



Fermisurfer Documentation

リリース 1.10.0

kawamura

2017 年 10 月 10 日

Contents

1	はじめに	1
2	ディレクトリと主なファイル	2
3	インストール手順	3
3.1	Linux の場合	3
3.2	Mac OSX の場合	3
3.3	Windows の場合	4
4	Input file	5
4.1	input file の書式	5
4.2	BXSF 形式からの変換	6
4.2.1	Linux の場合	6
4.2.2	Windows の場合	7
4.3	C/fortran での入力ファイルの書き出し方	7
5	操作方法	9
5.1	起動	9
5.1.1	Linux の場合	9
5.1.2	Windows の場合	9
5.2	Background color	11
5.3	Band	11
5.4	Brillouin zone	11
5.5	Color bar	12
5.6	Color scale mode	12
5.7	Equator	13
5.8	Interpolation	13
5.9	Lighting	13
5.10	Line width	16
5.11	Mouse Drag	16
5.12	Nodal line	16
5.13	Section	16
5.14	Shift Fermi energy	17
5.15	Stereogram	18
5.16	Tetrahedron	18
5.17	View point	18
5.18	画像の保存方法	18
6	Quantum ESPRESSO を用いたチュートリアル	21
6.1	PostProcess ツールのビルド	21
6.2	SCF 計算	21
6.3	Fermi 速度の計算と描画	22
6.4	原子軌道射影の計算と描画	23

7	ギャラリー	25
8	謝辞	26
9	プログラムの再配布	27
9.1	自分のプログラムに FermiSurfer を含める	27
9.2	Autoconf を使わずに FermiSurfer をビルドする	27
9.3	MIT ライセンス	28
10	問い合わせ先	30

Chapter 1

はじめに

この文書では Fermi 面描画ソフト「Fermi Surfer」についての解説を行っています. Fermi Surfer は東京大学の河村光晶が 2012 年頃から開発を行っていたもので, 2014 年 11 月に公開されました. Fermi 面を描画しその上に各種物理量 (超伝導ギャップ関数や軌道キャラクターなど) をカラープロットするソフトウェアです.

Chapter 2

ディレクトリと主なファイル

- windows/ : Windows 用インストーラーのディレクトリ
- doc/ [マニュアルのディレクトリ]
 - doc/index.html : 目次ページ
- examples/ : サンプル入力ファイル
- src/ : ソースファイルのディレクトリ
- configure : ビルド設定スクリプト

Chapter 3

インストール手順

Linux の場合

1. 必要なパッケージをインストール (既にパッケージが入っている場合は何も起こりません.)

- Debian/Ubuntu 等

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

- Red Hat Enterprise Linux/CentOS 等

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

2. インストール

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

以上で実行可能ファイル `src/fermisurfer` および `src/bxsf2frmsf` が作られ, `/usr/local/bin/` にコピーされます.

Mac OSX の場合

1. Xcode をあらかじめインストールしておく

2. インストール

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

以上で実行可能ファイル `src/fermisurfer` および `src/bxsf2frmsf` が作られ, `/usr/local/bin/` にコピーされます.

Windows の場合

実行可能ファイル `bin/fermisurfer.exe` を直接実行します. `freeglut` ライブラリをダウンロードして自分でコンパイルすることも可能です.

Chapter 4

Input file

input file の書式

用意するデータは,

- Brillouin 領域分割数 (3 方向)
- 逆格子ベクトル
- バンド本数
- 軌道固有値 (以下エネルギーと呼びます) の各バンド, k グリッド点での値
- カラープロットしたい物理量 (以下物理量と呼びます) の各バンド, k グリッド点での値

です.

上記データを次のとおりの書式で並べます (サンプルファイル `mgb2_vfz.frmsf` の中身).

```
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. k グリッド数
2. k グリッドの指定方法 (0, 1, 2 のいずれか)

k グリッドを次のように表します.

$$\mathbf{k}_{i,j,k} = x_i \mathbf{b}_1 + y_j \mathbf{b}_2 + z_k \mathbf{b}_3 \quad (4.1)$$

ただし: i, j, k は 1 から各逆格子ベクトル方向の分割数 N_1, N_2, N_3 (上で読み取ったもの) とする

x_i, y_j, z_k は次の 3 つのとり方が可能です.

- 0 の場合 (Monkhorst-Pack グリッド): $x_i = \frac{2i-1-N_1}{2N_1}$
- 1 の場合: $x_i = \frac{i-1}{N_1}$
- 2 の場合: $x_i = \frac{2i-1}{2N_1}$

3. バンド本数
4. 逆格子ベクトル 1 (任意単位)
5. 逆格子ベクトル 2
6. 逆格子ベクトル 3
7. エネルギー (並び順は *Cfortran* での入力ファイルの書き出し方 参照)

fermisurfer はデフォルトでは Fermi エネルギーを 0.0 としています. ただし, 後述の Shift Fermi Energy メニューを用いて Fermi エネルギーを 0.0 以外の値に変更することも可能です.

8. 物理量 (並び順は *Cfortran* での入力ファイルの書き出し方 参照)

特にカラープロットをしたい量が無く, Fermi 面の描画のみを行いたい場合には, 後述の Color scale mode メニューから Unicolor を選択してください. その際, この”物理量”は利用されることはありませんが, ファイルの読み込みを完了させるために何らかの数字で埋めてください (すべて 1.0 とするなど).

BXSF 形式からの変換

付属のユーティリティ `bxs2frmsf` を使うと XCrysDen 用の入力ファイルから `fermisurfer` で読み取り可能な入力ファイルが作られます. また, そのときに Fermi 面上にカラープロットされる Fermi 速度が, エネルギーバンドから計算されます.

使い方は次のとおりです. `examples/` ディレクトリにある `pb.bxs2frmsf` ファイルを例として使います.

Linux の場合

作成した実行可能ファイル `bxs2frmsf` にパスが通っている状態で

```
$ bxs2frmsf pb.bxs2frmsf
```

とコマンド, スペース, 入力ファイル名とタイプします. その後, 次のファイルが生成されます.

- `pb_vf.frmsf`: Fermi 速度の絶対値
- `pb_vfx.frmsf`: Fermi 速度の x 成分
- `pb_vfy.frmsf`: Fermi 速度の y 成分
- `pb_vfz.frmsf`: Fermi 速度の z 成分
- `pb_vfa1.frmsf`: Fermi 速度の, 格子ベクトル \mathbf{a}_1 に平行な成分

- `pb_vfa2.frmsf`: Fermi 速度の, 格子ベクトル a_2 に平行な成分
- `pb_vfa3.frmsf`: Fermi 速度の, 格子ベクトル a_3 に平行な成分

Windows の場合

入力ファイル(この場合は `pb.bxsf`) を右クリックし, メニューから「プログラムから開く」を選択し, 実行ファイルを `bxsf2frmsf.exe` に設定してください.

C/fortran での入力ファイルの書き出し方

fortran

```
real(4) :: bvec1(3), bvec2(3), bvec3(3) !逆格子ベクトル
INTEGER :: nk1, nk2, nk3 !各逆格子ベクトルの方向の分割数
integer :: ishift !グリットをシフトさせるか (=1) 否か (=0)
integer :: nbnd !バンド数
real(4) :: eig(nk3,nk2,nk1,nbnd) !エネルギー
real(4) :: x(nk3,nk2,nk1,nbnd) !物理量

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

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
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 言語

```
float bvec1[3], bvec2[3], bvec3[3]; /*逆格子ベクトル*/
int nk1, nk2, nk3; /*各逆格子ベクトルの方向の分割数*/
int ishift; /*グリットをシフトさせるか (=1) 否か (=0)*/
int nbnd; /*バンド数*/
float eig[nbnd][nk1][nk2][nk3]; /*エネルギー*/
```

```
float x[nbnd][nk1][nk2][nk3]; /*物理量*/

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

操作方法

起動

Linux の場合

作成した実行可能ファイル `fermisurfer` にパスが通っている状態で

```
$ fermisurfer mgb2_vfz.frmsf
```

とコマンド、スペース、入力ファイル名とタイプします。(サンプルファイルの中身は MgB_2 の Fermi 速度の z 方向成分です。)

Windows の場合

入力ファイル(この場合は `mgb2_vfz.frmsf`) を右クリックし、メニューから「プログラムから開く」を選択し、実行ファイルを `fermisurfer.exe` に設定してください。

このあとは Linux, Windows 共通です。 `fermisurfer` が起動すると、まずファイルから読み取った情報が出力されます。

```
#####
##                                     ##
##  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 patches
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. それぞれのバンドにおけるエネルギーと物理量の最小値・最大値.
2. それぞれのバンドにおけるパッチ (Fermi 面を構成する平面) の数.
3. 物理量の Fermi 面における最大値と最小値. この数字がカラーバーの最大・最小に対応します. 下の例では一番青いところが -22.283419, 一番赤いところが 22.283419 となります. (1) で表示されているのは Brillouin 領域全体のものです.
4. それぞれのバンドにおける node line (後述) の本数.

次に操作方法が出力され, Fermi 面が描画されます (図 5.1).

- マウスのドラッグによる回転が出来ます.
- マウスのホイールを使つての拡大・縮小が出来ます.
- ウィンドウの大きさを変えることもできます.
- カーソルキーを使ってウィンドウ内で上下左右に図を動かせます.
- ウィンドウ内でマウスの右クリックをするとメニューが表示されます.

次から右クリックで表示されるメニューを説明します.

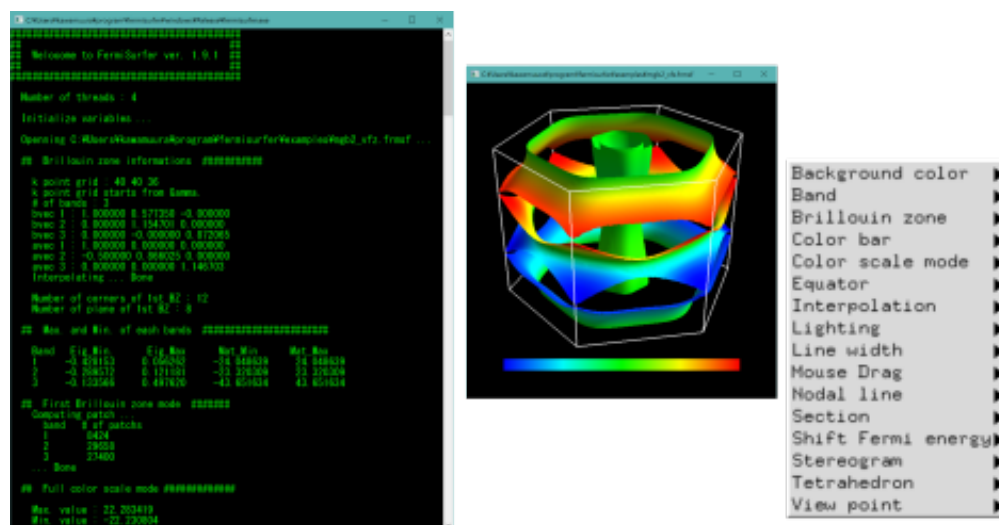


図 5.1: Fermisurfer を起動した直後の画面.

Background color

背景色を黒または白に切り替えます. Brillouin Zone の枠線も白/黒と切り替わります (図 5.2).

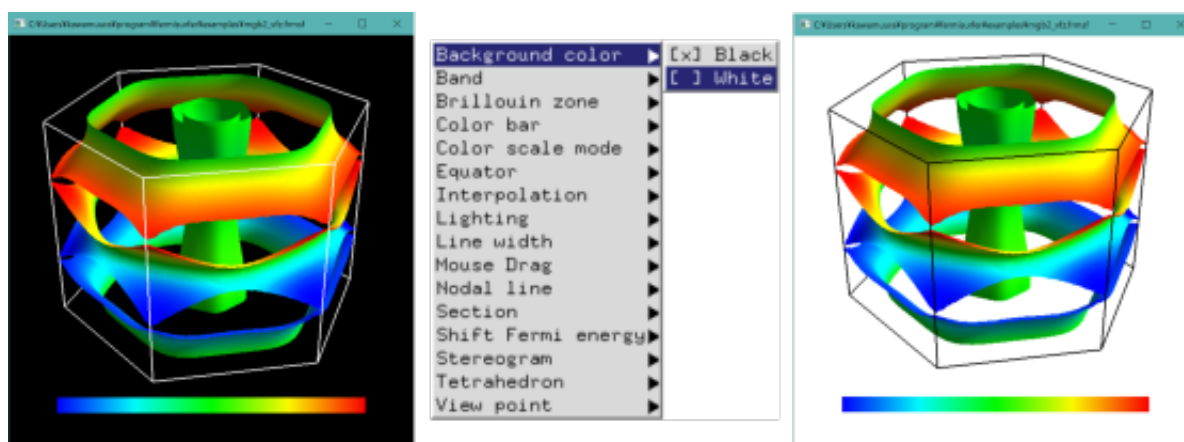


図 5.2: “Background color” メニューで背景色を白/黒に切り替える。

Band

バンド毎の表示 on/off を切り替えます (図 5.3).

Brillouin zone

描画範囲を First Brillouin Zone/Primitive Brillouin Zone と切り替える事が出来ます (図 5.4).

First Brillouin Zone Γ 点から一番近い Bragg 面で囲まれた領域

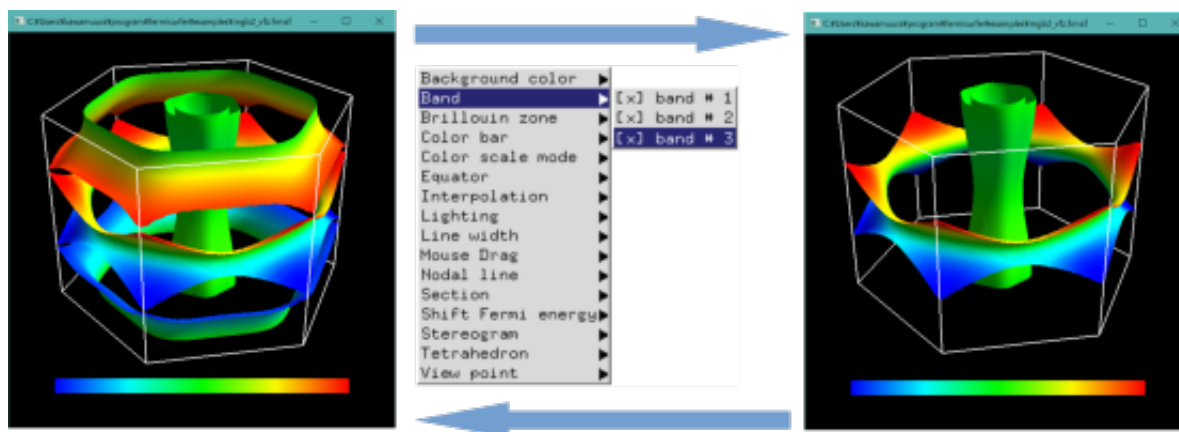


図 5.3: “Band” メニューで 3 番目のバンドの表示/非表示を切り替える。

Primitive Brillouine Zone 逆格子ベクトルを辺とする平行 6 面体領域

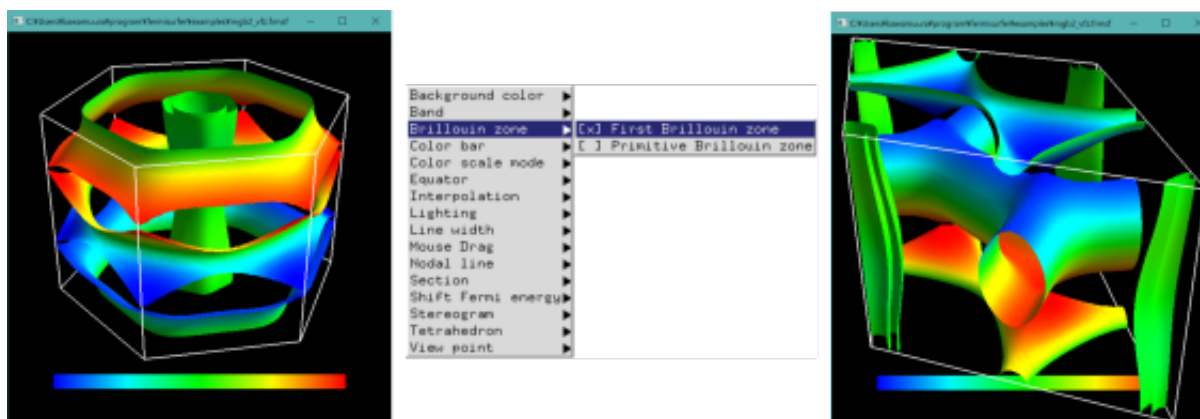


図 5.4: “Brillouin zone” メニューで Brillouin 領域のとり方を変更する。

Color bar

カラーバーの表示/非表示を切り替えます (図 5.5).

Color scale mode

Fermi 面の色表示のさせ方を変更します (図 5.6).

Auto(デフォルト) カラースケールの範囲を Fermi 面上での物理量の最小値から最大値までとします。

Manual カラースケールの範囲を標準入力から設定します。

Unicolor 物理量に関係なく、各バンド毎に単色で Fermi 面を塗ります。

Periodic 周期的な量のプロットに用います。物理量が $0 \rightarrow \pi/3 \rightarrow 2\pi/3 \rightarrow \pi \rightarrow 4\pi/3 \rightarrow 5\pi/3 \rightarrow 2\pi$ と変化するに連れて色が赤 \rightarrow 黄 \rightarrow 緑 \rightarrow シアン \rightarrow 青 \rightarrow マゼンタ \rightarrow 赤と変わります。

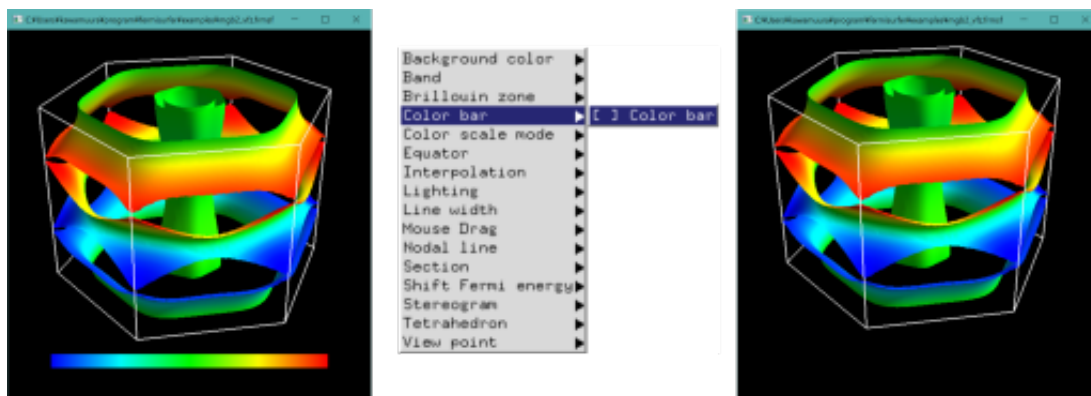


図 5.5: “Color bar On/Off” メニューでカラーバーの表示/非表示を切り替える.

Fermi velocity (Auto) エネルギーの差分から Fermi 速度 $v_F = \nabla_k \varepsilon_k$ を計算し, その絶対値をカラープロットする. カラースケールの範囲は Fermi 面上での $|v_F|$ の最小値から最大値までとする.

Fermi velocity (Manual) エネルギーの差分から Fermi 速度 $v_F = \nabla_k \varepsilon_k$ を計算し, その絶対値をカラープロットする. カラースケールの範囲は標準入力から設定する.

Gray scale (Manual), Gray scale (Auto) 黒色の濃淡でプロットする.

Equator

ある k に対して, $v_F \cdot k = 0$ となる線を表示します (図 5.7).

Equator Equator の表示・非表示を切り替えます.

Modify equator k を指定します. コンソールの

```
New Miller index :
```

の後に k ベクトル (フラクショナル座標) を入力してください.

Interpolation

補間により図の曲面を滑らかにします (図 5.8). コンソールの

```
New interpolation ratio :
```

の後に分点数を入力してください. ただし分点数を増やすと描画にかかる時間も増えます.

Lighting

光を当てる面を変更します (図 5.9).

Both side Fermi 面の表裏両面に光を当てます.

Unoccupied side 非占有領域側のみに光を当てます.

Occupied side 占有領域側のみに光を当てます.

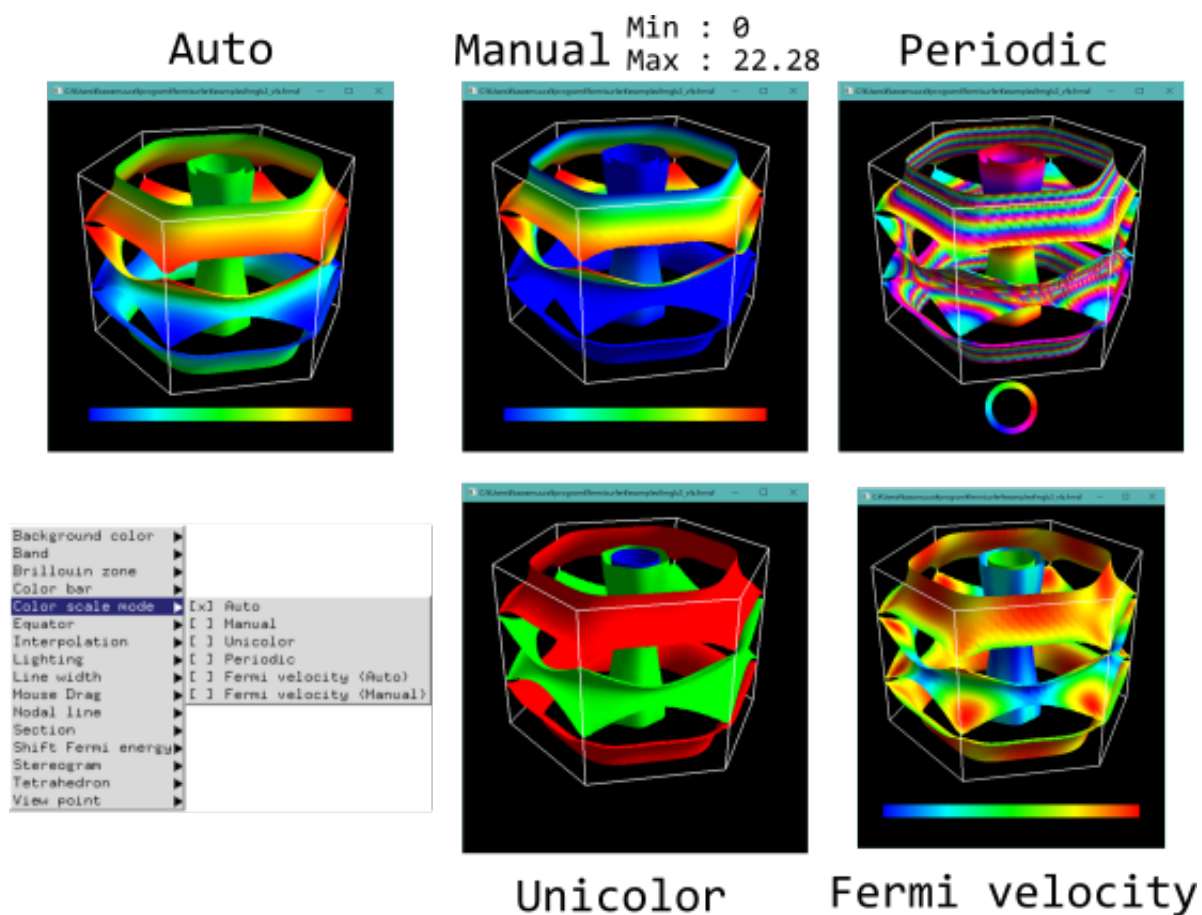


図 5.6: “Color scale mode” メニュー.

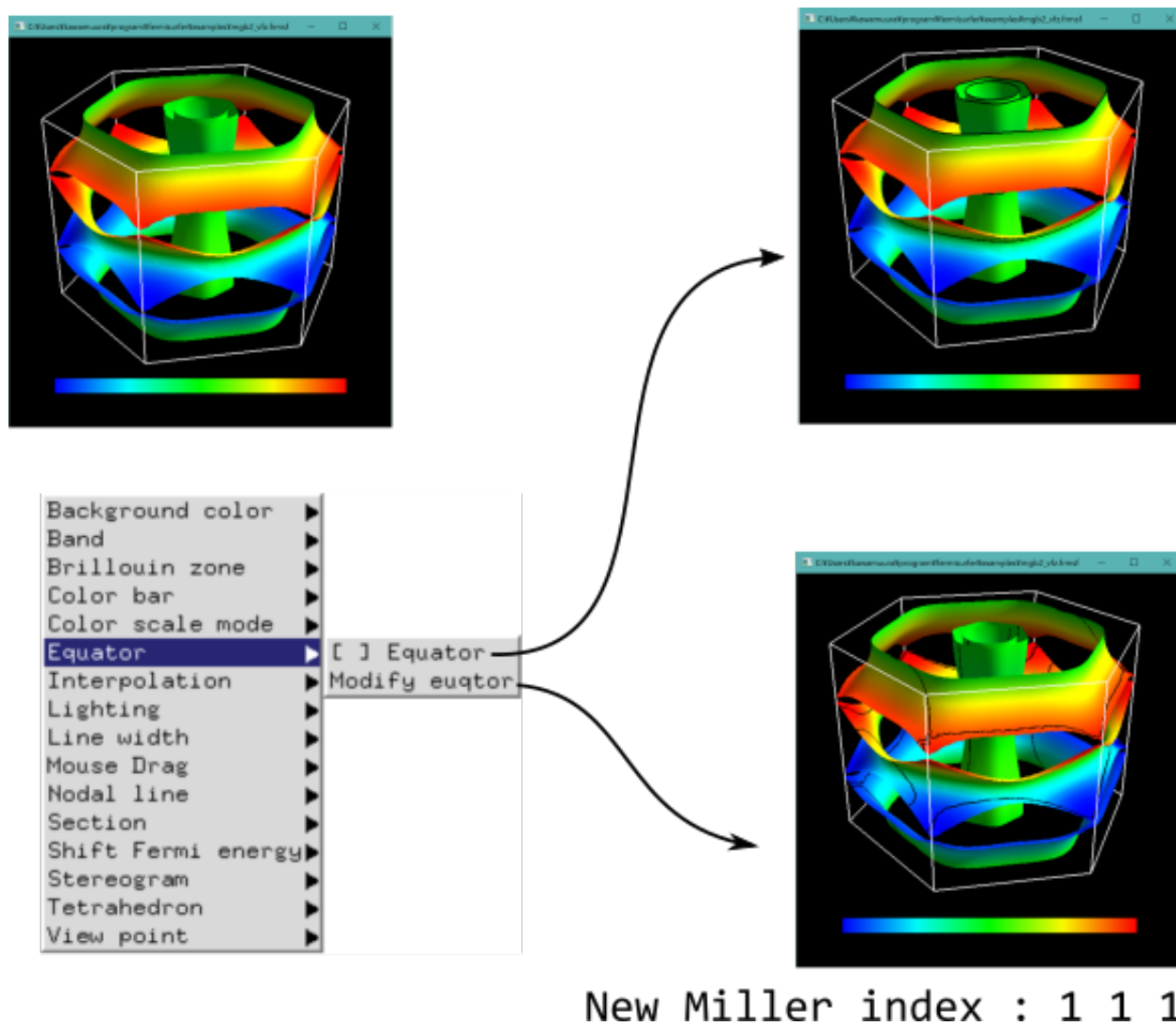


図 5.7: “Equator” メニューで Fermi 面の赤道 (Equator) を表示する。

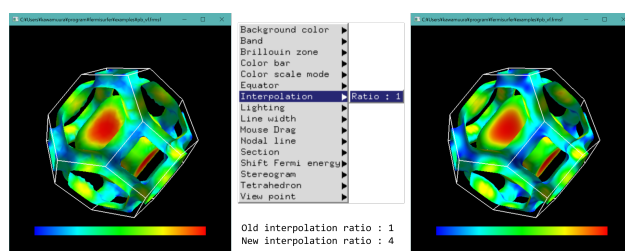


図 5.8: “Interpolate” メニューで 分点数を 1 から 4 に変える。

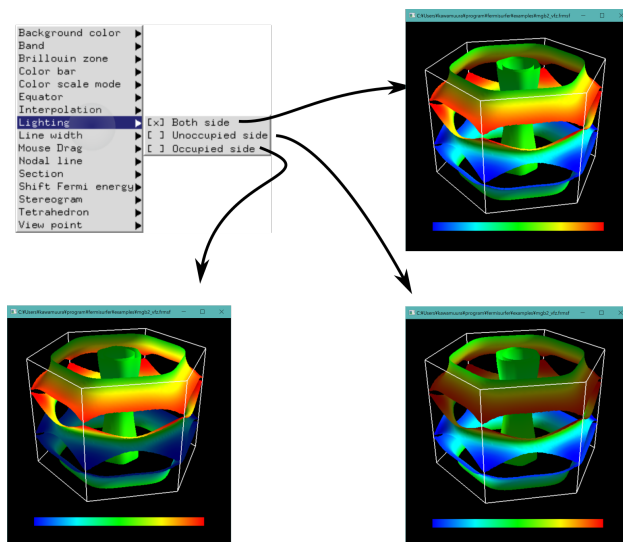


図 5.9: “Lighting” メニューで光を当てる Fermi 面を変更する。

Line width

ブリルアンゾーンの境界やノーダルライン等の線幅を変更します。

Mouse Drag

マウスの左ボタンドラッグを行った時の動作を変更します。

Rotate(デフォルト) ドラッグをした方向に図形を回転させます。

Scale 上方にドラッグすると図形を拡大, 下方にドラッグすると図形を縮小します。

Translate ドラッグした方向に図形を動かします。

Nodal line

物理量が 0 となるところに引く線 (ノーダルライン) の On/Off を切り替えます (図 5.10).

Section

Brillouin 領域を任意の断面で切り取り, 2 次元の Fermi 面 (線) を描画します (図 5.11).

Section 断面の表示・非表示を切り替えます。

Modify Section 断面を指定します. コンソールの

```
New Miller index :
```

の後に法線ベクトル (フラクショナル座標) を入力してください. 断面は法線ベクトルの先端を通ります。

Modify Section (across Gamma) 断面を指定します. コンソールの

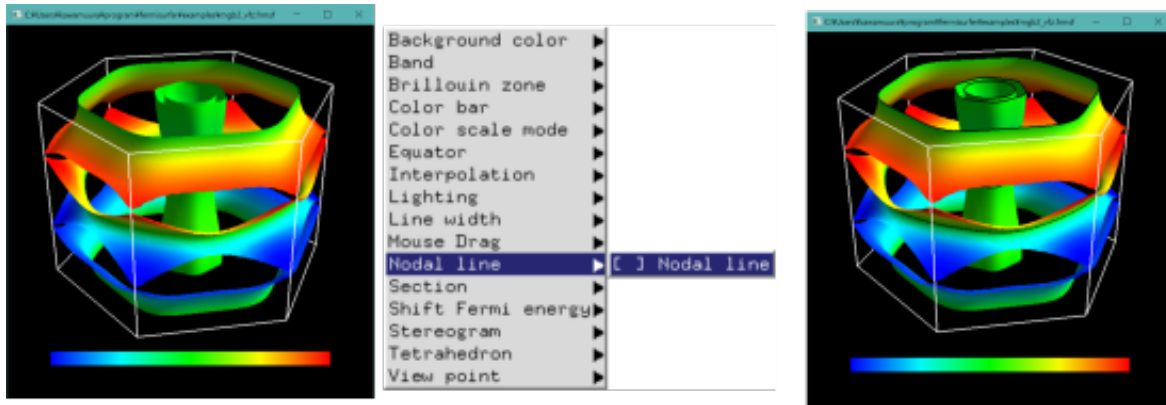


図 5.10: “Nodal line” メニューで nodal line の表示/非表示を切り替える。

New Miller index :

の後に法線ベクトル (フラクショナル座標) を入力してください。断面は Γ 点を通ります。

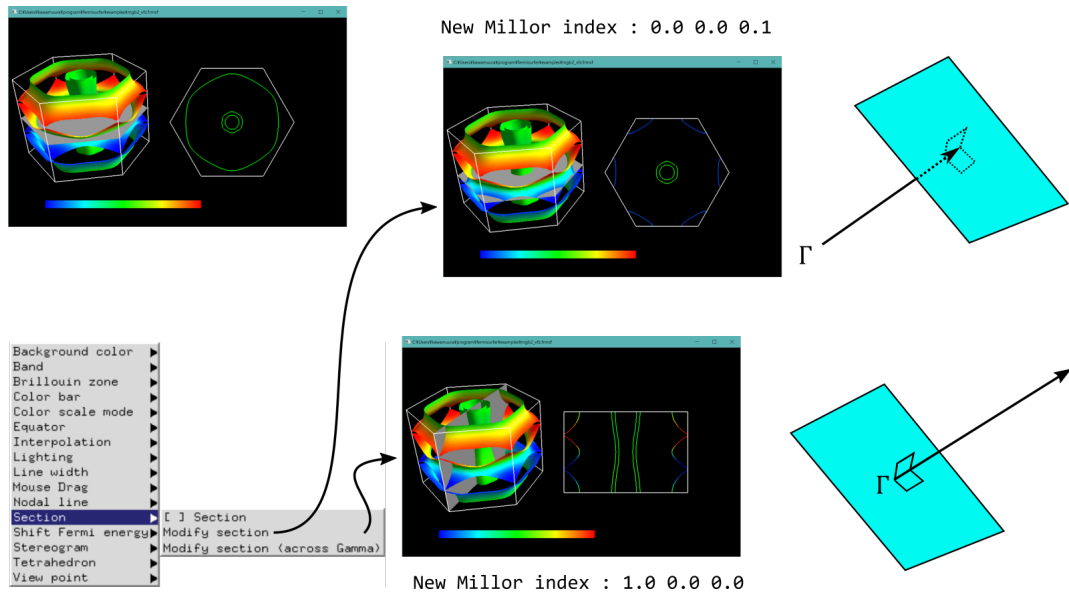


図 5.11: “Section” メニューで Fermi 面の断面を表示する。

Shift Fermi energy

Fermi エネルギー (デフォルトでは 0) を任意の値にずらしします。このメニューを選択すると次のようにインプット中の最小のエネルギー, 最大のエネルギー, 現在の Fermi エネルギーが標準出力として表示されます。

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

次に新しい Fermi エネルギーを入力すると, Fermi 面が再描画されます (図 5.12)。

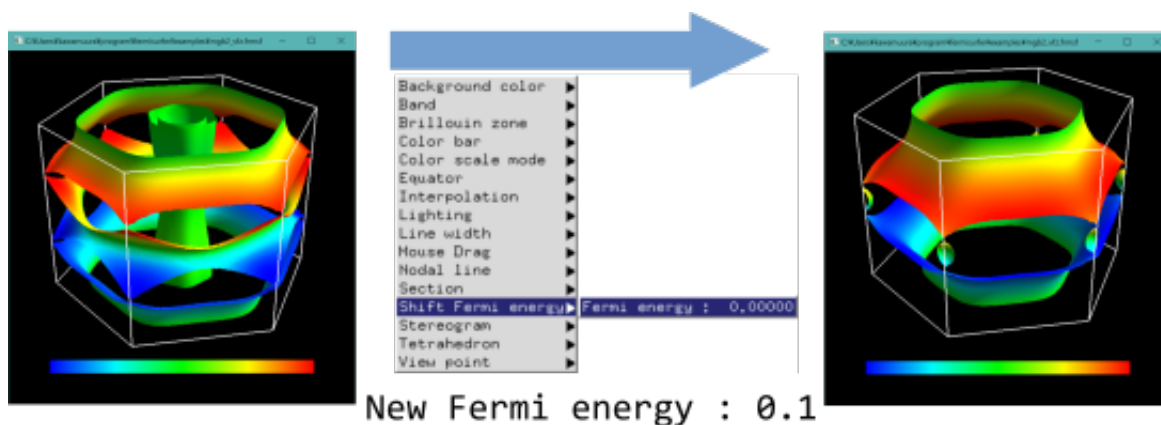


図 5.12: “Shift Fermi energy” メニューで Fermi エネルギーを 0.0 Ry から 0.1 Ry に変える。

Stereogram

裸眼立体視用の図の表示/非表示を切り替えます (図 5.13).

None (デフォルト) 立体視を無効にします。

Parallel 平行法用の図を表示します。

Cross 交差法用の図を表示します。

Tetrahedron

四面体の切り方を変えます (デフォルトは `tetra # 1`). 図が綺麗になる可能性があります, 多くの場合は逆に図がギザギザして汚くなるようです。

View point

視点を変更します。

Scale 図形のサイズを指定します。

Position 図形の上下位置を指定します。

Rotation x, y, z 軸周りの回転角を指定します. 回転操作は z 軸- y 軸- x 軸の順で行われます

それぞれのメニューを選択すると, はじめに現在の値が表示され, その後変更後の値を入れるプロンプトが現れます (図 5.14).

画像の保存方法

`fermisurfer` には画像をファイル出力する機能はありません. お使いの PC にあった方法でスクリーンショットを取得して (Printscreen キーを押すなど) ペイントブラシや `gimp` で編集して画像を作成してください。

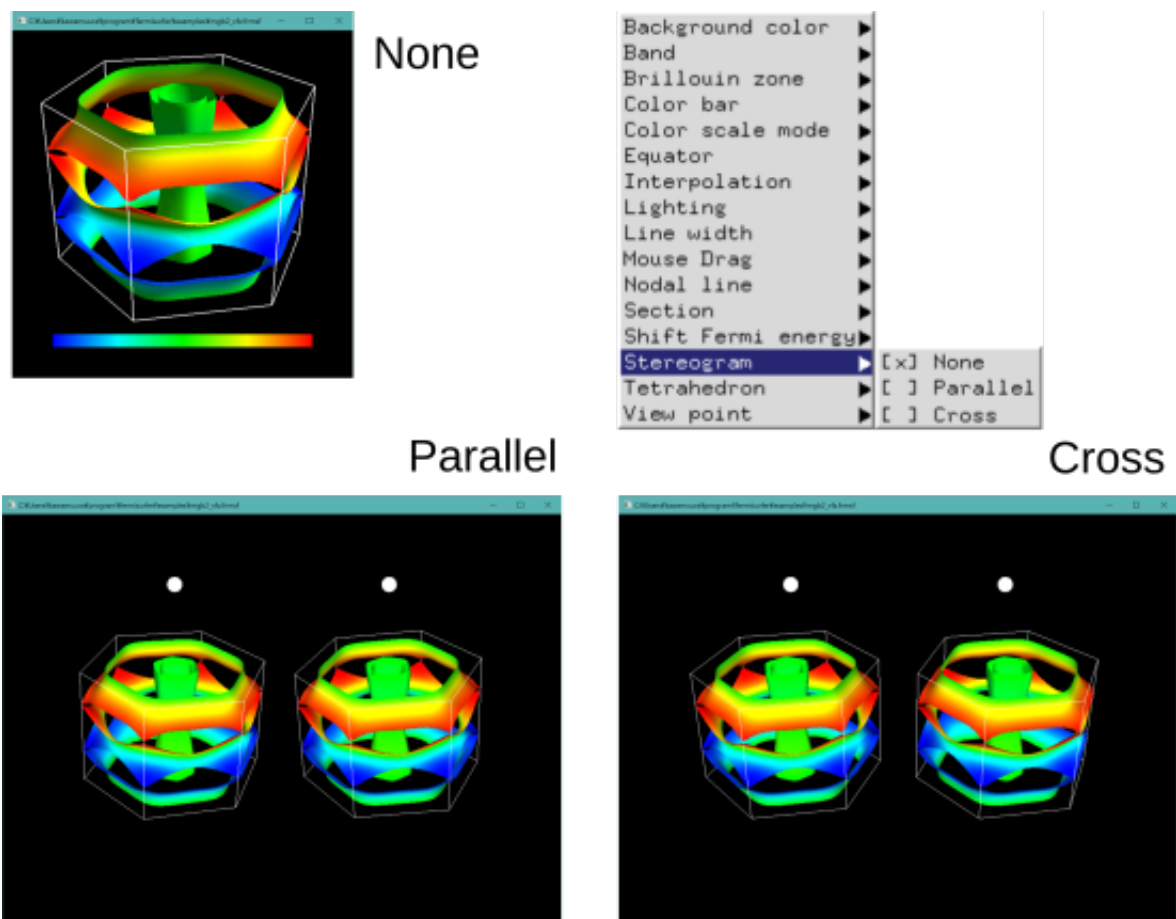


図 5.13: “Stereogram” メニューで立体視用画像を表示する.

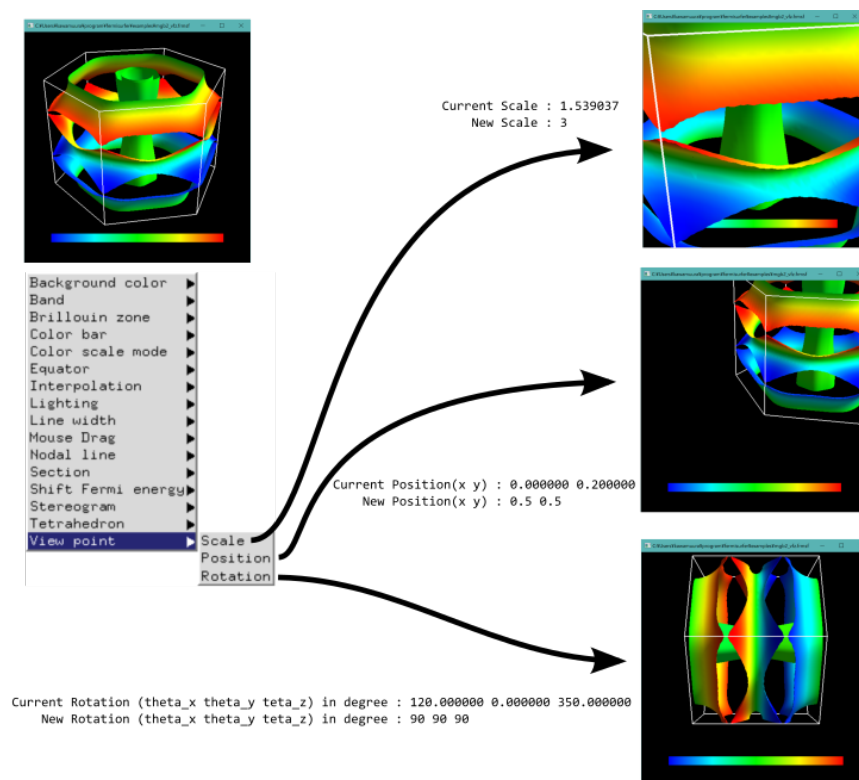


図 5.14: “View point” メニューで 視点を変更する。

Chapter 6

Quantum ESPRESSO を用いたチュートリアル

Quantum ESPRESSO version 6.2 から, FermiSurfer で読み込む事が出来るデータ形式でのファイルを出力出来るようになった. FermiSurfer でプロット出来るのは次の量である.

- Fermi 速度の絶対値 $|\mathbf{v}_F|$ (fermi_velocity.x)
- 各原子軌道への射影 $|\langle \phi_{nlm} | \psi_{nk} \rangle|^2$ (fermi_proj.x)

PostProcess ツールのビルド

上記の量を FermiSurfer で描画するためには, 次のようにして QuantumESPRESSO 内の PostProcess ツール (バンド図や電子密度をプロットするためのツール群) をビルドする必要がある.

```
$ make pp
```

SCF 計算

ここからチュートリアルに入る. まず初めに pw.x プログラムを用いて電子状態計算をおこなう. MgB_2 を題材として扱う. 入力ファイルは次の通りである.

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

```

この計算で使われている擬ポテンシャルファイルは [PS Library](http://theosrv1.epfl.ch/uploads/Main/NoBackup/Mg.pbe-n-kjpaw_psl.0.3.0.upf) に含まれているものであり, 以下のアドレスからダウンロードできる.

- http://theosrv1.epfl.ch/uploads/Main/NoBackup/Mg.pbe-n-kjpaw_psl.0.3.0.upf
- http://theosrv1.epfl.ch/uploads/Main/NoBackup/B.pbe-n-kjpaw_psl.0.1.upf

入力ファイルと擬ポテンシャルファイルを同じディレクトリに置き, そのディレクトリで `pw.x` を実行する.

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

プロセス数, k 点並列数 (`npool`) は自由. お好みで, 異なる k グリッドで Non-scf 計算を行っても良い.

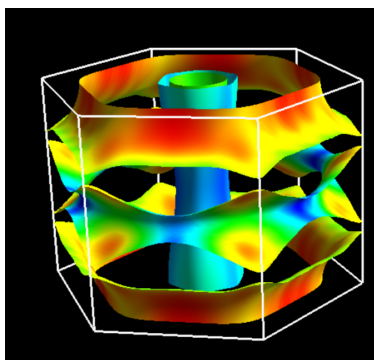
Fermi 速度の計算と描画

`fermi_velocity.x` プログラムを実行する. 入力ファイルは `pw.x` のものと同一である.

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

このとき, k 点並列数 (`npool`) は 1(もしくは指定しない) でなければならない. これにより, Fermi 速度のファイル `vfermi.frmsf` が作られるので, それを FermiSurfer で読み込む.

```
$ fermisurfer vfermi.frmsf
```



なお, コリニアスピン計算では各スピンについてそれぞれ `vfermi1.frmsf`, `vfermi2.frmsf` の 2 つのファイルが出力される.

原子軌道射影の計算と描画

原子軌道射影の描画では, まず部分状態密度等を計算するプログラム `projwfc.x` を実行する. 入力ファイルは次の通りである.

`proj.in`

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

PROJWFC ネームリストの終わり (/) 以降は `projwfc.x` では使われず, 後で `fermi_proj.x` を実行するときのみ使われる. `projwfc.x` を実行するときのプロセス数, k 点並列数 (`npool`) は直前の `pw.x` の実行時と同じ値にしなければならない.

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

ただし, `wf_collect=.true.` としていたときは除く.

`projwfc.x` の標準出力のはじめの方に次のような箇所がある.

```
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)
```

これは各原子軌道の番号 (`state #`) とその中身 (詳しくは QE に付属の `INPUT_PROJWFC.html` 等を参照) を表している. この後で Fermi 面上で描画する原子軌道射影を選ぶ時にはこの番号を用いる. 具体的な例を示す. 上記の `proj.in` を用いて

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

のように実行して FermiSurfer 用ファイル `proj.frmsf` を得るのだが, このとき `proj.in` の / 以降

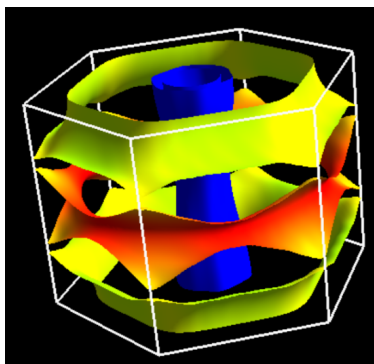
```
2
6 10
```

では, 最初の数字 (2) が足し合わされる原子軌道射影の総数を表し, その後に足し合わされる原子軌道射影に対応する番号が列挙される. この場合は 1 番目の B 原子の 2pz 軌道 (6) と 2 番目の B 原子の 2pz 軌道 (10) を足したものの

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

が出力される.

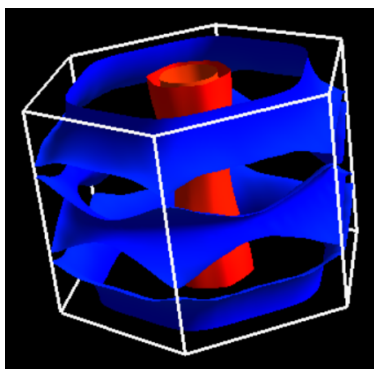
```
$ fermisurfer proj.frmsf
```



また例えば, すべての B 原子の 2px, 2py 軌道からの寄与を合わせたものをプロットしたい場合には,

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

のように `proj.in` を書き換えて, `fermi_proj.x` をもう一度実行すれば良い. `projwfc.x` を再度実行する必要は無い.



Chapter 7

ギャラリー

IrO_2 の Hall 係数に対する Fermi 面上の各軌道からの寄与 (図 7.1). 理化学研究所 有田グループ 佐野航氏提供).

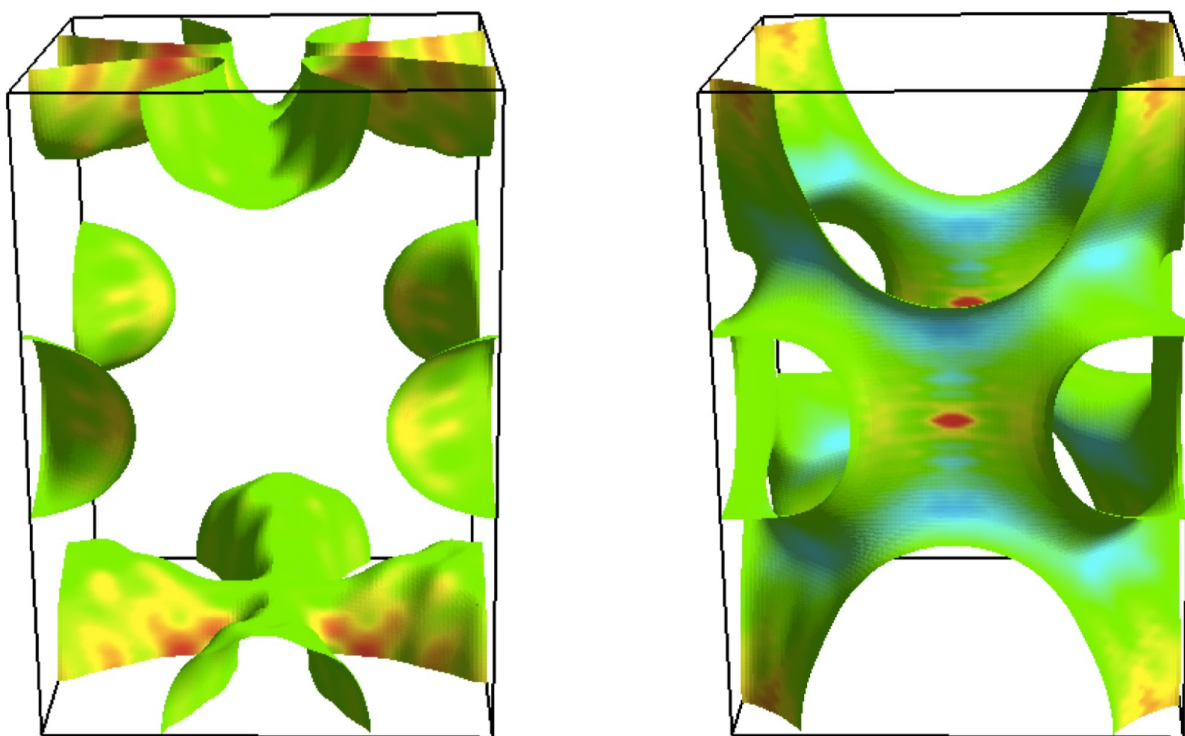


図 7.1: IrO_2 の Hall 係数に対する Fermi 面上の各軌道からの寄与.

Chapter 8

謝辞

東京大学物性研究所 小西優祐氏には, Mac OSX での動作チェックおよび Makefile, パッチの提供をしていただいたことに感謝する.

Chapter 9

プログラムの再配布

自分のプログラムに **FermiSurfer** を含める

FermiSurfer は下記の [MIT ライセンス](#) に基づいて配布されている。これはかいつまんで言うと、個人的 (研究室や共同研究者等のグループ) なプログラムであろうとも、公開したり売ったりするプログラムであろうとも自由にコピーしたり改変して良いし、どのようなライセンスで配布しても構わない、ということである。

Autoconf を使わずに **FermiSurfer** をビルドする

このパッケージでは Autotools (Autoconf, Aptomake, Libtool) を使って fermisurfer をビルドしている。もし再配布するソースコードに fermisurfer を含めるときに、Autoconf の使用に支障がある場合には、以下の簡易版の Makefile を使うと良い (タブに注意)。

```
CC = gcc
CFLAGS=-g -O0 -lglut -lGLU -lGL -lm -fopenmp -g -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 ライセンス

Copyright (c) 2014 Mitsuaki Kawamura

以下に定める条件に従い、
本ソフトウェアおよび関連文書のファイル（以下「ソフトウェア」）
の複製を取得するすべての人に対し、
ソフトウェアを無制限に扱うことを無償で許可します。これには、
ソフトウェアの複製を使用、複写、変更、結合、掲載、頒布、サブライセンス、
および/または販売する権利、
およびソフトウェアを提供する相手に同じことを許可する権利も無制限に含まれます。

上記の著作権表示および本許諾表示を、
ソフトウェアのすべての複製または重要な部分に記載するものとします。

ソフトウェアは「現状のまま」で、明示であるか暗黙であるかを問わず、何らの保証もなく提供されます。ここでいう保証とは、商品性、特定の目的への適合性、および権利非侵害についての保証も含みますが、それに限定されるものではありません。作者または著作権者は、契約行為、不法行為、またはそれ以外であろうと、ソフトウェアに起因または関連し、あるいはソフトウェアの使用またはその他の扱いによって生じる一切の請求、損害、その他の義務について何らの責任も負わないものとします。

Chapter 10

問い合わせ先

プログラムのバグや質問は以下のフォーラムへご投稿ください。

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

開発に参加したい方は以下の連絡先にて受け付けております。

東京大学物性研究所

河村光晶

mkawamura__at__issp.u-tokyo.ac.jp