

実空間差分法に基づく
第一原理計算プログラム RSPACE の使い方
(伝導特性計算編)

令和 4 年 12 月 28 日 第 3 版作成
令和 5 年 4 月 7 日 第 4 版作成
令和 5 年 5 月 2 日 第 5 版作成

神戸大学大学院工学研究科

1 はじめに

本手引書は，神戸大学大学院工学研究科小野研究室で開発した「実空間差分法に基づく第一原理計算プログラム RSPACE」の基本的な使い方をまとめたものである．本書で取り上げている，例題の計算はすべて Intel Fortran Compiler と Sun Grid Engine をインストールした PC クラスタで行うことを前提に記述してある．1.1 節に挙げる他の計算機で計算を行う場合，または 1.1 節に載っていない計算機で計算を行う場合は 1.2 節の連絡先まで連絡のこと．

1.1 動作確認機種

本プログラムは以下のマシンに関して動作確認を行っている．

- 富岳コンピュータ
- 東京大学物性研究所 ohtaka

1.2 連絡先

本プログラムの不具合，及び本手引書の誤植に関する連絡先

〒 657-8501 神戸市灘区六甲台町 1-1
神戸大学大学院工学研究科電気電子工学専攻
小野倫也
e-mail:t.ono@eedept.kobe-u.ac.jp

1.3 使用条件

RSPACE はフリーのソフトウェアであり，Apache License, Version 2.0 の使用条件に基づいて配布されています．使用条件の全文は，<https://www.apache.org/licenses/LICENSE-2.0.txt> もしくはフォルダ内の LICENSE.TXT を参照してください．

1.4 参考文献

RSPACE の伝導特性計算法を学術論文等で引用する際は，下記の論文を引用して下さい．

- K. Hirose, T. Ono, Y. Fujimoto, and S. Tsukamoto, *First-Principles Calculations in Real-Space Formalism, Electronic Configurations and Transport Properties of Nanostructures* (Imperial College, London, 2005).
 - T. Ono, Y. Egami, K. Hirose, Phys. Rev. B **86**, 195406 (2012).
- また，NCPS97 シリーズのノルム保存型擬ポテンシャルを用いた場合は，下記の論文も引用して下さい．
- K. Kobayashi, Comput. Mater. Sci. **14**, 72 (1999).

2 ソースプログラムのコンパイル

本節では、ソースファイルの入手方法とコンパイル方法、および使用する擬ポテンシャルについて説明する。

2.1 ソースプログラムのコピー

<https://github.com/onotmy/RSPACEdist> からダウンロードした RSPACEtr ディレクトリもしくは講習会で配布された RSPACEtr.zip を計算機上の適当なフォルダに展開する。展開したディレクトリの中に kukan, inverse, genbloch, negf_sclapck と NCPS というディレクトリが作成される。kukan, inverse, genbloch, negf_sclapck はソースファイルが格納されたディレクトリ、NCPS は練習用の擬ポテンシャルが格納されたディレクトリである。

2.2 makefile の設定

ソースプログラムをコンパイルするにあたり、ディレクトリ kukan, inverse, genbloch, negf_sclapck の中の makefile を変更する。展開したフォルダの中に makefile がある。初期値は、Intel Fortran Compiler をインストールした PC クラスター用に設定しており、makefile の 4 行目が

```
system = xeon
```

になっている。Intel Fortran Compiler をインストールした PC クラスターを使うのであれば、このまま変更する必要はない。1.1 で紹介した他のコンピュータを使うには、表 1 のように設定する。

表 1: makefile の設定。ohtaka は Intel コンパイラを使用しているので PC クラスターと同じコンパイルオプションで良い。

計算機	オプション
富岳コンピュータ	kei
Intel コンパイラを装備した PC クラスター	xeon
東京大学物性研究所 ohtaka	xeon

2.3 コンパイル

各ディレクトリの下で、

```
make
```

と入力する。コンパイル後、kukan, inverse, genbloch, negf というファイルが作成されていることを確認する。

2.4 擬ポテンシャルについて

伝導特性計算にはノルム保存型擬ポテンシャルを使う必要がある．伝導特性計算用の擬ポテンシャルは，NCPS というディレクトリに格納されている．paw.**の 1 行目が `# ncps pseudopotential` ... となっていることを確認すること．なお，ノルム保存型擬ポテンシャルであってもファイル名は paw.** である．また，`# ncps pseudopotential converted by ncps_convert` となっている擬ポテンシャルは，物質・材料研究機構 小林一昭氏の作製した擬ポテンシャルデータベース NCPS97 を RSPACE 用にフォーマットを改編したものである．擬ポテンシャルのデータベース NCPS シリーズについては，

<https://www.nims.go.jp/emsc/staff/kobayak/NCPS/ncps2kplusE.html> を参照のこと．ただし，現時点において NCPS97 を改編したポテンシャルは RSPACE の配布版に含まれていない．必要な場合は，1.2 節の連絡先に相談のこと．

3 伝導特性計算の概要

伝導特性計算のモデルは，図 1 に示すように左右電極領域と散乱領域の 3 つの部分からなる．図 1 に黒色で示す座標軸は全系の座標軸である．左側電極領域のポテンシャルを計算するには青色の，右側電極領域のポテンシャルを計算するには緑色の，座標軸とスーパーセルを用いて電子状態を計算する．散乱領域のポテンシャルを計算するには，赤色の座標軸とスーパーセルを用いて計算する．3 つの領域を図 1 のように合体させたときに，接続部のポテンシャルが滑らかに接続されるよう注意を払うこと．また，電極領域と散乱領域の接合面を擬ポテンシャルの非局所領域が跨がないようにモデルを作成すること．図 2 に左側電極領域と散乱領域の接合

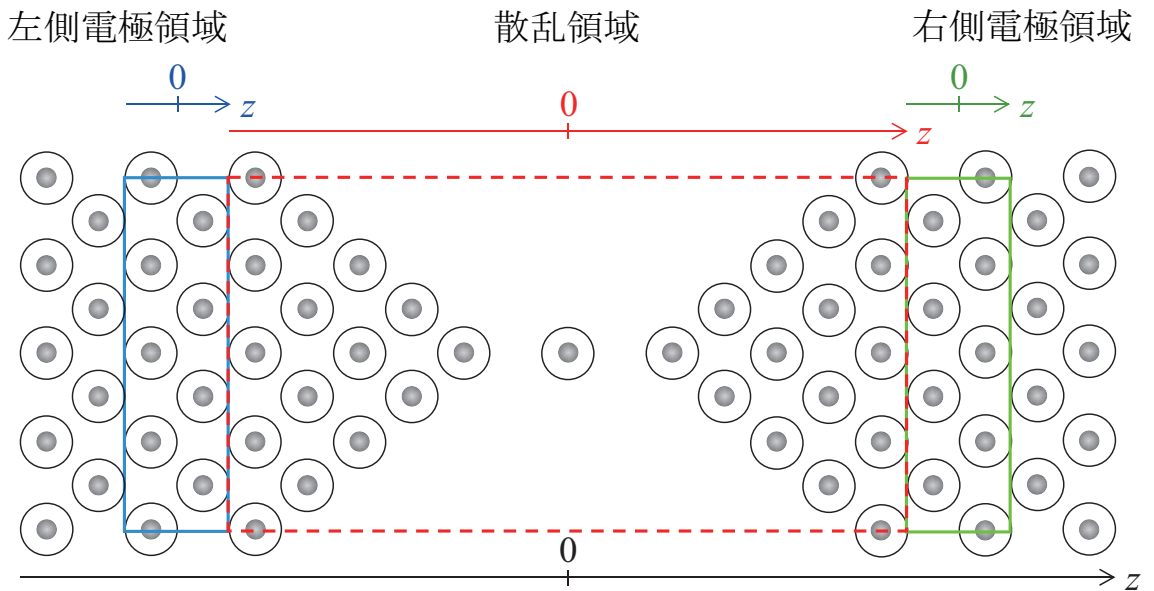


図 1: 伝導特性計算のモデル．青線は左側電極領域計算用の座標軸とスーパーセル．緑線は右側電極領域計算用の座標軸とスーパーセル．赤線は散乱領域計算用の座標軸とスーパーセル．灰色球は原子を表す．灰色球を囲む黒丸は，原子の擬ポテンシャルの非局所領域である．

界面の拡大図を示す．npzmax が擬ポテンシャルの非局所領域の半径に対応するグリッド数を表す．そのため，接合界面は原子から npzmax 分離した場所に設定する．

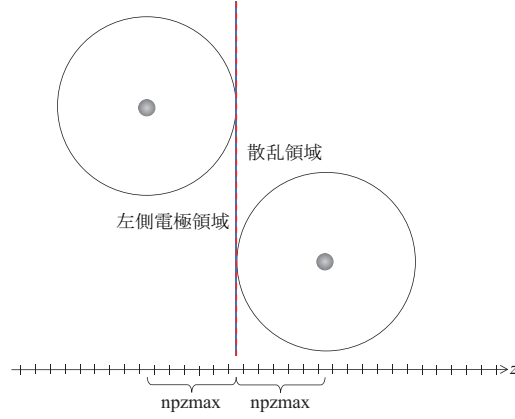


図 2: 左側電極領域と散乱領域の接合界面．

4 伝導特性計算の流れ

伝導計算は，1. 電極・散乱領域の電子状態計算，2. 電極・散乱領域の Green 関数計算，3. 電極領域の一般化 Bloch 波計算，4. 散乱波動関数計算 (透過特性計算) の手順で行われる．ここでは，C 原子単原子鎖の伝導特性を調べる．まず 5 章では，図 3 のように電極と散乱領域が全く同じモデルの伝導特性を計算する．次に 6 章では，5 章で用いた構造を電極として，図 4 のように散乱領域の C 原子の位置を少し変異させた，電極領域と散乱領域が異なるモデルの伝導特性を計算する．

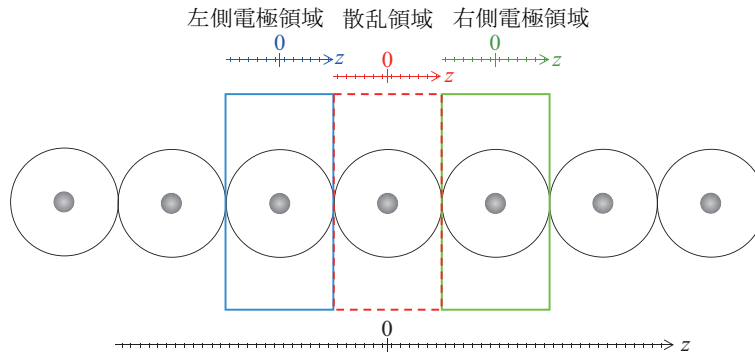


図 3: 電極と散乱領域の構造が同じモデル．灰色球は C 原子である． z 軸上の短い線は，grid point を表す．その他の symbol は，図 1 と同じである．

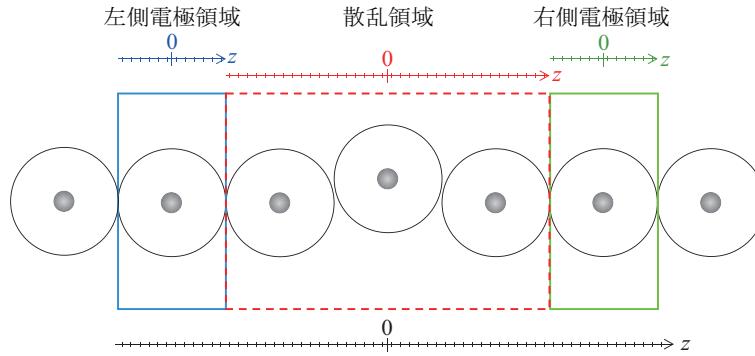


図 4: 電極と散乱領域の構造が異なるモデル. 図中の symbol は, 図 3 と同じである.

5 電極領域と散乱領域が同一の場合の伝導特性計算

5.1 電極 (散乱) 領域の電子状態計算

電子状態計算編を参考に, 電極領域用計算ディレクトリ `electrode` を作成し, その下に `kukan8` を `~/RSPACE/kukan` から実行ディレクトリにコピーする. また, 計算条件設定ファイル `parameters.inp`, 原子座標ファイル `atom.xyz` を下記のように作成する. 図 3 のように, 散乱領域と電極領域のそれぞれに座標の原点があることに注意すること.

`parameters.inp` ファイルの作成例

```
1: &nml_inp.prm_kukan
2: nprocx = 2
3: nprocy = 2
4: nprocz = 2
5: xmax = 5.2d0
6: ymax = 5.2d0
7: zmax = 1.4d0
8: nxmax = 8
9: nymax = 8
10: nzmax = 4
11: npxmax = 4
12: npymax = 4
13: npzmax = 4
14: neigmx = 4
15: natom = 1
16: lveffout = .true.
17: /
```

`atom.xyz` ファイルの作成例 (初めの 2 行のみ)

```
1: ! [x], [y], [z], [atom number], switch [x], [y], [z], [weight], switches [soc], [pp], [na]
2: 0.0000000000000000D+00 0.0000000000000000D+00 0.0000000000000000D+00 6 0 0 0 22056.00 00 1 1a
```

さらに, ディレクトリ `electrode` の下に, ディレクトリ `pspaw` を作成し, その下に C 原子用のノルム保存型擬ポテンシャルをにコピーする.

実行スクリプト `job.sh` を下記のように作成する.

`job.sh` ファイルの作成例

```
1: #!/bin/bash
2: ## -S /bin/bash
3: ## -pe mpi 64
4: ## -cwd
5: ## -V
6: export OMP_NUM_THREADS=8
7: i=0
8: for bbbb in $(cat $USER.$PE.$JOB_ID.2)
9: do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
```

```
15: mpiexec.hydra ./kukan8
```

その後、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。-q は、実行キューを指定するオプションである。投げるジョブに合ったキューを指定すること。

電子状態計算プログラムが終了すると、rho.0000*, vht.0000*, wfc.00000.00*, mdresult.dat, Potential.txt が作成される。*は 0~7 の数である。

```
cat mdresult.dat
```

と、入力すると計算結果を確認できる。以下に mdresult.dat の内容を示す。

mdresult.dat の出力例 (最後の 26 行のみ)

```
one electron energy      -1.98428247580907      (hartree)
exchange correlation energy 0.503128486008291      (hartree)
hartree energy          -3.87149552432593      (hartree)
ewald energy            -0.653017442586121      (hartree)
field correction         0.000000000000000E+000 (hartree)
B_con correction         0.000000000000000E+000 (hartree)
energy offset of ps. pot. 0.000000000000000E+000 (hartree)
energy jellium/ions       0.000000000000000E+000 (hartree)
helmholtz free energy     -6.00566695673548      (hartree)
Total energy             -6.00566695671283      (hartree)
atomic force
1 0.000000000000000D+00 0.000000000000000D+00 0.000000000000000D+00
mdstep= 1
atom coordinate (input)
0.000000000000000D+00 0.000000000000000D+00 0.000000000000000D+00 6 0 0 0 22056.00 11 0 1a
atom coordinate (output)
0.000000000000000D+00 0.000000000000000D+00 0.000000000000000D+00 6 0 0 0 22056.00 11 0 1a
real time 1.50830888748169 (sec)
com. time 0.363945007324219 (sec)
com. vol. 610557952.000000 (byte)
```

mdresult.dat の total energy と atom force が、上の内容と 10 桁程度一致していれば、計算は成功である。コンパイラによっても計算順序が異なるため、10 桁以降の部分については、上の結果と異なることがある。

5.2 電極 (散乱) 領域の Green 関数計算

inverse と gfcdir.sh を ~/RSPACE/inverse から電極領域計算用ディレクトリ electrode の下にコピーする。gfcdir.sh に

```
chmod +x gfcdir.sh
```

で実行特性を追加する。そして、5.1 節で作成した計算条件設定ファイル parameters.inp に下記を追記する。

parameters.inp ファイルの作成例 (追記部分のみ)

```
1: &nml_inp_prm_inv
2: nprocx = 2
3: nprocy = 2
4: nprocz = 2
5: nprocg = 1
6: nshift = 5
7: deshift = 0.00367485d0
8: /
```

ここで、nprocg は、Green 関数の列方向の並列 process 数であり、nprocx×nprocy×nprocz×nprocg が全 MPI process 数に対応する。nshift は、エネルギーのサンプル点数であり、index 0~nshift-1 に対し、nshift/2 (少数以下切り上げ) が Fermi level に対応する。deshift は、エネルギーメッシュ

の間隔である。また、実行スクリプト job.sh を下記のように作成する。

job.sh ファイルの作成例

```
1: #!/bin/bash
2: #SBATCH --S /bin/bash
3: #SBATCH --pe mpi 64
4: #SBATCH --cwd
5: #SBATCH --V
6: export OMP_NUM_THREADS=8
7: i=0
8: for bbbb in $(cat $USER.$PE.$JOB_ID.2)
9: do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./inverse > output_inverse.txt
```

上記の赤字は、5.1 節からの変更点である。最後に、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。

Green 関数計算プログラムが終了すると、ディレクトリ gfc11, gfc1r, gfc1l, gfc1r, gfc が作成され、その中に Green 関数の情報が保存される。

```
tail output_inverse.txt
```

と、入力すると以下の内容が表示される。

output_inverse.txt の出力例 (最後の 10 行のみ)

```
2045th Green function is converged. # of its      140 elapsed time    0.15705E-01(sec.) 2045/ 2048
2046th Green function is converged. # of its      140 elapsed time    0.15720E-01(sec.) 2046/ 2048
2047th Green function is converged. # of its      140 elapsed time    0.15687E-01(sec.) 2047/ 2048
2048th Green function is converged. # of its      134 elapsed time    0.14973E-01(sec.) 2048/ 2048
computation of Green function has been completed before the elapsed time limit.
Green functions have been calculated.
Elapsed time= 0.3294042E+02 (sec)
program ended.
carried out by the version of 12/24/2021-01
Total elapsed time= 0.3362157E+02 (sec.)
```

下から 6 行目、computation of Green function has been completed before the elapsed time limit. と表示されていれば、計算は成功である。

5.3 電極領域の一般化 Bloch 波計算

genbloch と gbwdir.sh を ~/RSPACE/genbloch から電極領域計算用ディレクトリ electrode の下にコピーする。gbwdir.sh に

```
chmod +x gbwdir.sh
```

で実行特性を追加する。そして、5.2 節で作成した計算条件設定ファイル parameters.inp に下記を追記する。

parameters.inp ファイルの作成例 (追記部分のみ)

```
1: &nml_inp_prm_genbl
2: nprocx = 1
3: nprocy = 1
4: nprocz = 1
5: nprochrpt = 8
6: /
```

ここで、nprochrpt は、Sakurai-Sugiura 法の周回積分における積分点に関する並列 process 数である。nprocx×nprocy×nprocz×nprochrpt が全 MPI process 数に対応する。また、実行スクリプト job.sh を下記のように作成する。

job.sh ファイルの作成例

```
1: #!/bin/bash
2: #SBATCH --S /bin/bash
```

```

3:  $$ -pe mpi 64
4:  $$ -cwd
5:  $$ -V
6:  export OMP_NUM_THREADS=8
7:  i=0
8:  for bbbb in $(cat $USER.$PE.$JOB_ID.2)
9:  do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./genbloch > output_genbloch.txt

```

上記の赤字は、5.2 節からの変更点である。最後に、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。

一般化 Bloch 波計算プログラムが終了すると、ディレクトリ gbwdir が作成され、その中に電極の情報が保存される。

```
tail output_genbloch.txt
```

と、入力すると以下の内容が表示される。

```

output_genbloch.txt の出力例 (最後の 10 行のみ)
Now simultaneous equations are being solved.
BiCG ended.
Simultaneous equations have been solved.
BiCG 0.1112206E+01 (sec.)
BL and BR are being calculated.
BL and BR have been calculated. 0.4599609E+00 (sec.)
SS method (incl. check) 0.2045255E+02 (sec.)
all ratio matrices are converged.
carried out by the version of 12/24/2021-01
total elapsed time 0.2205640E+02 (sec.)

```

下から 3 行目、all ratio matrices are converged. と表示されていれば、計算は成功である。

5.4 散乱波動関数計算 (透過特性計算)

genbloch 実行後に電極の一般化 Bloch 波が格納されているディレクトリ gbwdir ができているので、gbwdir を左側電極用リンク gbwl, 右側電極用リンク gbwr としてリンクを張る。

```
ln -s gbwdir gbwl; ln -s gbwdir gbwr
```

negf を ~/RSPACE/negf.sclpck から電極領域計算用ディレクトリ electrode の下にコピーする。そして、5.3 節で作成した計算条件設定ファイル parameters.inp に下記を追記する。

parameters.inp ファイルの作成例 (追記部分のみ)

```

1: &nml_inp_prm_negf
2: /

```

また、実行スクリプト job.sh を下記のように作成する。

job.sh ファイルの作成例

```

1: #!/bin/bash
2:  $$ -S /bin/bash
3:  $$ -pe mpi 64
4:  $$ -cwd
5:  $$ -V
6:  export OMP_NUM_THREADS=8
7:  i=0
8:  for bbbb in $(cat $USER.$PE.$JOB_ID.2)
9:  do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./negf > output_negf.txt

```

上記の赤字は、5.3 節からの変更点である。最後に、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。

散乱波動関数計算プログラムが終了すると、output_negf.txt.000*が作成され、その中に透過率の情報が保存される。*は0~4の数で、エネルギー点のindexに対応する。

```
tail output_negf.txt
```

と、入力すると以下の内容が表示される。

output_negf.txt の出力例 (最後の 10 行のみ)

```
lcalch : T
*****
Computed by negf.f90 03/26/2021-01. 0.8265972E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.8724928E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.8724928E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.4714772E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.4787703E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.5115573E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.5655939E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.5758747E+01(sec.)
```

Computed by negf.f90 という表示が、process 数分表示されていれば、計算は成功である。

```
cat output_negf.txt.0000
```

と入力すれば、0 番目のエネルギー点の透過率の情報が表示される。15 行目の transmission の後ろの数字が透過率である。

```
Generalized Bloch waves of electrodes are being read.
Generalized Bloch waves have been read. 0.1027203E-01(sec.)
Ratio matrices are being read.
Ratio matrices have been read. 0.3162122E-01(sec.)
offdiagonal elements are being read.
offdiagonal elements have been read. 0.1991701E-01(sec.)
Green functions are being read.
Green functions have been read. 0.6935692E-01 (sec)
Self energies are being computed.
Self energies have been computed. 0.3151560E+00(sec.)
Green functions are being computed.
Green functions have been computed. 0.1639958E+01(sec.)
Conductance is being computed by Fisher Lee formula.
*****
transmission= 2.000000000000000 (by Fisher Lee)
*****
```

output_negf.txt.0000 の中で、transmission は3回表示されるが、初級演習では15行目の数字のみ参考にすればよい。本節の演習では、電極領域と散乱領域が全く同じ原子構造であるため、散乱領域で散乱は起こらない。したがって、透過率が整数になることは自明である。

6 電極領域と散乱領域が異なる場合の伝導特性計算

6.1 散乱領域の電子状態計算

ここでは、整然とC原子が並んだC原子鎖の原子1個を少し移動させ、散乱領域とする。電極領域とPotentialができるだけ滑らかにつながるように、移動させた原子の両横にbuffer原子を1個ずつ置いた計3原子からなる構造を、散乱領域とする。散乱領域計算用ディレクトリscatをelectrodeと同じレベルに作成する。その下にkukan8を~/RSPACE/kukanから実行ディレクトリにコピーする。また、計算条件設定ファイルparameters.inp, 原子座標ファイルatom.xyzを下記のように作成する。図4のように、散乱領域と電極領域のそれぞれに座標の原点があることに注意すること。

parameters.inp ファイルの作成例

```
1: &nml_inp_prm_kukan
2: nprocx = 2
3: nprocy = 2
4: nprocz = 2
5: xmax = 5.2d0
6: ymax = 5.2d0
7: zmax = 4.2d0
8: nxmax = 8
9: nymax = 8
10: nzmax = 12
11: npxmax = 4
12: npymax = 4
13: npzmax = 4
14: neigmx = 12
15: natom = 3
16: lveffout = .true.
17: /
```

atom.xyz ファイルの作成例 (初めの 4 行のみ)

```
1:! [x], [y], [z], [atom number], switch [x], [y], [z], [weight], switches [soc], [pp], [na]
2: 0.0000000000000000D+00 0.0000000000000000D+00 -2.8000000000000000D+00 6 0 0 0 22056.00 00 1 1a
3: 0.5000000000000000D+00 0.5000000000000000D+00 0.0000000000000000D+00 6 0 0 0 22056.00 00 1 2a
4: 0.0000000000000000D+00 0.0000000000000000D+00 2.8000000000000000D+00 6 0 0 0 22056.00 00 1 3a
```

さらに、ディレクトリ scat の下に、ディレクトリ pspaw を作成し、その下に C 原子用のノルム保存型擬ポテンシャルをコピーする。

実行スクリプト job.sh を下記のように作成する。

job.sh ファイルの作成例

```
1: #!/bin/bash
2: #$ -S /bin/bash
3: #$ -pe mpi 64
4: #$ -cwd
5: #$ -V
6: export OMP_NUM_THREADS=8
7: i=0
8: for bbbb in $(cat $USER.$PE.$JOB_ID.2)
9: do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./kukan8
```

その後、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。

電子状態計算プログラムが終了すると、rho.0000*, vht.0000*, wfc.00000.00*, mdresult.dat, Potential.txt が作成される。*は 0~7 の数である。

```
cat mdresult.dat
```

と、入力すると計算結果を確認できる。以下に mdresult.dat の内容を示す。

mdresult.dat の出力例 (最後の 30 行のみ)

```
one electron energy      -4.42126517940423      (hartree)
exchange correlation energy  1.43546009002026      (hartree)
hartree energy          -11.4133984518028      (hartree)
ewald energy            -2.30899188167952      (hartree)
field correction         0.000000000000000E+000 (hartree)
B_con correction         0.000000000000000E+000 (hartree)
energy offset of ps. pot. 0.000000000000000E+000 (hartree)
energy jellium/ions       0.000000000000000E+000 (hartree)
helmholtz free energy    -16.7174051015430      (hartree)
Total energy            -16.7081954228663      (hartree)
atomic force
 1 -0.5004831433636606D-01 -0.5004831424587112D-01 -0.3428957091978465D-01
 2  0.1001011372392719D+00  0.1001011370422020D+00  0.1928804633492357D-04
 3 -0.5005282290290586D-01 -0.5005282279633087D-01  0.3427028287344972D-01
mdstep= 1
atom coordinate (input)
```

```

0.0000000000000000D+00 0.0000000000000000D+00 -0.2800000000000000D+01 6 0 0 0 22056.00 11 0 1a
0.5000000000000000D+00 0.5000000000000000D+00 0.0000000000000000D+00 6 0 0 0 22056.00 11 0 2a
0.0000000000000000D+00 0.0000000000000000D+00 0.2800000000000000D+01 6 0 0 0 22056.00 11 0 3a
atom coordinate (output)
0.0000000000000000D+00 0.0000000000000000D+00 -0.2800000000000000D+01 6 0 0 0 22056.00 11 0 1a
0.5000000000000000D+00 0.5000000000000000D+00 0.0000000000000000D+00 6 0 0 0 22056.00 11 0 2a
0.0000000000000000D+00 0.0000000000000000D+00 0.2800000000000000D+01 6 0 0 0 22056.00 11 0 3a
real time 17.5159118175507 (sec)
com. time 5.65586304664612 (sec)
com. vol. 10140364800.0000 (byte)

```

mdresult.dat の total energy と atom force が、上の内容と 10 桁程度一致していれば、計算は成功である。コンパイラによっても計算順序が異なるため、10 桁以降の部分については、上の結果と異なることがある。

6.2 散乱領域の Green 関数計算

inverse と gfcdir.sh を ~/RSPACE/inverse から散乱領域計算用ディレクトリ scat の下にコピーする。gfcdir.sh に

```
chmod +x gfcdir.sh
```

で実行特性を追加する。そして、6.1 節で作成した計算条件設定ファイル parameters.inp に下記を追記する。

parameters.inp ファイルの作成例 (追記部分のみ)

```

1: &nml_inp_prm_inv
2: nprocx = 2
3: nprocy = 2
4: nprocz = 2
5: nprocg = 1
6: nshift = 5
7: deshift = 0.00367485d0
8: /

```

エネルギーのサンプル点数 nshift とエネルギーメッシュ間隔 deshift は、電極を計算したものと一致させておく。また、実行スクリプト job.sh を下記のように作成する。

job.sh ファイルの作成例

```

1: #!/bin/bash
2: #$ -S /bin/bash
3: #$ -pe mpi 64
4: #$ -cwd
5: #$ -V
6: export OMP_NUM_THREADS=8
7: i=0
8: for bbbb in $(cat $USER.$PE.$JOB.ID.2)
9: do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./inverse > output_inverse.txt

```

上記の赤字は、6.1 節からの変更点である。最後に、実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる。

Green 関数計算プログラムが終了すると、ディレクトリ gfccl, gfcclr, gfcrl, gfcrr, gfc が作成され、その中に Green 関数の情報が保存される。

```
tail output_inverse.txt
```

と、入力すると以下の内容が表示される。

output_inverse.txt の出力例 (最後の 10 行のみ)

```

2047th Green function. Seed is switched to 3 at itr 384
2047th Green function is converged. # of its 435 elapsed time 0.62642E-01(sec.) 2047/ 2048

```

```

2048th Green function. Seed is switched to 3 at itr 358
2048th Green function is converged. # of its 375 elapsed time 0.55043E-01(sec.) 2048/ 2048
computation of Green function has been completed before the elapsed time limit.
Green functions have been calculated.
Elapsed time= 0.1555100E+03 (sec)
program ended.
carried out by the version of 12/24/2021-01
Total elapsed time= 0.1562588E+03 (sec.)

```

下から6行目, computation of Green function has been completed before the elapsed time limit. と表示されていれば, 計算は成功である.

6.3 散乱波動関数計算 (透過特性計算)

散乱領域に対しては, 一般化 Bloch 波を計算する必要はない. 電極の一般化 Bloch 波は, すでに 5.3 節で計算を終えたものがあるので, electrode の下の gbwdir を左側電極用リンク gbwl, 右側電極用リンク gbwr としてリンクを張る.

```
ln -s ../electrode/gbwdir gbwl; ln -s ../electrode/gbwdir gbwr
```

negf を ~/RSPACE/negf_sclpcck から散乱領域計算用ディレクトリ scat の下にコピーする. そして, 6.2 節で作成した計算条件設定ファイル parameters.inp に下記を追記する.

```

parameters.inp ファイルの作成例 (追記部分のみ)
1: &nml_inp_prm_negf
2: /

```

また, 実行スクリプト job.sh を下記のように作成する.

```

job.sh ファイルの作成例
1: #!/bin/bash
2: # $ -S /bin/bash
3: # $ -pe mpi 64
4: # $ -cwd
5: # $ -V
6: export OMP_NUM_THREADS=8
7: i=0
8: for bbbb in $(cat $USER.$PE.$JOB.ID.2)
9: do
10: i=$(( i+1 ))
11: done
12: export I_MPI_PERHOST=$(( NSLOTS/OMP_NUM_THREADS/i ))
13: export I_MPI_HYDRA_HOST_FILE=hostfile
14: cut $PE_HOSTFILE -d " " -f 1 > hostfile
15: mpiexec.hydra ./negf

```

上記の赤字は, 6.2 節からの変更点である. 最後に, 実行スクリプト job.sh を

```
qsub -q mm job.sh
```

で投げる.

散乱波動関数計算プログラムが終了すると, output_negf.txt.000*が作成され, その中に透過率の情報が保存される. *は 0~4 の数で, エネルギー点の index に対応する.

```
tail output_negf.txt
```

と, 入力すると以下の内容が表示される.

```
output_negf.txt の出力例 (最後の 10 行のみ)
```

```

lcalch : T
*****
Computed by negf.f90 03/26/2021-01. 0.6978035E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.7871866E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.8123875E-02(sec.)
Computed by negf.f90 03/26/2021-01. 0.8749112E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.9594602E+01(sec.)
Computed by negf.f90 03/26/2021-01. 0.1020803E+02(sec.)
Computed by negf.f90 03/26/2021-01. 0.1038002E+02(sec.)
Computed by negf.f90 03/26/2021-01. 0.1057339E+02(sec.)

```

Computed by negf.f90 という表示が，process 数分表示されていれば，計算は成功である．

```
cat output_negf.txt.0000
```

と入力すれば，0 番目のエネルギー点の透過率の情報が表示される．15 行目の transmission の後ろの数字が透過率である．

```
output_negf.txt.0000 の出力例 (最初の 16 行のみ)
Generalized Bloch waves of electrodes are being read.
Generalized Bloch waves have been read.  0.2075791E-01(sec.)
Ratio matrices are being read.
Ratio matrices have been read.  0.2291341E+01(sec.)
offdiagonal elements are being read.
offdiagonal elements have been read.  0.1054244E+01(sec.)
Green functions are being read.
Green functions have been read.  0.1806390E+01 (sec)
Self energies are being computed.
Self energies have been computed.  0.3305230E+00(sec.)
Green functions are being computed.
Green functions have been computed.  0.1577935E+01(sec.)
Conductance is being computed by Fisher Lee formula.
*****
transmission=  1.88977404371895      (by Fisher Lee)
*****
```

output_negf.txt.0000 の中で，transmission は 3 回表示されるが，初級演習では 15 行目の数字のみ参考にすればよい．本節の演習では，電極領域と散乱領域が異なる原子構造であるため，散乱領域で散乱が起こる．したがって，透過率が 5.4 節の結果より小さくなる．