



Fermisurfer Documentation

リリース *2.0.0*

kawamura

2019 年 06 月 27 日

目次

第 1 章	はじめに	1
第 2 章	ディレクトリと主なファイル	3
第 3 章	インストール手順	5
3.1	Linux の場合	5
3.2	Mac, Windows の場合	5
第 4 章	Input file	7
4.1	input file の書式	7
4.2	BXSF 形式	8
4.3	C/fortran での入力ファイルの書き出し方	9
4.4	2 次元量のカラープロットを行う場合	10
4.5	カラープロットしたい量を省略する場合	11
第 5 章	操作方法	13
5.1	起動	13
5.1.1	Linux・Unix・Mac の場合	13
5.1.2	Windows の場合	13
5.2	背景色	14
5.3	バンド毎の表示・非表示の切り替え	14
5.4	Brillouin zone (要 Update)	14
5.5	Color bar	15
5.6	カラープロットの種類と範囲 (要 Update)	16
5.7	Equator (要 Update)	17
5.8	補間の細かさ (要 Update)	17
5.9	Fermi 面のどちら側に光を当てるか	18
5.10	Line width	19
5.11	マウスドラッグをしたときの振る舞い	19
5.12	ノーダルライン	19
5.13	ブリルアンゾーンの断面 (要 Update)	19
5.14	Fermi エネルギーの変更 (要 Update)	21
5.15	立体視	22

5.16	四面体の切り方 (要 Update)	23
5.17	サイズ・角度・位置の数値での調整	23
5.18	画像の保存方法	23
第 6 章	バッチ・モード	25
第 7 章	Quantum ESPRESSO を用いたチュートリアル	27
7.1	PostProcess ツールのビルド	27
7.2	SCF 計算	27
7.3	Fermi 速度の計算と描画	28
7.4	原子軌道射影の計算と描画	29
第 8 章	謝辞	33
第 9 章	プログラムの再配布	35
9.1	自分のプログラムに FermiSurfer を含める	35
9.2	MIT ライセンス	35
第 10 章	問い合わせ先	37

第 1 章

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.

第 2 章

ディレクトリと主なファイル

- `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

第 3 章

インストール手順

3.1 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 libwxgtk3.0-dev
```

- Red Hat Enterprise Linux/CentOS 等

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

2. インストール

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

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

3.2 Mac, Windows の場合

それぞれの OS に応じたバイナリ実行ファイルをダウンロードする.

また wxWidgets ライブラリをインストールしたのちに FermiSurfer を自前でビルドすることも可能. Mac では Linux と同様に `configure` を用いてビルドする. Windows では VisualStudio 用ファイル `fermisurfer.vcxproj` が利用できる.


```
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 [C/fortran](#) 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 [C/fortran](#) での入力ファイルの書き出し方 参照) (オプションル)

0 個から 3 個のブロックで, カラープロットしたい量を入力する. 省略することも可能.

4.2 BXSf 形式

XCrysDen 用の入力ファイルを fermisurfer で読み取ることも可能. その場合は前節の「物理量」を省略した場合と同じ振る舞いになる.

4.3 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
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);

```

4.4 2次元量のカラープロットを行う場合

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 !グリッドをシフトさせるか (=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,2) !物理量 (2次元量、複素数など)

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
end do
close(fo)

```

4.5 カラープロットしたい量を省略する場合

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 !グリッドをシフトさせるか (=1) 否か (=0)
integer :: nbnd !バンド数
real(4) :: eig(nk3,nk2,nk1,nbnd) !エネルギー

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
```

第 5 章

操作方法

5.1 起動

5.1.1 Linux・Unix・Mac 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.)

5.1.2 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.

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る.
- マウスのホイールを使っでの拡大・縮小が出来る.
- ウィンドウの大きさを変えることもできる.
- カーソルキー (Windows では wasd) を使ってウィンドウ内で上下左右に図を動かせる.
- 右側のパネルを用いて様々な操作ができる.

次から右側のパネルでの操作をを説明する.

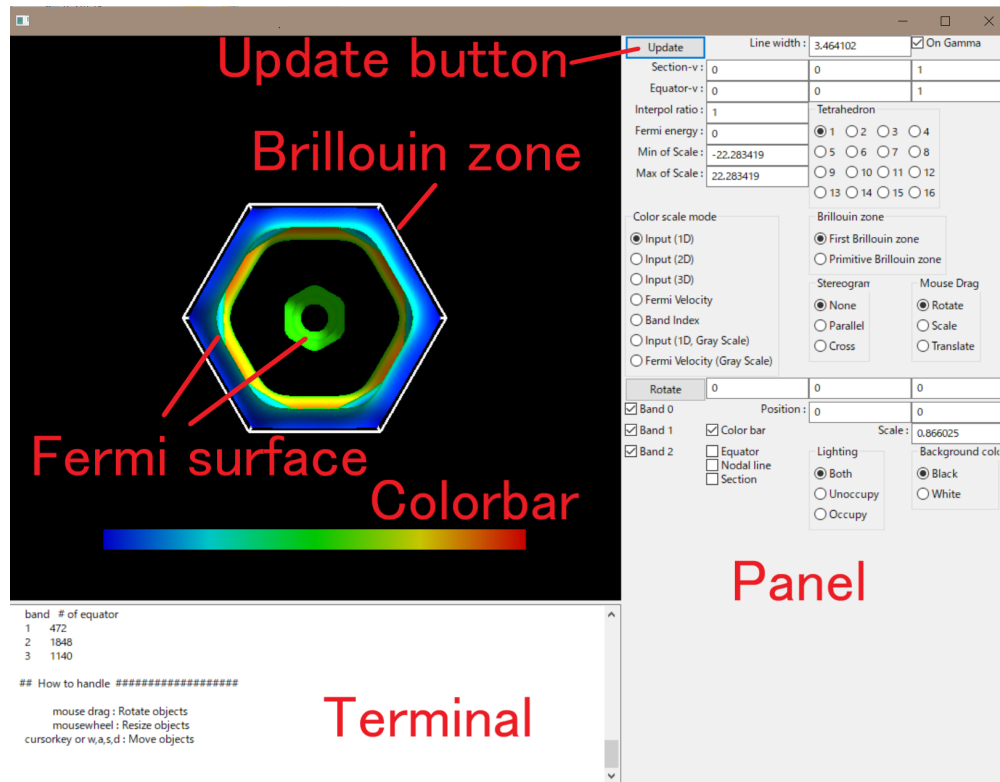


図 5.1 Fermisurfer を起動した直後の画面.

注釈: いくつかの操作はすぐに反映されず「Update」ボタンを押すことによって反映される. そのような操作については「要 Update」と明記してある.

5.2 背景色

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.

5.3 バンド毎の表示・非表示の切り替え

バンド毎の表示 on/off を切り替えます (図 5.2).

5.4 Brillouin zone (要 Update)

描画範囲を First Brillouin Zone/Primitive Brillouin Zone と切り替える事が出来ます (図 5.3).

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

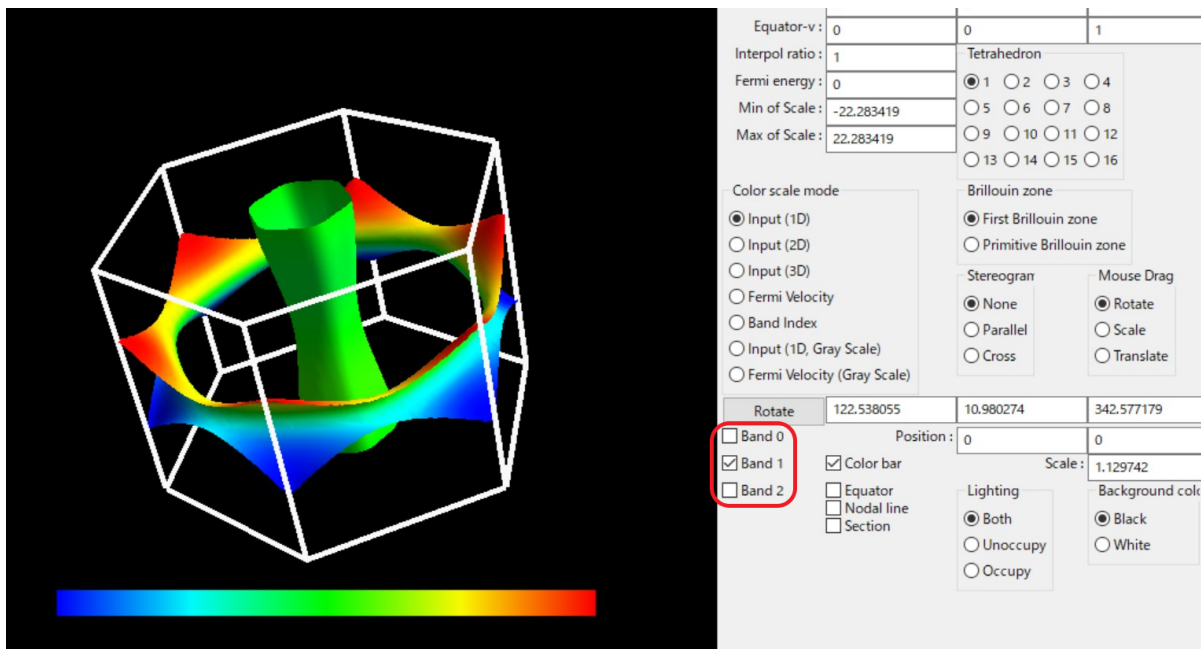
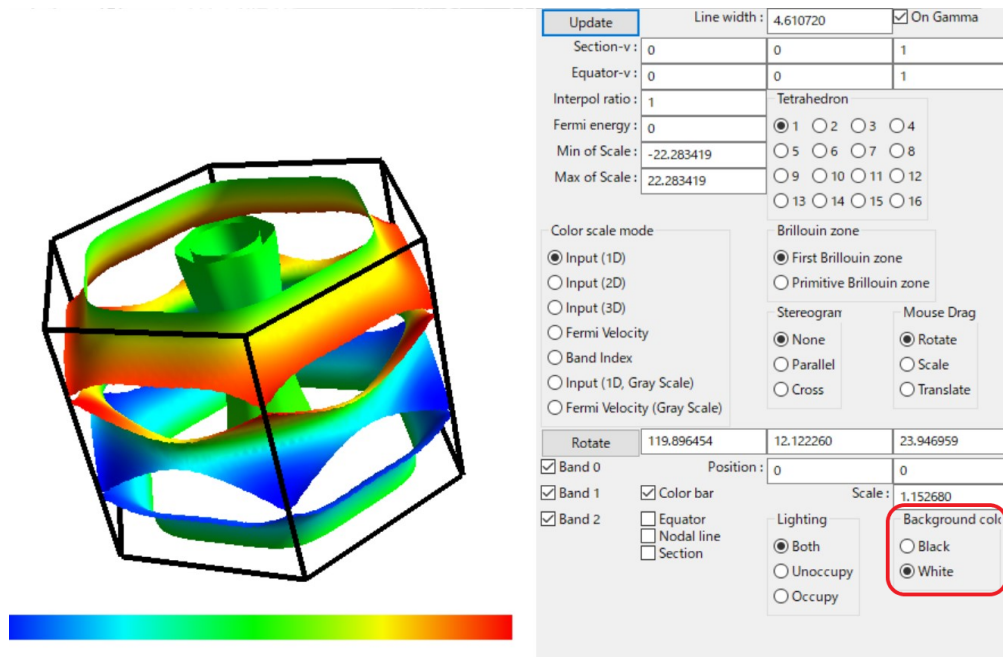


図 5.2 "Band" メニューで 3 番目のバンドの表示/非表示を切り替える。

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

5.5 Color bar

カラーバーの表示/非表示を切り替えます (図 5.4).

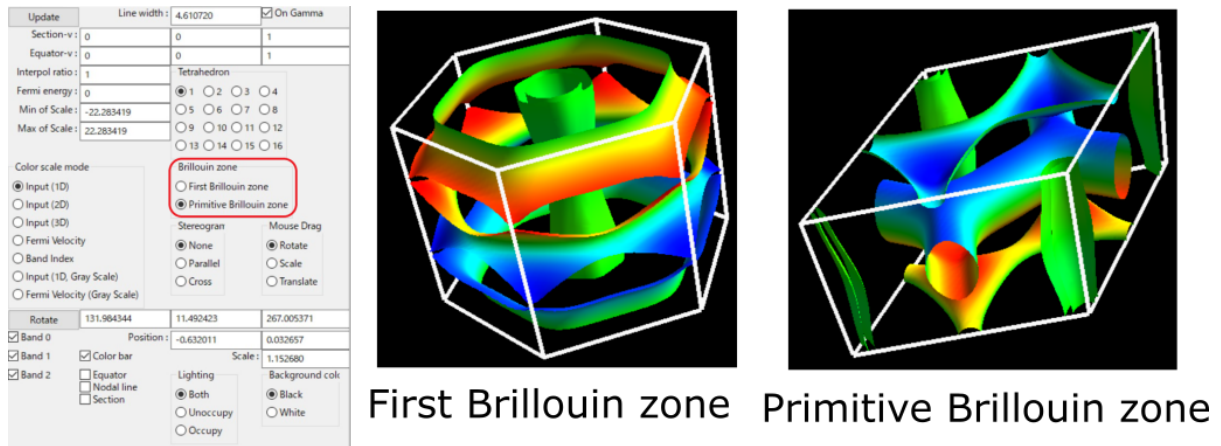


図 5.3 "Brillouin zone" メニューで Brillouin 領域のとり方を変更する。

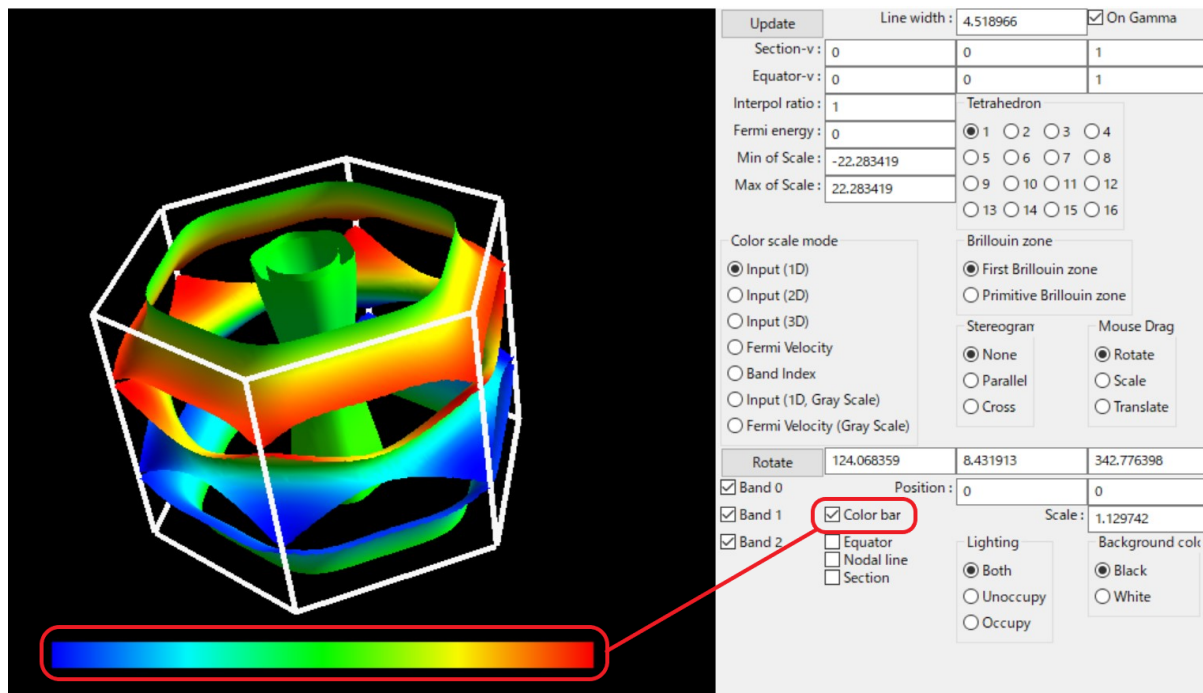


図 5.4 "Color bar On/Off" メニューでカラーバーの表示/非表示を切り替える。

5.6 カラープロットの種類と範囲 (要 Update)

Fermi 面の色表示のさせ方を変更します (図 5.5).

Input (1D) (入力ファイルのデータが 1 個のときのデフォルト): 青 → 緑 → 赤の範囲でカラープロットする。

Input (2D) (入力ファイルのデータが 2 個のときのデフォルト): カラーサークル (図参照) 上の色でカラープロットする。

Input (3D) (入力ファイルのデータが 3 個のときのデフォルト): フェルミ面上の線としてプロットする。フェル

ミ面の色は「Band Index」の場合と同様.

Fermi Velocity (入力ファイルのデータがエネルギーだけのときのデフォルト): 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.

Band Index: 物理量に関係なく, 各バンド毎に単色で Fermi 面を塗る.

Input (1D, Gray), Fermi Velocity (Gray) 黒色の濃淡でプロットする.

また, カラープロットの色範囲や 3D 線プロットの線の長さは「Min of Scale」および「Max of Scale」のテキストボックスに入力することで変更できる.

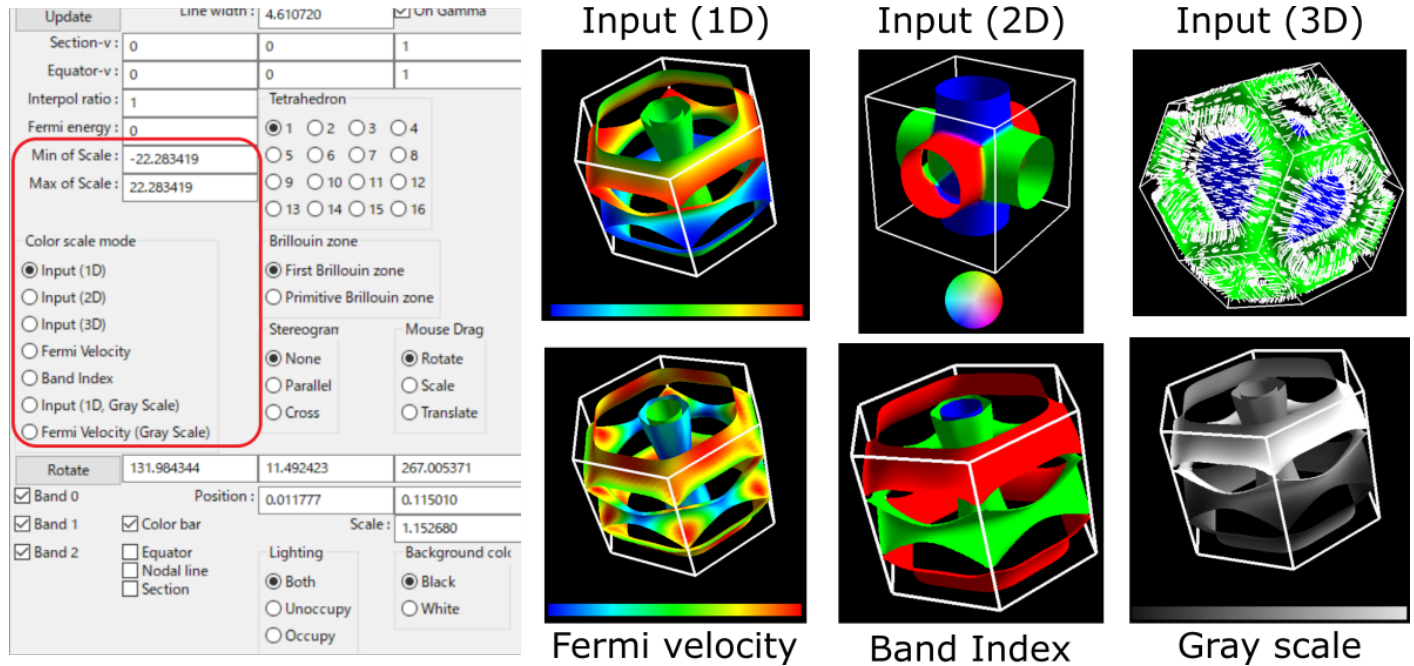


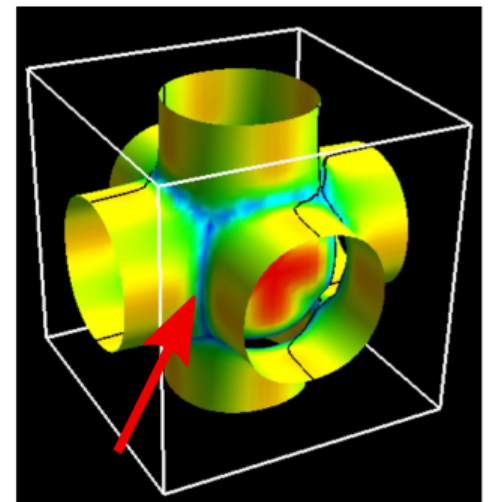
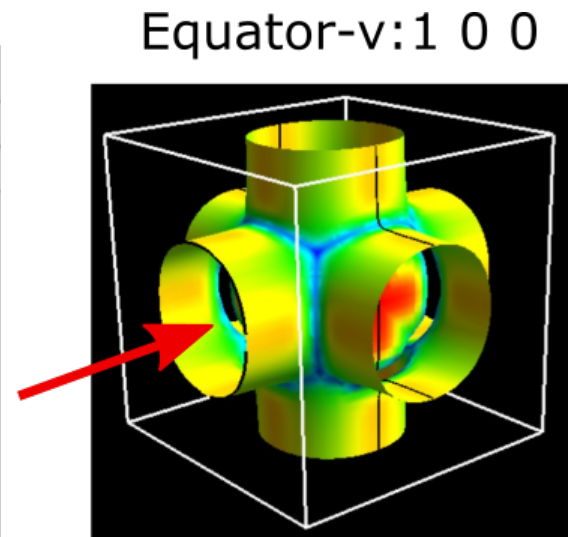
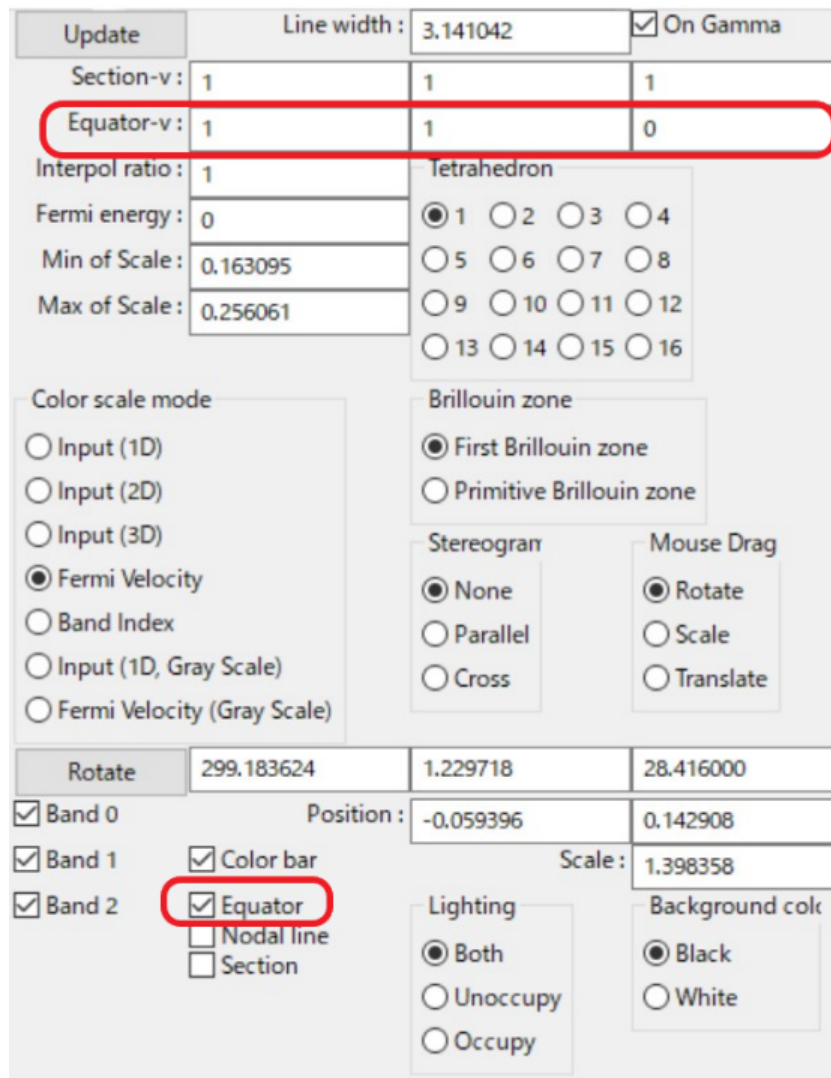
図 5.5 "Color scale mode" メニュー.

5.7 Equator (要 Update)

ある k に対して, $v_F \cdot k = 0$ となる線 (Equator:極軌道, もしくは Extremal orbit) を表示, 変更する. (図 5.6). 「Equator」チェックボックスで表示・非表示を切り替える. この時「Update」を押す必要はない. 「Equator-v:」のテキストボックスで k を指定する. k ベクトルはフラクショナル座標で入力する.

5.8 補間の細かさ (要 Update)

補間により図の曲面を滑らかにする (図 5.7). ただし分点数を増やすと描画にかかる時間も増えるので注意.



Equator-v:1 1 0

図 5.6 "Equator"メニューで Fermi 面の極軌道 (Equator) を表示する.

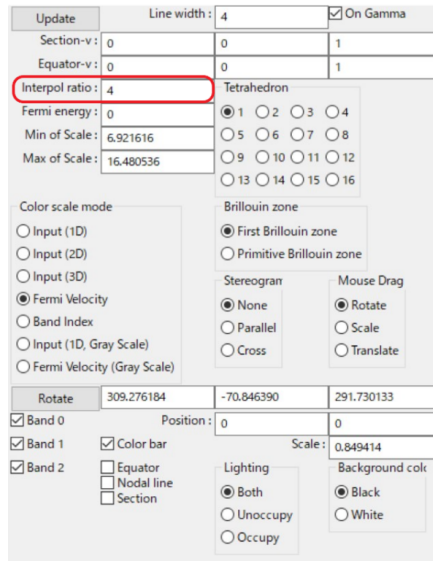
5.9 Fermi 面のどちら側に光を当てるか

光を当てる面を変更します (図 5.8).

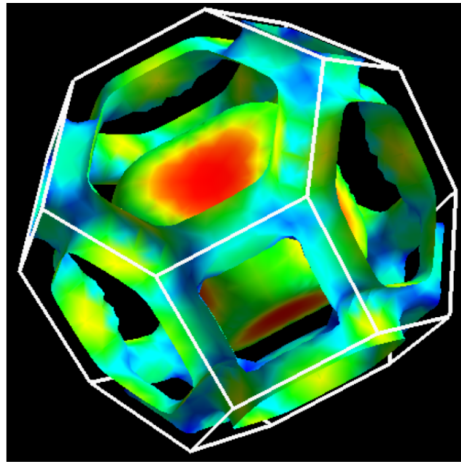
Both : Fermi 面の表裏両面に光を当てます.

Unoccupy : 非占有領域側のみに光を当てます.

Occupy : 占有領域側のみに光を当てます.



Interpol ratio : 1



Interpol ratio : 4

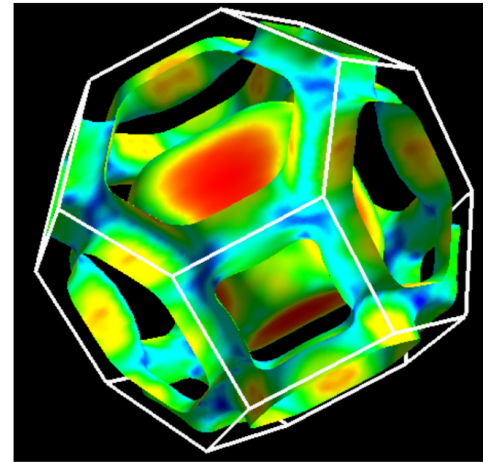


図 5.7 "Interpolate"メニューで 分点数を 1 から 4 に変える.

5.10 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.

5.11 マウสดラッグをしたときの振る舞い

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.

5.12 ノーダルライン

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 5.9).

5.13 ブリルアンゾーンの断面 (要 Update)

Brillouin 領域を任意の断面で切り取り, 2 次元の Fermi 面 (線) を描画する (図 5.13).

「Section」チェックボックスで断面の表示・非表示を切り替える (この操作は「Update」の必要はない). 断面の法線の指定は「Section-v:」のテキストボックスで行う. 法線ベクトルはフラクショナル座標で指定する.

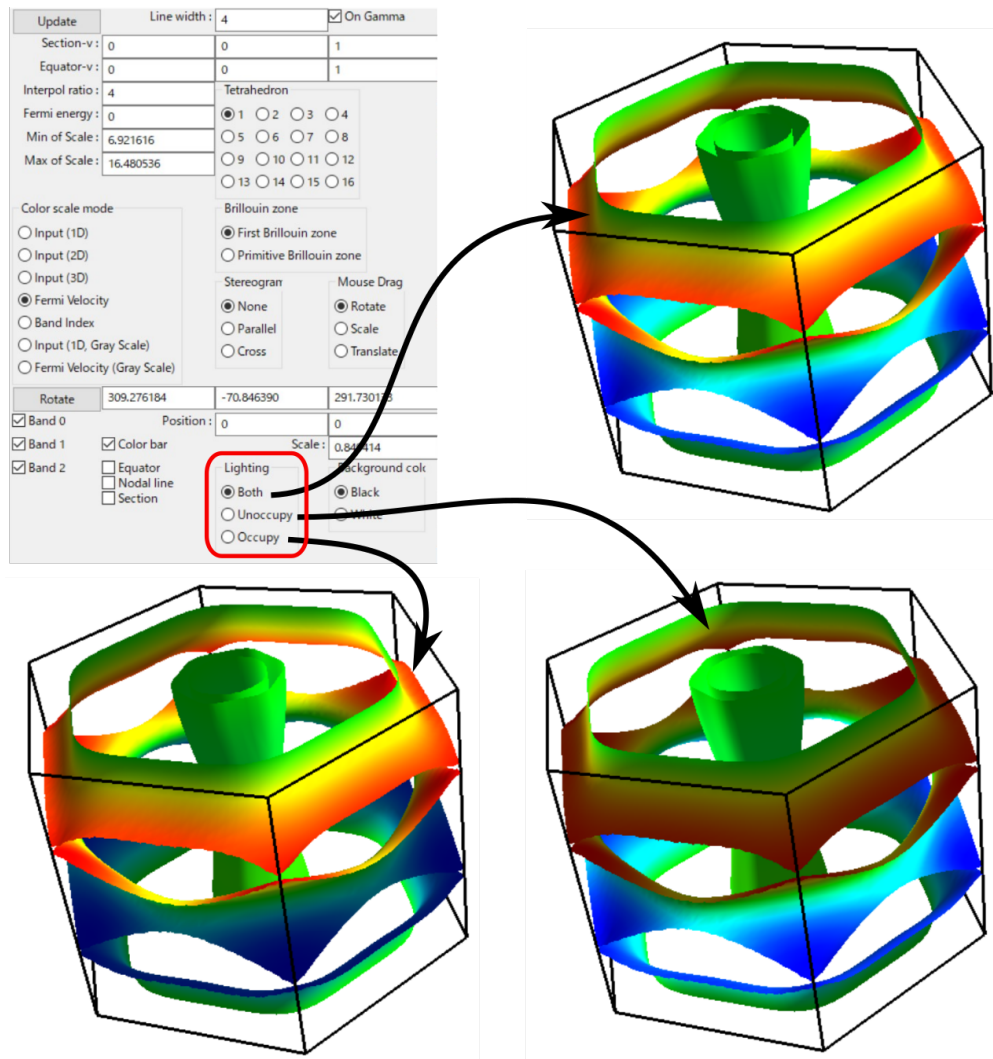
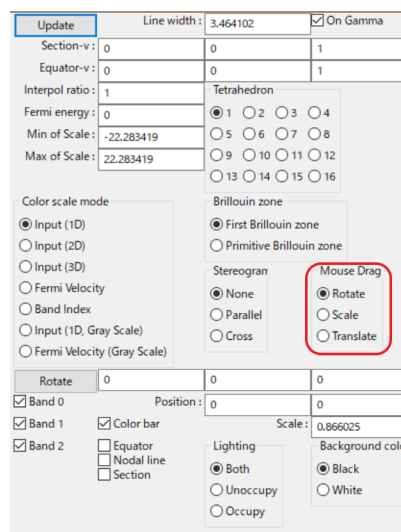


図 5.8 "Lighting"メニューで光を当てる Fermi 面を変更する.



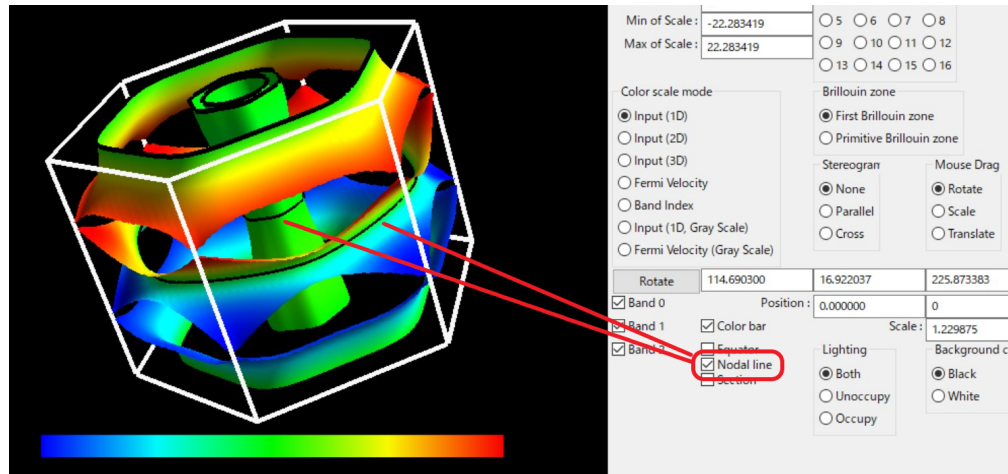
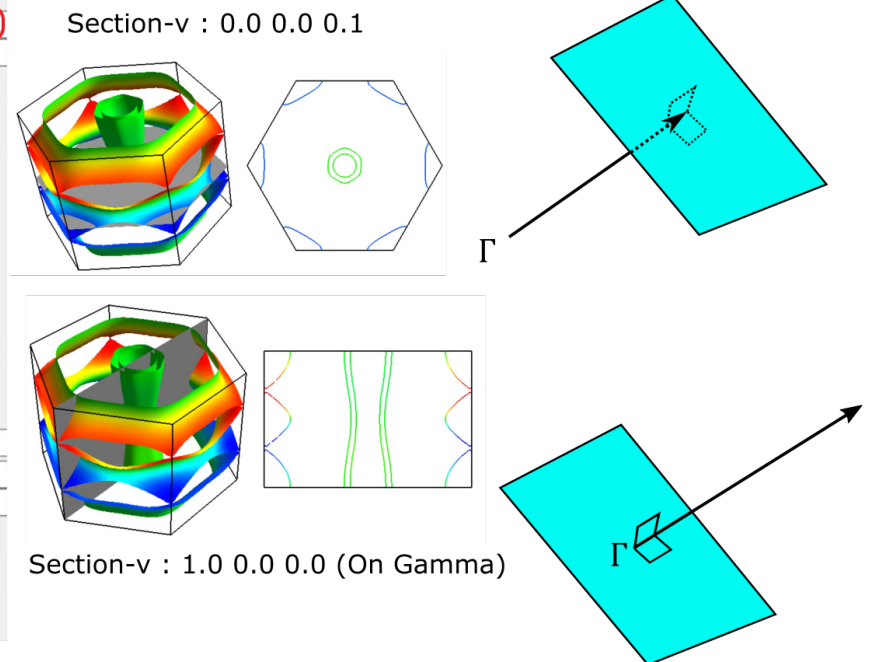
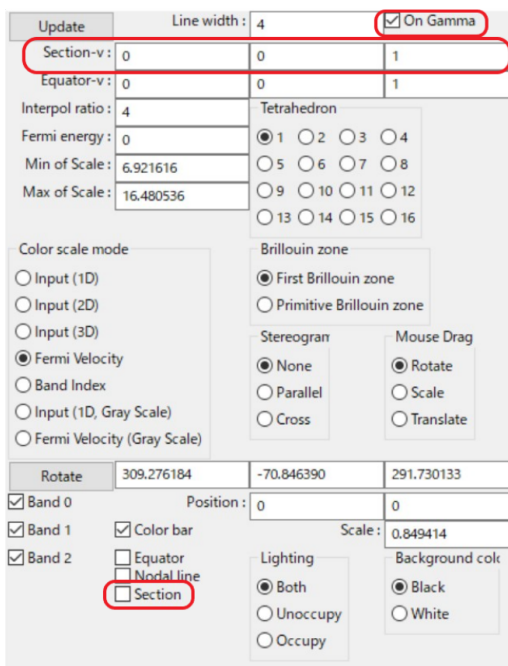


図 5.9 "Nodal line" メニューで nodal line の表示/非表示を切り替える。

また、「On Gamma」のチェックボックスがオンになっているときには断面は Γ 点を通る。



5.14 Fermi エネルギーの変更 (要 Update)

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. (図 5.10).

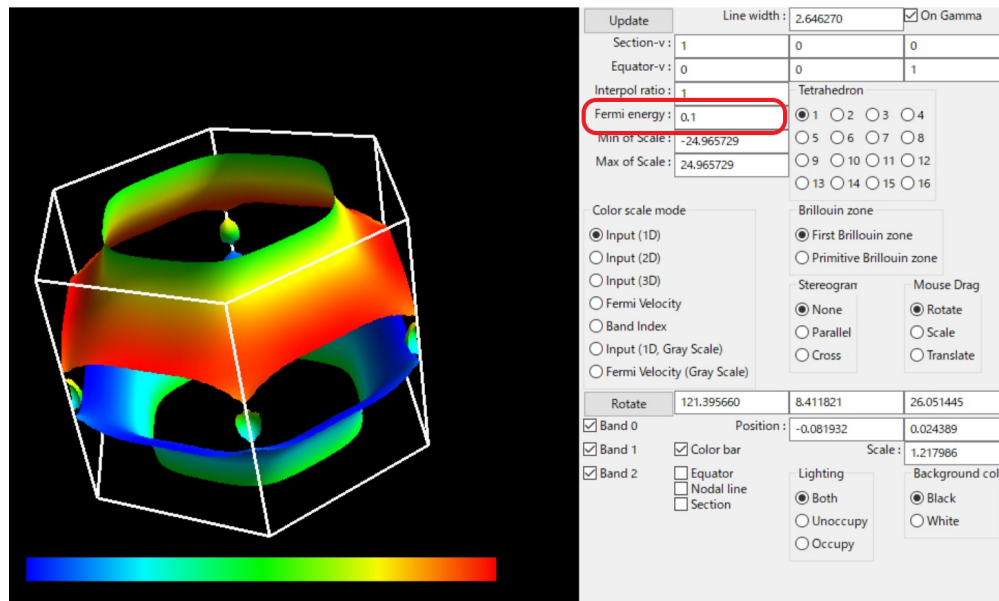


図 5.10 "Shift Fermi energy"メニューで Fermi エネルギーを 0.0 Ry から 0.1 Ry に変える。

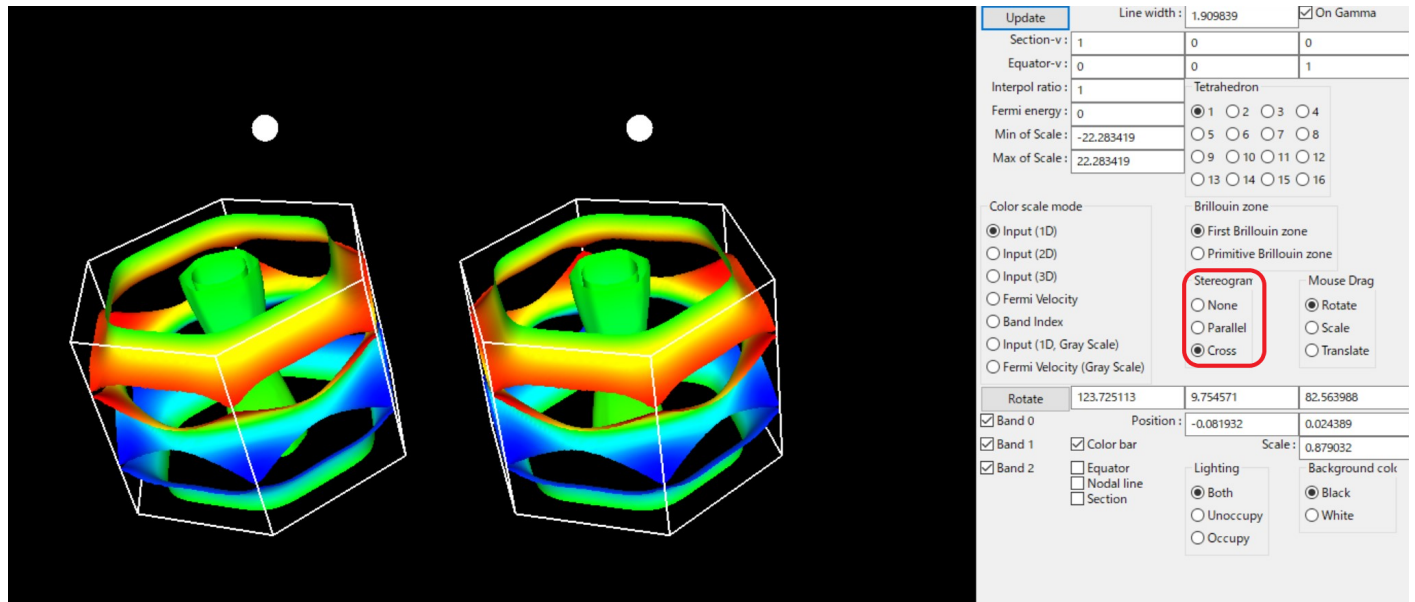
5.15 立体視

裸眼立体視用の図の表示/非表示を切り替えます (図 5.15).

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。



5.16 四面体の切り方 (要 Update)

四面体の切り方を変えます。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。

The screenshot shows the 'Update' dialog box in the Fermisurfer software. The 'Tetrahedron' section is highlighted with a red box, showing radio buttons for 1 through 16. Other sections include 'Color scale mode', 'Brillouin zone', 'Stereogram', 'Mouse Drag', 'Rotate', 'Band', 'Position', 'Scale', 'Lighting', and 'Background color'.

5.17 サイズ・角度・位置の数値での調整

視点を変更します (図 5.11).

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。

Rotate : x, y, z 軸周りの回転角を指定し, "Roate" ボタンを押すと回転する。回転操作は z 軸- y 軸- x 軸の順で行われます。

5.18 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。

Update	Line width :	4	<input checked="" type="checkbox"/> On Gamma
Section-v :	0	0	1
Equator-v :	0	0	1
Interpol ratio :	4	Tetrahedron	
Fermi energy :	0	<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4	
Min of Scale :	6.921616	<input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8	
Max of Scale :	16.480536	<input type="radio"/> 9 <input type="radio"/> 10 <input type="radio"/> 11 <input type="radio"/> 12	
		<input type="radio"/> 13 <input type="radio"/> 14 <input type="radio"/> 15 <input type="radio"/> 16	
Color scale mode		Brillouin zone	
<input type="radio"/> Input (1D)		<input checked="" type="radio"/> First Brillouin zone	
<input type="radio"/> Input (2D)		<input type="radio"/> Primitive Brillouin zone	
<input type="radio"/> Input (3D)		Stereogram	
<input checked="" type="radio"/> Fermi Velocity		<input checked="" type="radio"/> None	
<input type="radio"/> Band Index		<input type="radio"/> Parallel	
<input type="radio"/> Input (1D, Gray Scale)		<input type="radio"/> Cross	
<input type="radio"/> Fermi Velocity (Gray Scale)		Mouse Drag	
		<input checked="" type="radio"/> Rotate	
		<input type="radio"/> Scale	
		<input type="radio"/> Translate	
Rotate	309.276184	-70.846390	291.730133
<input checked="" type="checkbox"/> Band 0	Position :		0
<input checked="" type="checkbox"/> Band 1	<input checked="" type="checkbox"/> Color bar	Scale :	0.849414
<input checked="" type="checkbox"/> Band 2	<input type="checkbox"/> Equator	Lighting	
	<input type="checkbox"/> Nodal line	<input checked="" type="radio"/> Both	
	<input type="checkbox"/> Section	<input type="radio"/> Unoccupy	
		<input type="radio"/> Occupy	
		Background color	
		<input checked="" type="radio"/> Black	
		<input type="radio"/> White	

図 5.11 "View point"メニューで 視点を変更する.

第 6 章

バッチ・モード

コマンドライン操作のみで FermiSurfer で描画した図を画像 (PNG) ファイルに保存する, 「バッチ・モード」について説明する. バッチ・モードを用いるとこのような沢山の図を簡単に作ることができる.

例として, example/ ディレクトリ内で

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

を実行すると, 画像ファイル `frmsf.in.png` が出力される. 末尾の二つの数字はウィンドウの幅および高さである. `frmsf.in` はバッチ・モード用の描画設定ファイルで, 次のように書かれている.

```
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
```

これは前節のパネル操作に対応しており, 指定可能なキーワードは次の通りである. また, 指定しなかったキーワードについてはデフォルト値が使われる.

キー ワード	指定可能なパラメーター	デフォルト値	説明
background	black, white	black	背景色
band	1 または 0 をバンド本数分	1 1 1 1 ...	各バンドの表示 (1) または非表示 (0)
brillouinzone	first, primitive	first	ブリルアンゾーンの種類
colorbar	0, 1	1	カラーバーの表示 (1) または非表示 (0)
colormap	input1d, input2d, input3d, fermivelocity, bandindex, inputgray, fermivelocitygray	input1d	カラープロットの種類
minmax	実数 実数	フェルミ面全体の最小値, 最大値	カラースケールの範囲
equator	実数 実数 実数	指定しない場合は極軌道を表示しない	極軌道の接ベクトル (フラクショナル座標)
interpol	自然数	1	補間の細かさ
linewidth	実数	3.0	線幅
lighting	both, unoccupied, occupied	both	どちらの面に照光するか
nodalline	0, 1	0	ノードラインの表示 (1) または非表示 (0)
section	実数 実数 実数	指定しない場合は断面を表示しない	断面の法線ベクトル (フラクショナル座標)
acrossgamma	0, 1	1	断面が Γ 点を通るか (1) 否か (0)
position	実数 実数	0.0, 0.0	図形の描画位置
scale	実数	1.0	図形の拡大率
rotation	実数 実数 実数	0.0, 0.0, 0.0	図形の x-, y-, z-軸周りの回転角
fermienergy	実数	0.0	Fermi エネルギー
stereogram	none, parallel, cross	none	立体視
tetrahedron	0 から 15 の整数	0	四面体の切り方

注釈: この機能は ImageMagic のスクリーンショットを取るコマンド「import」を用いている. そのためこの機能を使うには ImageMagic がインストールされていなければならない.

第 7 章

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)

7.1 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
```

7.2 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](https://www.quantum-espresso.org/Doc/pslibrary.html) 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.

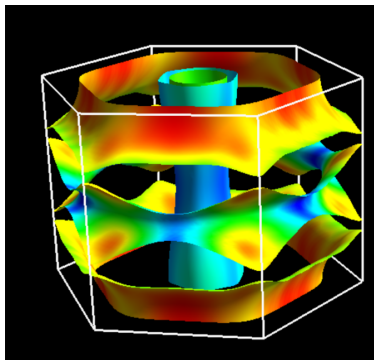
7.3 Fermi 速度の計算と描画

`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.

7.4 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

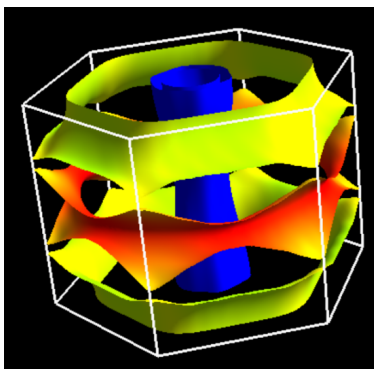
```
2
6 10
```

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

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

が出力される。

```
$ fermisurfer proj.frmsf
```

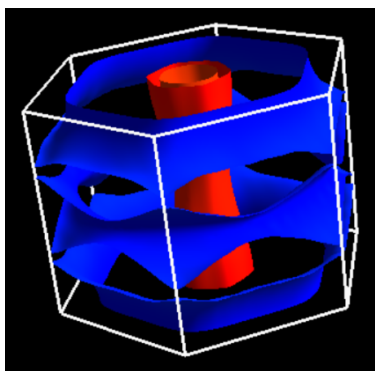


また例えば、すべての B 原子の 2px, 2py 軌道からの寄与を合わせたものをプロットしたい場合には、

```
&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.



第 8 章

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.

第 9 章

プログラムの再配布

9.1 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。

9.2 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.

第 10 章

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