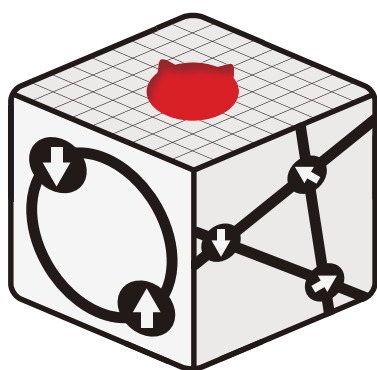


汎用多変数変分モンテカルロ mVMC
マニュアル ver. 1.0.3



mVMC

July 21, 2018

Contents

1	What is mVMC?	1
1.1	mVMC とは？	1
1.1.1	プログラム概要	1
1.1.2	ライセンス	2
1.1.3	コピーライト	2
1.1.4	開発貢献者	2
1.2	動作環境	3
2	How to use mVMC?	4
2.1	要件	4
2.2	インストール方法	4
2.2.1	mVMCconfig.sh を使う方法	5
2.2.2	cmake を使う方法	6
2.3	ディレクトリ構成	7
2.4	基本的な使い方	9
2.4.1	バージョン番号の確認	10
3	チュートリアル	11
3.1	サンプルファイル一覧	11
3.2	Heisenberg 模型	11
3.3	エキスパートユーザー向け	16
3.4	相関関数のフーリエ変換	17
4	ファイル仕様	18
4.1	スタンダードモード用入力ファイル	18
4.1.1	計算の種類に関する必須パラメーター	18
4.1.2	格子に関するパラメーター	19
4.1.3	副格子	23

4.1.4	ハミルトニアン of 各項の係数	23
4.1.5	計算条件のパラメーター	26
4.2	エキスパートモード入力ファイル	30
4.2.1	入力ファイル指定用ファイル (namelist.def) . . .	33
4.2.2	ModPara ファイル (modpara.def)	35
4.2.3	LocSpin 指定ファイル (locspn.def)	40
4.2.4	Trans 指定ファイル (trans.def)	42
4.2.5	InterAll 指定ファイル	44
4.2.6	CoulombIntra 指定ファイル (coulombintra.def) .	46
4.2.7	CoulombInter 指定ファイル (coulombiter.def) . .	48
4.2.8	Hund 指定ファイル (hund.def)	50
4.2.9	PairHop 指定ファイル	52
4.2.10	Exchange 指定ファイル (exchange.def)	54
4.2.11	Gutzwiller 指定ファイル (gutzwiller.def)	56
4.2.12	Jastrow 指定ファイル (jastrow.def)	58
4.2.13	DH2 指定ファイル	60
4.2.14	DH4 指定ファイル	62
4.2.15	Orbital/OrbitalAntiParallel 指定ファイル (orbital-idx.def)	64
4.2.16	OrbitalParallel 指定ファイル	66
4.2.17	OrbitalGeneral 指定ファイル	68
4.2.18	TransSym 指定ファイル (qptransidx.def)	71
4.2.19	変分パラメータ初期値指定ファイル	73
4.2.20	OneBodyG 指定ファイル (greenone.def)	75
4.2.21	TwoBodyG 指定ファイル (greentwo.def)	77
4.3	出力ファイル	79
4.3.1	変分パラメータ出力ファイル (***_opt.dat) . . .	79
4.3.2	ステップ別変分パラメータ出力ファイル (xxx_var_yyy.dat)	79
4.3.3	Gutzwiller 因子出力ファイル (***_gutzwiller_opt.dat)	79
4.3.4	Jastrow 因子出力ファイル (***_jastrow_opt.dat)	80
4.3.5	2 サイトダブロンホロン相関因子出力ファイル (***_doublonHolon2site_opt.dat)	80
4.3.6	4 サイトダブロンホロン相関因子出力ファイル (***_doublonHolon4site_opt.dat)	80
4.3.7	ペア軌道出力ファイル (***_orbital_opt.dat) . .	80
4.3.8	xxx_out_yyy.dat	80

4.3.9	xxx_CalcTimer.dat	80
4.3.10	xxx_time_zzz.dat	82
4.3.11	xxx_cisajs_yyy.dat	82
4.3.12	xxx_cisajscktalt_yyy.dat	83
4.3.13	xxx_ls_out_yyy.dat	84
4.3.14	xxx_ls_cisajs_yyy.dat	84
4.3.15	xxx_ls_cisajscktalt_yyy.dat	84
5	アルゴリズム	85
5.1	変分モンテカルロ法	85
5.2	Bogoliubov 表現	85
5.3	パフィアン-スレーター行列式の性質	86
5.3.1	f_{ij} と $\Phi_{in\sigma}$ の関係 (スピン反平行の場合)	86
5.3.2	F_{IJ} と $\Phi_{In\sigma}$ の関係 (スピン平行も含めた場合)	87
5.3.3	f_{ij} の特異値分解	88
5.4	Power-Lanczos 法	88
5.4.1	α の決定	88
5.4.2	物理量の計算	89
6	謝辞	91
A	非制限 Hartree-Fock 近似プログラム	92
A.1	概要	92
A.1.1	ソースコード	92
A.1.2	コンパイル方法	92
A.1.3	必要な入力ファイル	92
A.2	使用方法	94

1

What is mVMC?

1.1 mVMC とは？

量子多体系の理論模型の高精度解析は高温超伝導・量子スピン液体に代表される新奇量子相の発現機構を解明するうえで重要な役割を果たすことが期待できます。また、現実の物質を記述する有効模型を非経験的に導出する手法も近年発展しており [1]、その有効模型の高精度解析を行うことは、現実物質の物性を非経験的に解明して、さらに制御につなげるうえで重要なステップとなっています。有効模型解析で最も信頼できる手法は厳密対角化法であるものの、その適用できるサイズには強い制限があるのが大きな問題でした。厳密対角化法を超えたシステムサイズに対して高精度な計算が行える計算手法の一つとして、変分モンテカルロ法があります [2]。従来の変分モンテカルロ法では、使用する変分波動関数の強い制限による精度の低下が問題となっていました。近年の理論手法及び計算機の発展によって、変分モンテカルロ法で使用する波動関数の制限を大幅に緩和することが可能になっており、変分モンテカルロ法の計算精度は劇的に向上しています [3–5]。

この背景のもと、多変数変分モンテカルロ法 (many-variable variational Monte Carlo method [mVMC]) は簡便かつ柔軟なユーザー・インタフェースとともに大規模並列に対応したソフトウェアとして開発されました。ハバード模型・ハイゼンベルグ模型・近藤格子模型などの基本的な模型に対しては、ユーザーは 10 行程度の一つのファイルを用意するだけで容易に計算を実行することができます。また、同一のファイルを用いて、 $\mathcal{H}\Phi$ [6] による厳密対角化法の計算も実行できることから、ユーザーは小さなシステムサイズで計算精度を確認しながら、厳密対角化では到達できないシステムサイズの計算を行なうことができます。mVMC を実験研究者や量子化学の研究者などの分野を超えた幅広いユーザーの方にご利用頂ければ幸いです。

1.1.1 プログラム概要

このプログラムを利用することで以下の事項が計算可能です。

- 与えられた変分自由度の範囲でハミルトニアン の期待値が最小 (極小) 値を持つような変分波動関数 を数値的に生成します。量子数で分割された部分空間に限定して計算することも可能です。
- 得られた変分波動関数における各種物理量 (相関関数など) の期待値を計算することができます。

mVMC では以下の流れで計算を行います。

1. 入力ファイル (*.def) の読込

2. $\langle \mathcal{H} \rangle$ を最小化するように変分パラメータ α を最適化
3. 一体・二体 Green 関数の計算
4. 変分パラメータ・期待値の出力

計算では「実空間配置 $|x\rangle$ の生成からサンプリングまでを並列して行い、期待値を計算する際に一つにまとめる」という単純な並列化を行っています。各計算機クラスターで所定の手続きに従って、並列数を指定すれば MPI を用いた並列計算を勝手に行われますが、並列を行わない場合 (single core) も MPI を呼び出すため、MPI ジョブを禁止してる環境 (物性研 system B のフロントエンドなど) ではプログラム実行することができません。なお、本プログラムではパフィアンの計算にあたり PFAPACK を利用した計算を行っています [7]。

1.1.2 ライセンス

本ソフトウェアのプログラムパッケージおよびソースコード一式は GNU General Public License version 3 (GPL v3) に準じて配布されています。

mVMC を引用する際には、以下の URL を記載してください (mVMC に関する代表論文執筆後は、そちらへの引用に変更する予定です)。

URL: <https://github.com/issp-center-dev/mVMC>

1.1.3 コピーライト

©2016- The University of Tokyo. All rights reserved.

本ソフトウェアは 2016 年度 東京大学物性研究所 ソフトウェア高度化プロジェクトの支援を受け開発されています。

1.1.4 開発貢献者

本ソフトウェアは以下の開発貢献者により開発されています。

- ver.1.0.3 (2018/7/23 リリース)
- ver.1.0.2 (2017/8/25 リリース)
- ver.1.0.1 (2017/6/8 リリース)
- ver.1.0.0 (2017/5/23 リリース)
- ver.0.2.0 (2017/3/16 リリース)
- ver.0.1.1 (2016/12/16 リリース)
- ver.0.1.0 (2016/10/26 リリース)

— 開発者

* 三澤 貴宏 (東京大学 物性研究所)

- * 森田 悟史 (東京大学 物性研究所)
- * 大越 孝洋 (東京大学 大学院工学系研究科)
- * 井戸 康太 (東京大学 大学院工学系研究科)
- * 今田 正俊 (東京大学 大学院工学系研究科)
- * 本山 裕一 (東京大学 物性研究所)
- * 河村 光晶 (東京大学 物性研究所)
- * 吉見 一慶 (東京大学 物性研究所)
- プロジェクトコーディネーター
 - * 加藤 岳生 (東京大学 物性研究所)

1.2 動作環境

以下の環境で動作することを確認しています。

- 東京大学物性研究所スーパーコンピューターシステム B 「sekirei」
- 京コンピューター
- OpenMPI + Intel Compiler + MKL
- MPICH + Intel Compiler + MKL
- MPICH + GNU Compiler + MKL

2

How to use mVMC?

2.1 要件

mVMC のコンパイル・使用には次のものがが必要です。

- C コンパイラ (インテル、富士通、GNU など)
- MPI ライブラリ
- LAPACK ライブラリ (インテル MKL, 富士通, ATLAS など)
- オプション: ScaLAPACK ライブラリ

Tips

例/ intel コンパイラーでの設定

intel コンパイラを使用する場合には、コンパイラに付属の設定用スクリプトを使用するのが簡単です。

64 ビット OS で bash を使っている場合には

```
source /opt/intel/bin/compilervars.sh intel64
```

または

```
source /opt/intel/bin/iccvars.sh intel64
```

```
source /opt/intel/mkl/bin/mklvars.sh
```

等を ~/.bashrc に記載してください。詳しくはお手持ちのコンパイラ、ライブラリのマニュアルをお読みください。

2.2 インストール方法

mVMC は次の場所からダウンロードできます。

<https://github.com/issp-center-dev/mVMC/releases>

ダウンロードしたファイルを次のように展開してください。

```
$ tar xzvf mVMC-xxx.tar.gz
```

mVMC は次の 2 通りの方法でインストールできます。

2.2.1 mVMCconfig.sh を使う方法

展開したディレクトリのなかにある `mVMCconfig.sh` スクリプトを次のように実行してください。(物性研システム B "sekirei" の場合)

```
$ bash mVMCconfig.sh sekirei
```

これによりコンパイル環境設定ファイル `make.sys` が `src/` ディレクトリに作られます。`mVMCconfig.sh` の引数は次のものに対応しています。

- `sekirei` : 物性研究所システム B "sekirei"
- `kei` : 京コンピューターおよび物性研究所システム C "maki" (FX10)
- `intel-openmpi` : Intel コンパイラ + OpenMPI
- `intel-mpich` : Intel コンパイラ + MPICH2
- `intel-intelmpi` : Intel コンパイラ + IntelMPI
- `gcc-mpich-mkl` : GCC + MPICH + MKL
- `gcc-openmpi` : GCC + OpenMPI

`make.sys` の中身は次のようになっています (物性研システム B "sekirei" の場合)。

```
CC = mpicc
F90 = mpif90
CFLAGS = -O3 -no-prec-div -xHost -qopenmp -Wno-unknown-pragmas
FFLAGS = -O3 -implicitnone -xHost
LIBS = -L $(MKLR00T)/lib/intel64 -lmkl_scalapack_lp64 -lmkl_intel_lp64 \
      -lmkl_intel_thread -lmkl_core -lmkl_blacs_sgimpt_lp64 -lpthread -lm
SFMTFLAGS = -no-ansi-alias -DHAVE_SSE2
```

となります。それぞれのマクロ (変数) の説明は次のとおりです。

- `CC` : C コンパイラー (`mpicc`, `mpifccpx` など)
- `F90` : fortran コンパイラー (`ifort`, `frtpx` など)
- `LIBS` : リンカーオプション。
- `CFLAGS` : C コンパイルオプション。
- `FFLAGS` : fortran コンパイルオプション。

これでコンパイルのための準備が整います。その後

```
$ make mvmc
```

とすることで実行可能ファイル `vmc.out`、`vmcdry.out` が `src/` 内に生成されるので、このディレクトリにパスを通すか、パスの通っている場所にシンボリックリンクを作ってください。

Tips

実行ファイルにパスを通す時には、次のようにします。

```
$ export PATH=${PATH}:mVMCのディレクトリ/src/
```

この設定を常に残すには、例えばログインシェルが `bash` の場合には `~/.bashrc` ファイルに上記のコマンドを記載します。

2.2.2 cmake を使う方法

mVMC を展開したディレクトリのパスを `$PathTomvmc`、ビルドディレクトリを `$HOME/build/mvmc` (任意の場所を指定可能) とした場合に、

```
cd $HOME/build/mvmc
cmake -DCONFIG=gcc $PathTomvmc
make
```

でコンパイルすることができます。コンパイル後、`$HOME/build/mvmc` 直下に `src` フォルダが作成され、実行ファイルである `vmc.out` がそのフォルダ内に作成されます。

なお、上の例では `gcc` コンパイラを前提としたコンパイルになっていますが、

- `sekirei` : 物性研究所システム B "sekirei"
- `fujitsu` : 富士通コンパイラ
- `intel` : intel コンパイラ + Linux PC
- `gcc` : GCC + Linux PC

のオプションが用意されています。以下、mVMC を展開したディレクトリでビルドする例を示します (intel コンパイラの場合)。

```
mkdir ./build
cd ./build
cmake -DCONFIG=intel ../
make
```

実行後、`build` フォルダ直下に `src` フォルダが作成され、`vmc.out` が `src` フォルダ内に作成されます。また、LAPACK に代わり ScaLAPACK を計算に使用することが可能です。その場合には、

```
-DUSE_SCALAPACK=ON -DSCALAPACK_LIBRARIES="xxx"
```

を `cmake` をする際に付け加えてください (xxx には ScaLAPACK を利用するためのライブラリー式を指定します)。なお、コンパイラを変更しコンパイルし直したい場

合には、都度 build フォルダごと削除を行った上で、新規に上記作業を行うことをお勧めします。

Tips

sekirei で cmake を利用するには

```
source /home/issp/materiapps/tool/env.sh
```

をあらかじめ実行する必要があります。

また ScaLAPACK を利用するには

```
cmake -DCONFIG=sekirei ../ -DUSE_SCALAPACK=ON
```

を行うと、デフォルトで

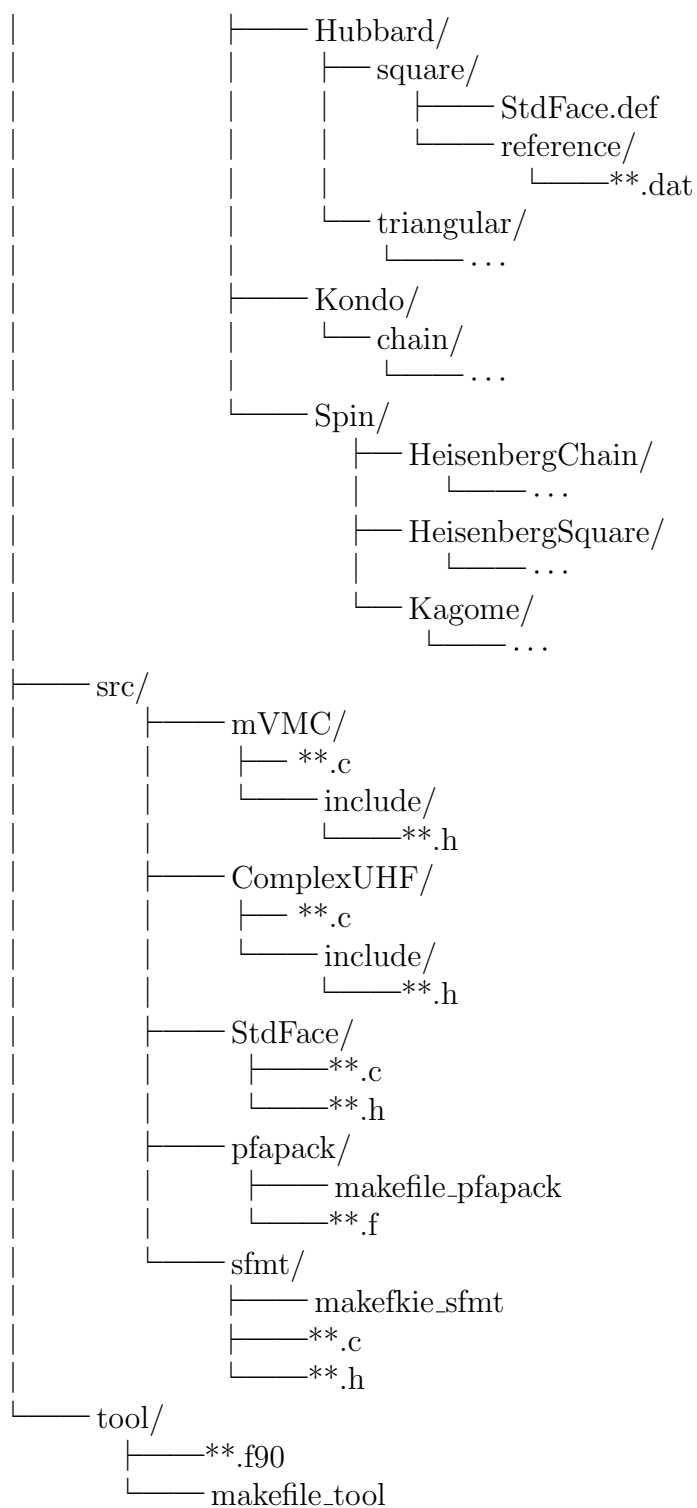
```
-DSCALAPACK_LIBRARIES="\${MKLR00T}/lib/intel64 -lmkl_scalapack_lp64  
-lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core  
-lmkl_blacs_sgimpt_lp64"
```

が指定されます。ライブラリへのパスが異なる場合には、
-DSCALAPACK_LIBRARIES を適宜変更してください。

2.3 ディレクトリ構成

mVMC-xxx.gz を解凍後に構成されるディレクトリ構成を以下に示します。

```
├── COPYING
├── mVMCconfig.sh
├── doc/
│   ├── bib/
│   │   ├── elsart-num_mod.bst
│   │   └── userguide.bib
│   ├── figs/
│   │   ├── *.pdf
│   │   └── *.xbb
│   ├── fourier/
│   │   ├── en/
│   │   ├── figs/
│   │   └── ja/
│   ├── jp/
│   │   └── *.tex
│   └── en/
│       └── *.tex
├── sample/
│   └── Standard/
```



2.4 基本的な使い方

mVMC は次の二つのいずれかのモードで動作します。

- エキスパートモード

mVMC では一般的な格子フェルミオン/スピン系に対応しており、各サイト毎にホッピング等を別々に指定することが出来ます。これにより計算対象の柔軟な指定が可能となりますが、用意する入力ファイルは多く、計算のセットアップは比較的煩雑になります。

- スタンダードモード

典型的なモデル (正方格子上の Heisenberg モデルなど) では、計算するセルのサイズや共通の相互作用項の大きさなど少数のパラメーターのみを入力してエキスパートモード用の入力ファイルを自動生成し、計算をすることが出来ます。計算対象はエキスパートモードに比べて限られますが、比較的容易に計算をセットアップすることが出来ます。また、エキスパートモード用の入力ファイルを自動生成した後、計算をする前にそれらを手動で編集してより広範なモデルに対応させることも可能です。

これらのモードを用いて次の用に計算を行います。

1. 計算用ディレクトリの作成

計算シナリオ名を記載したディレクトリを作成します。

2. スタンダードモードの入力ファイルの作成

あらかじめ用意されたいくつかのモデル (Heisenberg モデルや Hubbard モデル) や格子 (正方格子など) を指定し、それらに対するいくつかのパラメーター (最近接・次近接スピン結合やオンサイトクーロン積分など) を設定します。各ファイルは Sec. 4.1 に従い記載してください。

3. 実行

作成した入力ファイル名を引数として `vmc.out` を実行します。このとき入力ファイル名の前にオプション `-s` を付けます。

```
$ mpiexec -np プロセス数 パス/vmc.out -s 入力ファイル
```

ワークステーションやスパコン等でキューイングシステムを利用している場合はプロセス数をジョブ投入コマンドの引数として与える場合があります。詳しくはお使いのシステムのマニュアルをご参照ください。

4. 途中経過

計算実行の経過についてカレントディレクトリ直下の `output/` ディレクトリ (無ければ作られる) にログファイルが出力されます。出力されるファイルの詳細に関しては Sec. 4.3 を参考にしてください。

5. 最終結果

計算が正常終了した場合、計算モードに従い `output/` ディレクトリに計算結果ファイルが出力されます。出力されるファイルの詳細に関しては Sec. 4.3 を参考にしてください。

6. エキスパートモードの入力ファイルの作成と実行

上の例ではエキスパートモードのファイルを自動生成した後そのまま計算を開始していますが、エキスパートモードのファイルの生成のみを行う場合には `vmcdry.out` を実行します。MPI は使用しません。

```
$ パス/vmcdry.out 入力ファイル
```

このとき生成されたファイルを必要に応じて手動で編集したのち、`-e` というオプションの後に `namelist/def` というファイルを引数として `vmcd.out` を実行します。

```
$ mpiexec -np プロセス数 パス/vmc.out -e namelist.def
```

以降はスタンダードモードと同様です。

Tips

OpenMP スレッド数の指定

実行時の OpenMP のスレッド数を指定する場合は、`vmc.out` を実行する前に以下の様にしてください (16 スレッドの場合)。

```
export OMP_NUM_THREADS=16
```

2.4.1 バージョン番号の確認

次のように `-v` オプションをつけて `vmc.out`, `vmcdry.out` を実行すると、バージョン番号を標準出力した後終了します。

```
$ パス/vmcdry.out -v
```

```
$ パス/vmc.out -v
```

3

チュートリアル

3.1 サンプルファイル一覧

mVMC では `sample/Standard/` 以下に次のサンプルを用意しています。

- 2次元正方格子 Hubbard モデル
(`sample/Standard/Hubbard/square/`)
- 2次元三角格子 Hubbard モデル
(`sample/Standard/Hubbard/triangular/`)
- 1次元近藤格子モデル
(`sample/Standard/Kondo/chain/`)
- 1次元反強磁性的 Heisenberg モデル
(`sample/Standard/Spin/HeisenbergChain/`)
- 2次元正方格子反強磁性的 Heisenberg モデル
(`sample/Standard/Spin/HeisenbergSquare/`)
- 2次元カゴメ格子反強磁性的 Heisenberg モデル
(`sample/Standard/Spin/Kagome/`)

これらのチュートリアルの実行方法は全て同じ手順で実行することが可能です。
以下では Heisenberg 模型について説明します。

3.2 Heisenberg 模型

以下のチュートリアルはディレクトリ

`sample/Standard/Spin/HeisenbergChain/`

内で行います。このディレクトリには以下のファイルがあります。

Heisenberg 模型におけるサンプル入力ファイル: `StdFace.def`
参照用出力ディレクトリ: `reference/`

この例では1次元の Heisenberg 鎖 (最近接サイト間の反強磁性的スピン結合のみを持つ) を考察します。

$$\hat{H} = J \sum_{i=1}^L \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1} \quad (3.1)$$

インプットファイルの中身は次のとおりです。

```
L = 16
Lsub=4
model = "Spin"
lattice = "chain lattice"
J = 1.0
2Sz = 0
NMPtrans=1
```

この例ではスピン結合 $J = 1$ (任意単位) とし、サイト数は 16 としました。

計算実行

実行コマンドは次のとおりです。

```
$ mpiexec -np プロセス数 パス/vmc.out -s StdFace.def
```

使っているシステムによっては mpiexec コマンドではなく mpirun や mpijob、poe となる場合もあります。

この実行による標準出力は次のとおりです。

```
##### Standard Interface Mode STARTS #####
```

```
Open Standard-Mode Inputfile StdFace.def
```

```
KEYWORD : 1                | VALUE : 16
KEYWORD : lsub              | VALUE : 4
KEYWORD : model             | VALUE : spin
KEYWORD : lattice          | VALUE : chain
KEYWORD : j                 | VALUE : 1.0
KEYWORD : nmptrans         | VALUE : 1
```

```
##### Parameter Summary #####
```

```
@ Lattice Size & Shape
```

```
      L = 16
    Lsub = 4
      L = 16
      W = 1
phase0 = 0.00000 ##### DEFAULT VALUE IS USED #####
```

```
@ Hamiltonian
```



```

        2S = 1          ##### DEFAULT VALUE IS USED #####
        h = 0.00000    ##### DEFAULT VALUE IS USED #####
        Gamma = 0.00000 ##### DEFAULT VALUE IS USED #####
        D = 0.00000    ##### DEFAULT VALUE IS USED #####
        J0x = 1.00000
        J0y = 1.00000
        J0z = 1.00000

@ Numerical conditions

        Lsub = 4
        Wsub = 1
        ioutputmode = 1          ##### DEFAULT VALUE IS USED #####

##### Print Expert input files #####

qptransidx.def is written.
        filehead = zvo          ##### DEFAULT VALUE IS USED #####
        filehead = zqp          ##### DEFAULT VALUE IS USED #####
        NVMCCalMode = 0          ##### DEFAULT VALUE IS USED #####
        NLanczosMode = 0          ##### DEFAULT VALUE IS USED #####
        NDataIdxStart = 1          ##### DEFAULT VALUE IS USED #####
        NDataQtySmp = 1          ##### DEFAULT VALUE IS USED #####
        NSPGaussLeg = 8          ##### DEFAULT VALUE IS USED #####
        NMPTrans = 1
        NSROptItrStep = 1000      ##### DEFAULT VALUE IS USED #####
        NSROptItrSmp = 100        ##### DEFAULT VALUE IS USED #####
        NVMCWarmUp = 10           ##### DEFAULT VALUE IS USED #####
        NVMCInterval = 1          ##### DEFAULT VALUE IS USED #####
        NVMCSample = 1000         ##### DEFAULT VALUE IS USED #####
        NExUpdatePath = 2
        RndSeed = 123456789       ##### DEFAULT VALUE IS USED #####
        NSplitSize = 1            ##### DEFAULT VALUE IS USED #####
        NStore = 0                ##### DEFAULT VALUE IS USED #####
        DSOptRedCut = 0.00100     ##### DEFAULT VALUE IS USED #####
        DSOptStaDel = 0.02000     ##### DEFAULT VALUE IS USED #####
        DSOptStepDt = 0.02000     ##### DEFAULT VALUE IS USED #####
        NSPStot = 0               ##### DEFAULT VALUE IS USED #####
        ComplexType = 0           ##### DEFAULT VALUE IS USED #####
locspn.def is written.
trans.def is written.
interall.def is written.
jastrowidx.def is written.
coulombintra.def is written.
coulombinter.def is written.
hund.def is written.
exchange.def is written.

```

```

orbitalidx.def is written.
gutzwilleridx.def is written.
namelist.def is written.
modpara.def is written.
greenone.def is written.
greentwo.def is written.

##### Input files are generated. #####

-----
Start: Read *def files.
  Read File namelist.def .
  Read File 'modpara.def' for ModPara.
  Read File 'locspn.def' for LocSpin.
  Read File 'trans.def' for Trans.
  Read File 'coulombintra.def' for CoulombIntra.
  Read File 'coulombinter.def' for CoulombInter.
  Read File 'hund.def' for Hund.
  Read File 'exchange.def' for Exchange.
  Read File 'gutzwilleridx.def' for Gutzwiller.
  Read File 'jastrowidx.def' for Jastrow.
  Read File 'orbitalidx.def' for Orbital.
  Read File 'qptransidx.def' for TransSym.
  Read File 'greenone.def' for OneBodyG.
  Read File 'greentwo.def' for TwoBodyG.
End : Read *def files.
Start: Read parameters from *def files.
End : Read parameters from *def files.
Start: Set memories.
End : Set memories.
Start: Initialize parameters.
End : Initialize parameters.
Start: Initialize variables for quantum projection.
End : Initialize variables for quantum projection.
Start: Optimize VMC parameters.
End : Optimize VMC parameters.
-----

```

この実行でははじめにエキスパートモード用の入力ファイルとして、ハミルトニアンの詳細を記述するファイル

- locspin.def
- trans.def
- coulombinter.def
- coulombintra.def

- exchange.def
- hund.def
- namelist.def
- modpara.def

と、変分パラメータを設定するファイル

- gutzwilleridx.def
- jastrowidx.def
- orbitalidx.def
- qptransidx.def

結果として出力する相関関数の要素を指定するファイル

- greenone.def
- greentwo.def

が生成されます。各ファイルの詳細については Sec. 4.2 をご覧ください。

その後実際に計算が行われ、以下のファイルが情報として output/ ディレクトリに出力されます。

```
zvo_SRinfo.dat
zvo_out_001.dat
zvo_time_001.dat
zvo_var_001.dat
zvo_CalcTimer.dat
```

なお、zvo_out_001.dat には、ビン毎の計算情報として、

$$\langle H \rangle, \langle H^2 \rangle, \frac{\langle H^2 \rangle - \langle H \rangle^2}{\langle H \rangle^2}$$

が順に出力されますので、収束性の目安として利用することが可能です。gnuplot を用いる場合には、次のようにして表示することが出来ます ($\langle H \rangle$ の場合)。

```
plot "zvo_out_001.dat" u 1
```

各ファイルの詳細については Sec. 4.3 をご覧ください。

計算結果出力

計算が正常終了すると、エネルギー、エネルギーの分散、変分パラメータおよび計算実行時間を記載したファイルが `output/` ディレクトリに出力されます。以下に、このサンプルでの出力ファイルを記載します。

```
gutzwiller_opt.dat
jastrow_opt.dat
orbital_opt.dat
zqp_opt.dat
ClacTimer.dat
```

各ファイルの詳細については Sec. 4.3 をご覧ください。

Green 関数の計算

`modpara.def` ファイル中の `NVMCCalMode` を 0 から 1 に変更の上、以下のコマンドを実行します。下記のように実行時のコマンドライン引数として "`namelist.dat`" の後ろに "`zqp_opt.dat`" を付け加えることで、一つ前の計算で最適化された変分パラメータを使用した計算が行われます。

```
$ パス/vmc.out -e namelist.def output/zqp_opt.dat
```

計算が終了すると以下のファイルが `output/` ディレクトリに出力されます。

```
zvo_cisajs_001.dat
zvo_cisajsckalt_001.dat
```

各ファイルの詳細については Sec. 4.3 をご覧ください。

3.3 エキスパートユーザー向け

mVMC では、以下の 6 つに分類される入力ファイルを読み込み、計算実行を行います。

- (1) **List:** 詳細入力ファイルの種類と名前を指定するファイル
- (2) **Basic parameters:** 基本的なパラメータを指定するファイル
- (3) **Set Hamiltonian:** ハミルトニアンを指定するファイル
- (4) **Set condition of variational parameters :** 最適化する変分パラメータを指定するファイル
- (5) **Initial variational parameters:** 変分パラメータの初期値を指定するファイル
- (6) **Output:** 出力する一体・二体グリーン関数の成分を指定するファイル

上記で分類されるファイルを直接作成・指定することで、より複雑な計算を行うことが可能です。ファイルの詳細については Sec. 4.2 をご覧ください。

3.4 相関関数のフーリエ変換

このパッケージには、上で求めた相関関数をフーリエ変換し、プロットするユーティリティが付属しています。このユーティリティに関するマニュアルは

```
doc/fourier/ja/_build/html/index.html
doc/fourier/ja/_build/latex/fourier.pdf
doc/fourier/en/_build/html/index.html
doc/fourier/en/_build/latex/fourier.pdf
```

にありますので、そちらを参照してください。

4

ファイル仕様

4.1 スタンダードモード用入力ファイル

スタンダードモード用入力ファイルは次のような格好をしています。

```
W = 2
L = 4
model = "spin"

lattice = "triangular lattice"
//mu = 1.0
// t = -1.0
// t' = -0.5
// U = 8.0
//V = 4.0
//V'=2.0
J = -1.0
J'=-0.5
// nelec = 8
```

大まかなルールは次のとおりです。

- 各行にはひと組ずつキーワード(=の前)とパラメーター(=の後)が書かれており間は=で区切られています。
- 各キーワードは順不同に記述できます。
- 空白行、または//で始まる行(コメントアウト)は読み飛ばされます。
- 各キーワード、パラメーターの大文字・小文字は区別されません。ダブルクオート、空白は無視されます。
- 必ず指定しなければいけないパラメーター、指定しない場合デフォルト値が使われるパラメーター、(他のパラメーターの組み合わせによっては)使われないパラメーターが存在します。使われないパラメーターが指定された場合にはプログラムは終了し、入力ファイルをチェックするようにというメッセージが英語で表示されます。

次に各キーワードの説明をします。

4.1.1 計算の種類に関する必須パラメーター

- model

形式: 文字列("Fermion Hubbard", "Spin", "Kondo Lattice", "Fermion HubbardGC", "SpinGC", "Kondo LatticeGC" のいずれか)

説明：計算対象のモデルを指定します。文字列"Fermion Hubbard"は、カノニカル集団のフェルミ粒子 Hubbard 模型

$$H = -\mu \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma} - \sum_{i \neq j\sigma} t_{ij} c_{i\sigma}^\dagger c_{j\sigma} + \sum_i U n_{i\uparrow} n_{i\downarrow} + \sum_{i \neq j} V_{ij} n_i n_j, \quad (4.1)$$

文字列"Spin"はカノニカル集団のスピン模型 ($\{\sigma_1, \sigma_2\} = x, y, z$)

$$H = -h \sum_i S_{iz} - \Gamma \sum_i S_{ix} + D \sum_i S_{iz} S_{iz} \\ + \sum_{ij, \sigma_1} J_{ij\sigma_1} S_{i\sigma_1} S_{j\sigma_1} + \sum_{ij, \sigma_1 \neq \sigma_2} J_{ij\sigma_1\sigma_2} S_{i\sigma_1} S_{j\sigma_2}, \quad (4.2)$$

文字列"Kondo Lattice"はカノニカル集団の近藤格子模型 (Hubbard 模型と同様に U と J を入れることも可能)

$$H = -\mu \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma} - t \sum_{\langle ij \rangle \sigma} c_{i\sigma}^\dagger c_{j\sigma} + \frac{J}{2} \sum_i \left\{ S_i^+ c_{i\downarrow}^\dagger c_{i\uparrow} + S_i^- c_{i\uparrow}^\dagger c_{i\downarrow} + S_{iz} (n_{i\uparrow} - n_{i\downarrow}) \right\} \\ + \sum_i U n_{i\uparrow} n_{i\downarrow} + \sum_{i \neq j} V_{ij} n_i n_j, \quad (4.3)$$

に対応します。また、"Fermion HubbardGC"、"SpinGC"、"Kondo LatticeGC"はそれぞれ S_z 非保存でのフェルミ粒子 Hubbard 模型 [式 (4.1)]、スピン模型 [式 (4.2)]、近藤格子模型 [式 (4.3)] に対応します。 $\mathcal{H}\Phi$ との互換性から GC (= グランドカノニカル) と付いていますが、粒子数は保存していますのでご注意ください。

- lattice

形式：文字列 ("Chain Lattice", "Square Lattice", "Triangular Lattice", "Honeycomb Lattice", "Kagome", "Ladder" のいずれか)

説明：格子の形状を指定します。上記文字列はそれぞれ 1 次元鎖 (Fig. 4.1(a))、2 次元正方格子 (Fig. 4.1(b))、2 次元三角格子 (Fig. 4.1(c))、2 次元異方的蜂の巣格子 (Fig. 4.2)、カゴメ格子 (Fig. 4.3)、梯子格子 (Fig. 4.4) に対応します。

4.1.2 格子に関するパラメーター

1 次元鎖 [Fig. 4.1(a)]

- L

形式：自然数

説明：鎖の長さを指定します。

梯子格子 (Fig. 4.4)

- L

形式：自然数

説明：梯子の長さを指定します。

- W

形式：自然数

説明：梯子の本数を指定します。

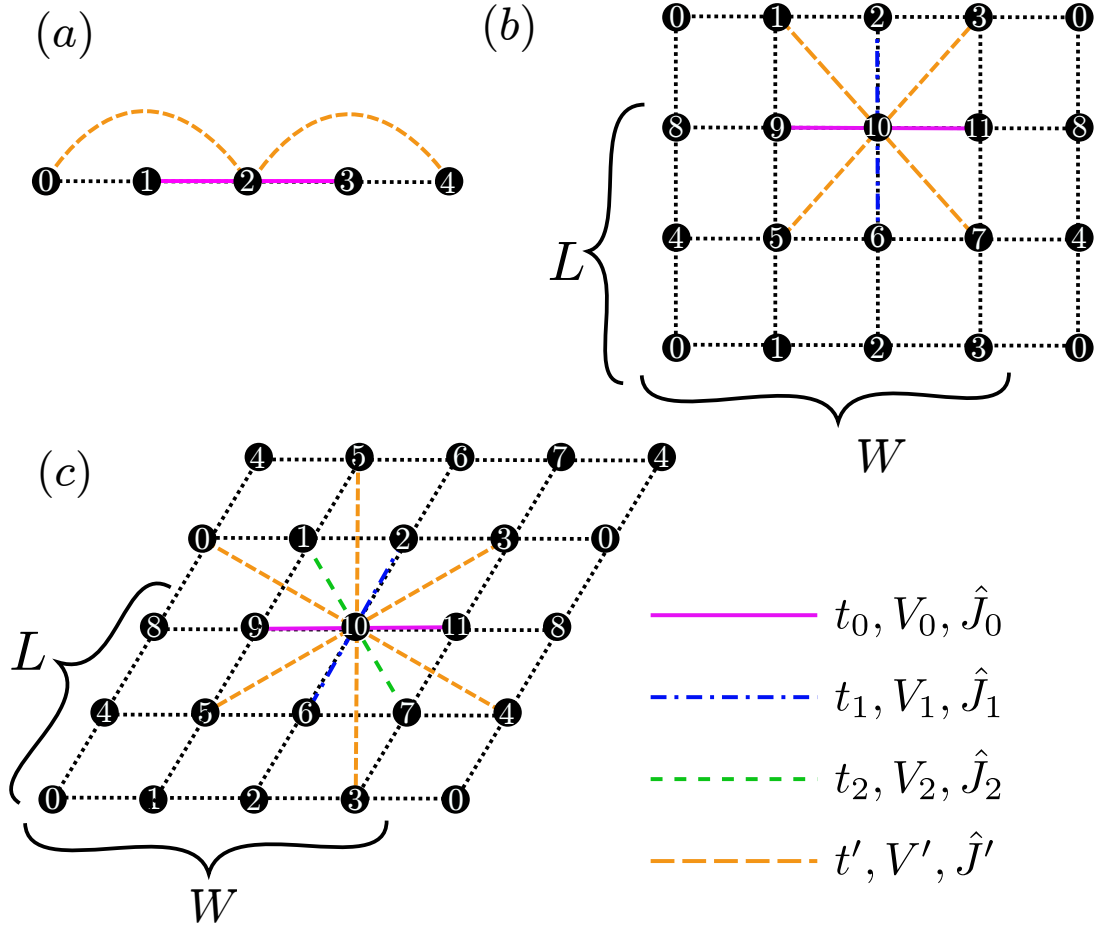


Figure 4.1: (a)1次元鎖、(b)2次元正方格子、(c)2次元三角格子の模式図. ホッピング積分、オフサイトクーロン積分、スピン結合は、再近接サイト間(マゼンタの実線)ではそれぞれ t, V, J となり、次近接サイト間(緑の破線)ではそれぞれ t', V', J' となります。

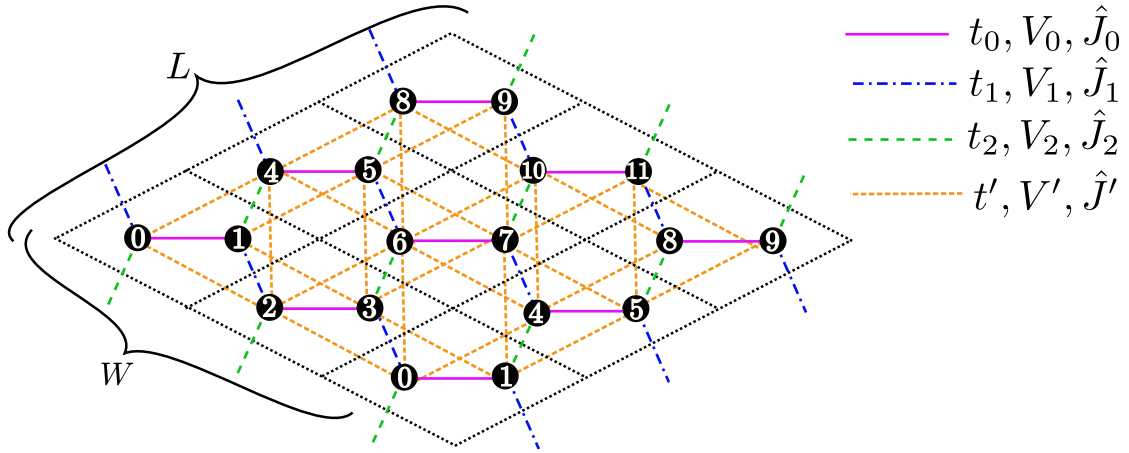


Figure 4.2: 2次元異方的蜂の巣格子の模式図. ホッピング積分、オフサイトクーロン積分、スピン結合は、ボンドの方向によって異なります。また、次近接のホッピング積分、オフサイトクーロン積分、スピン結合には対応していません。

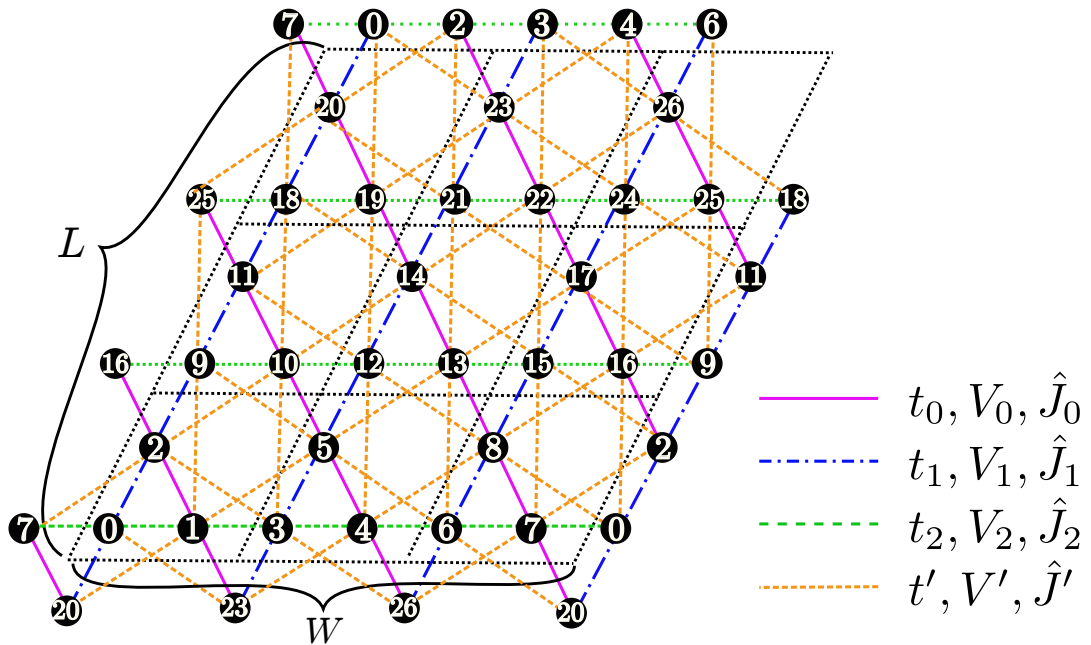


Figure 4.3: カゴメ格子の模式図.

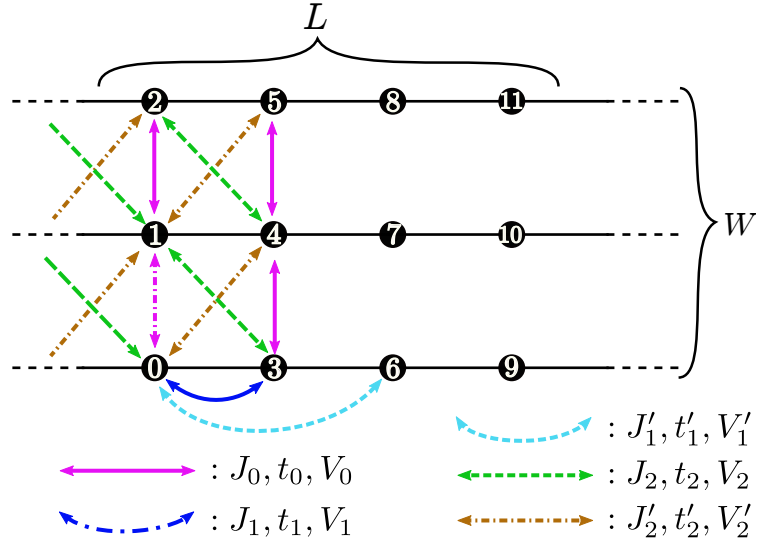
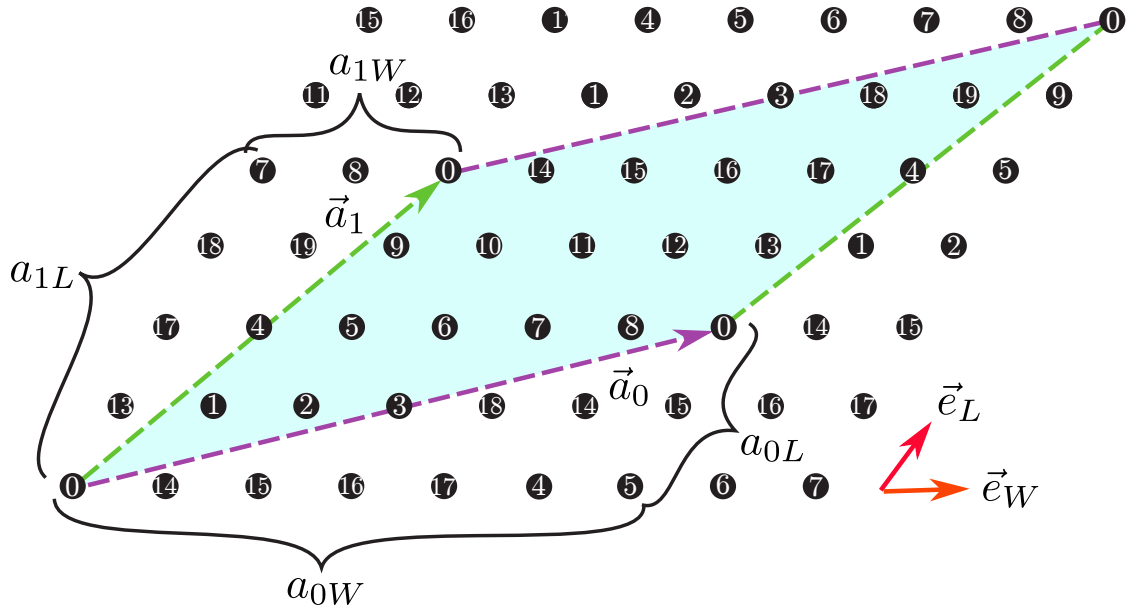


Figure 4.4: 梯子格子の模式図.

Figure 4.5: 三角格子において、 $\mathbf{a}_0 = (6, 2)$, $\mathbf{a}_1 = (2, 4)$ とした場合のセル形状。 \mathbf{a}_0 (マゼンタ) および \mathbf{a}_1 (グリーン) で囲まれた部分 (サイト数は 20) が計算するセルとなる。

正方格子 [Fig. 4.1(b)], 三角格子 [Fig. 4.1(c)], 蜂の巣格子 (Fig. 4.2)、カゴメ格子 (Fig. 4.3)

これらの格子では、標準の単位胞 (図中の黒の破線を参照) を用いて格子形状を指定する方法と、それらとは別の方向に格子ベクトルを取る方法が選択できます。また、両方を指定した場合にはプログラムを終了します。

- W, L

形式 : 自然数

説明 : 標準の単位胞の並び方を指定します。

- a_0W, a_0L, a_1W, a_1L

形式 : 自然数

説明 : 格子を指定する 2 本のベクトル ($\mathbf{a}_0, \mathbf{a}_1$) を指定します (Fig. 4.5)。これらのベクトルは標準の並進ベクトルを基底とした座標 (Fractional coordinate) で指定されます。

スタンダードモードで出力される `lattice.gp` (1 次元鎖、梯子格子では出力されません) というファイルを使うと、自分の意図した通りの格子のとり方になっているかどうかを確認することが出来ます。このファイルは、次のようにして `gnuplot` に読み込ませることが出来ます。

```
$ gnuplot lattice.gp
```

4.1.3 副格子

以下パラメータを用いると変分波動関数のペア軌道部分に副格子の周期性を持たせることが出来ます。

- $a_0W_{\text{sub}}, a_0L_{\text{sub}}, a_1W_{\text{sub}}, a_1L_{\text{sub}}, W_{\text{sub}}, L_{\text{sub}}$

形式 : 自然数。デフォルトでは $a_0W_{\text{sub}}=a_0W, a_0L_{\text{sub}}=a_0L, a_1W_{\text{sub}}=a_1W, a_1L_{\text{sub}}=a_1L, W_{\text{sub}}=W, L_{\text{sub}}=L$ となる。すなわち副格子を用いず、変分波動関数のすべてのパラメーターが独立に変化する。

説明 : これらのパラメーターの指定の仕方は $a_0W, a_0L, a_1W, a_1L, W, L$ と同様です。ただし、元の計算セルが副格子に整合しない場合にはプログラムを終了します。

4.1.4 ハミルトニアン各项の係数

デフォルト値は特に記載されていないものについては 0 に設定してあります。型が複素数のパラメータは「実部, 虚部」(間に “,”) の形式で指定し、実数の場合には「実部」で指定が可能です。

局所項

- μ

形式 : 実数

説明 : Hubbard および近藤格子模型での化学ポテンシャルを指定します。

- U

形式：実数

説明：Hubbard および近藤格子模型でのオンサイトクーロン積分を指定します。

- Jx, Jy, Jz, Jxy, Jyx, Jxz, Jzx, Jyz, Jzy

形式：実数

説明：近藤格子模型での、局在電子と遍歴電子のスピン結合を指定します。また対角項について、Jx, Jy, Jz を指定する代わりに、パラメータ J を指定すると $J_x = J_y = J_z = J$ が代入されます。J を指定した上で Jx, Jy, Jz を指定した場合はプログラムを終了します。

- h, Gamma, D

形式：実数

説明：スピン模型での縦磁場、横磁場、異方性パラメータを指定します。

下記の非局所項は、梯子格子の場合とそれ以外 (1次元鎖、矩形格子、三角格子、蜂の巣格子、カゴメ格子) の場合で指定の仕方が異なります。また、各格子で指定可能なパラメータを Table 4.1 に表します。

相互作用	1次元鎖	矩形格子	三角格子	蜂の巣格子	カゴメ格子	梯子格子
J, t, V(省略形)	○	○	○	○	○	-
J', t', V'	○	○	○	○	○	-
J0, t0, V0	○	○	○	○	○	○
J1, t1, V1	-	○	○	○	○	○
J2, t2, V2	-	-	○	○	○	○
J1', t1', V1'	-	-	-	-	-	○
J2', t2', V2'	-	-	-	-	-	○

Table 4.1: 各格子で定義可能な相互作用一覧。ただし、スピン結合については行列として与えることが可能。

非局所項 [梯子格子 (Fig. 4.4)]

- t0, t1, t1', t2, t2'

形式：複素数

説明：梯子格子でのホッピング (Fig. 4.4 参照) を指定します。

- V0, V1, V1', V2, V2'

形式：実数

説明：梯子格子でのオフサイトクーロン積分 (Fig. 4.4 参照) を指定します。

- J0x, J0y, J0z, J0xy, J0yx, J0xz, J0zx, J0yz, J0zy

- J1x, J1y, J1z, J1xy, J1yx, J1xz, J1zx, J1yz, J1zy

- J1'x, J1'y, J1'z, J1'xy, J1'yx, J1'xz, J1'zx, J1'yz, J1'zy

- $J_{2x}, J_{2y}, J_{2z}, J_{2xy}, J_{2yx}, J_{2xz}, J_{2zx}, J_{2yz}, J_{2zy}$
- $J_{2'x}, J_{2'y}, J_{2'z}, J_{2'xy}, J_{2'yx}, J_{2'xz}, J_{2'zx}, J_{2'yz}, J_{2'zy}$

形式：実数

説明：梯子格子でのスピン相互作用 (Fig. 4.4 参照) を指定します。また対角項について、例えば J_{0x}, J_{0y}, J_{0z} を指定する代わりにパラメータ J_0 を指定すると $J_{0x} = J_{0y} = J_{0z} = J_0$ が代入されます。 J_0 を指定した上で J_{0x}, J_{0y}, J_{0z} 等も指定した場合はプログラムを終了します。 J_1, J_1', J_2, J_2' についても同様です。

非局所項 [梯子格子以外 (Figs. 4.1, 4.2, 4.3)]

- t_0, t_1, t_2

形式：複素数

説明：Hubbard および近藤格子模型での、最近接サイト間の各方向のホッピングを指定します。Figs. 4.1, 4.2, 4.3 のそれぞれのボンド (異なる線種で表されています) に対応するホッピングに関して別々の値を取るように刷ることが可能です。また、最近接ホッピング (t_0, t_1, t_2) に関して、ボンド方向依存性 (もしくは異方性) がない場合は t_0, t_1, t_2 を別々に指定する代わりにパラメータ t で一括指定 ($t_0 = t_1 = t_2 = t$) を行うことができます。 t と t_0 等の両方が指定された場合にはプログラムを終了します。

- V_0, V_1, V_2

形式：実数

説明：Hubbard および近藤格子模型での、最近接サイト間の Coulomb 積分を指定します。また、最近接サイト間の Coulomb 積分 (V_0, V_1, V_2) に関して、ボンド方向依存性 (もしくは異方性) がない場合は V_0, V_1, V_2 を別々に指定する代わりにパラメータ V で一括指定 ($V_0 = V_1 = V_2 = V$) を行うことができます。 V と V_0 等の両方が指定された場合にはプログラムを終了します。

- $J_{0x}, J_{0y}, J_{0z}, J_{0xy}, J_{0yx}, J_{0xz}, J_{0zx}, J_{0yz}, J_{0zy}$
- $J_{1x}, J_{1y}, J_{1z}, J_{1xy}, J_{1yx}, J_{1xz}, J_{1zx}, J_{1yz}, J_{1zy}$
- $J_{2x}, J_{2y}, J_{2z}, J_{2xy}, J_{2yx}, J_{2xz}, J_{2zx}, J_{2yz}, J_{2zy}$

形式：実数

説明：スピン模型での、最近接サイト間のスピン相互作用を指定します。また対角項について、例えば J_{0x}, J_{0y}, J_{0z} を指定する代わりにパラメータ J_0 を指定すると $J_{0x} = J_{0y} = J_{0z} = J_0$ が代入されます。 J_0 を指定した上で J_{0x}, J_{0y}, J_{0z} 等も指定した場合はプログラムを終了します。 J_1, J_2 についても同様です。

最近接スピン間相互作用のボンド方向依存性がない場合には、 $J_x, J_y, J_z, J_{xy}, J_{yx}, J_{xz}, J_{zx}, J_{yz}, J_{zy}$ を指定すると、 $J_{0x} = J_{1x} = J_{2x} = J_x$ のようにすべてのボンド方向のスピン間相互作用に同じ値を代入することができます。 $J_x \sim J_{zy}$ 系列のどれかと $J_{0x} \sim J_{2zy}$ 系列のどれかを両方指定した場合にはプログラムを終了します。以下に最近接間スピン相互作用の指定方法の例を挙げます。

- ボンド方向依存性、スピン方向依存性、相互作用の非対角成分 (J_{xy} 等) がない場合
J を指定

- ボンド方向依存性、相互作用の非対角成分がなく、スピン方向依存性がある場合 J_x, J_y, J_z のうち 0 でないものを指定
- ボンド方向依存性がなく、スピン方向依存性、相互作用の非対角成分がある場合 $J_x, J_y, J_z, J_{xy}, J_{yz}, J_{xz}, J_{yx}, J_{zy}, J_{zx}$ のうち 0 でないものを指定
- スピン方向依存性、相互作用の非対角成分がなく、ボンド方向依存性がある場合 J_0, J_1, J_2 のうち 0 でないものを指定
- スピン方向依存性がなく、ボンド方向依存性、相互作用の非対角成分がある場合 $J_{0x}, J_{0y}, J_{0z}, J_{1x}, J_{1y}, J_{1z}, J_{2x}, J_{2y}, J_{2z}$ のうち 0 でないものを指定
- ボンド方向依存性、スピン方向依存性、相互作用の非対角成分がある場合 $J_{0x} \sim J_{2zy}$ のすべてのうち 0 でないものを指定

- t'

形式：複素数

説明：Hubbard および近藤格子模型での、次近接サイト間の各方向のホッピングを指定します。

- V'

形式：実数

説明：Hubbard および近藤格子模型での、次近接サイト間の Coulomb 積分を指定します。

- $J'_x, J'_y, J'_z, J'_{xy}, J'_{yx}, J'_{xz}, J'_{zx}, J'_{yz}, J'_{zy}$

形式：実数

説明：スピン模型での、次近接サイト間のスピン相互作用を指定します。また対角項について、 J'_x, J'_y, J'_z を指定する代わりにパラメータ J' を指定すると $J'_x = J'_y = J'_z = J'$ が代入されます。 J' を指定した上で J'_x, J'_y, J'_z も指定した場合はプログラムを終了します。

- phase0, phase1

形式：実数 (デフォルトでは 0.0)

説明：計算するセルの境界をまたいだホッピング項に付く因子の位相を指定することが出来ます (単位:度)。 \mathbf{a}_0 方向、 \mathbf{a}_1 方向それぞれ別の位相因子を用いることが出来ます。1次元系では phase0 のみ使用できます。例えば、 i サイトから j サイトへのホッピングで、正の方向に境界をまたいだ場合には次のようになります。

$$\exp(i \times \text{phase0} \times \pi/180) \times t \hat{c}_{j\sigma}^\dagger \hat{c}_{i\sigma} + \exp(-i \times \text{phase0} \times \pi/180) \times t^* \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \quad (4.4)$$

4.1.5 計算条件のパラメーター

- nelec

形式：int 型 (1 以上、必須)

説明：伝導電子の数。 \uparrow 電子と \downarrow 電子の個数を足したものを入力してください。

- **NVMCCalMode**
 形式 : int 型 (デフォルト値 = 0)
 説明 : [0] 変分パラメータの最適化、[1] 1 体・2 体のグリーン関数の計算。
- **NDataIdxStart**
 形式 : int 型 (デフォルト値 = 1)
 説明 : 出力ファイルの付加番号。NVMCCalMode = 0 の場合は NDataIdxStart が出力され、NVMCCalMode = 1 の場合は、NDataIdxStart から連番で NDataQtySmp 個のファイルを出力します。
- **NDataQtySmp**
 形式 : int 型 (デフォルト値 = 1)
 説明 : 出力ファイルのセット数。NVMCCalMode = 1 の場合に使用します。
- **NSPGaussLeg**
 形式 : int 型 (1 以上、デフォルト値は S_z 保存系で 2Sz=0 の時は 8)
 説明 : スピン量子数射影の β 積分 (S^y 回転) の Gauss-Legendre 求積法の分点数。 S_z 保存系で 2Sz=0 ではない場合と S_z 非保存系では使われません。
- **NSPStot**
 形式 : int 型 (0 以上、デフォルト値 = 0)
 説明 : 全スピン量子数。
- **2Sz**
 形式 : int 型 (0 以上、デフォルト値 = 0)
 説明 : スピン量子数 S_z 。
- **NMPTrans**
 形式 : int 型 (1 以上、デフォルト値は 1)
 説明 : 運動量・格子対称性の量子数射影の個数。TransSym ファイルで指定した重みで上から NMPTrans 個まで使用する。射影を行わない場合は 1 に設定する。
- **NSROptItrStep**
 形式 : int 型 (1 以上、デフォルト値 = 1000)
 説明 : SR 法で最適化する場合の全ステップ数。NVMCCalMode=0 の場合のみ使用されます。
- **NSROptItrSmp**
 形式 : int 型 (1 以上数、デフォルト値 = NSROptItrStep/10)
 説明 : NSROptItrStep ステップ中、最後の NSROptItrSmp ステップでの各変分パラメータの平均値を最適値とする。NVMCCalMode=0 の場合のみ使用されます。
- **DSROptRedCut**
 形式 : double 型 (デフォルト値 = 0.001)
 説明 : SR 法安定化因子。手法論文 [3] の ε_{wf} に対応。

- **DSROptStaDel**
 形式 : double 型 (デフォルト値 = 0.02)
 説明 : SR 法安定化因子。手法論文 [3] の ε に対応。
- **DSROptStepDt**
 形式 : double 型 (デフォルト値 = 0.02)
 説明 : SR 法で使用する刻み幅。手法論文 [3] の Δt に対応。
- **NVMCWarmUp**
 形式 : int 型 (1 以上、デフォルト値=10)
 説明 : マルコフ連鎖の空回し回数。
- **NVMCInterval**
 形式 : int 型 (1 以上、デフォルト値=1)
 説明 : サンプル間のステップ間隔。ローカル更新を $N_{\text{site}} \times \text{NVMCInterval}$ 回行います。
- **NVMCSample**
 形式 : int 型 (1 以上、デフォルト値=1000)
 説明 : 期待値計算に使用するサンプル数。
- **RndSeed**
 形式 : int 型 (デフォルト値 = 123456789)
 説明 : 乱数の初期 seed。MPI 並列では各計算機に $\text{RndSeed} + \text{my rank} + 1$ で初期 seed が与えられます。
- **NSplitSize**
 形式 : int 型 (1 以上、デフォルト値=1)
 説明 : MPI 内部並列を行う場合の並列数。
- **NStore**
 形式 : int 型 (0 もしくは 1、デフォルト値=1)
 説明 : 期待値 $\langle O_k O_l \rangle$ を計算するとき行列-行列積にして高速化するオプション (1 で機能 On、モンテカルロサンプリング数に応じてメモリの消費が増大します*1)。
- **NSRCG**
 形式 : int 型 (0 もしくは 1、デフォルト値=0)
 説明 : SR 法で連立一次方程式 $Sx = g$ を解くときに、 S を陽に構築せずに解くことでメモリを削減する*2オプション [8](1 で機能 On, NStore は 1 に固定されます)。
- **ComplexType**
 形式 : int 型 (0 もしくは 1、デフォルト値は S_z 保存系では 0、非保存系では 1)
 説明 : 0 のとき変分パラメータの実部のみを、1 のとき実部/虚部両方を最適化します。

*1使用メモリ量が、 $O(N_p^2)$ から $O(N_p^2) + O(N_p N_{\text{MCS}})$ になります。

*2使用メモリ量は、 $O(N_p) + O(N_p N_{\text{MCS}})$ です。

- **OutputMode**

形式 : "none", "correlation", "full" のいずれか (デフォルトは **correlation**)

説明 : 計算を行う相関関数を指定します。"none" の場合は相関関数を計算しません。"correlation" を指定した場合には、付属のユーティリティ **fourier** でサポートするものに対応した相関関数を計算します。詳しくは **doc/fourier/** 内のマニュアルを参照してください。"full" を指定した場合には、1 体部分はすべての i, j, σ, σ' について $\langle c_{i\sigma}^\dagger c_{j\sigma'} \rangle$ を、2 体部分はすべての $i_1, i_2, i_3, i_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4$ について $\langle c_{i_1\sigma_1}^\dagger c_{i_2\sigma_2} c_{i_3\sigma_3}^\dagger c_{i_4\sigma_4} \rangle$ を計算します。スピン系の演算子は Bogoliubov 表現により生成消滅演算子で表されています。詳しくは Sec. 5.2 をご覧ください。

- **CDataFileHead**

形式 : string 型 (デフォルト値 **"zvo"**)

説明 : アウトプットファイルのヘッダ。例えば、一体の Green 関数の出力ファイル名が **xxx_cisajs.dat** として出力されます (xxx に CDataFileHead で指定した文字が記載)。

- **CParaFileHead**

形式 : string 型 (デフォルト値 **"zqp"**)

説明 : 最適化された変分パラメータの出力ファイル名のヘッダ。最適化された変分パラメータが **xxx_opt.dat** ファイルとして出力されます (xxx に CParaFileHead で指定した文字が記載)。

4.2 エキスパートモード入力ファイル

ここでは mVMC で使用する詳細入力ファイル (*def) のフォーマットに関して説明します。入力ファイルの種別は以下の 6 つで分類されます。なお、キーワードの後にある括弧内に記載されているファイル名は vmcdry.out により作成されるファイル名を表します。

(1) リスト:

キーワード指定なし (**namelist.def**): 使用する input file の名前のリストを書きます。なお、ファイル名は任意に指定することができます。

(2) 基本パラメータ:

ModPara (modpara.def): 計算時に必要な基本的なパラメーター (サイトの数、電子数など) を設定します。

LocSpin (locspn.def): 局在スピンの位置を設定します。

(3) ハミルトニアン:

電子系の表式で記載されるハミルトニアン

$$\mathcal{H} = \mathcal{H}_T + \mathcal{H}_U + \mathcal{H}_V + \mathcal{H}_H + \mathcal{H}_E + \mathcal{H}_P + \mathcal{H}_I, \quad (4.5)$$

$$\mathcal{H}_T = - \sum_{i,j} \sum_{\sigma_1, \sigma_2} t_{ij\sigma_1\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2}, \quad (4.6)$$

$$\mathcal{H}_U = \sum_i U_i n_{i\uparrow} n_{i\downarrow}, \quad (4.7)$$

$$\mathcal{H}_V = \sum_{i,j} V_{ij} n_i n_j, \quad (4.8)$$

$$\mathcal{H}_H = - \sum_{i,j} J_{ij}^{\text{Hund}} (n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}), \quad (4.9)$$

$$\mathcal{H}_E = \sum_{i,j} J_{ij}^{\text{Ex}} (c_{i\uparrow}^\dagger c_{j\uparrow}^\dagger c_{j\downarrow}^\dagger c_{i\downarrow} + c_{i\downarrow}^\dagger c_{j\downarrow}^\dagger c_{j\uparrow}^\dagger c_{i\uparrow}), \quad (4.10)$$

$$\mathcal{H}_P = \sum_{i,j} J_{ij}^{\text{Pair}} c_{i\uparrow}^\dagger c_{j\uparrow}^\dagger c_{i\downarrow}^\dagger c_{j\downarrow}, \quad (4.11)$$

$$\mathcal{H}_I = \sum_{i,j,k,l} \sum_{\sigma_1, \sigma_2, \sigma_3, \sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}, \quad (4.12)$$

について設定します。ここで、 $n_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$ はスピン σ を持つサイト i の電子密度演算子を、 $n_i = n_{i\uparrow} + n_{i\downarrow}$ はサイト i の電子密度演算子をそれぞれ表します。各ハミルトニアンのパラメータは以下のファイルで指定します。

Trans (trans.def): \mathcal{H}_T 内の $t_{ij\sigma_1\sigma_2}$ を指定します。

CoulombIntra (coulombintra.def): \mathcal{H}_U 内の U_i を指定します。

CoulombInter (coulombinter.def): \mathcal{H}_V 内の V_{ij} を指定します。

Hund (hund.def): \mathcal{H}_H 内の J_{ij}^{Hund} を指定します。

Exchange (exchange.def): \mathcal{H}_E 内の J_{ij}^{Ex} を指定します。

PairHop: \mathcal{H}_P 内の J_{ij}^{Pair} を指定します。

InterAll: \mathcal{H}_I 内の $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ を指定します。

(4) 最適化対象変分パラメータ:

最適化する変分パラメータを指定します。変分波動関数は

$$|\psi\rangle = \mathcal{P}_G \mathcal{P}_J \mathcal{P}_{d-h}^{(2)} \mathcal{P}_{d-h}^{(4)} \mathcal{L}^S \mathcal{L}^K \mathcal{L}^P |\phi_{\text{pair}}\rangle, \quad (4.13)$$

$$\mathcal{P}_G = \exp \left[\sum_i g_i n_{i\uparrow} n_{i\downarrow} \right], \quad (4.14)$$

$$\mathcal{P}_J = \exp \left[\frac{1}{2} \sum_{i \neq j} v_{ij} (n_i - 1)(n_j - 1) \right], \quad (4.15)$$

$$\mathcal{P}_{d-h}^{(2)} = \exp \left[\sum_t \sum_{n=0}^2 (\alpha_{2nt}^d \sum_i \xi_{i2nt}^d + \alpha_{2nt}^h \sum_i \xi_{i2nt}^h) \right], \quad (4.16)$$

$$\mathcal{P}_{d-h}^{(4)} = \exp \left[\sum_t \sum_{n=0}^4 (\alpha_{4nt}^d \sum_i \xi_{i4nt}^d + \alpha_{4nt}^h \sum_i \xi_{i4nt}^h) \right], \quad (4.17)$$

$$\mathcal{L}_S = \frac{2S+1}{8\pi^2} \int d\Omega P_S(\cos \beta) \hat{R}(\Omega), \quad (4.18)$$

$$\mathcal{L}_K = \frac{1}{N_s} \sum_{\mathbf{R}} e^{i\mathbf{K} \cdot \mathbf{R}} \hat{T}_{\mathbf{R}}, \quad (4.19)$$

$$\mathcal{L}_P = \sum_{\alpha} p_{\alpha} \hat{G}_{\alpha}, \quad (4.20)$$

で与えられます。ここで、 $\Omega = (\alpha, \beta, \gamma)$ はオイラー角、 $\hat{R}(\Omega)$ は回転演算子、 $P_S(x)$ は S 次のルジャンドル多項式、 \mathbf{K} は全運動量、 $\hat{T}_{\mathbf{R}}$ は並進ベクトル \mathbf{R} に対応する並進演算子、 \hat{G}_{α} は格子の点群演算子、 p_{α} はパリティをそれぞれ表します。ダブロン・ホロン相関因子に関する詳細は文献 [3] の説明を参照してください。また、一体部分は実空間のペア関数

$$|\phi_{\text{pair}}\rangle = \left[\sum_{i,j=1}^{N_s} \sum_{\sigma_1, \sigma_2} f_{i\sigma_1 j\sigma_2} c_{i\sigma_1}^{\dagger} c_{j\sigma_2}^{\dagger} \right]^{N/2} |0\rangle, \quad (4.21)$$

を用いた波動関数で表されます。ここで N は全電子数、 N_s は全サイト数です。最適化する変分パラメータは以下のファイルを用いて指定します (\mathcal{L}_S は **ModPara** ファイルでパラメータの指定をします)。

Gutzwiller (gutzwilleridx.def): \mathcal{P}_G のうち、最適化の対象とする変分パラメータ g_i を指定します。

Jastrow (jastrowidx.def): \mathcal{P}_J のうち、最適化の対象とする変分パラメータ v_{ij} を指定します。

DH2: $\mathcal{P}_{d-h}^{(2)}$ で表される 2 サイトのダブロン・ホロン相関因子を指定します。

DH4: $\mathcal{P}_{d-h}^{(4)}$ で表される 4 サイトのダブロン・ホロン相関因子を指定します。

Orbital/OrbitalAntiParallel (orbitalidx.def): スピンが反平行のペア軌道 $|\phi_{\text{pair}}\rangle$ を設定します。

OrbitalParallel: スピンが平行のペア軌道 $|\phi_{\text{pair}}\rangle$ を設定します。

OrbitalGeneral: ペア軌道 $|\phi_{\text{pair}}\rangle$ を設定します。

TransSym (qptransidx.def): 運動量射影 \mathcal{L}_K と格子対称性射影 \mathcal{L}_P に関する指定を行います。

(5) 変分パラメータ初期値:

変分パラメータに関する初期値を与えます。キーワード指定されない場合には 0 が初

期値として設定されます。

InGutzwiller: \mathcal{P}_G 内の変分パラメータ g_i の初期値を設定します。

InJastrow: \mathcal{P}_J 内の変分パラメータ v_{ij} の初期値を設定します。

InDH2: $\mathcal{P}_{d-h}^{(2)}$ 内の 2 サイトのダブロン・ホロン相関因子 $\alpha_{2nt}^{d(h)}$ の初期値を設定します。

InDH4: $\mathcal{P}_{d-h}^{(4)}$ 内の 4 サイトのダブロン・ホロン相関因子 $\alpha_{4nt}^{d(h)}$ の初期値を設定します。

InOrbital/InOrbitalAntiParallel: ペア軌道 $|\phi_{\text{pair}}\rangle$ の $f_{i\uparrow j\downarrow}$ に関する初期値を設定します。

InOrbitalParallel: ペア軌道 $|\phi_{\text{pair}}\rangle$ の $f_{i\sigma j\sigma}$ に関する初期値を設定します。

InOrbitalGeneral: ペア軌道 $|\phi_{\text{pair}}\rangle$ の $f_{i\sigma j\sigma_1}$ に関する初期値を設定します。

(6) 出力:

OneBodyG (greenone.def): 出力する一体 Green 関数を指定します。

TwoBodyG (greentwo.def): 出力する二体 Green 関数を指定します。

4.2.1 入力ファイル指定用ファイル (namelist.def)

計算で使用する入力ファイル一式を指定します。ファイル形式に関しては、以下のようなフォーマットをしています。

```
ModPara  modpara.def
LocSpin  zlocspn.def
Trans    ztransfer.def
InterAll zinterall.def
Orbital  orbitalidx.def
OneBodyG zcisajs.def
TwoBodyG zcisajscktaltdc.def
```

ファイル形式

[string01] [string02]

パラメータ

- [string01]
形式 : string 型 (固定)
説明 : キーワードを指定します。
- [string02]
形式 : string 型
説明 : キーワードにひも付けられるファイル名を指定します (任意)。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- キーワードを記載後、半角空白 (複数可) を開けた後にファイル名を書きます。ファイル名は自由に設定できます。
- ファイル読込用キーワードは Table4.2 により指定します。
- 必ず指定しなければいけないキーワードは ModPara, LocSpin, Orbital, TransSym です。それ以外のキーワードについては、指定がない場合はデフォルト値が採用されます (変分パラメータについては最適化されず、固定する設定となります)。詳細は各ファイルの説明を参照してください。
- 各キーワードは順不同に記述できます。
- 指定したキーワード、ファイルが存在しない場合はエラー終了します。
- # で始まる行は読み飛ばされます。

Keywords	対応するファイルの概要
ModPara*	計算用のパラメータを指定します。
LocSpin*	局在・遍歴スピンを指定します。
Trans	一般的な一体相互作用を指定します。
InterAll	一般的な二体相互作用を指定します。
CoulombIntra	内部クーロン相互作用を指定します。
CoulombInter	サイト間クーロン相互作用を指定します。
Hund	フント結合を指定します。
PairHop	ペアホッピング相互作用を指定します。
Exchange	交換相互作用を指定します。
Gutzwiller	最適化する Gutzwiller 因子を設定します。
Jastrow	最適化する電荷 Jastrow 因子を指定します。
DH2	最適化する 2 サイトダブロン・ホロン相関因子を指定します。
DH4	最適化する 4 サイトダブロン・ホロン相関因子を指定します。
Orbital*	反平行のスピンを持つペア軌道因子を指定します。
OrbitalAntiParallel	反平行のスピンを持つペア軌道因子を指定します。
OrbitalParallel	平行のスピンを持つペア軌道因子を指定します。
OrbitalGeneral	ペア軌道因子を指定します。
TransSym*	並進・格子対称演算子を設定します。
InGutzwiller	Gutzwiller 因子の初期値を設定します。
InJastrow	電荷 Jastrow 因子の初期値を設定します。
InDH2	2 サイトダブロン・ホロン相関因子の初期値を設定します。
InDH4	4 サイトダブロン・ホロン相関因子の初期値を設定します。
InOrbital	ペア軌道因子 $f_{i\uparrow j\downarrow}$ の初期値を設定します。
InOrbitalAntiParallel	ペア軌道因子 $f_{i\uparrow j\downarrow}$ の初期値を設定します。
InOrbitalParallel	ペア軌道因子 $f_{i\sigma j\sigma}$ の初期値を設定します。
InOrbitalGeneral	ペア軌道因子 $f_{i\sigma j\sigma'}$ の初期値を設定します。
OneBodyG	出力する一体グリーン関数を指定します。
TwoBodyG	出力する二体グリーン関数を指定します。

Table 4.2: *.def ファイルの一覧。* が付いているファイルは実行時に必ず必要となります。

4.2.2 ModPara ファイル (modpara.def)

計算で使用するパラメータを指定します。以下のようなフォーマットをしています。

```

-----
Model_Parameters
-----
VMC_Cal_Parameters
-----
CDataFileHead  zvo
CParaFileHead  zqp
-----
NVMCCalMode    0
NLanczosMode    0
-----
NDataIdxStart  1
NDataQtySmp    1
-----
Nsite          16
Nelectron      8
NSPGaussLeg    1
NSPStot        0
NMPTrans       1
NSROptItrStep  1200
NSROptItrSmp   100
DSROptRedCut   0.001
DSROptStaDel   0.02
DSROptStepDt   0.02
NVMCWarmUp     10
NVMCInterval   1
NVMCSample     1000
NExUpdatePath  0
RndSeed        11272
NSplitSize     1
NStore         1

```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 - 5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行: [string01] [string02]
- 7 行: [string03] [string04]
- 8 行: ヘッダ (何が書かれても問題ありません)
- 9 行以降: [string05] [int01] (もしくは [double01])

各項目の対応関係は以下の通りです。

- [string01]
形式 : string 型 (空白不可)
説明 : アウトプットファイルのヘッダを表すためのキーワード。何を指定しても問題ありません。
- [string02]
形式 : string 型 (空白不可)
説明 : アウトプットファイルのヘッダ。例えば、一体の Green 関数の出力ファイル名が **xxx_cisajs.dat** として出力されます (xxx に指定した文字が記載)。
- [string03]
形式 : string 型 (空白不可)
説明 : 最適化された変分パラメータの出力ファイル名のヘッダを表すためのキーワード。何を指定しても問題ありません。
- [string04]
形式 : string 型 (空白不可)
説明 : 最適化された変分パラメータの出力ファイル名のヘッダ。最適化された変分パラメータが **xxx_opt.dat** ファイルとして出力されます (xxx に指定した文字が記載)。
- [string05]
形式 : string 型 (固定)
説明 : キーワードの指定を行います。
- [int01] ([double01])
形式 : int (double) 型 (空白不可)
説明 : キーワードでひも付けられるパラメータを指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 9 行目以降ではキーワードを記載後、半角空白 (複数可) を開けた後に整数値を書きます。
- 9 行目以降では”-”で始まる行は読み込まれません。

キーワード

- NVMCCalMode
形式 : int 型 (デフォルト値 = 0)
説明 : [0] 変分パラメータの最適化、[1] 1 体・2 体のグリーン関数の計算。

- **NLanczosMode**

形式 : int 型 (デフォルト値 = 0)

説明 : [0] 何もしない、[1] Single Lanczos Step でエネルギーまで計算、[2] Single Lanczos Step でエネルギー、1 体・2 体のグリーン関数まで計算 (条件: 1, 2 は NVMCCalMode = 1 のみ使用可能。)

- **NDataIdxStart**

形式 : int 型 (デフォルト値 = 0)

説明 : 出力ファイルの付加番号。NVMCCalMode = 0 の場合は NDataIdxStart が出力され、NVMCCalMode = 1 の場合は、NDataIdxStart から連番で NDataQtySmp 個のファイルを出力します。

- **NDataQtySmp**

形式 : int 型 (デフォルト値 = 1)

説明 : 出力ファイルのセット数。NVMCCalMode = 1 の場合に使用します。

- **Nsite**

形式 : int 型 (1 以上、必須)

説明 : サイト数を指定する整数。

- **Nelectron**

形式 : int 型 (1 以上、必須)

説明 : 電子のペア数 (電子数は 2 Nelectron で与えられる)。

- **NCond**

形式 : int 型 (0 以上)

説明 : 伝導電子の数。

- **2Sz**

形式 : int 型

説明 : $2S_z$ の値。電子がペアを組むため、 $2S_z$ は偶数で指定する必要がある。

- **NSPGaussLeg**

形式 : int 型 (1 以上、デフォルト値 = 8)

説明 : スピン量子数射影の β 積分 (S^y 回転) の Gauss-Legendre 求積法の分点数。

- **NSPStot**

形式 : int 型 (0 以上、デフォルト値 = 0)

説明 : スピン量子数。

- **NMPTrans**

形式 : int 型 (デフォルト値 = 1)

説明 : NMPTrans の絶対値で並進・格子対称性の量子数射影の個数を指定する。負の場合は反周期境界条件を与える。TransSym ファイルで指定した重みで上から NMPTrans 個まで使用する。射影を行わない場合は 1 に設定する必要があります。

- **NSROptItrStep**
形式 : int 型 (1 以上、デフォルト値 = 1000)
説明 : SR 法で最適化する場合の全ステップ数。NVMCCalMode=0 の場合のみ使用されます。
- **NSROptItrSmp**
形式 : int 型 (1 以上数、デフォルト値 = NSROptItrStep/10)
説明 : NSROptItrStep ステップ中、最後の NSROptItrSmp ステップでの各変分パラメータの平均値を最適値とする。NVMCCalMode=0 の場合のみ使用されます。
- **DSROptRedCut**
形式 : double 型 (デフォルト値 = 0.001)
説明 : SR 法安定化因子。手法論文 [3] の ε_{wf} に対応。
- **DSROptStaDel**
形式 : double 型 (デフォルト値 = 0.02)
説明 : SR 法安定化因子。手法論文 [3] の ε に対応。
- **DSROptStepDt**
形式 : double 型
説明 : SR 法で使用する刻み幅。手法論文 [3] の Δt に対応。
- **NSROptCGMaxIter**
形式 : int 型 (デフォルト値 = 0)
説明 : SR-CG 法での、CG 法の繰り返し回数の上限。0 以下を指定した場合、最大で S 行列のサイズの数だけ実行するようになります。NSRCG!=0 の場合のみ使用されます。
- **DSROptCGTol**
形式 : double 型 (デフォルト値 = 1.0e-10)
説明 : SR-CG 法での、CG 法の収束判定条件。残差ベクトルの要素の自乗平均平方根がこの値以下になったら CG 法を終了します。NSRCG!=0 の場合のみ使用されます。
- **NVMCWarmUp**
形式 : int 型 (1 以上、デフォルト値=10)
説明 : マルコフ連鎖の空回し回数。
- **NVMCInterval**
形式 : int 型 (1 以上、デフォルト値=1)
説明 : サンプル間のステップ間隔。ローカル更新を $N_{site} \times NVMCInterval$ 回行います。
- **NVMCSample**
形式 : int 型 (1 以上、デフォルト値=1000)
説明 : 期待値計算に使用するサンプル数。

- NExUpdatePath

形式 : int 型 (0 以上)

説明 : 電子系でローカル更新で 2 電子交換を [0] 認めない、[1] 認めるの設定をします。スピン系の場合には 2 に設定する必要があります。

- RndSeed

形式 : int 型

説明 : 乱数の初期 seed。MPI 並列では各計算機に $\text{RndSeed} + \text{my rank} + 1$ で初期 seed が与えられます。

- NSplitSize

形式 : int 型 (1 以上、デフォルト値=1)

説明 : MPI 内部並列を行う場合の並列数。

- NStore

形式 : int 型 (0 もしくは 1、デフォルト値=1)

説明 : 期待値 $\langle O_k O_l \rangle$ を計算するとき行列-行列積にして高速化するオプション (1 で機能 On、モンテカルロサンプリング数に応じてメモリの消費が増大します*3)。

- NSRCG

形式 : int 型 (0 もしくは 1、デフォルト値=0)

説明 : SR 法で連立一次方程式 $Sx = g$ を解くときに、 S を陽に構築せずに解くことでメモリを削減する*4オプション [8](1 で機能 On, NStore は 1 に固定されます)。

*3使用メモリ量が、 $O(N_p^2)$ から $O(N_p^2) + O(N_p N_{\text{MCS}})$ になります。

*4使用メモリ量は、 $O(N_p) + O(N_p N_{\text{MCS}})$ です。

4.2.3 LocSpin 指定ファイル (locspn.def)

局在スピンを指定します。以下のようなフォーマットをしています。

```
=====
NlocalSpin      6
=====
=====i_0LocSpn_1IteElc =====
=====
      0      1
      1      0
      2      1
      3      0
      4      1
      5      0
      6      1
      7      0
      8      1
      9      0
     10      1
     11      0
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : 局在スピンの総数を示すキーワード (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : 局在スピンの総数を指定する整数。
- [int02]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。

- [int03]

形式 : int 型 (空白不可)

説明 : 遍歴電子か局在スピンを指定する整数 (0: 遍歴電子, 1: 局在スピン)。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と [int03] で指定される局在電子数の総数が異なる場合はエラー終了します。
- [int02] の総数が全サイト数と異なる場合はエラー終了します。
- [int02] が全サイト数以上もしくは負の値をとる場合はエラー終了します。

4.2.4 Trans 指定ファイル (trans.def)

一般的な一体相互作用をハミルトニアンに付け加え、ハミルトニアン中の相互作用パラメータ $t_{ij\sigma_1\sigma_2}$ を指定します。付け加える項は以下で与えられます。

$$\mathcal{H}_T = - \sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2} \quad (4.22)$$

以下にファイル例を記載します。

```
=====
NTransfer      24
=====
=====i_j_s_tijs=====
=====
  0      0      2      0  1.000000  0.000000
  2      0      0      0  1.000000  0.000000
  0      1      2      1  1.000000  0.000000
  2      1      0      1  1.000000  0.000000
  2      0      4      0  1.000000  0.000000
  4      0      2      0  1.000000  0.000000
  2      1      4      1  1.000000  0.000000
  4      1      2      1  1.000000  0.000000
  4      0      6      0  1.000000  0.000000
  6      0      4      0  1.000000  0.000000
  4      1      6      1  1.000000  0.000000
  6      1      4      1  1.000000  0.000000
  6      0      8      0  1.000000  0.000000
  8      0      6      0  1.000000  0.000000
...
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [int04] [int05] [double01] [double02]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : 定義するパラメータの総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 定義するパラメータの総数を指定します。
- [int02], [int04]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int03], [int05]
形式 : int 型 (空白不可)
説明 : スピンを指定する整数。
0: アップスピン
1: ダウンスピン
を選択することが出来ます。
- [double01]
形式 : double 型 (空白不可)
説明 : $t_{ij\sigma_1\sigma_2}$ の実部を指定します。
- [double02]
形式 : double 型 (空白不可)
説明 : $t_{ij\sigma_1\sigma_2}$ の虚部を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- 空行は許されません。
- [int01] と定義されている Trasfer の総数が異なる場合はエラー終了します。
- [int02]-[int05] を指定する際、範囲外の整数を指定した場合はエラー終了します。
- Hamiltonian がエルミートという制限から $t_{ij\sigma_1\sigma_2} = t_{ji\sigma_2\sigma_1}^\dagger$ の関係を満たす必要があります。

4.2.5 InterAll 指定ファイル

一般的な二体相互作用をハミルトニアンに付け加え、ハミルトニアン中の相互作用パラメータを指定します。付け加える項は以下で与えられます。

$$\mathcal{H}_I = \sum_{i,j,k,l} \sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \quad (4.23)$$

以下にファイル例を記載します。

```
=====
NInterAll      36
=====
=====zInterAll=====
=====
0   0   0   1   1   1   1   0   0.50  0.0
0   1   0   0   1   0   1   1   0.50  0.0
0   0   0   0   1   0   1   0   0.25  0.0
0   0   0   0   1   1   1   1  -0.25  0.0
0   1   0   1   1   0   1   0  -0.25  0.0
0   1   0   1   1   1   1   1   0.25  0.0
2   0   2   1   3   1   3   0   0.50  0.0
2   1   2   0   3   0   3   1   0.50  0.0
2   0   2   0   3   0   3   0   0.25  0.0
2   0   2   0   3   1   3   1  -0.25  0.0
2   1   2   1   3   0   3   0  -0.25  0.0
2   1   2   1   3   1   3   1   0.25  0.0
4   0   4   1   5   1   5   0   0.50  0.0
4   1   4   0   5   0   5   1   0.50  0.0
4   0   4   0   5   0   5   0   0.25  0.0
4   0   4   0   5   1   5   1  -0.25  0.0
4   1   4   1   5   0   5   0  -0.25  0.0
4   1   4   1   5   1   5   1   0.25  0.0
...
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [int04] [int05] [int06] [int07] [int08] [int09] [double01] [double02]

パラメータ

- [string01]

形式 : string 型 (空白不可)

説明 : 二体相互作用の総数のキーワード名を指定します (任意)。

- [int01]

形式 : int 型 (空白不可)

説明 : 二体相互作用の総数を指定します。

- [int02], [int04], [int06], [int08]

形式 : int 型 (空白不可)

説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。

- [int03], [int05], [int07], [int09]

形式 : int 型 (空白不可)

説明 : スピンを指定する整数。

0: アップスピン

1: ダウンスピン

を選択することが出来ます。

- [double01]

形式 : double 型 (空白不可)

説明 : $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ の実部を指定します。

- [double02]

形式 : double 型 (空白不可)

説明 : $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ の虚部を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- ハミルトニアンがエルミートという制限から $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} = I_{lkji\sigma_4\sigma_3\sigma_2\sigma_1}^\dagger$ の関係を満たす必要があります。
- [int01] と定義されている InterAll の総数が異なる場合はエラー終了します。
- [int02]-[int09] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.6 CoulombIntra 指定ファイル (coulombintra.def)

オンサイトクーロン相互作用をハミルトニアンに付け加えます。付け加える項は以下で与えられます。

$$\mathcal{H}_U = \sum_i U_i n_{i\uparrow} n_{i\downarrow} \quad (4.24)$$

以下にファイル例を記載します。

```
=====
NCoulombIntra 6
=====
=====i_0LocSpn_1IteElc =====
=====
0 4.000000
1 4.000000
2 4.000000
3 4.000000
4 4.000000
5 4.000000
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [double01]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : オンサイトクーロン相互作用の総数のキーワード名を指定します (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : オンサイトクーロン相互作用の総数を指定します。
- [int02]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [double01]
形式 : double 型 (空白不可)
説明 : U_i を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されているオンサイトクーロン相互作用の総数が異なる場合はエラー終了します。
- [int02] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.7 CoulombInter 指定ファイル (coulombiter.def)

サイト間クーロン相互作用をハミルトニアンに付け加えます。付け加える項は以下で与えられます。

$$\mathcal{H}_V = \sum_{i,j} V_{ij} n_i n_j \quad (4.25)$$

以下にファイル例を記載します。

```
=====
NCoulombInter 12
=====
=====CoulombInter =====
=====
    0      1      1.5000000000000000
    0      3      1.5000000000000000
    1      2      1.5000000000000000
    1      4      1.5000000000000000
    2      0      1.5000000000000000
    2      5      1.5000000000000000
    3      4      1.5000000000000000
    3      0      1.5000000000000000
    4      5      1.5000000000000000
    4      1      1.5000000000000000
    5      3      1.5000000000000000
    5      2      1.5000000000000000
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1行: ヘッダ (何が書かれても問題ありません)。
- 2行: [string01] [int01]
- 3-5行: ヘッダ (何が書かれても問題ありません)。
- 6行以降: [int02] [int03] [double01]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: サイト間クーロン相互作用の総数のキーワード名を指定します (任意)。
- [int01]
形式: int 型 (空白不可)
説明: サイト間クーロン相互作用の総数を指定します。

- [int02], [int03]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 `Nsite` 未満で指定します。
- [double01]
形式 : double 型 (空白不可)
説明 : V_{ij} を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されているオフサイトクーロン相互作用の総数が異なる場合はエラー終了します。
- [int02]-[int03] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.8 Hund 指定ファイル (hund.def)

Hund カップリングをハミルトニアンに付け加えます。付け加える項は以下で与えられます。

$$\mathcal{H}_H = - \sum_{i,j} J_{ij}^{\text{Hund}} (n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}) \quad (4.26)$$

以下にファイル例を記載します。

```
=====
NHund 6
=====
=====Hund =====
=====
    0      1 -0.250000
    1      2 -0.250000
    2      3 -0.250000
    3      4 -0.250000
    4      5 -0.250000
    5      0 -0.250000
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [double01]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : Hund カップリングの総数のキーワード名を指定します (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : Hund カップリングの総数を指定します。
- [int02], [int03]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [double01]
形式 : double 型 (空白不可)
説明 : J_{ij}^{Hund} を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている Hund カップリングの総数が異なる場合はエラー終了します。
- [int02]-[int03] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.9 PairHop 指定ファイル

PairHop カップリングをハミルトニアンに付け加えます。付け加える項は以下で与えられます。

$$\mathcal{H}_P = \sum_{i,j} J_{ij}^{\text{Pair}} (c_{i\uparrow}^\dagger c_{j\uparrow} c_{i\downarrow}^\dagger c_{j\downarrow} + c_{j\downarrow}^\dagger c_{i\downarrow} c_{j\uparrow}^\dagger c_{i\uparrow}) \quad (4.27)$$

以下にファイル例を記載します。

```
=====
NPairhop 6
=====
=====Pairhop =====
=====
  0      1  0.50000
  1      2  0.50000
  2      3  0.50000
  3      4  0.50000
  4      5  0.50000
  5      0  0.50000
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [double01]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : PairHop カップリングの総数のキーワード名を指定します (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : PairHop カップリングの総数を指定します。
- [int02], [int03]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [double01]
形式 : double 型 (空白不可)
説明 : J_{ij}^{Pair} を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている PairHop カップリングの総数が異なる場合はエラー終了します。
- [int02]-[int03] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.10 Exchange 指定ファイル (exchange.def)

Exchange カップリングをハミルトニアンに付け加えます。電子系の場合には

$$\mathcal{H}_E = \sum_{i,j} J_{ij}^{\text{Ex}} (c_{i\uparrow}^\dagger c_{j\uparrow}^\dagger c_{j\downarrow}^\dagger c_{i\downarrow} + c_{i\downarrow}^\dagger c_{j\downarrow}^\dagger c_{j\uparrow}^\dagger c_{i\uparrow}) \quad (4.28)$$

が付け加えられ、スピン系の場合には

$$\mathcal{H}_E = \sum_{i,j} J_{ij}^{\text{Ex}} (S_i^+ S_j^- + S_i^- S_j^+) \quad (4.29)$$

が付け加えられます。以下にファイル例を記載します。

```
=====
NExchange 6
=====
=====Exchange =====
=====
  0      1  0.50000
  1      2  0.50000
  2      3  0.50000
  3      4  0.50000
  4      5  0.50000
  5      0  0.50000
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [double01]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : Exchange カップリングの総数のキーワード名を指定します (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : Exchange カップリングの総数を指定します。
- [int02], [int03]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。

- [double01]
形式 : double 型 (空白不可)
説明 : J_{ij}^{Ex} を指定します。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている Exchange カップリングの総数が異なる場合はエラー終了します。
- [int02]-[int03] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.11 Gutzwiller 指定ファイル (gutzwiller.def)

Gutzwiller 因子

$$\mathcal{P}_G = \exp \left[\sum_i g_i n_{i\uparrow} n_{i\downarrow} \right] \quad (4.30)$$

の設定を行います。指定するパラメータはサイト番号 i と g_i の変分パラメータの番号です。以下にファイル例を記載します。

```
=====
NGutzwillerIdx 2
ComplexType 0
=====
=====
0      0
1      0
2      0
3      1
(continue...)
12     1
13     0
14     0
15     0
0      1
1      0
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_g は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - (5+ N_s) 行: [int03] [int04]
- (6+ N_s) - (5+ N_s + N_g) 行: [int05] [int06]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: g_i の変分パラメータの種類の総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : g_i の変分パラメータの種類の総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : g_i の変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定する整数。0 が実数、1 が複素数に対応します。
- [int03]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 N_{site} 未満で指定します。
- [int04]
形式 : int 型 (空白不可)
説明 : g_i の変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int05]
形式 : int 型 (空白不可)
説明 : g_i の変分パラメータの種類を表します (最適化有無の設定用)。0 以上 [int01] 未満で指定します。
- [int06]
形式 : int 型 (空白不可)
説明 : [int05] で指定した g_i の変分パラメータの最適化有無を設定します。最適化する場合に 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int06] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.12 Jastrow 指定ファイル (jastrow.def)

Jastrow 因子

$$\mathcal{P}_J = \exp \left[\frac{1}{2} \sum_{i \neq j} v_{ij} (n_i - 1)(n_j - 1) \right] \quad (4.31)$$

の設定を行います。指定するパラメータはサイト番号 i, j と v_{ij} の変分パラメータの番号です。以下にファイル例を記載します。

```
=====
NJastrowIdx 5
ComplexType 0
=====
=====
  0      1      0
  0      2      1
  0      3      0
  (continue...)
  0      1
  1      1
  2      1
  3      1
  4      1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_j は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - $(5 + N_s \times (N_s - 1))$ 行: [int03] [int04] [int05]
- $(6 + N_s \times (N_s - 1)) - (5 + N_s \times (N_s - 1) + N_j)$ 行: [int06] [int07]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: v_{ij} の変分パラメータの種類の総数のキーワード名を指定します (任意)。
- [int01]
形式: int 型 (空白不可)
説明: v_{ij} の変分パラメータの種類の総数を指定します。

- [string02]
形式 : string 型 (空白不可)
説明 : v_{ij} の変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : v_{ij} の変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int04]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int05]
形式 : int 型 (空白不可)
説明 : v_{ij} の変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int06]
形式 : int 型 (空白不可)
説明 : v_{ij} の変分パラメータの種類を表します (最適化有無の設定用)。0 以上 [int01] 未満で指定します。
- [int07]
形式 : int 型 (空白不可)
説明 : [int06] で指定した v_{ij} の変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int07] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.13 DH2 指定ファイル

$$\mathcal{P}_{d-h}^{(2)} = \exp \left[\sum_t \sum_{n=0}^2 (\alpha_{2nt}^d \sum_i \xi_{i2nt}^d + \alpha_{2nt}^h \sum_i \xi_{i2nt}^h) \right] \quad (4.32)$$

で表される 2 サイトの doublon-holon 相関因子の設定を行います。指定するパラメータはサイト番号 i とその周囲 2 サイト、 $\alpha_{2nt}^{d,h}$ の変分パラメータの番号で、変分パラメータは各サイト毎に t 種類設定します。各パラメータ、演算子に関する詳細は文献 [3] をご覧ください。以下にファイル例を記載します。

```
=====
NDoublonHolon2siteIdx 2
ComplexType 0
=====
=====
  0      5    15    0
  0     13     7    1
  1      6    12    0
  1     14     4    1
  (continue...)
 15      0    10    0
 15      8     2    1
  0      1
  1      1
  2      1
  (continue...)
 10      1
 11      1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_{dh2} は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - $(5 + N_s \times N_{dh2})$ 行: [int03] [int04] [int05] [int06]
- $(6 + N_s \times N_{dh2}) - (5 + (N_s + 6) \times N_{dh2})$ 行: [int07] [int08]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: 変分パラメータのセット総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 変分パラメータのセット総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : 変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int04], [int05]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int06]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int07]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します (最適化有無の設定用)。値は
 - n : 周囲の doublon(holon) 数 (0, 1, 2)
 - s : 中心が doublon の場合 0, 中心が holon の場合 1
 - t : 変分パラメータのセット番号 (0, \dots [int1]-1)
 として、 $(2n + s) \times [\text{int01}] + t$ を設定します。
- [int08]
形式 : int 型 (空白不可)
説明 : [int07] で指定した変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int08] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.14 DH4 指定ファイル

$$\mathcal{P}_{d-h}^{(4)} = \exp \left[\sum_t \sum_{n=0}^4 (\alpha_{4nt}^d \sum_i \xi_{i4nt}^d + \alpha_{4nt}^h \sum_i \xi_{i4nt}^h) \right] \quad (4.33)$$

で表される 4 サイトの doublon-holon 相関因子の設定を行います。指定するパラメータはサイト番号 i とその周囲 4 サイト、 $\alpha_{4nt}^{d,h}$ の変分パラメータの番号で、変分パラメータは各サイト毎に t 種類設定します。各パラメータ、演算子に関する詳細は文献 [3] をご覧ください。以下にファイル例を記載します。

```
=====
NDoublonHolon4siteIdx 1
ComplexType 0
=====
=====
  0      1      3      4      12      0
  1      2      0      5      13      0
  2      3      1      6      14      0
  3      0      2      7      15      0
(continue...)
 14      15      13      2      10      0
 15      12      14      3      11      0
  0      1
  1      1
(continue...)
  8      1
  9      1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_{dh4} は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - $(5 + N_s \times N_{dh4})$ 行: [int03] [int04] [int05] [int06] [int07] [int08]
- $(6 + N_s \times N_{dh4})$ - $(5 + (N_s + 10) \times N_{dh4})$ 行: [int09] [int10]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: 変分パラメータのセット総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 変分パラメータのセット総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : 変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int04], [int05], [int06], [int07]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int08]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int09]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します (最適化有無の設定用)。値は
 - n : 周囲の doublon(holon) 数 (0, 1, 2, 3, 4)
 - s : 中心が doublon の場合 0, 中心が holon の場合 1
 - t : 変分パラメータのセット番号 (0, \dots [int1]-1)
 として、 $(2n + s) \times [\text{int01}] + t$ を設定します。
- [int10]
形式 : int 型 (空白不可)
説明 : [int09] で指定した変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int10] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.15 Orbital/OrbitalAntiParallel指定ファイル (orbitalidx.def)

$$|\phi_{\text{pair}}\rangle = \left[\sum_{i,j=1}^{N_s} f_{ij} c_{i\uparrow}^\dagger c_{j\downarrow}^\dagger \right]^{N/2} |0\rangle \quad (4.34)$$

で表されるペア軌道の設定を行います。指定するパラメータはサイト番号 i, j と変分パラメータの種類を設定します。以下にファイル例を記載します。

```
=====
NOrbitalIdx 64
ComplexType 0
=====
=====
    0      0      0
    0      1      1
    0      2      2
    0      3      3
  (continue...)
   15      9     62
   15     10     63
    0      1
    1      1
  (continue...)
   62      1
   63      1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_o は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - $(5+N_s^2)$ 行: [int03] [int04] [int05] [int06]
- $(6+N_s^2) - (5+N_s^2 + N_o)$ 行: [int07] [int08]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: 変分パラメータのセット総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 変分パラメータのセット総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : 変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int04]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int05]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int06]
形式 : int 型
説明 : 反周期境界条件モードが ON (ModPara ファイルで NMPTrans が負の場合に有効) の場合、変分パラメータ f_{ij} の番号の他に符号を反転するか否かを直接指定する。
[int06]= ± 1 により符号を指定する。反周期境界条件モードが OFF の場合は省略可能。
- [int07]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します (最適化有無の設定用)。0 以上 [int01] 未満で指定します。
- [int08]
形式 : int 型 (空白不可)
説明 : [int06] で指定した変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int08] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.16 OrbitalParallel 指定ファイル

$$|\phi_{\text{pair}}\rangle = \left[\sum_{i,j=1}^{N_s} f_{i\sigma j\sigma} c_{i\sigma}^\dagger c_{j\sigma}^\dagger \right]^{N/2} |0\rangle \quad (4.35)$$

で表されるペア軌道の設定を行います。指定するパラメータはサイト番号 i, j と変分パラメータの種類を設定します。以下にファイル例を記載します。

```
=====
NOrbitalIdx 120
ComplexType 0
=====
=====
  0      1      0
  0      2      1
  0      3      2
  (continue...)
 15     13    118
 15     14    119
  0      1
  1      1
  (continue...)
 118     1
 119     1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_o は変分パラメータの種類の数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - $(5 + N_s * (N_s - 1)/2)$ 行: [int03] [int04] [int05] [int06]
- $(6 + N_s * (N_s - 1)/2) - (5 + N_s * (N_s - 1)/2 + N_o)$ 行: [int07] [int08]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: 変分パラメータのセット総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 変分パラメータのセット総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : 変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int04]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。
- [int05]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int06]
形式 : int 型
説明 : 反周期境界条件モードが ON (ModPara ファイルで NMPTrans が負の場合に有効) の場合、変分パラメータ f_{ij} の番号の他に符号を反転するか否かを直接指定する。
[int06]= ± 1 により符号を指定する。反周期境界条件モードが OFF の場合は省略可能。
- [int07]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します (最適化有無の設定用)。0 以上 [int01] 未満で指定します。
- [int08]
形式 : int 型 (空白不可)
説明 : [int06] で指定した変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int08] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.17 OrbitalGeneral 指定ファイル

$$|\phi_{\text{pair}}\rangle = \left[\sum_{i,j=1}^{N_s} \sum_{\sigma_1, \sigma_2} f_{i\sigma_1 j\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2}^\dagger \right]^{N/2} |0\rangle \quad (4.36)$$

で表されるペア軌道の設定を行います。指定するパラメータはサイト番号 i, j 、スピン σ_1, σ_2 と変分パラメータの種類を設定します。 $i + \sigma_1 N_s < j + \sigma_2 N_s (\sigma = 0, 1)$ を満たすように指定する必要があります。以下にファイル例を記載します。

```
=====
NOrbitalIdx 255
ComplexType 0
=====
=====
  0  0  0  1  0
  0  0  1  1  1
(continue...)
14  0 15 1 253
15  0 15 1 254
  0   1
  1   1
(continue...)
253  1
254  1
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_o は変分パラメータの種類の数)。変分パラメータの総数 N_p は $i + \sigma_1 N_s < j + \sigma_2 N_s (\sigma = 0, 1)$ を満たすペアの総数に対応し、 $S_z = 0$ の場合は $N_p = N_s^2$ 、 S_z 非保存の場合は $N_p = 2N_s^2 - N_s$ 個となります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3 行: [string02] [int02]
- 4-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - (5+ N_p) 行: [int03] [int04] [int05] [int06] [int07] [int08]
- (6+ N_p) - (5+ N_p + N_o) 行 : [int09] [int10]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : 変分パラメータのセット総数のキーワード名を指定します (任意)。

- [int01]
形式 : int 型 (空白不可)
説明 : 変分パラメータのセット総数を指定します。
- [string02]
形式 : string 型 (空白不可)
説明 : 変分パラメータの型を指定するためのキーワード名を指定します (任意)。
- [int02]
形式 : int 型 (空白不可)
説明 : 変分パラメータの型を指定します。0 が実数、1 が複素数に対応します。
- [int03], [int05]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 `Nsite` 未満で指定します。
- [int04], [int06]
形式 : int 型 (空白不可)
説明 : スピンを指定する整数。0 が↑スピン, 1 が↓スピンに対応します。
- [int07]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します。0 以上 [int01] 未満で指定します。
- [int08]
形式 : int 型
説明 : 反周期境界条件モードが ON(`ModPara` ファイルで `NMPTrans` が負の場合に有効) の場合、変分パラメータ f_{ij} の番号の他に符号を反転するか否かを直接指定する。
[int08]= ± 1 により符号を指定する。反周期境界条件モードが OFF の場合は省略可能。
- [int09]
形式 : int 型 (空白不可)
説明 : 変分パラメータの種類を表します (最適化有無の設定用)。0 以上 [int01] 未満で指定します。
- [int10]
形式 : int 型 (空白不可)
説明 : [int06] で指定した変分パラメータの最適化有無を設定します。最適化する場合は 1、最適化しない場合は 0 とします。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの種類の総数が異なる場合はエラー終了します。
- [int02]-[int10] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.18 TransSym 指定ファイル (qptransidx.def)

運動量射影 $\mathcal{L}_K = \frac{1}{N_s} \sum_{\mathbf{R}} e^{i\mathbf{K} \cdot \mathbf{R}} \hat{T}_{\mathbf{R}}$ と格子対称性射影 $\mathcal{L}_P = \sum_{\alpha} p_{\alpha} \hat{G}_{\alpha}$ について、重みとサイト番号に関する指定を行います。射影するパターンは (α, \mathbf{R}) で指定されます。射影を行わない場合も重み 1.0 で“恒等演算”を指定してください。以下にファイル例を記載します。

```
=====
NQPTrans 4
=====
== TrIdx_TrWeight_and_TrIdx_i_xi ==
=====
  0  1.000000
  1  1.000000
  2  1.000000
  3  1.000000
  0    0    0
  (continue...)
  3   12    1
  3   13    2
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_s はサイト数、 N_{TS} は射影演算子の種類の総数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - (5+ N_{TS}) 行: [int02] [double01]
- (6+ N_{TS}) - (5+($N_s + 1$) \times N_{TS}) 行: [int03] [int04] [int05] [int06]

パラメータ

- [string01]
形式: string 型 (空白不可)
説明: 射影パターンの総数に関するキーワード名を指定します (任意)。
- [int01]
形式: int 型 (空白不可)
説明: 射影パターンの総数を指定します。
- [int02]
形式: int 型 (空白不可)
説明: 射影パターン (α, \mathbf{R}) を指定する整数。0 以上 [int01] 未満で指定します。

- [double01]
 形式 : double 型 (空白不可)
 説明 : 射影パターン (α, \mathbf{R}) の重み $p_\alpha \cos(\mathbf{K} \cdot \mathbf{R})$ を指定します。
- [int03]
 形式 : int 型 (空白不可)
 説明 : 射影パターン (α, \mathbf{R}) を指定する整数。0 以上 [int01] 未満で指定します。
- [int04], [int05]
 形式 : int 型 (空白不可)
 説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。[int03] で指定した並進・点群移動をサイト番号 [int04] に作用させた場合の行き先が、サイト番号 [int05] となるように設定します。
- [int06]
 形式 : int 型 (空白不可)
 説明 : 反周期境界条件モードが ON(ModPara ファイルで NMPTrans が負の場合に有効) の場合、並進演算で生成消滅演算子の符号が反転するか否かを直接指定する。[int06]= ± 1 により符号を指定する。反周期境界条件モードが OFF の場合は省略可能。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている射影パターンの総数が異なる場合はエラー終了します。
- [int02]-[int06] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.19 変分パラメータ初期値指定ファイル

各変分パラメータの初期値を設定することが可能です。
 変分パラメータの種類は入力ファイル指定用ファイル (Sec. 4.2.1) において
 InGutzwiller, InJastrow, InDH2, InDH4, InOrbital, InOrbitalAntiParallel, InOrbitalParallel,
 InOrbitalGeneral
 をキーワードとして指定することで区別します。なお、ファイルフォーマットは全て共通で
 す。以下、InJastrow ファイルの例を記載します。

```
=====
NJastrowIdx  28
=====
== i_j_JastrowIdx  ==
=====
0 -8.909963465082626488e-02  0.000000000000000000e+00
1  5.521681211878626955e-02  0.000000000000000000e+00
(continue...)
27 -9.017586139930480749e-02  0.000000000000000000e+00
```

ファイル形式

以下のように行数に応じ異なる形式をとります (N_v は変分パラメータの種類の総数)。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 - (5+ N_v) 行: [int02] [double01] [double02]

パラメータ

- [string01]
 形式 : string 型 (空白不可)
 説明 : 変分パラメータ総数のキーワード名を指定します (任意)。
- [int01]
 形式 : int 型 (空白不可)
 説明 : 変分パラメータ総数を指定します。
- [int02]
 形式 : int 型 (空白不可)
 説明 : 変分パラメータの種類を指定する整数。0 以上 [int01] で指定します。
- [double01]
 形式 : double 型 (空白不可)
 説明 : 変分パラメータの初期値の実部を与えます。
- [double02] 形式 : double 型 (空白不可)
 説明 : 変分パラメータの初期値の虚部を与えます。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている変分パラメータの総数が異なる場合はエラー終了します。

4.2.20 OneBodyG 指定ファイル (greenone.def)

計算・出力する一体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ を指定します。以下にファイル例を記載します。

```
=====
NCisAjs          24
=====
===== Green functions =====
=====
  0      0      0      0
  0      1      0      1
  1      0      1      0
  1      1      1      1
  2      0      2      0
  2      1      2      1
  3      0      3      0
  3      1      3      1
  4      0      4      0
  4      1      4      1
  5      0      5      0
  5      1      5      1
  6      0      6      0
  6      1      6      1
  7      0      7      0
  7      1      7      1
  8      0      8      0
  8      1      8      1
  9      0      9      0
  9      1      9      1
 10      0     10      0
 10      1     10      1
 11      0     11      0
 11      1     11      1
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1 行: ヘッダ (何が書かれても問題ありません)。
- 2 行: [string01] [int01]
- 3-5 行: ヘッダ (何が書かれても問題ありません)。
- 6 行以降: [int02] [int03] [int04] [int05]

パラメータ

- [string01]

形式 : string 型 (空白不可)

説明 : 一体グリーン関数成分総数のキーワード名を指定します (任意)。

- [int01]

形式 : int 型 (空白不可)

説明 : 一体グリーン関数成分の総数を指定します。

- [int02], [int04]

形式 : int 型 (空白不可)

説明 : サイト番号を指定する整数。0 以上 Nsite 未満で指定します。

- [int03], [int05]

形式 : int 型 (空白不可)

説明 : スピンを指定する整数。

0: アップスピン

1: ダウンスピン

を選択することが出来ます。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている一体グリーン関数成分の総数が異なる場合はエラー終了します。
- [int02]-[int05] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.2.21 TwoBodyG 指定ファイル (greentwo.def)

計算・出力する二体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ を指定します。以下にファイル例を記載します。

```
=====
NCisAjsCktAltDC          576
=====
===== Green functions for Sq AND Nq =====
=====
  0    0    0    0    0    0    0    0
  0    0    0    0    0    1    0    1
  0    0    0    0    1    0    1    0
  0    0    0    0    1    1    1    1
  0    0    0    0    2    0    2    0
  0    0    0    0    2    1    2    1
  0    0    0    0    3    0    3    0
  0    0    0    0    3    1    3    1
  0    0    0    0    4    0    4    0
  0    0    0    0    4    1    4    1
  0    0    0    0    5    0    5    0
  0    0    0    0    5    1    5    1
  0    0    0    0    6    0    6    0
  0    0    0    0    6    1    6    1
  0    0    0    0    7    0    7    0
  0    0    0    0    7    1    7    1
  0    0    0    0    8    0    8    0
  0    0    0    0    8    1    8    1
  0    0    0    0    9    0    9    0
  0    0    0    0    9    1    9    1
  0    0    0    0   10    0   10    0
  0    0    0    0   10    1   10    1
  0    0    0    0   11    0   11    0
  0    0    0    0   11    1   11    1
  0    1    0    1    0    0    0    0
  ...
```

ファイル形式

以下のように行数に応じ異なる形式をとります。

- 1行: ヘッダ (何が書かれても問題ありません)。
- 2行: [string01] [int01]
- 3-5行: ヘッダ (何が書かれても問題ありません)。
- 6行以降: [int02] [int03] [int04] [int05] [int06] [int07] [int08] [int09]

パラメータ

- [string01]
形式 : string 型 (空白不可)
説明 : 二体グリーン関数成分総数のキーワード名を指定します (任意)。
- [int01]
形式 : int 型 (空白不可)
説明 : 二体グリーン関数成分の総数を指定します。
- [int02], [int04], [int06], [int08]
形式 : int 型 (空白不可)
説明 : サイト番号を指定する整数。0 以上 `Nsite` 未満で指定します。
- [int03], [int05], [int07], [int09]
形式 : int 型 (空白不可)
説明 : スピンを指定する整数。
0: アップスピン
1: ダウンスピン
を選択することが出来ます。

使用ルール

本ファイルを使用するにあたってのルールは以下の通りです。

- 行数固定で読み込みを行う為、ヘッダの省略はできません。
- [int01] と定義されている二体グリーン関数成分の総数が異なる場合はエラー終了します。
- [int02]-[int09] を指定する際、範囲外の整数を指定した場合はエラー終了します。

4.3 出力ファイル

出力ファイルの一覧は下記の通りです。***にはModPara ファイルの CParaFileHead で指定されるヘッダが、xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。また、zzz には ModPara ファイルの NDataIdxStart が記載されます。

ファイル名	対応するファイルの中身
***_opt.dat	最適化された全パラメータ.
***_gutzwiller_opt.dat	最適化された Gutzwiller 因子.
***_jastrow_opt.dat	最適化された Jastrow 因子.
***_doublonHolon2site_opt.dat	最適化された 2 サイトダブロン-ホロン相関因子.
***_doublonHolon4site_opt.dat	最適化された 4 サイトダブロン-ホロン相関因子.
***_orbital_opt.dat	最適化されたペア軌道因子.
xxx_out_yyy.dat	エネルギーとその分散.
xxx_var_yyy.dat	パラメータ最適化過程の情報.
xxx_CalcTimer.dat	各プロセスに対する計算時間に関する情報.
xxx_time_zzz.dat	モンテカルロサンプリングの過程に関する情報.
xxx_cisajs_yyy.dat	一体グリーン関数.
xxx_cisajsckalt_yyy.dat	二体グリーン関数.

Table 4.3: List of the output files.

4.3.1 変分パラメータ出力ファイル (***_opt.dat)

SR 法で最適化された変分パラメータとエネルギーが一斉出力されます。変分パラメータが一斉に読み込めるため、変分パラメータ最適化後の物理量の計算を行う場合に使用します。出力されるデータは

$$\langle H \rangle, \langle H^2 \rangle, g_i, v_{ij}, \alpha_{2nt}^{d(h)}, \alpha_{4nt}^{d(h)}, f_{ij}$$

で、それぞれの平均値と標準偏差が出力されます (平均値は実数、虚数の順に、標準偏差は実数のみ出力)。なお、全データが 1 行で出力され、数値の間は半角空白で区切られます。*** には ModPara ファイルの CParaFileHead で指定されるヘッダが記載されます。

4.3.2 ステップ別変分パラメータ出力ファイル (xxx_var_yyy.dat)

SR 法の各ステップにおける変分パラメータとエネルギーが zqp_opt.dat と同形式で“追記しながら”出力されます (標準偏差はゼロ) が出力されます。xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。

4.3.3 Gutzwiller 因子出力ファイル (***_gutzwiller_opt.dat)

SR 法で最適化された Gutzwiller 因子が出力されます。出力形式は Sec. 4.2.19 の InGutzwiller 指定ファイルと同じです。

4.3.4 Jastrow 因子出力ファイル (***_jastrow_opt.dat)

SR 法で最適化された Jastrow 因子が出力されます。出力形式は Sec. 4.2.19 の InJastrow 指定ファイルと同じです。

4.3.5 2 サイトダブロンホロン相関因子出力ファイル (***_doublonHolon2site_opt.dat)

SR 法で最適化された 2 サイトの doublon-holon 相関因子が出力されます。出力形式は Sec. 4.2.19 の InDH2 指定ファイルと同じです。

4.3.6 4 サイトダブロンホロン相関因子出力ファイル (***_doublonHolon4site_opt.dat)

SR 法で最適化された 4 サイトの doublon-holon 相関因子が出力されます。出力形式は Sec. 4.2.19 の InDH4 指定ファイルと同じです。

4.3.7 ペア軌道出力ファイル (***_orbital_opt.dat)

SR 法で最適化されたペア軌道の変分パラメータが出力されます。出力形式は Sec. 4.2.19 の InOrbital 指定ファイルと同じです。

4.3.8 xxx_out_yyy.dat

ビン毎の計算情報として、

$$\langle H \rangle, \langle H^2 \rangle, \frac{\langle H^2 \rangle - \langle H \rangle^2}{\langle H \rangle^2}, \langle S^z \rangle, \langle (S^z)^2 \rangle.$$

が順に出力されます。 $\langle H \rangle$ については実部と虚部がそれぞれ出力され、それ以外は実部のみ出力されます。xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。以下に出力例を記載します。

```
1.151983765704212992e+01  8.124622418360909482e-01  \
1.619082955438887268e+02  2.019905203939084959e-01
1.288482613817423150e+01  5.006903733262847433e-01  \
1.972000325276957824e+02  1.824505193695792893e-01
1.308897206011880421e+01  5.701244886956570168e-01  \
2.072610167083121837e+02  2.029162857569105916e-01
...
```

4.3.9 xxx_CalcTimer.dat

計算終了後に、処理毎の計算処理時間が処理名、処理に割り当てられた識別番号、実行秒数の順に出力されます。出力例は以下の通りです。

All	[0]	15.90724
Initialization	[1]	0.04357
read options	[10]	0.00012
ReadDefFile	[11]	0.00082
SetMemory	[12]	0.00002
InitParameter	[13]	0.03026
VMCParaOpt	[2]	15.86367
VMCMakeSample	[3]	12.85650
makeInitialSample	[30]	0.20219
make candidate	[31]	0.02553
hopping update	[32]	12.51967
UpdateProjCnt	[60]	7.41864
CalculateNewPfM2	[61]	3.67098
CalculateLogIP	[62]	0.07599
UpdateMAll	[63]	1.27466
exchange update	[33]	0.00000
UpdateProjCnt	[65]	0.00000
CalculateNewPfMTwo2	[66]	0.00000
CalculateLogIP	[67]	0.00000
UpdateMAllTwo	[68]	0.00000
recal PfM and InvM	[34]	0.08294
save electron config	[35]	0.00232
VMCMainCal	[4]	2.45481
CalculateMAll	[40]	0.47556
LocEnergyCal	[41]	0.79754
CalHamiltonian0	[70]	0.00259
CalHamiltonian1	[71]	0.18765
CalHamiltonian2	[72]	0.00107
ReturnSlaterElmDiff	[42]	0.40035
calculate OO and HO	[43]	0.68045
StochasticOpt	[5]	0.30489
preprocess	[50]	0.02587
stcOptMain	[51]	0.25471
initBLACS	[55]	0.06564
calculate S and g	[56]	0.05603
DPOSV	[57]	0.09833
gatherParaChange	[58]	0.02774
postprocess	[52]	0.02372
UpdateSlaterElm	[20]	0.02556
WeightAverage	[21]	0.06676
outputData	[22]	0.10554
SyncModifiedParameter	[23]	0.02151

4.3.10 xxx_time_zzz.dat

計算情報としてビン毎にサンプリング数、hopping および exchange のアップデートに対する acceptance ratio (acc_hopp, acc_ex)、それぞれのアップデートの試行回数 (n_hopp, n_ex) および実行した際の時間を順に出力します。xxx には CDataFileHead で指定されるヘッダが、zzz には ModPara ファイルの NDataIdxStart が記載されます。出力例は以下の通りです。

```
00000  acc_hop acc_ex  n_hop    n_ex      : Mon Jul 25 14:03:29 2016
00001  0.59688 0.00000 320      0        : Mon Jul 25 14:03:30 2016
00002  0.47727 0.00000 176      0        : Mon Jul 25 14:03:30 2016
00003  0.50000 0.00000 176      0        : Mon Jul 25 14:03:30 2016
00004  0.49432 0.00000 176      0        : Mon Jul 25 14:03:30 2016
00005  0.57386 0.00000 176      0        : Mon Jul 25 14:03:30 2016
00006  0.55114 0.00000 176      0        : Mon Jul 25 14:03:30 2016
...
```

4.3.11 xxx_cisajs_yyy.dat

OneBodyG 指定ファイルで指定された一体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ の計算結果を出力します。xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。以下にファイル例を記載します。

```
0  0  0  0  0.4452776740 0.0000000000
0  1  0  1  0.4452776740 0.0000000000
1  0  1  0  0.5000000000 0.0000000000
1  1  1  1  0.5000000000 0.0000000000
2  0  2  0  0.4452776740 0.0000000000
2  1  2  1  0.4452776740 0.0000000000
3  0  3  0  0.5000000000 0.0000000000
3  1  3  1  0.5000000000 0.0000000000
...
```

ファイル形式

- [int01] [int02] [int03] [int04] [double01] [double02]

パラメータ

- [int01], [int03]

形式 : int 型

説明 : サイト番号を指定する整数。[int01] が i サイト、[int03] が j サイトを表します。

- [int02], [int04]

形式 : int 型

説明 : スピンを指定する整数。[int02] が σ_1 、[int03] が σ_2 に対応します。

0: アップスピン

1: ダウンスピン
を表します。

- [double01], [double02]

形式 : double 型

説明 : $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ の値を表します。

[double01] が実部、[double02] が虚部を表します。

4.3.12 xxx_cisajsktalt_yyy.dat

TwoBodyG 指定ファイルで指定された二体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ の計算結果を出力します。xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。以下にファイル例を記載します。

```

0    0    0    0    0    0    0    0 0.4452776740 0.0000000000
0    0    0    0    0    1    0    1 0.1843355815 0.0000000000
0    0    0    0    1    0    1    0 0.1812412105 0.0000000000
0    0    0    0    1    1    1    1 0.2640364635 0.0000000000
0    0    0    0    2    0    2    0 0.0279690007 0.0000000000
0    0    0    0    2    1    2    1 0.2009271524 0.0000000000
0    0    0    0    3    0    3    0 0.2512810778 0.0000000000
0    0    0    0    3    1    3    1 0.1939965962 0.0000000000
...
```

ファイル形式

- [int01] [int02] [int03] [int04] [int05] [int06] [int07] [int08] [double01] [double02]

パラメータ

- [int01], [int03],[int05], [int07]

形式 : int 型

説明 : サイト番号を指定する整数。[int01] が i サイト、[int03] が j サイト、[int05] が k サイト、[int07] が l サイトを表します。

- [int02], [int04],[int06], [int08]

形式 : int 型

説明 : スピンを指定する整数。[int02] が σ_1 、[int04] が σ_2 、[int06] が σ_3 、[int08] が σ_4 に対応します。

0: アップスピン

1: ダウンスピン

を表します。

- [double01], [double02]

形式 : double 型

説明： $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ の値を表します。
 [double01] が実部、[double02] が虚部を表します。

4.3.13 xxx_ls_out_yyy.dat

Power Lanczos 法により求めた $\langle H \rangle$, $\langle H^2 \rangle$ および最適化パラメータ α の順に出力されます。ModPara 指定ファイルで NVMCCalMode=1, NLanczosmode=1, 2 に設定することで計算されます。xxx には CDataFileHead で指定されるヘッダが、yyy には ModPara ファイルの NDataIdxStart, NDataQtySmp に従い NDataIdxStart...NDataIdxStart+NDataQtySmp の順に記載されます。

4.3.14 xxx_ls_cisajs_yyy.dat

Power Lanczos 法により求めた, OneBodyG 指定ファイルで指定された一体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ の計算結果を出力します。ファイル形式は xxx_cisajs_yyy.dat ファイルと同じです。ModPara 指定ファイルで NVMCCalMode=1, NLanczosmode=2 に設定することで計算されます。

4.3.15 xxx_ls_cisajsktalt_yyy.dat

Power Lanczos 法により求めた, TwoBodyG 指定ファイルで指定された二体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ の計算結果を出力します。ファイル形式は xxx_cisajsktalt_yyy.dat ファイルと同じです。ModPara 指定ファイルで NVMCCalMode = 1, NLanczosmode = 2 に設定することで計算されます。

5

アルゴリズム

5.1 変分モンテカルロ法

変分モンテカルロ法では、試行波動関数を用意して、その試行波動関数を含むパラメータを変分原理に従って最適化することで量子多体系の基底状態 (または低励起エネルギー状態) の波動関数を近似的に求めます。試行波動関数に対する物理量の期待値を計算する部分で、マルコフ連鎖モンテカルロ法を利用し、効率よく重み付きサンプリングを行います。

本パッケージでは、サンプリングに用いる完全系として電子の実空間配置 $\{|x\rangle\}$ をとっています:

$$|x\rangle = \prod_{n=1}^{N/2} c_{r_{n\uparrow}}^\dagger \prod_{n=1}^{N/2} c_{r_{n\downarrow}}^\dagger |0\rangle \quad (5.1)$$

ここで、 $r_{n\sigma}$ は n 番目の電子 (スピン σ) の位置、 $c_{r_{n\sigma}}^\dagger$ はその位置での電子 (スピン σ) の生成演算子を表します。この基底を用いると、演算子 A の期待値は

$$\langle A \rangle = \frac{\langle \psi | A | \psi \rangle}{\langle \psi | \psi \rangle} = \sum_x \frac{\langle \psi | A | x \rangle \langle x | \psi \rangle}{\langle \psi | \psi \rangle} \quad (5.2)$$

となるため、マルコフ連鎖の重みを

$$\rho(x) = \frac{|\langle x | \psi \rangle|^2}{\langle \psi | \psi \rangle} \geq 0, \quad \sum_x \rho(x) = 1 \quad (5.3)$$

と定義して、

$$\langle A \rangle = \sum_x \rho(x) \frac{\langle \psi | A | x \rangle}{\langle \psi | x \rangle} \quad (5.4)$$

と書き直した後、 x に関する和をマルコフ連鎖モンテカルロ法により評価しています。Local Green's function $G_{ij\sigma\sigma'}(x)$ は

$$G_{ij\sigma\sigma'}(x) = \frac{\langle \psi | c_{i\sigma}^\dagger c_{j\sigma'} | \psi \rangle}{\langle \psi | x \rangle} \quad (5.5)$$

と定義されますが、これも演算子 A を $c_{i\sigma}^\dagger c_{j\sigma'}$ とすることで、同じ方法により重み付きサンプリングを行うことができます。なお、サンプリングに使用する乱数生成については、メルセンヌツイスター法を使用しています [9]。

5.2 Bogoliubov 表現

スピン系の計算において一体項 (transfer), InterAll 形式での相互作用、相関関数のインデックスの指定には Bogoliubov 表現が使われています。一般に、スピンの演算子は次の

ようにフェルミオンの生成・消滅演算子 $c_{i\sigma}$, $c_{i\sigma}^\dagger$ によって書き換えることができます:

$$S_{iz} = \sum_{\sigma=-S}^S \sigma c_{i\sigma}^\dagger c_{i\sigma} \quad (5.6)$$

$$S_i^+ = \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma+1}^\dagger c_{i\sigma} \quad (5.7)$$

$$S_i^- = \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma}^\dagger c_{i\sigma+1} \quad (5.8)$$

本パッケージでは、 $S = 1/2$ のスピン系のみ取り扱っており、上記の式で $S = 1/2$ と置いたものを用いています。

5.3 パフィアン-スレーター行列式の性質

この節では、パフィアン-スレーター行列式のもつ性質について簡単にまとめます。5.3.1 節と 5.3.2 節でパフィアン-スレーター行列式と単一スレーター行列式との関係を導出し、5.3.3 節で f_{ij} の特異値分解の意味について説明します。

5.3.1 f_{ij} と $\Phi_{in\sigma}$ の関係 (スピン反平行の場合)

多変数変分モンテカルロ法で試行波動関数の一部部分として用いられるパフィアン-スレーター行列式は

$$|\phi_{\text{Pf}}\rangle = \left(\sum_{i,j=1}^{N_s} f_{ij} c_{i\uparrow}^\dagger c_{j\downarrow}^\dagger \right)^{N_e/2} |0\rangle, \quad (5.9)$$

のように定義されます。ここで、 N_s はサイト数、 N_e は全電子数、 f_{ij} は変分パラメータです。簡単化のため、以降 f_{ij} は実数と仮定します。また、単一スレーター行列式として

$$|\phi_{\text{SL}}\rangle = \left(\prod_{n=1}^{N_e/2} \psi_{n\uparrow}^\dagger \right) \left(\prod_{m=1}^{N_e/2} \psi_{m\downarrow}^\dagger \right) |0\rangle, \quad (5.10)$$

$$\psi_{n\sigma}^\dagger = \sum_{i=1}^{N_s} \Phi_{in\sigma} c_{i\sigma}^\dagger. \quad (5.11)$$

を定義します。ただし、 Φ は正規直交基底であり、クロネッカーのデルタ δ_{nm} を用い

$$\sum_{i=1}^{N_s} \Phi_{in\sigma} \Phi_{im\sigma} = \delta_{nm}, \quad (5.12)$$

で表されます。この直交性の関係から、以下の関係式

$$[\psi_{n\sigma}^\dagger, \psi_{m\sigma}]_+ = \delta_{nm}, \quad (5.13)$$

$$G_{ij\sigma} = \langle c_{i\sigma}^\dagger c_{j\sigma} \rangle = \frac{\langle \phi_{\text{SL}} | c_{i\sigma}^\dagger c_{j\sigma} | \phi_{\text{SL}} \rangle}{\langle \phi_{\text{SL}} | \phi_{\text{SL}} \rangle} \quad (5.14)$$

$$= \sum_n \Phi_{in\sigma} \Phi_{jn\sigma}. \quad (5.15)$$

が導かれます。

次に、 $|\phi_{\text{SL}}\rangle$ を変形し、 f_{ij} と $\Phi_{in\sigma}$ の間に成り立つ関係式を示します。 $\psi_{n\sigma}^\dagger$ の交換関係を用いると、 $|\phi_{\text{SL}}\rangle$ は

$$|\phi_{\text{SL}}\rangle \propto \prod_{n=1}^{N_e/2} \left(\psi_{n\uparrow}^\dagger \psi_{\mu(n)\downarrow}^\dagger \right) |0\rangle, \quad (5.16)$$

と書き換えられます。ここで、 $\mu(n)$ は $n = 1, 2, \dots, N_e/2$ の置換を表します。議論を簡単にするため、同一のペア $n = \mu(n)$ となる場合を考えましょう。このとき、 $K_n^\dagger = \psi_{n\uparrow}^\dagger \psi_{n\downarrow}^\dagger$ として、 $K_n^\dagger K_m^\dagger = K_m^\dagger K_n^\dagger$ の関係を用いることで、

$$|\phi_{\text{SL}}\rangle \propto \prod_{n=1}^{N_e/2} \left(\psi_{n\uparrow}^\dagger \psi_{n\downarrow}^\dagger \right) |0\rangle = \prod_{n=1}^{N_e/2} K_n^\dagger |0\rangle \quad (5.17)$$

$$\propto \left(\sum_{n=1}^{N_e/2} K_n^\dagger \right)^{N_e/2} |0\rangle = \left(\sum_{i,j=1}^{N_s} \left[\sum_{n=1}^{N_e/2} \Phi_{in\uparrow} \Phi_{jn\downarrow} \right] c_{i\uparrow}^\dagger c_{j\downarrow}^\dagger \right)^{N_e/2} |0\rangle, \quad (5.18)$$

の関係が得られます。これより f_{ij} は単一スレーター行列式の係数により

$$f_{ij} = \sum_{n=1}^{N_e/2} \Phi_{in\uparrow} \Phi_{jn\downarrow}. \quad (5.19)$$

として表されることが分かります。なお、この形式は単一スレーター行列式で与えられる f_{ij} の表式の一つであり、実際にはペアを組む自由度 (どの $\mu(n)$ を選ぶか) およびゲージの自由度 (すなわち $\Phi_{in\sigma}$ の符号の自由度) に依存します。この自由度の多さが f_{ij} の冗長性につながっています。

5.3.2 F_{IJ} と $\Phi_{In\sigma}$ の関係 (スピン平行も含めた場合)

前節で考察したパフィアン-スレーター波動関数と単一スレーター波動関数の間の関係は、同種スピンのペアリングも考えた場合に拡張することができます。パフィアン-スレーター波動関数とスレーター波動関数をそれぞれ

$$|\phi_{\text{Pf}}\rangle = \left(\sum_{I,J=1}^{2N_s} F_{IJ} c_I^\dagger c_J^\dagger \right)^{N_e/2} |0\rangle, \quad (5.20)$$

$$|\phi_{\text{SL}}\rangle = \left(\prod_{n=1}^{N_e} \psi_n^\dagger \right) |0\rangle, \quad \psi_n^\dagger = \sum_{I=1}^{2N_s} \Phi_{In} c_I^\dagger. \quad (5.21)$$

と定義します。ここで I, J はスピン自由度も含めたサイトのインデックスです。スピン反平行の場合とはほぼ同様の議論を用いることで、

$$F_{IJ} = \sum_{n=1}^{N_e/2} \left(\Phi_{In} \Phi_{Jn+1} - \Phi_{Jn} \Phi_{In+1} \right). \quad (5.22)$$

の関係を示すことができます。これはスピン反平行のペアリングにもそのまま適用できるので、mVMC ver1.0 以降ではこの表式を使用しています。

5.3.3 f_{ij} の特異値分解

行列 $F, \Phi_\uparrow, \Phi_\downarrow, \Sigma$ を

$$(F)_{ij} = f_{ij}, \quad (\Phi_\uparrow)_{in} = \Phi_{in\uparrow}, \quad (\Phi_\downarrow)_{in} = \Phi_{in\downarrow}, \quad (5.23)$$

$$\Sigma = \text{diag}[\underbrace{1, \dots, 1}_{N_e/2}, 0, 0, 0], \quad (5.24)$$

として定義します。前節のように f_{ij} (すなわち F) が単一スレーター行列と関係づけられているとき、 F の特異値分解は

$$F = \Phi_\uparrow \Sigma \Phi_\downarrow^t. \quad (5.25)$$

となることを示すことができます。この結果は、一般に F を特異値分解したとき、非ゼロの特異値が $N_e/2$ 個存在し、かつ全ての F の非ゼロの特異値が 1 であった場合、 f_{ij} が単一スレーター波動関数を記述すること (つまり平均場近似解として記述できること) を表しています。言い換えると、特異値の非ゼロ成分の数とその値が、シングルスレーター行列式からパフィアンスレーター行列式がどのようにしてずれるのか、という点について定量的な基準を与えることを示しています。

5.4 Power-Lanczos 法

このセクションでは、Power-Lanczos 法での最適化パラメータ α の決定方法について述べます。また、ここではシングルステップの Lanczos 法を適用した後の物理量の計算についても説明します。

5.4.1 α の決定

最初に、変分モンテカルロ法のサンプリングに関して簡単に説明します。物理量 \hat{A} は以下の手順で計算されます:

$$\langle \hat{A} \rangle = \frac{\langle \phi | \hat{A} | \phi \rangle}{\langle \phi | \phi \rangle} = \sum_x \rho(x) F(x, \hat{A}), \quad (5.26)$$

$$\rho(x) = \frac{|\langle \phi | x \rangle|^2}{\langle \phi | \phi \rangle}, \quad F(x, \hat{A}) = \frac{\langle x | \hat{A} | \phi \rangle}{\langle x | \phi \rangle}. \quad (5.27)$$

演算子の積 $\hat{A}\hat{B}$ を計算する場合には、以下の二通りの方法があります。

$$\langle \hat{A}\hat{B} \rangle = \sum_x \rho(x) F(x, \hat{A}\hat{B}), \quad (5.28)$$

$$\langle \hat{A}\hat{B} \rangle = \sum_x \rho(x) F^\dagger(x, \hat{A}) F(x, \hat{B}). \quad (5.29)$$

後述するように、後者の表記の方が数値的には安定します。例えば、エネルギーの期待値の分散 $\sigma^2 = \langle (\hat{H} - \langle \hat{H} \rangle)^2 \rangle$ を考えてみると、分散は以下の2通りの方法で計算できます。

$$\sigma^2 = \sum_x \rho(x) F(x, (\hat{H} - \langle \hat{H} \rangle)^2) = \sum_x \rho(x) F(x, \hat{H}^2) - \left[\sum_x \rho(x) F(x, \hat{H}) \right]^2, \quad (5.30)$$

$$\begin{aligned} \sigma^2 &= \sum_x \rho(x) F^\dagger(x, \hat{H} - \langle \hat{H} \rangle) F(x, \hat{H} - \langle \hat{H} \rangle) \\ &= \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{H}) - \left[\sum_x \rho(x) F(x, \hat{H}) \right]^2 \end{aligned} \quad (5.31)$$

この定義から、後者の方法では常に正の値となることが保証されているのに対して、前者の方法では分散が正の値になることが必ずしも保証されないことが分かります。次に、シングルステップでの power-Lanczos 波動関数 $|\phi\rangle = (1 + \alpha \hat{H})|\psi\rangle$ に対するエネルギーの期待値とその分散を考えます。エネルギーは以下の式で計算されます：

$$E_{LS}(\alpha) = \frac{\langle \phi | \hat{H} | \phi \rangle}{\langle \phi | \phi \rangle} = \frac{h_1 + \alpha(h_{2(20)} + h_{2(11)}) + \alpha^2 h_{3(12)}}{1 + 2\alpha h_1 + \alpha^2 h_{2(11)}}, \quad (5.32)$$

ここで、 h_1 , $h_{2(11)}$, $h_{2(20)}$, and $h_{3(12)}$ を以下のように定義しました：

$$h_1 = \sum_x \rho(x) F^\dagger(x, \hat{H}), \quad (5.33)$$

$$h_{2(11)} = \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{H}), \quad (5.34)$$

$$h_{2(20)} = \sum_x \rho(x) F^\dagger(x, \hat{H}^2), \quad (5.35)$$

$$h_{3(12)} = \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{H}^2). \quad (5.36)$$

$\frac{\partial E_{LS}(\alpha)}{\partial \alpha} = 0$ の条件から α の二次方程式が導出され、それを解くことで α が決定されます。分散に関しても同様の手法で計算することが可能です。

5.4.2 物理量の計算

最適化パラメータ α を用いることで、演算子 \hat{A} の期待値を以下の式から計算することが出来ます：

$$A_{LS}(\alpha) = \frac{\langle \phi | \hat{A} | \phi \rangle}{\langle \phi | \phi \rangle} = \frac{A_0 + \alpha(A_{1(10)} + A_{1(01)}) + \alpha^2 A_{2(11)}}{1 + 2\alpha h_1 + \alpha^2 h_{2(11)}}, \quad (5.37)$$

ここで、 A_0 , $A_{1(10)}$, $A_{1(01)}$, and $A_{2(11)}$ は

$$A_0 = \sum_x \rho(x) F(x, \hat{A}), \quad (5.38)$$

$$A_{1(10)} = \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{A}), \quad (5.39)$$

$$A_{1(01)} = \sum_x \rho(x) F(x, \hat{A} \hat{H}), \quad (5.40)$$

$$A_{2(11)} = \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{A} \hat{H}). \quad (5.41)$$

として定義される変数を表します。プログラムでは、この表式に基づき一体グリーン関数および二体グリーン関数の計算を行っています。

6

謝辞

mVMCの開発は田原大資氏が開発された変分モンテカルロ法のコードをもとに行いました。コードを提供していただいた田原大資氏に感謝します。また、田原大資氏のコードを改良していく段階で、品岡寛氏、山地洋平氏、栗田萌氏、金子隆威氏に協力して頂きました。この場を借りて、この方々に感謝します。

mVMC ver.0.1, ver. 0.2, ver. 1.0 は、東京大学物性研究所 ソフトウェア高度化プロジェクト (2016 年度) の支援を受け開発されました。この場を借りて感謝します。

A

非制限 Hartree-Fock 近似プログラム

mVMC では補助プログラムとして、多変数変分モンテカルロ法のペア軌道 f_{ij} の初期値を非制限 Hartree-Fock(UHF) 近似から与えるためのプログラムを用意しています (対応関係は 5.3 を参照)。なお、本プログラムは遍歴電子系を対象としており、スピン系、近藤系では正しく動作しません。

A.1 概要

UHF 近似では揺らぎ $\delta A \equiv A - \langle A \rangle$ の一次までを考慮することで、二体項を一体項へと近似します。たとえば、サイト間クーロン相互作用

$$\mathcal{H}_V = \sum_{i,j,\sigma,\sigma'} V_{ij} n_{i\sigma} n_{j\sigma'} \quad (\text{A.1})$$

について考えます。簡単化のため、 $i \equiv (i, \sigma)$, $j \equiv (j, \sigma')$ とすると相互作用の項は揺らぎの二次を落とすことで、

$$\begin{aligned} n_i n_j &= (\langle n_i \rangle + \delta n_i)(\langle n_j \rangle + \delta n_j) - \left[\langle c_i^\dagger c_j \rangle + \delta(c_i^\dagger c_j) \right] \left[\langle c_j^\dagger c_i \rangle + \delta(c_j^\dagger c_i) \right] \\ &\sim \langle n_i \rangle n_j + \langle n_j \rangle n_i - \langle c_i^\dagger c_j \rangle c_j^\dagger c_i - \langle c_j^\dagger c_i \rangle c_i^\dagger c_j - \langle n_i \rangle \langle n_j \rangle + \langle c_j^\dagger c_i \rangle \langle c_i^\dagger c_j \rangle \end{aligned} \quad (\text{A.2})$$

と近似されます。このような形式で、その他の相互作用についても近似を行うことで、一体問題に帰着させることができます。計算では、上記の各平均値が self-consistent になるまで計算を行います。

A.1.1 ソースコード

ソースコード一式は `src/ComplexUHF/src` 以下に入っています。

A.1.2 コンパイル方法

コンパイルは mVMC のコンパイルと同様に mVMC のフォルダ直下で

```
$ make mvmc
```

を実行することで行われます。コンパイルが終了すると、`src/ComplexUHF/src` に実行ファイル UHF が作成されます。

A.1.3 必要な入力ファイル

入力ファイル指定用ファイル (`namelist.def`)

UHF で指定するファイルは以下のファイルです。`namelist.def` は 4.2.1 で定義されているファイルと同じ様式です。

- ModPara
- LocSpin
- Trans
- CoulombIntra
- CoulombInter
- Hund
- PairHop
- Exchange
- Orbital/OrbitalAntiParallel
- OrbitalParallel
- OrbitalGeneral
- Initial

基本的には mVMC と同じファイルとなりますが、

- ModPara ファイルで指定されるパラメータ
- Initial ファイルの追加

が mVMC と異なります。以下、その詳細を記載します。

ModPara ファイルで指定するパラメータ

UHF で指定するパラメータは以下のパラメータです。

- Nsite
- Ne
- Mix
- EPS
- IterationMax

Nsite, Ne は mVMC と共通のパラメータで、以下の三つが UHF 独特のパラメータです。

- Mix
linear mixing を double 型で指定します。mix=1 とすると完全に新しい Green 関数に置き換えられます。
- EPS
収束判定条件を int 型で指定します。新しく計算された Green 関数と一つ前の Green 関数の残差が $10^{-\text{EPS}}$ の場合に、計算が打ち切られます。
- IterationMax
ループの最大数を int 型で指定します。

なお、mVMC で使用するその他パラメータが存在する場合は Warning が標準出力されます (計算は中断せずに実行されます)。

Initial ファイル

グリーン関数 $G_{ij\sigma_1\sigma_2} \equiv \langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ の初期値を与えます。ファイル様式は **Trans** ファイルと同じで、 $t_{ij\sigma_1\sigma_2}$ の代わりに $G_{ij\sigma_1\sigma_2}$ の値を記述します。なお、値を指定しないグリーン関数には 0 が入ります。

A.2 使用方法

UHF 自体は mVMC と同じように

```
$ UHF namelist.def
```

で動きます。計算の流れは以下の通りです。

1. ファイル読み込み
2. ハミルトニアン作成
3. グリーン関数の計算 (self-consistent になるまで)
4. f_{ij} 、各種ファイルの出力

計算後に出力されるファイルおよび出力例は以下の通りです。

- zvo_result.dat: エネルギーと粒子数が出力されます。

```
energy -15.2265348135
num    36.0000000000
```

- zvo_check.dat: イタレーションのステップ数、グリーン関数の残差の絶対値の平均、収束過程のエネルギー、粒子数を順に出力します。

```
0  0.004925645652 -544.963484605164 36.000000
1  0.002481594941 -278.304285708488 36.000000
2  0.001274395448 -147.247026925130 36.000000
3  0.000681060599 -82.973664527606 36.000000
...
```

- zvo_UHF_cisajs.dat: 収束した一体グリーン関数 $G_{ij\sigma_1\sigma_2} \equiv \langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ 。全成分について $i, \sigma_1, j, \sigma_2, \text{Re}[G_{ij\sigma_1\sigma_2}], \text{Im}[G_{ij\sigma_1\sigma_2}]$ の順に出力されます。

```
0  0  0  0  0.5037555283 0.0000000000
0  0  0  1  0.4610257618 0.0003115503
0  1  0  0  0.4610257618 -0.0003115503
0  1  0  1  0.4962444717 0.0000000000
...
```

- zvo_eigen.dat: 収束したハミルトニアン固有値が低エネルギー順に出力されます。

```
1  -2.9425069199
2  -2.9425069198
3  -1.5005359205
...
```

- zvo_gap.dat: 全電子数を N_{tot} とした場合に、 $\Delta E = E(N_{\text{tot}} + 1) - E(N_{\text{tot}})$ が出力されます。

```
5.2208232631
```

- zvo_orbital_opt.dat: スレータ行列式から生成した f_{ij} 。InOrbital, InOrbitalAntiParallel, InOrbitalParallel, InOrbitalAntiGeneral ファイルと同じ形式のファイルが出力されます。 f_{ij} が Orbital, OrbitalAntiParallel, OrbitalParallel, OrbitalAntiGeneral ファイルを参照し計算され、同種のパラメータについては平均化した値が採用されます。

References

- [1] M. Imada, T. Miyake, J. Phys. Soc. Jpn. **79**, 112001 (2010).
- [2] C. Gros, Ann. Phys. **189**, 53–88 (1989).
- [3] D. Tahara, M. Imada, Journal of the Physical Society of Japan **77**, 114701 (2008).
- [4] T. Misawa, M. Imada, Phys. Rev. B **90**, 115137 (2014).
- [5] S. Morita, R. Kaneko, M. Imada, J. Phys. Soc. Jpn. **84**, 024720 (2015).
- [6] <http://ma.cms-initiative.jp/ja/listapps/hphi>.
- [7] M. Wimmer: ACM Trans. Math. Software 38 30 (2012).
- [8] E. Neuscamman, C. J. Umrigar, G. K.-L. Chan, Phys. Rev. B **85**, 045103 (2012).
- [9] <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT>.