



Fermisurfer Documentation

リリース 2.4.0

kawamura

2023 年 08 月 18 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	2
第3章 インストール手順	3
3.1 Linux, macOS の場合	3
3.2 Windows の場合	3
第4章 Input file	4
4.1 input file の書式	4
4.2 BXSF 形式	5
4.3 C/fortran での入力ファイルの書き出し方	5
4.4 2次元量のカラープロットを行う場合	7
4.5 カラープロットしたい量を省略する場合	8
第5章 操作方法	10
5.1 起動	10
5.1.1 Linux・Unix・Mac の場合	10
5.1.2 Windows の場合	10
5.2 背景色	11
5.3 線幅	12
5.4 線色	12
5.5 バンド毎の表示・非表示の切り替え	12
5.6 Brillouin zone の形式 (要 Update)	13
5.7 Brillouin zone の数 (要 Update)	13
5.8 カラーバーの表示/非表示	14
5.9 カラープロットの種類と範囲 (要 Update)	14
5.10 遠近法の有効化・無効化	15
5.11 カラープロットの配色	15
5.12 Equator (要 Update)	16
5.13 補間の細かさ (要 Update)	16
5.14 Fermi 面のどちら側に光を当てるか	17
5.15 マウスドラッグをしたときの振る舞い	17
5.16 ノーダルライン	19
5.17 ブリルアンゾーンの断面 (要 Update)	19
5.18 ブリルアンゾーン断面のファイル出力	19
5.19 Fermi エネルギーの変更 (要 Update)	21
5.20 立体視	21
5.21 四面体の切り方 (要 Update)	21
5.22 サイズ・角度・位置の数値での調整	23
5.23 矢印	23

5.24 ワイヤーフレーム球	24
5.25 ネスティング関数	24
5.26 画像の保存方法	24
第 6 章 パッチ・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 FermiSurfer on Web	32
8.1 操作方法	32
8.2 ファイル入力	32
8.3 Web 上のファイルを開く	32
第 9 章 謝辞	34
第 10 章 プログラムの再配布	35
10.1 自分のプログラムに FermiSurfer を含める	35
10.2 MIT ライセンス	35
第 11 章 問い合わせ先	36

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, macOS 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
```

- macOS (Homebrew)

```
$ brew install wxmac
```

- macOS (Mac Ports)

```
$ port install wxWidgets-3.0
```

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 Windows の場合

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 を自分でビルドすることも可能。VisualStudio 用ファイル fermisurfer.vcxproj が利用できる。

第4章 Input file

4.1 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
3          (2)
(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
2.93242838E-02
2.98905596E-02
3.08193434E-02
:
:
0.14393796
0.12800488
0.0000000
0.36269817
0.71675694
1.0535113
1.3644149
:
:
-26.409407

```

(次のページに続く)

(前のページからの続き)

-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:math: 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) (オプショナル)

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;

```

(次のページに続く)

(前のページからの続き)

```

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")

```

(次のページに続く)

(前のページからの続き)

```

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

```

4.5 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

```

(次のページに続く)

(前のページからの続き)

```
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 MgB₂ 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 面が描画される(図 1)。

- マウスのドラッグによる回転ができる。
- マウスのホイールを使っての拡大・縮小ができる。
- 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。ダブルクリックでも移動可。
- 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説明する。

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。

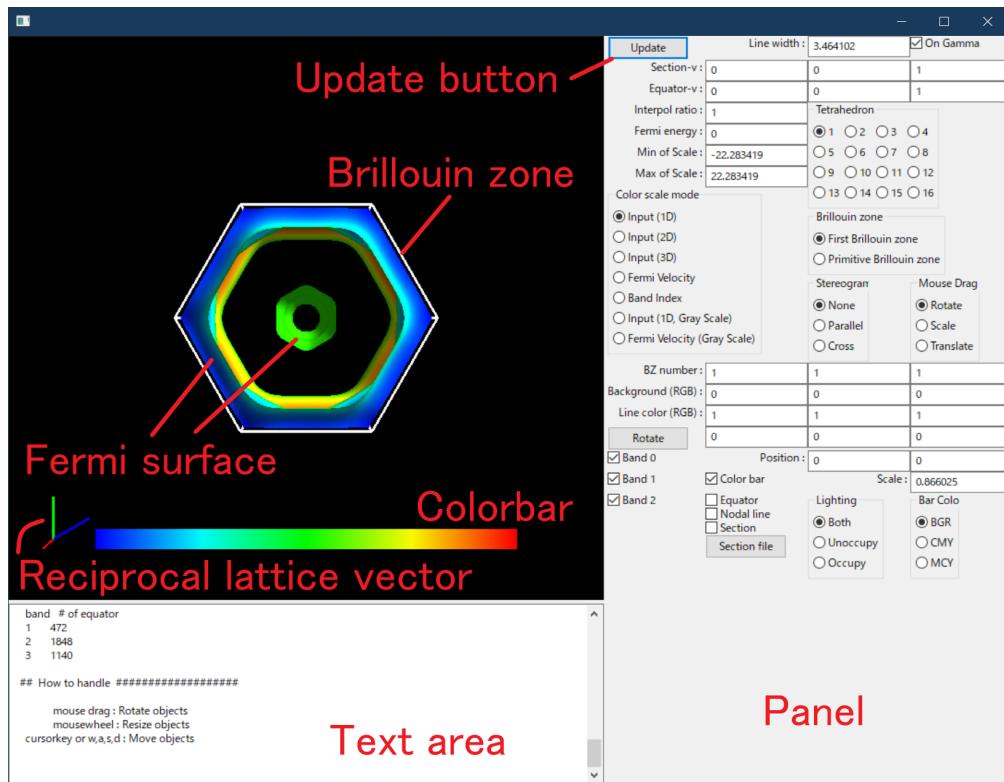
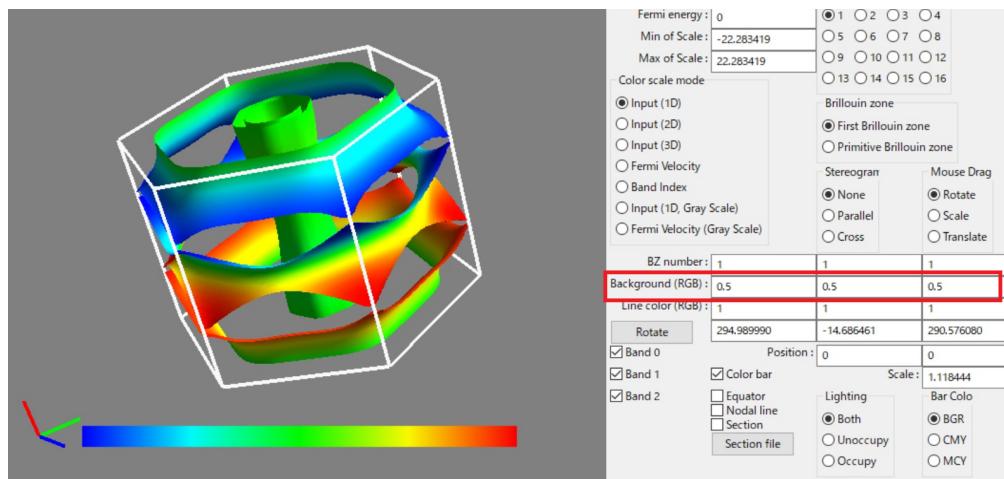


図 1: Fermisurfer を起動した直後の画面。

5.2 背景色

Background (RGB) : 背景色を RGB で指定する。

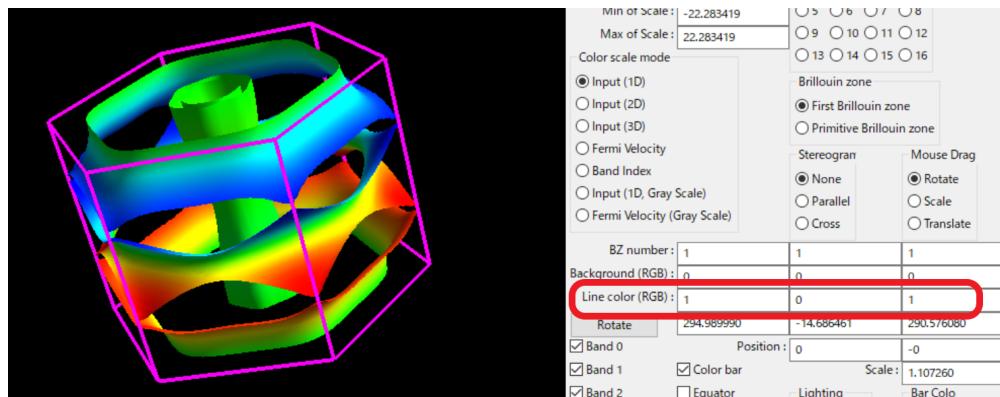


5.3 線幅

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する。

5.4 線色

Line color (RGB): 線色を RGB で指定する。



5.5 バンド毎の表示・非表示の切り替え

Band 0, RGB, Band 1, RGB ... : バンド毎の表示 on/off を切り替える (図 2)。

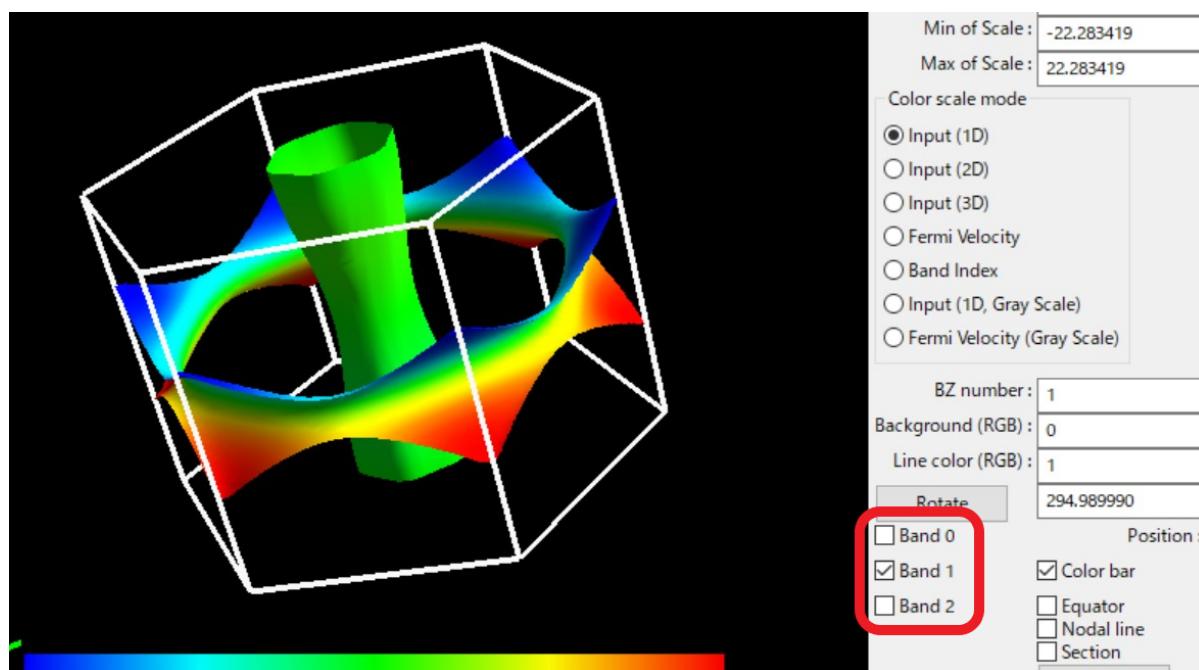


図 2: "Band" メニューで 3 番目のバンドの表示/非表示を切り替える。

5.6 Brillouin zone の形式 (要 Update)

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3)。

First Brillouin Zone

Γ 点から一番近い Bragg 面で囲まれた領域

Primitive Brillouine Zone

逆格子ベクトルを辺とする平行 6 面体領域

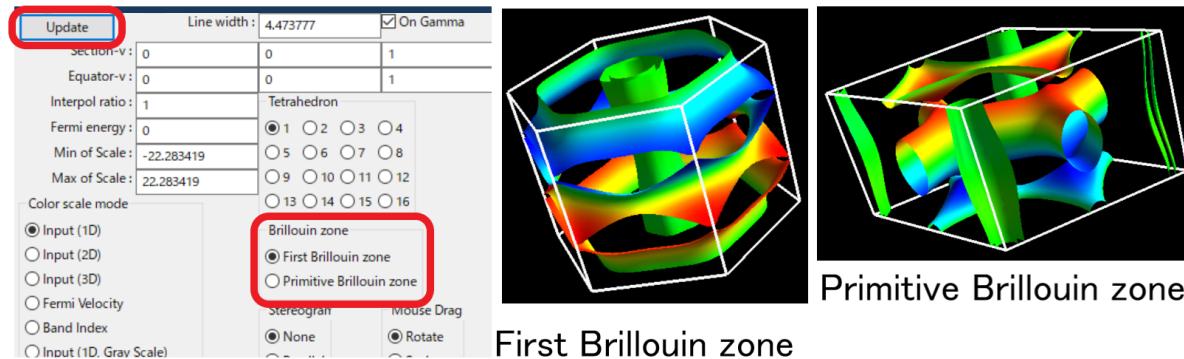
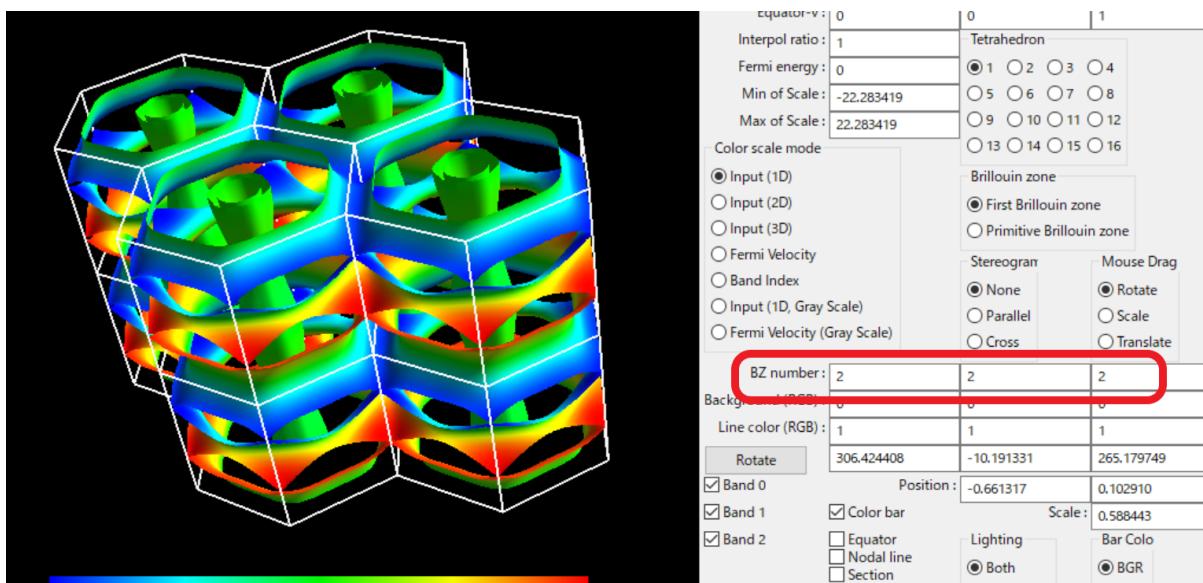


図 3: "Brillouin zone" メニューで Brillouin 領域のとり方を変更する。

5.7 Brillouin zone の数 (要 Update)

BZ number : 各逆格子ベクトルの方向にいくつ表示するかを指定する。



5.8 カラーバーの表示/非表示

Color bar : カラーバーの表示/非表示を切り替えます (図 4).

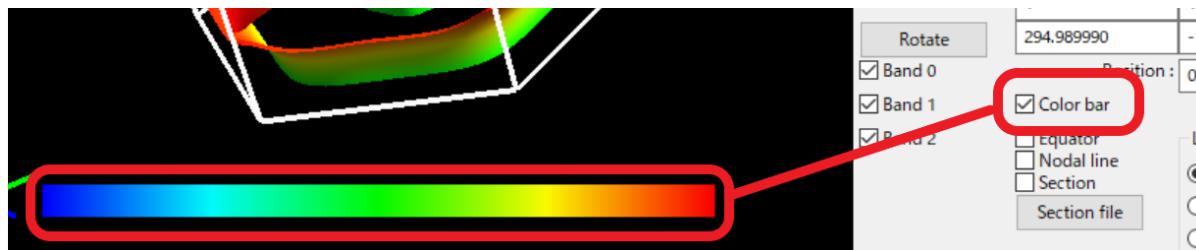


図 4: "Color bar On/Off" メニューでカラーバーの表示/非表示を切り替える.

5.9 カラープロットの種類と範囲 (要 Update)

Color scale mode : Fermi 面の色表示のさせ方を変更します (図 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) :

フェルミ面上の矢印 (細い三角形) 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.

Min of Scale, Max of Scale : カラープロットの色の範囲を指定する. 「Input (3D)」の場合は Max of Scale で三角形の長さを変更できる.

Arrow width : 「Input (3D)」の場合に矢印 (三角形) の幅を指定する.

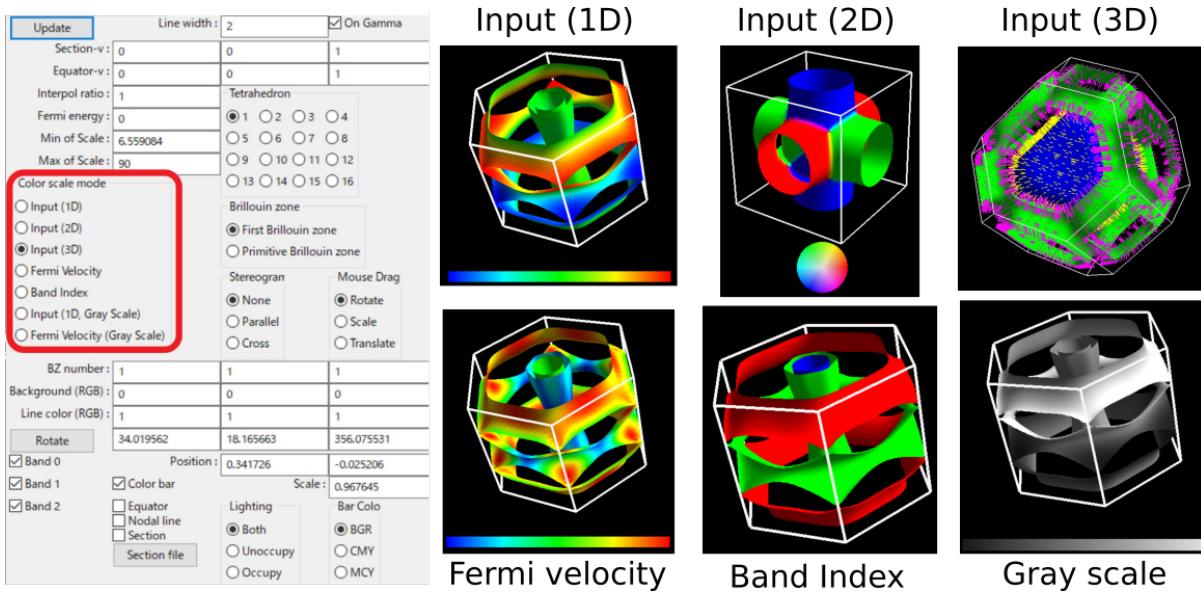


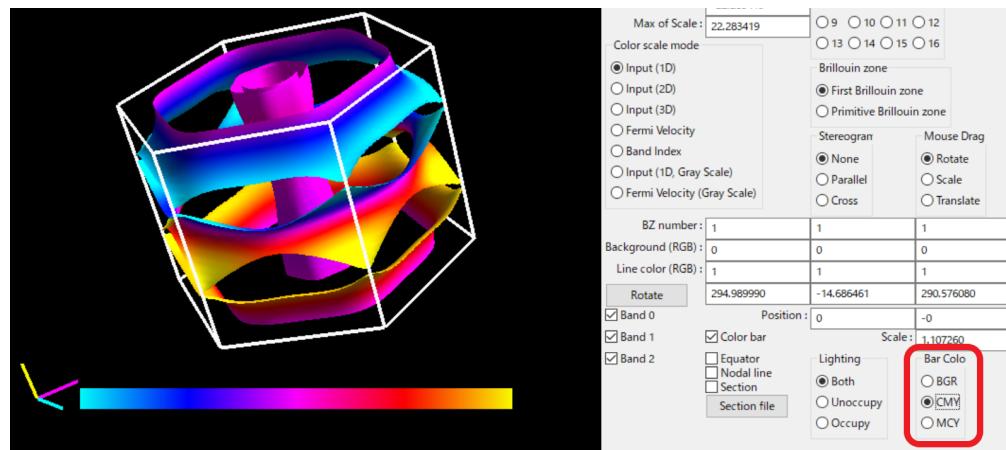
図 5: "Color scale mode" メニュー。

5.10 遠近法の有効化・無効化

Perspective: 遠近法の有効化・無効化を切り替える。

5.11 カラープロットの配色

Bar Color: カラープロットの配色を次の 3 つから選ぶことができる。"BGR":青 → 水 → 緑 → 黄 → 赤、"CMY":水 → 青 → 桃 → 赤 → 黄、"MCY":桃 → 青 → 水 → 緑 → 黄。

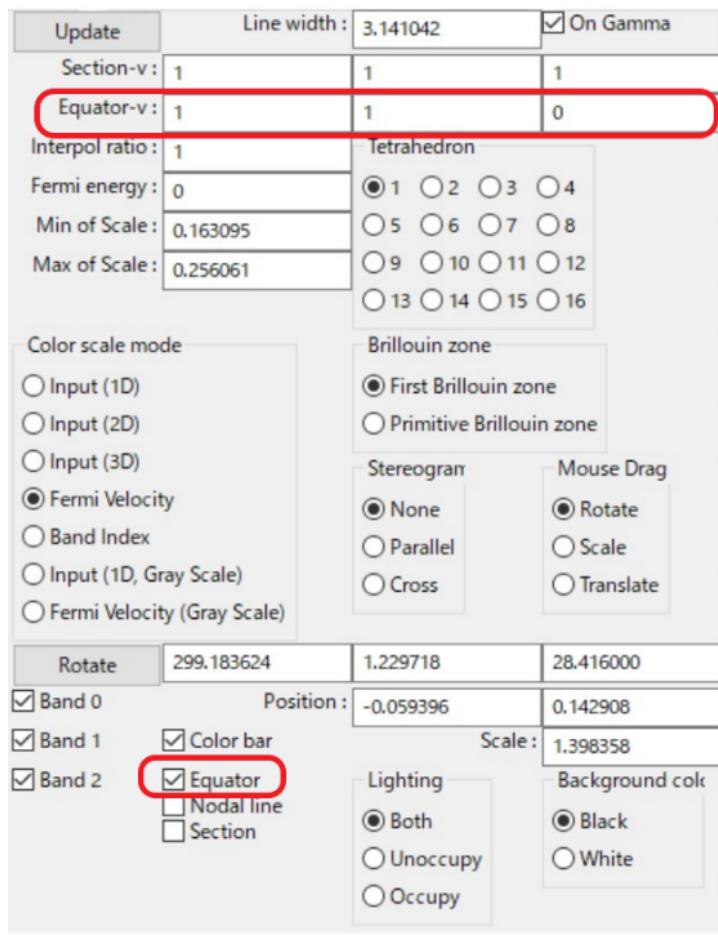


5.12 Equator (要 Update)

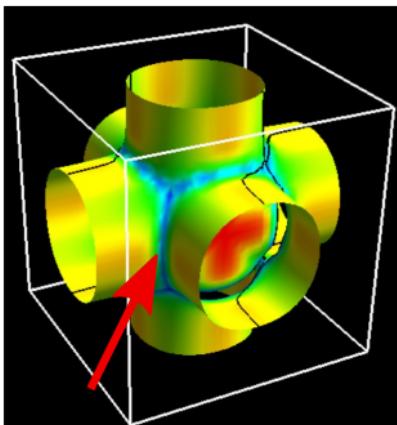
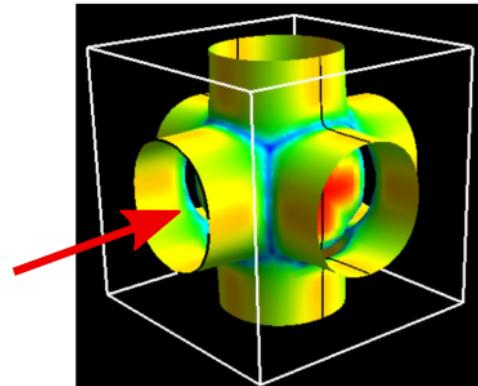
ある k に対して, $v_F \cdot k = 0$ となる線を表示, 変更する(図 6). 作った時点では作者が勘違いしていたが、dHvA で考慮される極軌道(extremal orbit)とは一致しない。超音波吸収とは関連すると考えられる。

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 : このテキストボックスで k を指定する。 k ベクトルはフракショナル座標で入力する。変更を反映させるには「Update」を押す。



Equator-v:1 0 0



Equator-v:1 1 0

図 6: "Equator"メニューで Fermi 面の極軌道(Equator)を表示する。

5.13 補間の細かさ (要 Update)

Interpol ratio : 補間ににより図の曲面を滑らかにする(図 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。

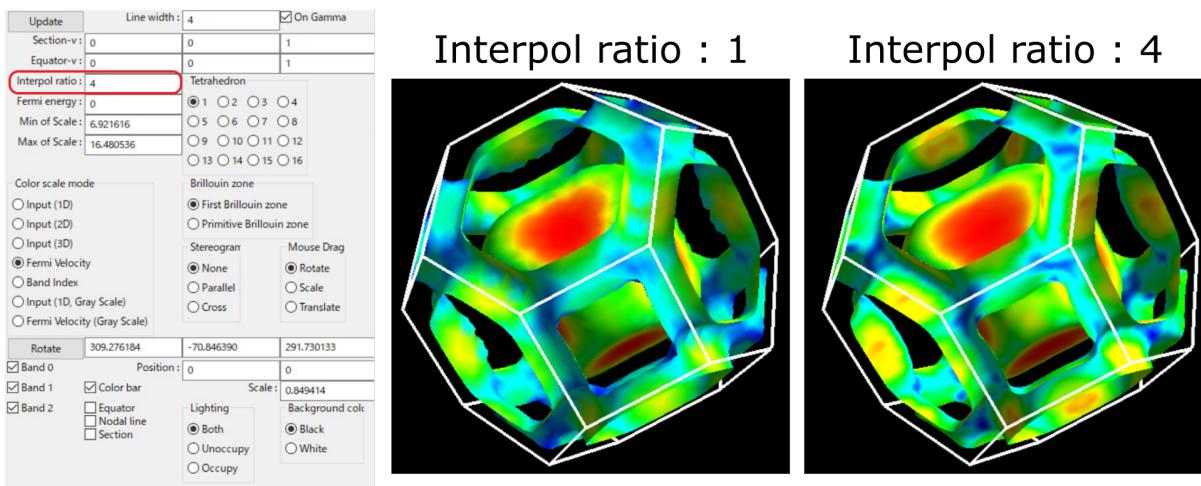


図 7: "Interpolate"メニューで分点数を 1 から 4 に変える。

5.14 Fermi 面のどちら側に光を当てるか

Lighting : 光を当てる面を変更します (図 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.

5.15 マウスドラッグをしたときの振る舞い

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.

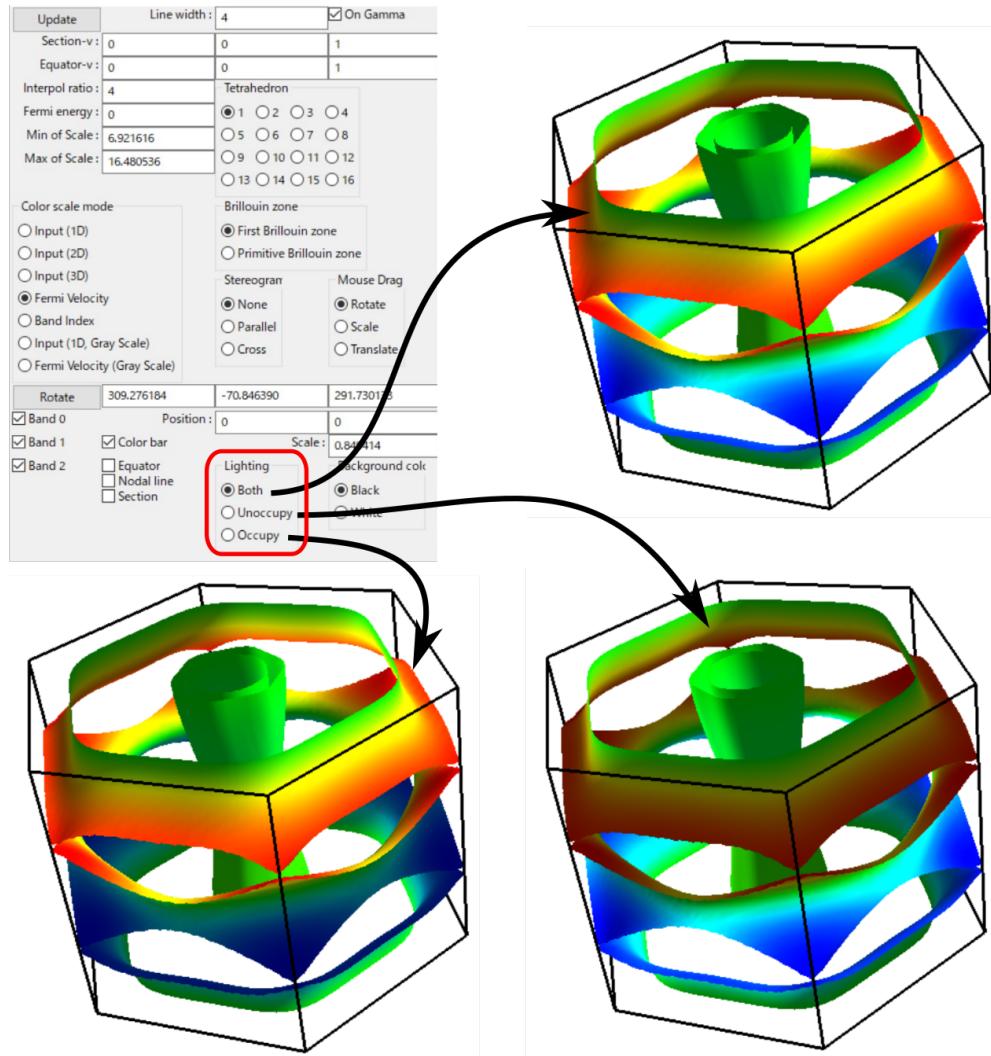
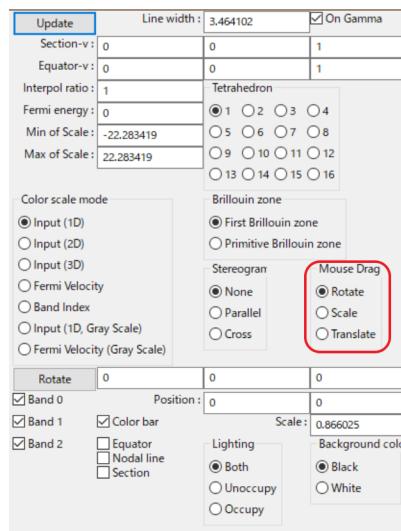


図 8: "Lighting"メニューで光を当てる Fermi 面を変更する。



5.16 ノーダルライン

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 9).

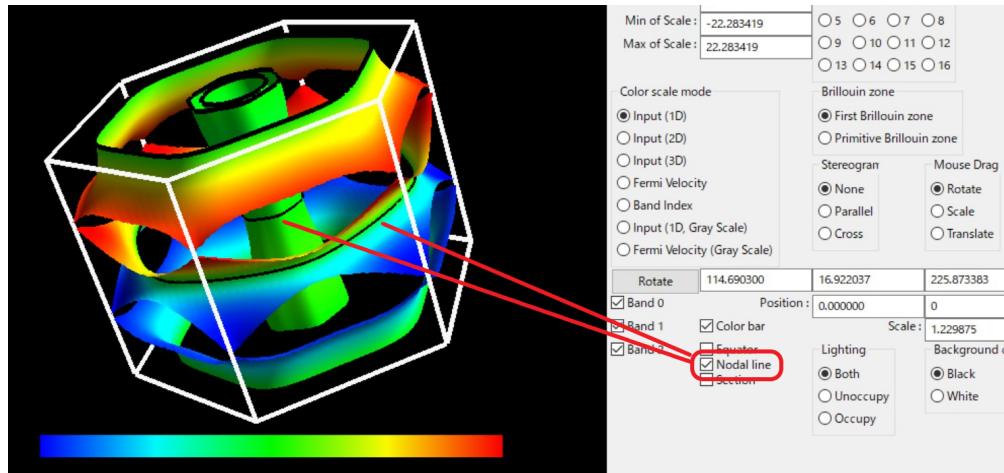


図 9: "Nodal line" メニューで nodal line の表示/非表示を切り替える.

5.17 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 ??).

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).

Section-v: 断面の法線の指定する. 法線ベクトルはフラクショナル座標で指定する. 変更を反映するには Update を押す。

On Gamma: こ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. 変更を反映するには Update を押す。

Section (RGB): ブリルアンゾーンの切断面を示す平面の色を指定する.

5.18 ブリルアンゾーン断面のファイル出力

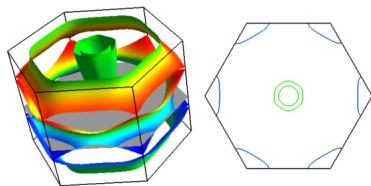
Section file: このボタンを押すと, 上記フェルミ面およびブリルアンゾーンの断面を Gnuplot や Igol などで図示するためのファイル("fermi_line.dat" と "bz_line.dat")を出力する.

gnuplot では次のように使うことができる.

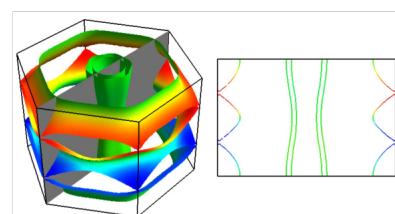
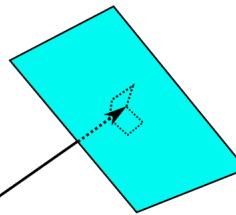
```
plot "fermi_line.dat" w l, "bz_line.dat" w l
```

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
Min of Scale :	6.921616	<input type="radio"/> 4	<input type="radio"/> 5	<input type="radio"/> 6
Max of Scale :	16.480536	<input type="radio"/> 7	<input type="radio"/> 8	<input type="radio"/> 9
Color scale mode				
<input type="radio"/> Input (1D)				
<input type="radio"/> Input (2D)				
<input type="radio"/> Input (3D)				
<input checked="" type="radio"/> Fermi Velocity				
<input type="radio"/> Band Index				
<input type="radio"/> Input (1D, Gray Scale)				
<input type="radio"/> Fermi Velocity (Gray Scale)				
Rotate	309.276184	-70.846390	291.730133	
<input checked="" type="checkbox"/> Band 0	Position :	0	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	<input type="checkbox"/> Nodal line	<input type="checkbox"/> Section	<input type="checkbox"/> Section file
Lighting				
<input checked="" type="radio"/> Both	<input type="radio"/> Unoccupy	<input type="radio"/> Occupy	Background colr	
<input type="radio"/> Black	<input checked="" type="radio"/> White			

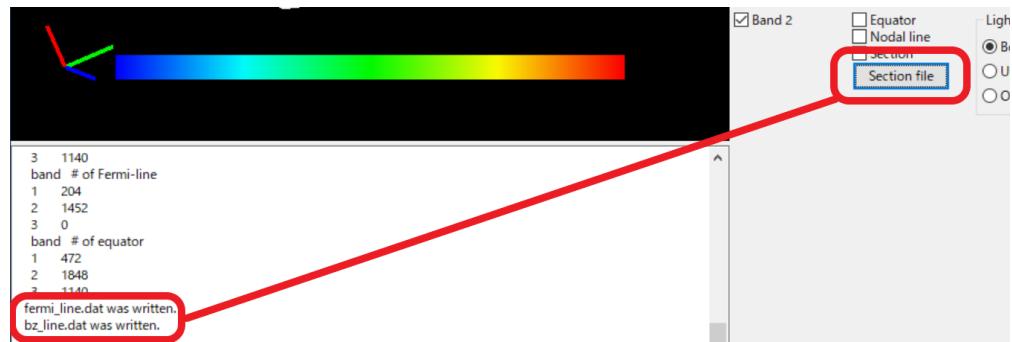
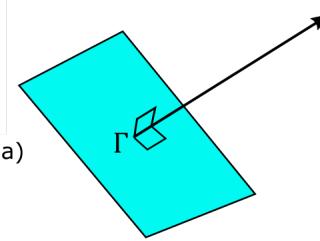
Section-v : 0.0 0.0 0.1



Γ



Section-v : 1.0 0.0 0.0 (On Gamma)



5.19 Fermi エネルギーの変更 (要 Update)

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ます. (図 10).

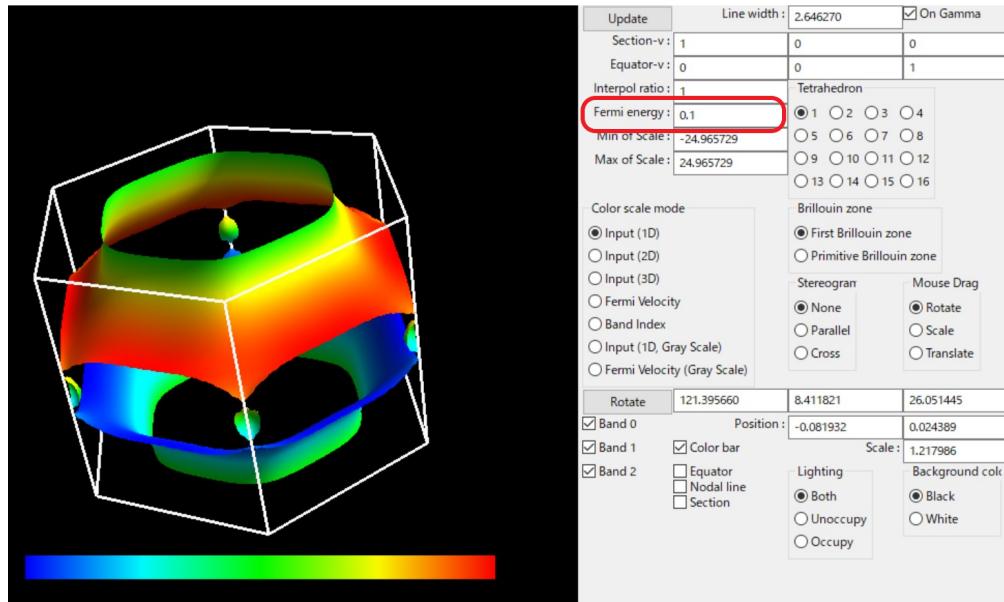


図 10: "Shift Fermi energy" メニューで Fermi エネルギーを 0.0 Ry から 0.1 Ry に変える.

5.20 立体視

Stereogram : 裸眼立体視用の図の表示/非表示を切り替えます (図 ??).

None (デフォルト) :

立体視を無効にします.

Parallel :

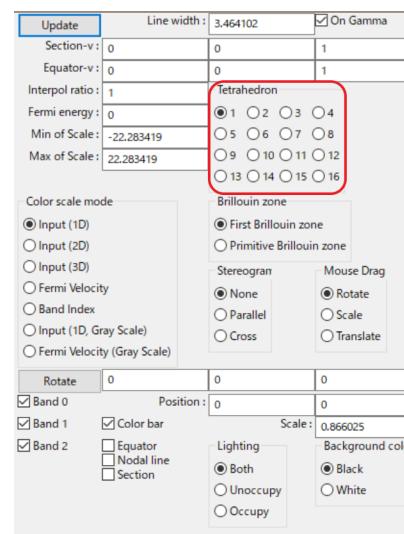
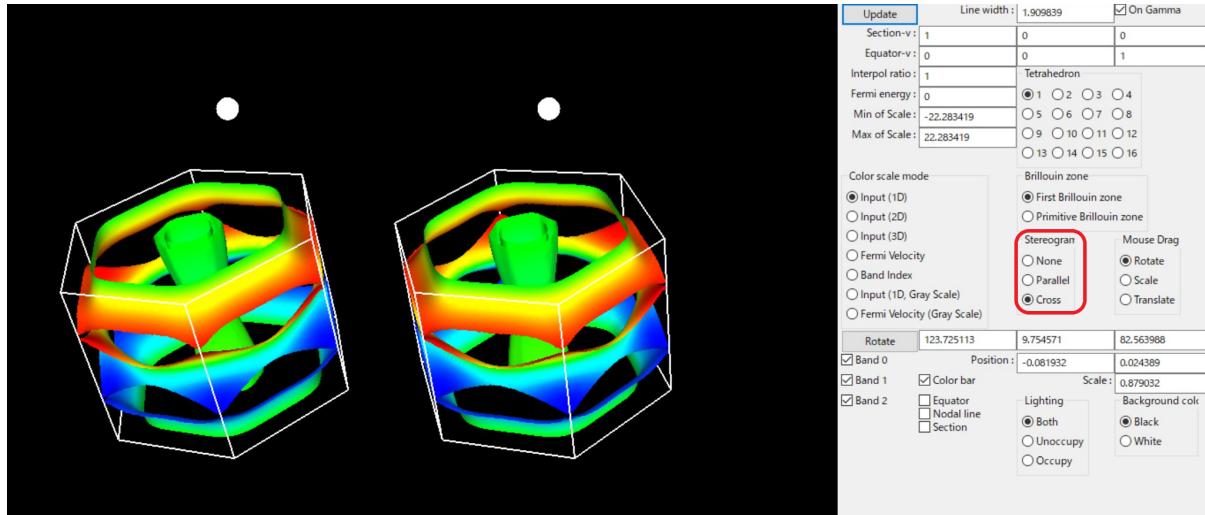
平行法用の図を表示します.

Cross :

交差法用の図を表示します.

5.21 四面体の切り方 (要 Update)

Tetraghedron : 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が、多くの場合は逆に図がギザギザして汚くなるようです.



5.22 サイズ・角度・位置の数値での調整

視点を変更します(図 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.

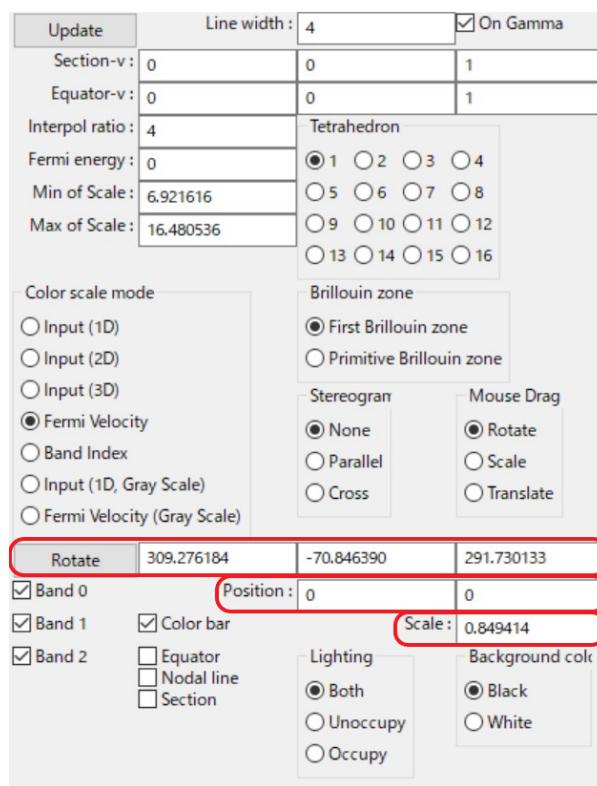


図 11: "View point" メニューで 視点を変更する.

5.23 矢印

任意の位置に矢印(実際には細長い三角形)を表示する. 以下は全てフランクショナル座標で指定する.

Arrow (Start) : 始点

Arrow (End) : 終点

Arrow (Diff) : 上記の差. Arrow (End) と Arrow (Diff) は連動する.

Arrow width : 矢印(三角形)の幅を指定する.

5.24 ワイヤーフレーム球

任意の位置に任意の大きさのワイヤーフレーム球を表示する。HiLAPWとの関連で使用する。

Sphere center : 球の中心をデカルト座標で指定する。

Sphere radius : 球の半径を上記と同じスケールで指定する。

5.25 ネスティング関数

次の二種類のネスティング関数を計算し, FermiSurfer で読み取り可能なファイルを出力する。

delta*delta : ファイル名は「doubledelta.frmsf」

$$\sum_{nn'k} \delta(\varepsilon_{n\mathbf{k}} - \varepsilon_F) \delta(\varepsilon_{n'\mathbf{k}+\mathbf{q}} - \varepsilon_F)$$

Lindhard : ファイル名は「lindhard.frmsf」

$$-\sum_{nn'k} \frac{\theta(\varepsilon_F - \varepsilon_{n'\mathbf{k}+\mathbf{q}}) - \theta(\varepsilon_F - \varepsilon_{n\mathbf{k}})}{\varepsilon_{n'\mathbf{k}+\mathbf{q}} - \varepsilon_{n\mathbf{k}}}$$

5.26 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。

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 output 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。

```
backgroundcolor 1 1 0
linecolor 0 0 1
    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	実数 実数 実数	0.0 0.0 0.0	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)
brillouin-zone	first, primitive	first	ブリルアンゾーンの種類
color-bar	0, 1	1	カラーバーの表示 (1) または非表示 (0)
colorscale	input1d, input2d, input3d, input1d fermivelocity, bandindex, inputgray, fermivelocitygray		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 (フランクショナル座標)
interpol	自然数	1	補間の細かさ
linewidth	実数	1.0	線幅
lighting	both, unoccupied, occupied	both	どちらの面に照光するか
nodalline	0, 1	0	ノーダルラインの表示 (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 (フランクショナル座標)
across-gamma	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 $|v_F|$ (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. MgB₂ 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 ,
```

(次のページに続く)

(前のページからの続き)

```

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](#) 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

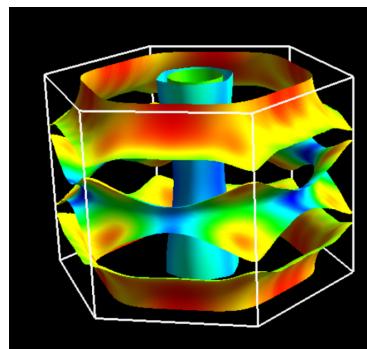
`fermi_velocity.x` プログラムを実行する。入力ファイルは `pw.x` のものと同一である。

```
$ 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) でなければならぬ。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)
```

(次のページに続く)

(前のページからの続き)

```
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

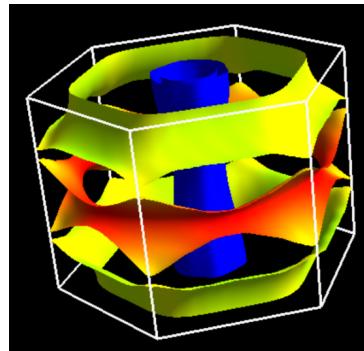
```
2
6 10
```

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、

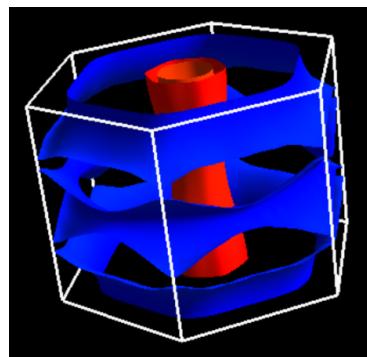
```
&PROJWFC
outdir = './'
prefix='mgb2'
Emin=-0.3422,
Emax=10.0578,
DeltaE=0.1
/
```

(次のページに続く)

(前のページからの続き)

[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 FermiSurfer on Web

下記のページにアクセスすると Web ブラウザ上で FermiSurfer を使うことができます。 <https://fermisurfer.osdn.jp/js/index.php>

8.1 操作方法

アプリ版 *FermiSurfer* の機能 と同等を目指していますが、一部の機能はまだサポートされていません。

8.2 ファイル入力

画面左上のファイル選択ボタンでローカルの FRMSF ファイル (BXSF ファイルはサポートしていません) を選択するとフェルミ面が表示されます。表示されるまでややタイムラグが生じる場合があります。

8.3 Web 上のファイルを開く

以下のように URL に引数をつけることにより、Web 上で公開されているフェルミ面の入力ファイルを、リンクをクリックするだけで開くことができます。

<https://fermisurfer.osdn.jp/js/index.php?frmsf=https://fermisurfer.osdn.jp/js/Pb.js>

<https://fermisurfer.osdn.jp/js/index.php?frmsf=> に続けてファイルの保存先の URL を付け足します。これは研究紹介やプレスリリースのページで使うとよいでしょう。

このときの入力ファイル Pb.js は、

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

のように、*FRMSF 形式* のファイルの改行をスペースに置き換えて 1 行の文字列にしたものを `frmsf` という変数に代入する javascript ソースになっています。 BXSF 形式には対応していません。

FRMSF 形式のファイルを上記の形式に変換するにはコマンドラインで

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

とします。これと同じ処理を行うシェルスクリプトファイルがこちらにあります。

<https://fermisurfer.osdn.jp/js/frmsf2js.sh>

使い方は下記のとおりです。 ANY.js (ANY の部分は任意) というファイルが生成されるので、それを Web サーバーに配置します。

```
$ bash frmsf2js.sh ANY.frmsf
```

第9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きいたことに感謝する。

第10章 プログラムの再配布

10.1 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。これはかいづまんで言うと、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ペ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、と言うことである。

10.2 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11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。

<https://github.com/mitsuaki1987/fermisurfer/issues>

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__ds.itc.u-tokyo.ac.jp`