: Main view.

Band

It makes each band enable/disable (Fig. 5.2).

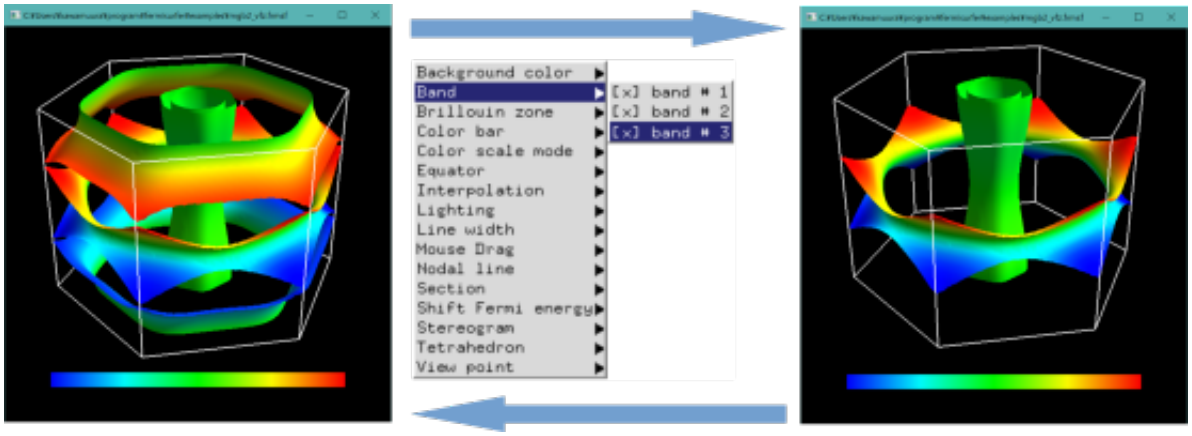


Fig. 5.2: You make each band enable/disable with “Band” menu.

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. 5.3).

Brillouin zone

You choose Brillouin-zone type as follows (Fig. 5.4):

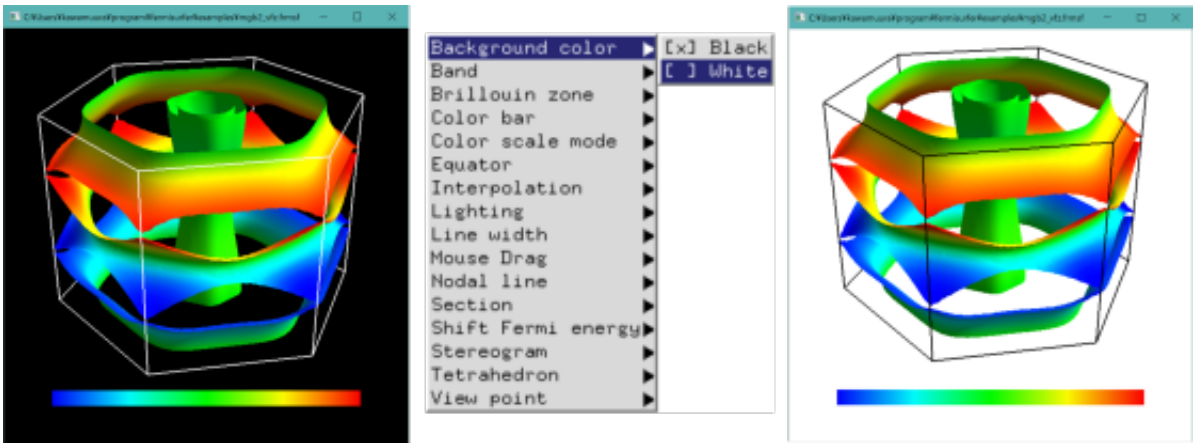


Fig. 5.3: The background color is toggled with "Background color" menu.

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.

Fig. 5.4: You can change the type of the Brillouin zone with "Brillouin zone" menu.

Color bar

The color bar becomes enable/disable (Fig. 5.5).

Color scale mode

It turns color pattern on Fermi surfaces (Fig. 5.6).

Auto(default) It makes blue as the minimum on Fermi surfaces and red as the maximum on them.

Manual You can set manually (from standard input) values corresponding to blue and red.

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

Periodic It makes periodic color plot enable. When the matrix element varies as $0 \to \pi/3 \to 2\pi/3 \to \pi \to 4\pi/3 \to 5\pi/3 \to 2\pi$, the color varies as red $\to$ yellow $\to$ green $\to$ cyan $\to$ blue $\to$ magenta $\to$ red.

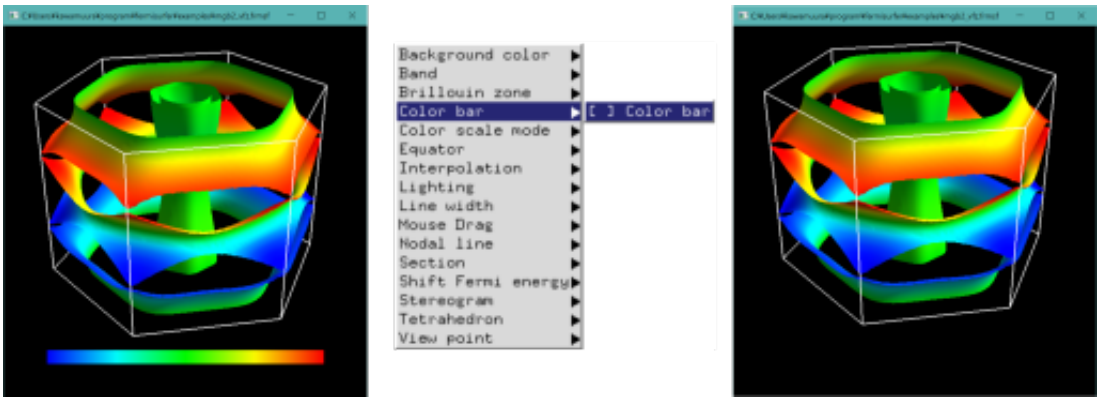


Fig. 5.5: Toggling the color bar with "Color bar On/Off" menu.

Fermi velocity (Auto) 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. The color scale spans from the minimum and the maximum of $|\mathbf{v}_\mathrm{F}|$ on the Fermi surface.

Fermi velocity (Manual) 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. The range of color scale can be specified manually from the terminal.

Gray scale (Manual), Gray scale (Auto) Plot with gray scale.

Equator

We can draw the line where $\mathbf{v}_\mathrm{F} \cdot \mathbf{k} = 0$ for a vector $\mathbf{k}$ (equator). See fig. 5.7

Equator Toggle the equator.

Modify equator We specify the $\mathbf{k}$ vector. Please type a vector (**fractional coordinate**) at the prompt

```
    New Miller index :
```

Interpolation

Smooth the Fermi surface with the interpolation (Fig. 5.8). Please type the number of interpolation points at the prompt

```
New interpolation ratio :
```

The time for the plot increases with the interpolation ratio.

Lighting

We can choose the lighted side of the Fermi surface (Fig. 5.9).

Both side Light both sides.

Unoccupied side Light unoccupied side.

Occupied side Light the occupied side.

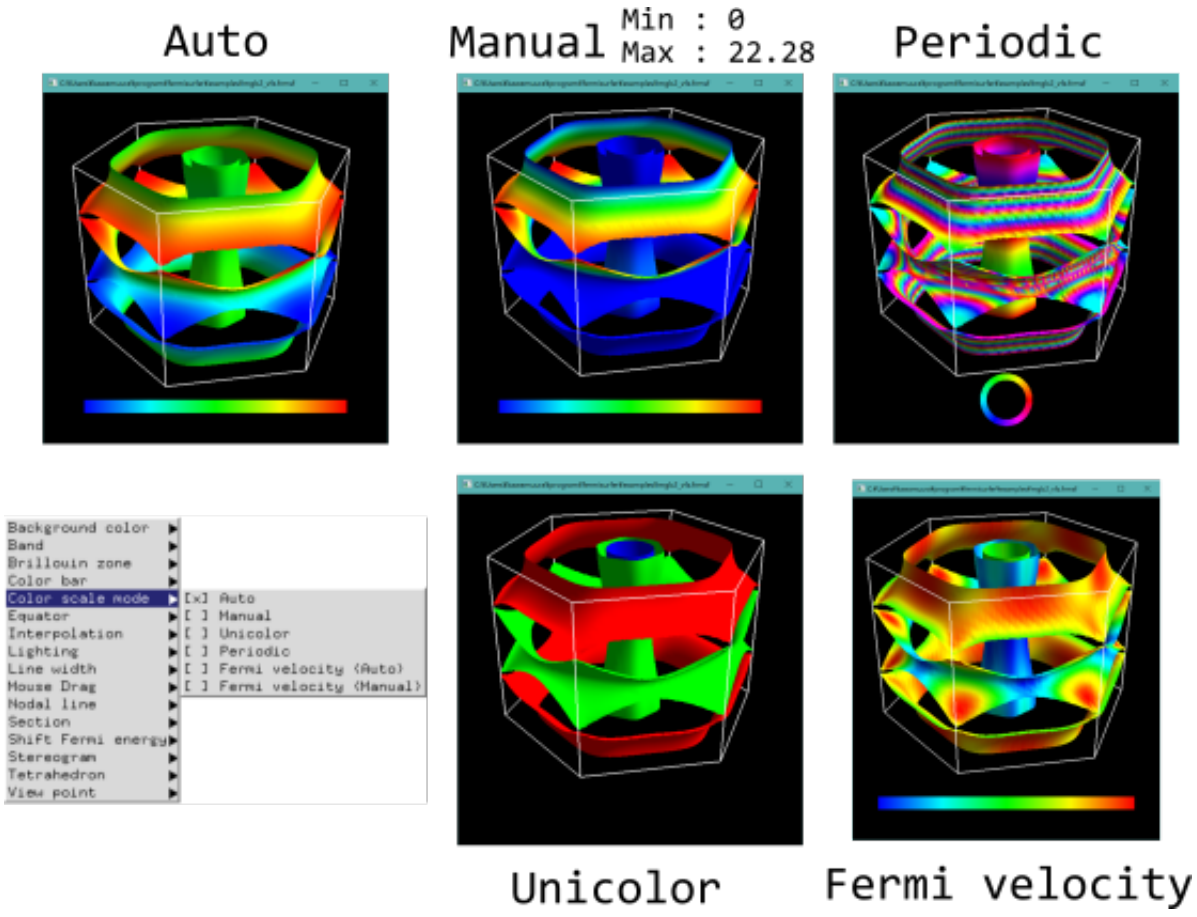


Fig. 5.6: “Color scale mode” menu.

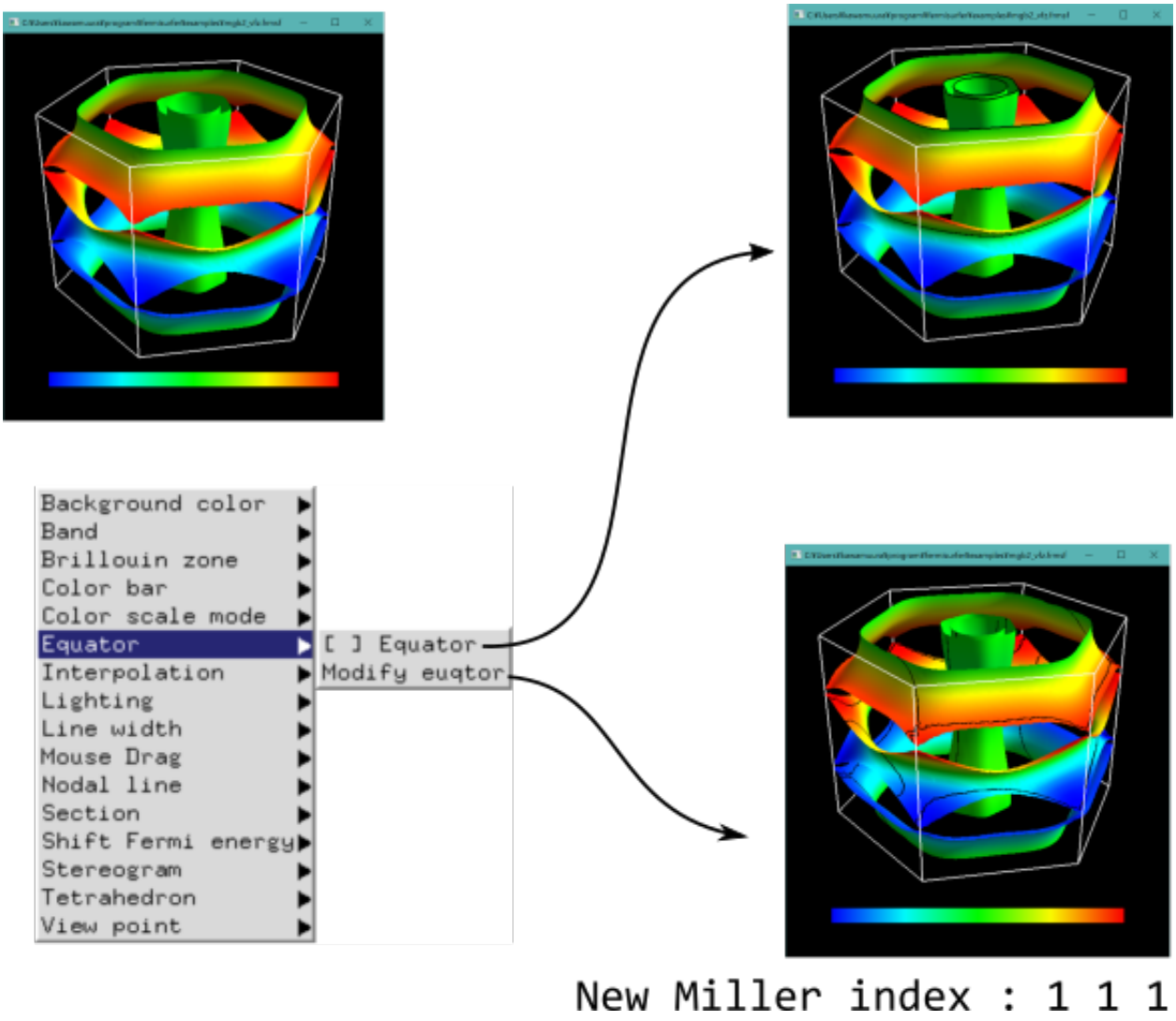


Fig. 5.7: Display the equator with the "Equator" menu.

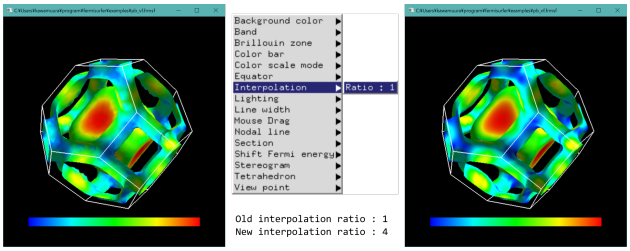


Fig. 5.8: Modify the number of interpolation points from 1 to 4 with "Interpolate" menu.

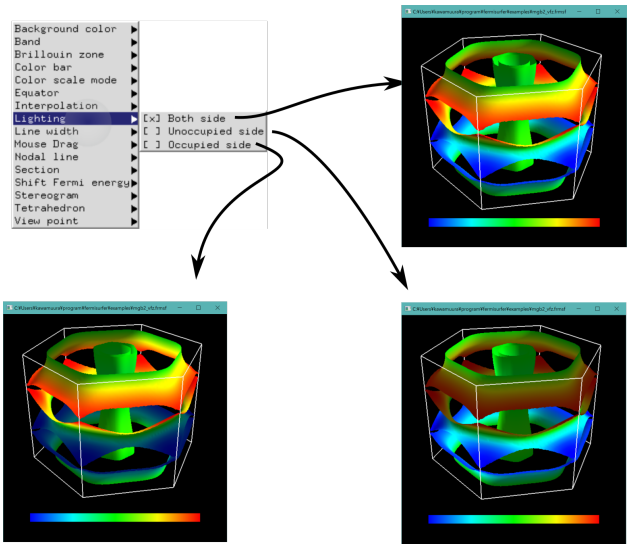


Fig. 5.9: Change the lighted side by using the "Lighting" menu.

Line width

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

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.

Nodal line

The line on which the matrix element becomes 0 (we call it nodal line) becomes enable/disable (Fig. 5.10).

Section

Display a 2D plot of the Fermi surface (line) on an arbitrary section of the Brillouin zone (Fig. 5.11).

Section Toggle the 2D plot of the Fermi surface (line).

Modify Section Specify the section. Please type the normal vector (**fractional coordinate**) at the prompt

```
New Miller index :
```

The section crosses the head of the normal vector.

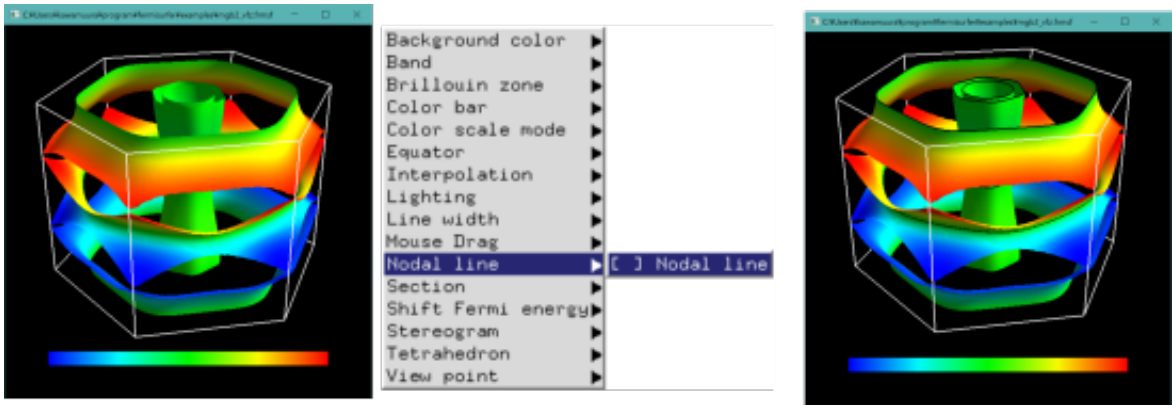


Fig. 5.10: Toggling the node line with "Nodal line" menu.

Modify Section (across Gamma) Specify the section. Please type the normal vector (**fractional coordinate**) at the prompt

```
New Miller index :
```

The section crosses Γ point.

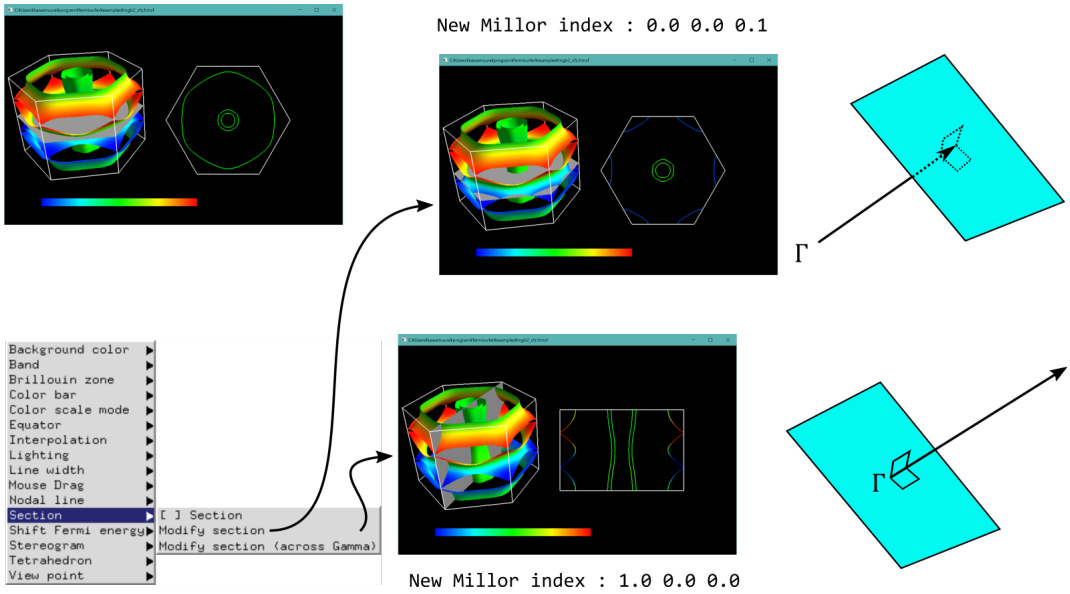


Fig. 5.11: Display 2D plot of the Fermi surface (line) with "Section" menu.

Shift Fermi energy

It shifts the Fermi energy (= 0 in default) to arbitrary value. When you use this menu, first, it displays minimum and maximum energy in the input file and the current Fermi energy;

```
Min   Max   E_F
-0.428153 0.497620 0.000000
Fermi energy shift :
```

Then, you should type the new Fermi energy; finally, the new Fermi surfaces are depicted (Fig. 5.12).

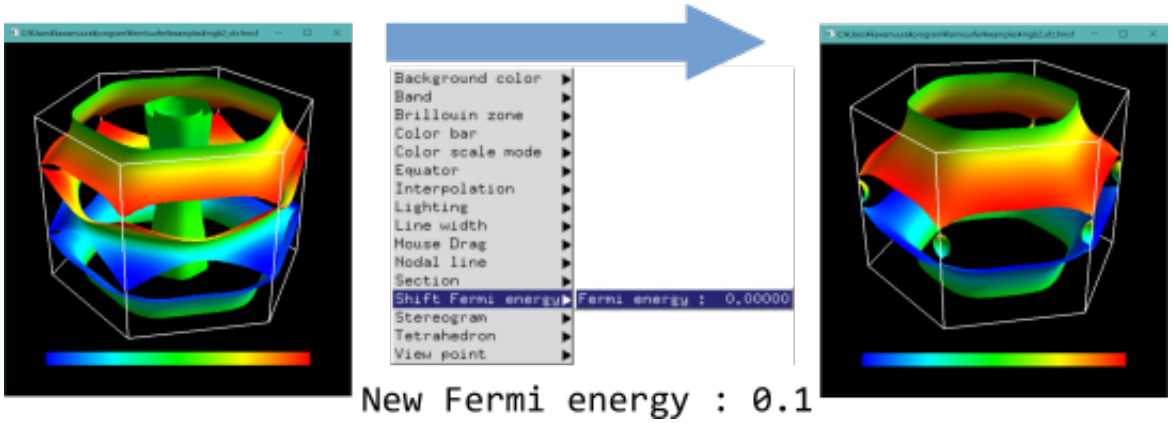


Fig. 5.12: The Fermi energy is set from 0 Ry to 0.1 Ry with "Shift Fermi energy" menu

Stereogram

The stereogram (parallel eyes and cross eyes) becomes enabled/disabled (Fig. 5.13).

None (Default)

Parallel Parallel-eyes stereogram

Cross Cross-eyes stereogram

Tetrahedron

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

View point

Changing the view point.

Scale Change the size of the figure.

Position Change the xy position of the figure.

Rotation Change angles at x-, y-, z- axis. Rotations are performed as z-y-x axis.

In each menu, first the current value is printed. then a prompt to input the new value appears (Fig. 5.17).

Saving images

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

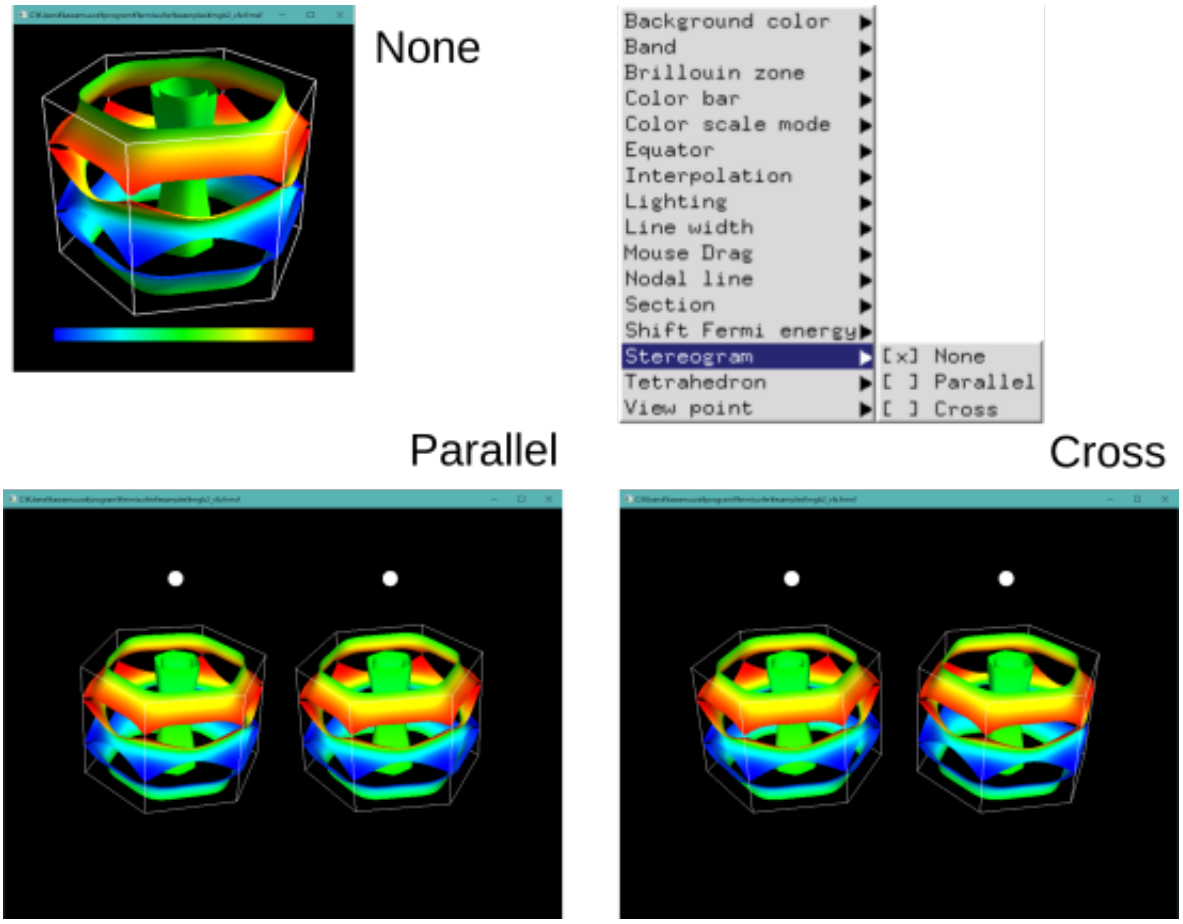
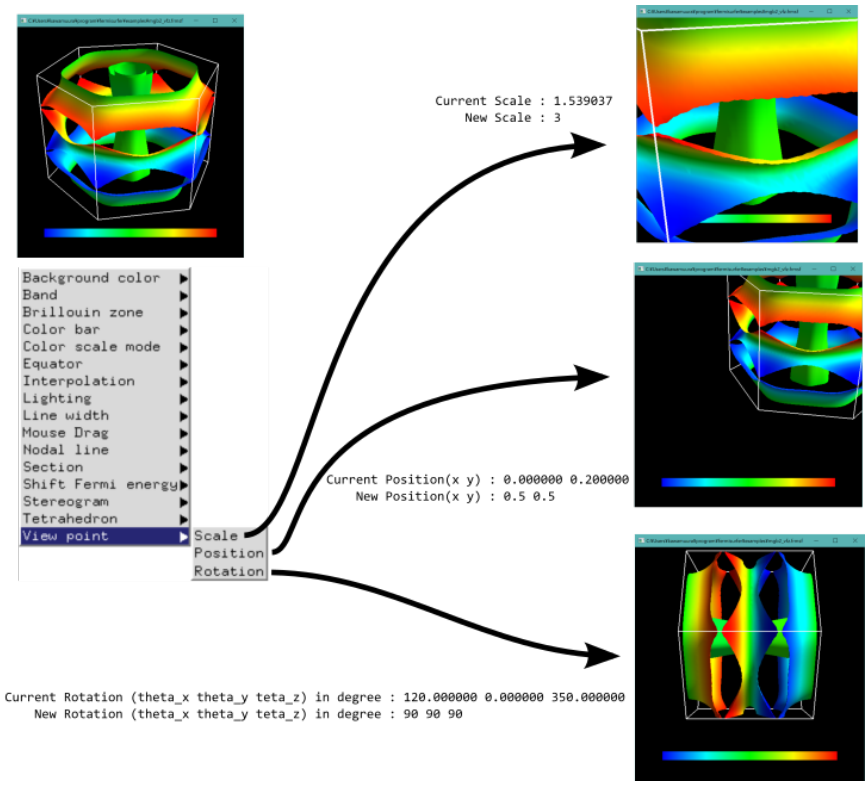


Fig. 5.13: The stereogram becomes enabled/disabled with “Stereogram” menu.


Background color
Band
Brillouin zone
Color bar
Color scale mode
Equator
Interpolation
Lighting
Line width
Mouse Drag
Nodal line
Section
Shift Fermi energy
Stereogram
Tetrahedron
View point
Scale
Position
Rotation
Current Scale : 1.539037
New Scale : 3
Current Position(x y) : 0.000000 0.200000
New Position(x y) : 0.5 0.5
Current Rotation (theta_x theta_y teta_z) in degree : 120.000000 0.000000 350.000000
New Rotation (theta_x theta_y teta_z) in degree : 90 90 90

CHAPTER 6

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`)

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
```

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.

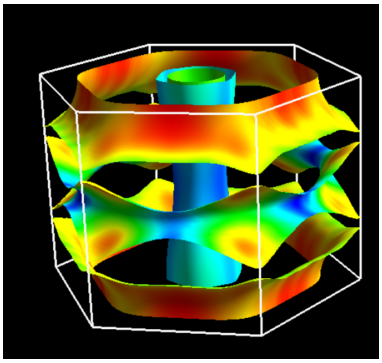
Compute and display Fermi velocity

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.

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`),

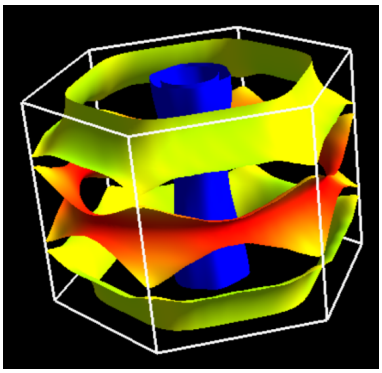
$$|\langle\phi_{B_1 2pz}|\psi_{nk}\rangle|^2 + |\langle\phi_{B_2 2pz}|\psi_{nk}\rangle|^2$$

is specified. We can display the Fermi surface as

```
$ fermisurfer proj.frmsf
```

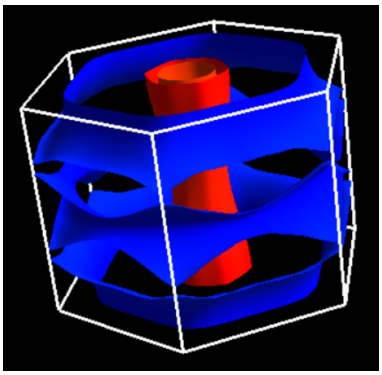
If we want to plot the projections onto 2px and 2py orbitals of all B atoms, the input file for `fermi_proj.x` becomes

```
&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 7

Gallery

Contributions of each Fermi surfaces to the Hall effect in IrO_2 (Fig. 7.1. Provided by Mr. Wataru Sano in Arita group, RIKEN)

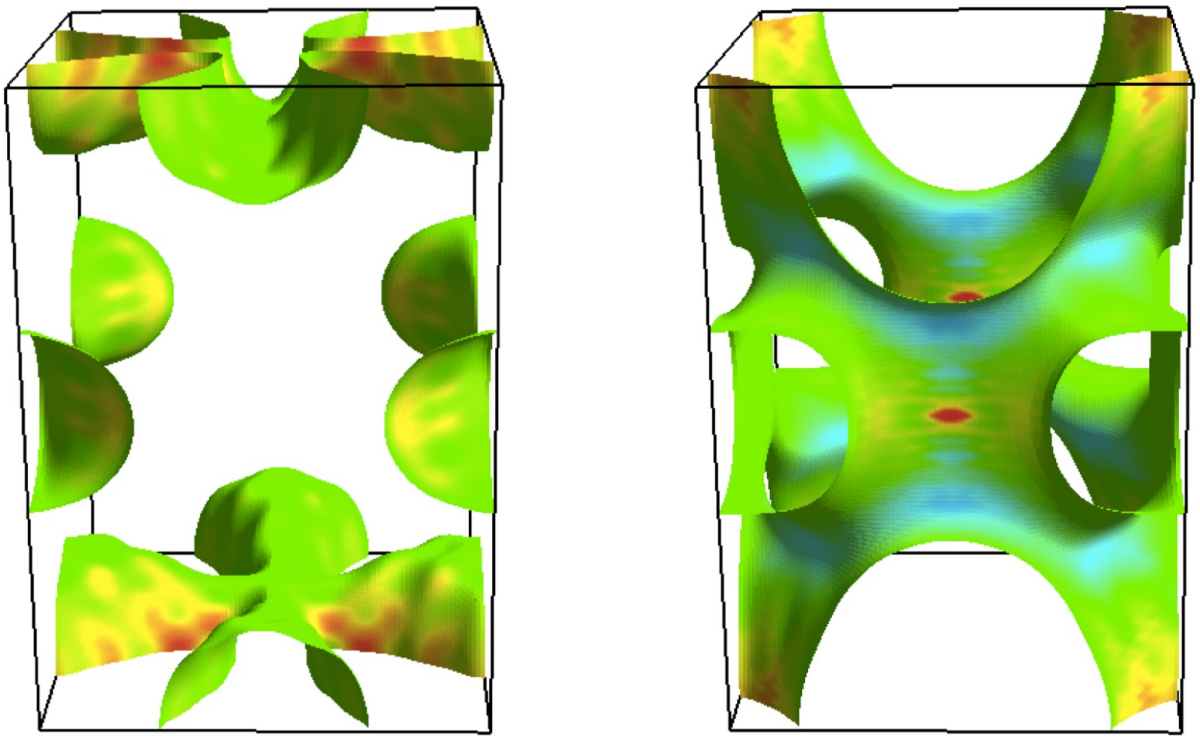


Fig. 7.1: Contributions of each Fermi surfaces to the Hall effect in IrO_2 .

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

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.

Build FermiSurfer without Autoconf

In this package, FermiSurfer is built with Autotools (Autoconf, Automake, Libtool). If you do not want to use Autotools for your distributed program with FermiSurfer's source, you can use the following simple Makefile (please care about TAB).

```
CC = gcc
CFLAGS = -g -O2 -lglut -lGLU -lGL -lm -fopenmp -DHAVE_GL_GLUT_H

OBJS= \
basic_math.o \
bz_lines.o \
calc_nodeline.o \
draw.o \
fermisurfer.o \
fermi_patch.o \
free_patch.o \
initialize.o \
kumo.o \
menu.o \
operation.o \
read_file.o \
section.o

HEADERS= \
basic_math.h \
bz_lines.h \
calc_nodeline.h \
draw.h \
fermi_patch.h \
free_patch.h \
initialize.h \
kumo.h \
```

```
menu.h \
operation.h \
read_file.h \
section.h \
variable.h

all:fermisurfer bxsf2frmsf

SUFFIXES: .o .c

.c.o:
    $(CC) $(CFLAGS) -c $<

fermisurfer:$(OBJS)
    $(CC) $(OBJS) $(CFLAGS) -o $@

bxsf2frmsf:bxsf2frmsf.o
    $(CC) $< $(CFLAGS) -o $@

clean:
    rm -rf *.o fermisurfer bxsf2frmsf

basic_math.o:$(HEADERS)
bz_lines.o:$(HEADERS)
calc_nodeline.o:$(HEADERS)
draw.o:$(HEADERS)
fermisurfer.o:$(HEADERS)
fermi_patch.o:$(HEADERS)
free_patch.o:$(HEADERS)
initialize.o:$(HEADERS)
kumo.o:$(HEADERS)
menu.o:$(HEADERS)
operation.o:$(HEADERS)
read_file.o:$(HEADERS)
section.o:$(HEADERS)
```

MIT License

OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
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