Fermisurfer Documentation

Release 2.1.0

kawamura

Nov 13, 2020
# CONTENTS

1 Introduction 1

2 Directories and important files 2

3 Install 3

3.1 Installation in Linux and macOS 3

3.2 Installation in Windows 3

4 Input file 4

4.1 input-file format 4

4.2 BXSF format 5

4.3 How to produce the input file in C and fortran programs 5

4.4 For the 2D color plot (See srvo3_t2g.frmsf in examples) 7

4.5 Omit the quantity for the color plot 8

5 Control FermiSurfer 9

5.1 Launch 9

5.1.1 For Linux, Unix, Mac 9

5.1.2 For Windows 9

5.2 Background color 10

5.3 Line width 10

5.4 Line color 10

5.5 Band 12

5.6 Brillouin zone (Update required) 12

5.7 Number of Brillouin zone 13

5.8 Color bar 13

5.9 Color scale mode (Update required) 13

5.10 Color sequence for plot 14

5.11 Equator (Update required) 15

5.12 Interpolation (Update required) 15

5.13 Which (or both) side of Fermi surface is illuminated 16

5.14 Mouse Drag 16

5.15 Nodal line 16

5.16 Section of the Brillouine zone (Update required) 18

5.17 Output section of the Brillouine zone 18

5.18 Shift Fermi energy (Update required) 18

5.19 Stereogram 18

5.20 Tetrahedron (Update required) 21

5.21 View point 21

5.22 Saving images 21
6 Batch mode

7 Tutorial with Quantum ESPRESSO
   7.1 Building PostProcess tool .................................................. 25
   7.2 SCF calculation ................................................................... 25
   7.3 Compute and display Fermi velocity ........................................... 26
   7.4 Compute and display projection onto the atomic orbital .......... 27

8 FermiSurfer for Android
   8.1 Install ................................................................. 29
   8.2 Input file ............................................................. 29
   8.3 Run ................................................................. 29

9 FermiSurfer on Web
   9.1 Control FermiSurfer on Web .................................................. 30
   9.2 How to input file ............................................................. 30
   9.3 Open online file ............................................................. 30

10 Acknowledgment ..................................................................... 32

11 Re-distribution of this program
   11.1 Contain Fermisurfer in your program ....................................... 33
   11.2 MIT License .............................................................. 33

12 Contact ................................................................................. 34
CHAPTER
ONE

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.
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
3.1 Installation in Linux and macOS

1. Install the required package

   • For Debian/Ubuntu
     
     ```bash
     $ sudo aptitude install libwxgtk3.0-dev
     ```

   • For Red Hat Enterprise Linux/CentOS
     
     ```bash
     $ sudo yum install wxGTK3-devel.x86_64
     ```

   • macOS (Homebrew)
     
     ```bash
     $ brew install wxmac
     ```

   • macOS (Mac Ports)
     
     ```bash
     $ port install wxWidgets-3.0
     ```

2. Install

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

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

3.2 Installation in Windows

Download the binary file.

Alternatively, we can build FermiSurfer by ourselves after we install wxWidgets library. We can use `fermisurfer.vcxproj` together with VisualStudio.
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  
0  
3  
1.000000 0.57735026 -0.000000  
0.000000 1.1547005 0.000000  
0.000000 0.000000 0.87206507  
2.91340202E-02  
2.93242838E-02  
2.98905596E-02  
3.08193434E-02  
:  
0.14393796  
0.12800488  
0.000000  
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:

\[
k_{i,j,k} = x_i b_1 + y_j b_2 + z_k b_3,
\]

(4.1)

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

```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))
```

(continues on next page)
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
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]);
      }
    }
  }
}

(continues on next page)
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
_write(fo,* real(eig(ik3,ik2,ik1,ibnd))
_write(fo,* real(eig(ik3,ik2,ik1,ibnd,2))
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
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
end do
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. 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)
- Operate by using the panel

Here, I will explain all menus.

**Note:** Some operations are not applied immediately, and after th “Update” button is pushed they are applied. Such operations are referred as “Update required”.

5.2 Background color

The background color is specified as RGB.

5.3 Line width

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

5.4 Line color

The line color is specified with RGB.
5.4. Line color
5.5 Band

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

5.6 Brillouin zone (Update required)

You choose Brillouin-zone type as follows (Fig. 2):

First Brillouin Zone  The region surrounded by Bragg’s planes the nearest to \( \Gamma \) point.

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

Fig. 2: You can change the type of the Brillouin zone with “Brillouin zone” menu.
5.7 Number of Brillouin zone

We can specify how many zones are displayed along each reciprocal lattice vector.

5.8 Color bar

The color bar becomes enable/disable (Fig. 3).

Fig. 3: Toggling the color bar with “Color bar On/Off” menu.

5.9 Color scale mode (Update required)

It turns color pattern on Fermi surfaces (Fig. 4).

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

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

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

**Fermi velocity (default for no input quantity)**: Compute the Fermi velocity $v_F = \nabla \varepsilon_k$ 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.

![Color scale mode menu](image.png)

**5.10 Color sequence for plot**

We can specify the sequence of color plot. “BGR” is Blue-Cyan-Green-Yellow-Red, “CMY” is Cyan-Blue-Magenta-Red-Yellow, “MCY” is Magenta-Blue-Cyan-Green-Yellow.
5.11 Equator (Update required)

We can draw the line where $v_F \cdot k = 0$ for a vector $k$ (equator or extremal orbit). See fig. 5. We can toggle equator with the checkbox “Equator” (this operation does not require the update, and modify the direction of the tangent vector $k$ by using the textbox at “Equator-v :” (fractional coordinate).

![Fig. 5: Display the equator with the “Equator” menu.](image1)

5.12 Interpolation (Update required)

Smooth the Fermi surface with the interpolation (Fig. 6). The time for the plot increases with the interpolation ratio.
5.13 Which (or both) side of Fermi surface is illuminated

We can choose the illuminated side of the Fermi surface (Fig. 7).

**Both** : Light both sides.

**Unoccupy** : Light unoccupied side.

**Occupy** : Light the occupied side.

5.14 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.

5.15 Nodal line

The line on which the matrix element becomes 0 (we call it nodal line) becomes enable/disable (Fig. 8).
Fig. 7: Change the lighted side by using the “Lighting” menu.
5.16 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. 9).

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 \( \Gamma \) point.

5.17 Output section of the Brillouine zone

Above section of the Brillouin zone and Fermi surfaces are outputted into files “fermi_line.dat” and “bz_line.dat” by pushing this button.

These files are plotted in gnuplot as follows:

```
plot "fermi_line.dat" w l, "bz_line.dat" w l
```

5.18 Shift Fermi energy (Update required)

It shifts the Fermi energy (= 0 in default) to arbitrary value (Fig. 10).

5.19 Stereogram

The stereogram (parallel eyes and cross eyes) becomes enabled/disabled (Fig. 11).

None (Default)
Parallel Parallel-eyes stereogram
Cross Cross-eyes stereogram
Fig. 9: Display 2D plot of the Fermi surface (line) with “Section” menu.

5.19. Stereogram
Fig. 10: The Fermi energy is set from 0 Ry to 0.1 Ry with “Shift Fermi energy” menu.

Fig. 11: The stereogram becomes enabled/disabled with “Stereogram” menu.
5.20 Tetrahedron (Update required)

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

5.21 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. 12).

5.22 Saving images

fermisurfer does not have any functions to save images to a file. Please use the screenshot on your PC.
Fig. 12: Modify the view point by using “View point” menu
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) :

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

**Note:** This function uses “import” command to get the screen-shot in ImageMagic. Therefore ImageMagic have to be installed to use this function.
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 $|v_F|$ (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:

```bash
$ 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.

```ini
&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
```

(continues on next page)
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/Mg.pbe-n-kjpaw_psl.0.3.0.upf)
- [http://theossrv1.epfl.ch/uploads/Main/NoBackup/B.pbe-n-kjpaw_psl.0.1.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 \(k\)-parallelization (npool) can be arbitrary numbers. We also can perform additional non-scf calculation 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 \(k\)-parallelization (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_12pz}|\psi_{nk}\rangle|^2 + |\langle \phi_{B_22pz}|\psi_{nk}\rangle|^2
```

is specified. We can display the Fermi surface as

```
7.4. Compute and display projection onto the atomic orbital
```
If we want to plot the projections onto 2px and 2py orbitals of all B atoms, the input file for `fermi_proj.x` becomes:

```plaintext
$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.
Functionally limited version of FermiSurfer can be used in Android.

8.1 Install

Download APK file from the following page
https://osdn.net/projects/fermisurfer/releases/p16366
then install it.

8.2 Input file

FermiSurfer for Android reads an input file whose name and path are fixed to /Download/frmsf. Therefor, we need to change the file name whenever we read each input file. For example, if we download the following file,
http://fermisurfer.osdn.jp/Nb1_Im-3m_151406/Nb4D.frmsf
we need to change the file name from Nb4D.frmsf to frmsf before we run FermiSurfer app.

8.3 Run

When FermiSurfer app runs, the above input file is read automatically, and Fermi surfaces are displayed. We can do the following operations:
  • Rotate objects by swiping.
  • Magnify/shrink objects by swiping up/down the edge of the display. Both right and left edges can be used.
We can use FermiSurfer on Web at the following URL: https://fermisurfer.osdn.jp/js/index.php

9.1 Control FermiSurfer on Web

Although we are planning to port all functions of the app version of FermiSurfer, part of them have not been supported yet.

9.2 How to input file

Fermi surfaces are displayed by choosing a local FRMSF file (BXSF is not supported yet) through the file-explore at left top of the window. Several time-lag may be expected.

9.3 Open online file

We can open an input file placed online by just clicking the link by a URL with an argument: https://fermisurfer.osdn.jp/js/index.php?frmsf=https://fermisurfer.osdn.jp/js/Pb.js

We add the URL where the input file locates after https://fermisurfer.osdn.jp/js/index.php?frmsf=.

The input file Pb.js is as follows:

```
frmsf="16 16 16 1 2 -0.67303315756516724 0.67303315756516724 ... ";
```

This is a javascript source in which a single-lined string generated by replacing new-line letters in FRMSF-formatted file with spaces is inputted into a variable frmsf. BXSF format has not been supported.

We can convert a FRMSF-formatted file into the above format as

```
sed -e '1i frmsf=""' -e '$a ";' ANY.frmsf | perl -pe 's/
/ /g' | sed -E -e 's/ +/ /g' -e 's/ /"/g' -e 's/"/"/g' > ANY.js
```

The following bash script file also do this conversion

```
https://fermisurfer.osdn.jp/js/frmsf2js.sh
```

Usage:
where ANY is an arbitrary string. Then a file ANY.js which should be uploaded onto a Web server is generated.
ACKNOWLEDGMENT

I thank Dr. Yusuke Konishi in ISSP; he performed a test in Mac OS X, and proposed Makefiles and a patch.
11.1 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.

11.2 MIT License

Copyright (c) 2014 Mitsuaki Kawamura

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS 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.
Please post bug 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