実行例: Cu の電子状態

鳥取大学 吉本 芳英

平成 25 年 4 月 16 日

1 はじめに

この実行例では Cu の電子状態を計算し、バンド図、波動関数の k 点依存性、フェルミ面を描画する。なおこの配布パッケージには入力例と出力例が同梱されている。`sample/example-cu` を見ること。

2 入力データの用意

この節で作成する各セクションはまとめて cu.cg に収納する。

2.1 原子構造データの構成

fcc の対称性を指定する symmetry data セクションは

```
# symmetry data
&symmetry
symmetry_format = reciprocal,
number_sym_op = 24
/
    1  0  0    0  1  0    0  0  1    0  0  0
    0  1  0    0  0  1    1  0  0    0  0  0
    0  0  1    1  0  0    0  1  0    0  0  0
    1  0  0    0  0  1    0  1  0    0  0  0
    0  0  1    0  1  0    1  0  0    0  0  0
    0  1  0    1  0  0    0  0  1    0  0  0
   -1 -1 -1    0  1  0    0  0  1    0  0  0
   -1 -1 -1    0  0  1    0  1  0    0  0  0
   -1 -1 -1    1  0  0    0  0  1    0  0  0
   -1 -1 -1    0  0  1    1  0  0    0  0  0
   -1 -1 -1    1  0  0    0  1  0    0  0  0
   -1 -1 -1    0  1  0    1  0  0    0  0  0
    0  1  0   -1 -1 -1    0  0  1    0  0  0
    0  0  1   -1 -1 -1    0  1  0    0  0  0
    1  0  0   -1 -1 -1    0  0  1    0  0  0
    0  0  1   -1 -1 -1    1  0  0    0  0  0
```

```
  1  0  0   -1 -1 -1    0  1  0     0  0  0
  0  1  0   -1 -1 -1    1  0  0     0  0  0
  0  1  0    0  0  1   -1 -1 -1     0  0  0
  0  0  1    0  1  0   -1 -1 -1     0  0  0
  1  0  0    0  0  1   -1 -1 -1     0  0  0
  0  0  1    1  0  0   -1 -1 -1     0  0  0
  1  0  0    0  1  0   -1 -1 -1     0  0  0
  0  1  0    1  0  0   -1 -1 -1     0  0  0
```

となる。次に原子の位置情報などを指定する atom data セクションであるが、使用する擬ポテンシャルは 11 価の銅原子である。原子の位置は原点であるから、

```
# atom data
 11  29
 1   0.0  0.0  0.0
```

となる。

2.2 サンプル k 点データの構成

$16 \times 16 \times 16$ のサンプル k 点を構成する。Γ 点を通らないメッシュを作る。バンドの分散関係 $\epsilon_n(k)$ の cos 展開を $32 \times 32 \times 32$ のメッシュで行う。

バンド図の計算だけなら、このように濃いメッシュは必要ないがフェルミ面の描画をするために濃いメッシュを使う。

サンプル k 点関連の情報を指定する k-points data セクションは

```
# k-points data
&smpl_kpt
dos_mode = COS,
dos_mesh = 32, 32, 32,
bz_mesh = 32,
bz_number_tile = 1
/
    17    17    17
     2     2     2
```

となる。

2.3 計算条件のセットアップ

平面波基底のカットオフエネルギーを 49Ry に取るには、波数を 7 に取れば良い。また、スピンなしの計算を行うこととする。その他の条件を、

- 格子定数 $a = 6.90772$ bohr にとる。
- 元素の数は 1 個。原子数は全部で 1 個。

- 価電子帯に必要なバンドを 12 本とする。
- SCF は 40 回も回せばこの場合収束するはずである。
- 交換相関汎関数に PBE を使う。
- 多めに計算の打ち切り時間を 7200 秒にセット。
- 終了時に波動関数を書き出しておく。
- 初期電荷を原子の電荷データから作成する。

とすると、main data セクションは

```
# main data
&tappinput
lattice_factor = 6.90772,
lattice_list = 0.5,  0.5,  0.0,
               0.5,  0.0,  0.5,
               0.0,  0.5,  0.5,
cutoff_wave_function = 7.0,
number_element = 1,
number_atom = 1,
number_band = 12,
store_wfn = 1,
initial_lpt = 2,
scf_number_iter_1st = 40,
scf_number_iter = 40,
xc_type = PBE,
control_uptime = 7200.0
/
```

とする。

構造最適化は行わないので、struct_opt data セクションを以下のようにセットする。

```
# struct_opt data
&struct_opt
number_cycle = 0
/
```

これで、与えた構造についての電子状態計算のみが実行される。

構造最適化のための制約条件は特に必要なく、すべてデフォルトの制約しないでよいから、str_opt_constr data セクションに

```
# str_opt_constr data
 1
 0
```

としておく。

3　構造最適化プログラムの実行

擬ポテンシャルは xTAPP-test に付属の ps-Cu-pbe を用いる。これに対応する原子電荷のデータは ps-Cu-pbe.ichr である。これらは PS ディレクトリ以下にある。

まず、初期化プログラム `inipot` を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。同様な構造で一度動かせば良く、毎回実行する必要はない。

`inipot` では擬ポテンシャルを 1 番目から順に論理機番 $34, 35, \cdots$ に設定する必要があり、また、原子電荷のデータは 1 番目から順に論理機番 $28, 29, \cdots$ に設定する必要がある。そして作った入力ファイルは論理機番 10 に設定する必要があるので sh 系での入力コマンドは

```
$ export FORT10=./cu.cg FORT34=./ps-Cu-pbe FORT28=./ps-Cu-pbe.ichr
$ mpirun ./inipot > inipot.log
```

である。このコマンドで 1 MPI process をつかって `inipot` が動く。ログは inipot.log に回収している。実行が正常に終わるとカレントディレクトリにいくつかの中間ファイルが作成される。これら中間ファイルは Fortran の実装によって異なる名前がついているはずであるが同一の Fortran の実装を全部のプログラムで使っているのであれば問題ない。

次に構造最適化を行うプログラム `cgmrpt` を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。mpirun に渡すランクファイルを 6corefproc.rkf としておく。これの例は xTAPP-test にあるが本質的に実行環境依存である。例えばバッチシステムが動いている場合には必要ないはずである。

また実行においては波動関数、ローカルポテンシャル、電荷密度、結果のサマリをそれぞれ

```
cu.wfn
cu.lpt
cu.rho
cu.str
```

に保存するものとする。

2 MPI process で計算を実行させる入力コマンドは

```
$ export FORT10=./cu.cg FORT11=./cu.lpt FORT25=./cu.rho
$ export FORT96=./cu.wfn FORT99=./cu.str
$ mpirun -np 2 -rf 6corefproc.rkf ./cgmrpt > cgmrpt.log
```

である。このコマンドで 2 MPI process をつかって `cgmrpt` が動く。ログは cgmrpt.log に回収している。

3.1　フェルミ面を描画する

波動関数ファイルにある軌道エネルギーからフェルミ面を描画するためのツールが xTAPP-util/fldtool にある wfn2ee である。

このツールでは、拡張ゾーンに拡大して描画できる。立方体の拡張ゾーンに拡張し、そのゾーンを $32 \times 32 \times 32$ で分割して、ゾーンの中心をちょうど Gamma 点になるようにすると、wfn2ee の使用方法は

```
$ wfn2ee cu '1 1 0' '1 0 1' '0 1 1' '-16 -16 -16' '32 32 32'
```

である。これでカレントディレクトリにある cu.wfn が処理されて、そこから OpenDX へ渡せるデータファイルが作成される。ファイル名は、`cu.([0-9]+).ee.dx` の形をとっており、() 内はバンドインデックスである。

これらのファイルにはそれぞれのバンドの軌道エネルギーの k 点依存性が入っているから、これらのデータの isosurface をフェルミエネルギーにおいて描画すればそれはフェルミ面である。

なお、フェルミ面は 6 番目のバンドにある。フェルミエネルギーは cu.str に保存されている。また描画を行うための OpenDX のネットワーク例が xTAPP-test/opendx にある plotee.net である。

4 バンド図と波動関数の k 点依存性を描く

4.1 トレースする k 点の決定

バンド図を描くために必要なトレースする特殊 k 点の位置を、逆格子を単位に

1. G (1,0,0)
2. X (1, 1/2, 1/2)
3. K (3/4, 3/8, 3/8)
4. G (0,0,0)
5. L (1/2, 1/2, 1/2)
6. K (3/4, 3/8, 3/8)
7. W (3/4, 1/4, 1/2)
8. X (1/2, 0, 1/2)

と設定し、それぞれの間の分点を

1. GX 10
2. XK 5
3. KG 10
4. GL 10
5. LK 10
6. KW 5
7. WX 5

と設定する。最初の Γ 点, X 点の位置が第一ブリュアンゾーンの中にないのは追跡する k 点を一筆書きにするためである。

4.2 入力データの準備

次にバンドを計算するプログラム vbpef が必要とするセクション trace band data を作成し、構造最適化に使った入力データに付け加える。

作成するデータは 7 区間分であるから

```
# trace band data
 &trace_band
 distrib_mode = none,
 output_wave_function = 1,
 number_trace_block = 7
 /
    G     X     K       G     L     K       W      X
   1.0   1.0   0.750   0.0   0.5   0.750   0.75   0.5
   0.0   0.5   0.375   0.0   0.5   0.375   0.25   0.0
   0.0   0.5   0.375   0.0   0.5   0.375   0.50   0.5
      10     5      10    10    10       5      5
```

である。各列に k 点のシンボルと位置を指定している。最後の列は区間毎の分割数である。これをもとの cu.cg に付け加えて cu.pef としておく。

次に main data セクションにおいて、namelist の&tappinput の initial_lpt を 1 に変更して、vbpef が構造最適化時に計算済みのローカルポテンシャルを読み込むように設定しておく。また store_wfn を 0 に変更しておく。

4.3 バンド計算プログラムの実行

擬ポテンシャルは xTAPP-test に付属の ps-Cu-pbe を用いる。これは PS ディレクトリ以下にある。

初期化プログラム inipot を動かす必要はない。同様な構造で一度動かせば良いからである。

バンド計算プログラム vbpef を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。このプログラムは収束済みのローカルポテンシャルを必要とするので注意すること。

vbpef の追跡した k 点の固有値データを cu.band に、その k 点の波動関数を cu.wfk にそれぞれ格納させることにすると、sh 系での入力コマンドは

```
$ export FORT10=./cu.pef FORT50=./cu.band FORT58=./cu.wfk
$ mpirun -np 2 -rf 6corefproc.rkf ./vbpef > vbpef.log
```

である。このコマンドで 2 MPI process をつかって vbpef が動く。ログは vbpef.log に回収している。

4.4 バンド図を作成する

諏訪氏が開発したものを多少改変したバンド図作成ツール vbpef2gp-lsda によって、追跡した k 点の固有値データからバンド図を描くことができる。Cu の場合、フェルミエネルギーを 0 に

とってプロットしたいので、cu.str にあるフェルミエネルギーを読み取って eV 単位に書き直したものを [EF] とすると、vbpef2gp-lsda の使用方法は

```
$ vbpef2gp-lsda -fcu -e[EF] ./cu.band
```

である。これでカレントディレクトリにある cu.band が処理されて cu.gp、cu.dat、cu.kpt が作成される。次に gnuplot で cu.gp を処理するとバンド図が gnuplot で作成される。

4.5 波動関数を描画する

xTAPP-util/fldtool にある wfk2dx によって cu.wfk にある追跡した k 点の波動関数（厳密には擬波動関数）を描画できる。

描画は基本セルではなく、立方体に拡張したセルで行う。また、cell periodic function, $u_{i,\mathbf{k}}(\mathbf{r})$, ではなくブロッホ関数, $\exp(i\mathbf{k}\cdot\mathbf{r})u_{i,\mathbf{k}}(\mathbf{r})$, の方を出力させる。そして、この拡張したセルを 50,50,50 の実空間メッシュで切ることにする。また、波動関数の位相はできるだけ実にとる。

wfk2dx の使用方法は

```
$ wfk2dx -p -r cu.wfk '1 1 -1' '1 -1 1' '-1 1 1' '0 0 0' '50 50 50' > cu.wfk.log
```

である。これでカレントディレクトリにある cu.wfk が処理されて、そこから OpenDX へ渡せるデータファイルが作成される。ファイル名は、cu.wfk.([0-9]+).([0-9]+).dx の形をとっており、1 番目の () 内はバンドインデックス、2 番目の () 内は k 点の番号である。このファイルの中の軌道エネルギーは eV 単位である。

xTAPP-test にはこれらの OpenDX 用ファイルを可視化するためのネットワークの例が opendx/plotwfk.net として含まれている。

描画においては、ブロッホ関数を描画しているため、セルの中で波動関数は周期的ではないことに注意すること。周期的にしたければ k 点の分割と合致するセルが必要であるが、それは大抵大きなセルである。