

HPhi++ Documentation

リリース **3.2.0**

HPhi++ team

2019 年 05 月 06 日

目次

第 1 章	HPhi++ とは?	1
第 2 章	ライセンス	3
第 3 章	コピーライト	5
第 4 章	ダウンロード	7
第 5 章	Contents	9
5.1	What is HPhi++?	9
5.1.1	HPhi++ とは?	9
5.2	How to use HPhi++	12
5.2.1	要件	12
5.2.2	インストール方法	12
5.2.3	ディレクトリ構成	14
5.2.4	基本的な使い方	14
5.3	チュートリアル	17
5.3.1	スタンダードモード	17
5.3.2	エキスパートモード	24
5.3.3	相関関数のフーリエ変換	30
5.4	ファイルフォーマット	30
5.4.1	スタンダードモード用入力ファイル	30
5.4.2	エキスパートモード用入力ファイル	50
5.4.3	出力ファイル	98
5.4.4	エラーメッセージ一覧	147
5.5	アルゴリズム	148
5.5.1	Lanczos 法	148
5.5.2	完全対角化	151
5.5.3	熱的純粋量子状態による有限温度計算	151
5.5.4	動的グリーン関数	152
5.5.5	実時間発展	153
5.5.6	Bogoliubov 表現	153
5.6	Calculation of physical quantities	153

5.6.1	Calculation of specific heat	154
5.6.2	Calculation of spectrum function	154
5.6.3	Calculation of magnetization curve	154
.1	相関関数の Fourier 変換ユーティリティー	154
.1.1	概要	154
.1.2	静的相関関数についてのチュートリアル	155
.1.3	動的相関関数についてのチュートリアル	157
.1.4	ファイルフォーマット	159
.1.5	greenr2k ユーティリティーの動作について	162
.1.6	dynamicalr2k ユーティリティーの動作について	163
.1.7	Contact	163
.2	Wannier 関数を用いたダウンフォールディング	164
.2.1	概要	164
.2.2	チュートリアル	164
.2.3	スタンダードモードの入力パラメーター	171
.2.4	ファイルフォーマット	172
.2.5	Contact	173
.3	謝辞	173

第 1 章

HPhi++ とは？

並列計算機に対応した数値厳密対角化法による有効モデルソルバーパッケージ。広汎な多体量子系の有効モデル (多軌道ハバード模型、ハイゼンベルグ模型、近藤格子模型など) の基底状態及び低励起状態の波動関数を並列計算によって求められます。ランチョス法による基底状態計算、熱的純粋量子状態を利用した比熱・帯磁率の温度依存性計算が可能です。さらに、ver.2.0 より数値ライブラリ $K\omega$ が接続され、先端的数値アルゴリズム (シフト型クリロフ部分空間理論) による動的グリーン関数の計算が可能となっています。

第 2 章

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。

HPhi++ (hphi) を引用する際には、以下の文献を引用してください。

Mitsuaki Kawamura, Kazuyoshi Yoshimi, Takahiro Misawa, Youhei Yamaji, Synge Todo, and Naoki Kawashima

Comp. Phys. Commun. 217 (2017) 180-192.

第 3 章

コピーライト

© 2015- The University of Tokyo. All rights reserved.

本ソフトウェアは 2015, 2016, 2017 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おり、その著作権は東京大学が所持しています。

第 4 章

ダウンロード

HPhi++ のソースコードは [GitHub page](#) or [release page](#) からダウンロードできます。

第 5 章

Contents

5.1 What is HPhi++?

5.1.1 HPhi++ とは？

実験データと理論模型の解析結果との直接比較は、物質科学の研究プロセスの一つの核となっています。例えば、低エネルギー励起構造を反映する比熱の温度依存性や、帯磁率から見積もった有効スピン・モーメントおよびキュリー・ワイス温度等の実験データからその背後にある電子状態を理解するためには、理論模型の解析結果との比較が必要不可欠です。

理論模型の解析手法としては、厳密対角化法 \hat{A}^{*1} が近似を用いず定量的な計算ができる最も信頼できる手法です。この目的のためには、東京工業大学の西森秀稔教授が作られた固有値問題ソルバパッケージ TITPACK が長年広く利用されてきました。これまでは計算機資源の制約もあり、その利用は比較的小さな系への適用に限られてきました。しかし、計算機の進歩によって、並列化を利用してハバード模型 18 格子点、スピン 1/2 の格子スピン模型 36 格子点程度の計算が容易に実行可能となっています。また、量子統計力学の発展 $\hat{A}^{*2*3*4*5}$ により、基底状態計算と同程度のコストでの有限温度計算が可能となり、比熱や帯磁率の実験との定量的比較が十分可能となってきています \hat{A}^{*6} 。これらの計算が可能になった背景には低バンド幅で多数の分散メモリ計算コアを持つ新しい計算機アーキテクチャがあり、その性能を最大限に生かしつつ、簡便に利用可能な新しいソフトウェアの登場が待たれていました。

このような背景に基づき、汎用量子模型ソルバーパッケージ HPhi++ は、ランチョス法に基づく量子多体模型の基底状態および低励起状態に対する厳密対角化法と、熱的純粋量子状態 \hat{A}^{*5} を基礎とした有限温度計算を、簡便かつ柔軟なユーザー・インターフェイスとともに並列実装されたアプリケーションを目指して開発されました。シンブルなハバード模型やハイゼンベルグ模型に始まり、多軌道ハバード模型やジャロシンスキー-守谷相互作用やキタエフ項のように SU(2) 対称性を破る相互作用を持つ量子スピン模型、さらに遍歴電子と局在スピンが結合した近

^{*1} E. Dagotto, Rev. Mod. Phys. **66**, 763-840 (1994).

^{*2} M. Imada, M. Takahashi, Journal of the Physical Society of Japan **55**, 3354-3361 (1986).

^{*3} J. Jakli, P. Prelov, Phys. Rev. B **49**, 5065-5068 (1994).

^{*4} A. Hams, H. De Raedt, Phys. Rev. E **62**, 4365-4377 (2000).

^{*5} S. Sugiura, A. Shimizu, Phys. Rev. Lett. **108**, 240401 (2012).

^{*6} Y. Yamaji, Y. Nomura, M. Kurita, R. Arita, M. Imada, Phys. Rev. Lett. **113**, 107201 (2014).

藤格子模型まで、ユーザーの興味に応じて広汎な量子格子模型を解析することができます。基底状態および有限温度での内部エネルギーはもちろん、比熱や電荷・スピン構造因子を始めとする様々な物理量が計算可能となっています。実験研究者を含む幅広いユーザーに気楽にご利用いただければ幸いです。

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。HPhi++ (hphi) を引用する際には、以下の文献を引用してください。

Mitsuaki Kawamura, Kazuyoshi Yoshimi, Takahiro Misawa, Youhei Yamaji, Synge Todo, and Naoki Kawashima, *Comp. Phys. Commun.* 217 (2017) 180-192.

コピーライト

Â©2015- The University of Tokyo. All rights reserved.

本ソフトウェアは 2015, 2016, 2017 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おり、その著作権は東京大学が所持しています。

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.3.1 (2018/9/3 リリース)
 - 開発者
 - * 三澤 貴宏 (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)
 - * 井戸 康太 (東京大学大学院 工学系研究科)
 - * 山地 洋平 (東京大学大学院 工学系研究科)
 - * 藤堂 眞治 (東京大学 理学系研究科)
 - * 小西 優祐 (株式会社 アカデメイア)
 - プロジェクトコーディネーター
 - * 川島 直輝 (東京大学 物性研究所)

- ver.3.0 (2017/12/25 リリース)
 - 開発者

- * 三澤 貴宏 (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)
- * 井戸 康太 (東京大学大学院 工学系研究科)
- * 山地 洋平 (東京大学大学院 工学系研究科)
- * 藤堂 眞治 (東京大学 理学系研究科)
- プロジェクトコーディネーター
 - * 川島 直輝 (東京大学 物性研究所)
- ver.2.0 (2017/4/11 リリース)
- ver.1.2 (2016/11/14 リリース)
- ver.1.1 (2016/5/13 リリース)
- ver.1.0 (2016/4/5 リリース)
- 開発者
 - * 三澤 貴宏 (東京大学大学院 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)
 - * 山地 洋平 (東京大学大学院 工学系研究科)
 - * 藤堂 眞治 (東京大学 理学系研究科)
- プロジェクトコーディネーター
 - * 川島 直輝 (東京大学 物性研究所)

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」
- 富士通 FX-10 および京コンピューター
- Linux PC + intel コンパイラ
- Linux PC + gcc

- Mac + gcc

5.2 How to use HPhi++

5.2.1 要件

HPhi++ のコンパイルおよび使用には次のものがが必要です。

- C/fortran コンパイラ (インテル、富士通、GNU など)
- BLAS/LAPACK ライブラリ (インテル MKL, 富士通, ATLAS など)
- MPI ライブラリ (MPI 並列を行わない場合は必要ありません)
- ScaLAPACK ライブラリ (全対角化で使わない場合は必要ありません)
- MAGMA ライブラリ (全対角化で使わない場合は必要ありません)

ちなみに:

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。

5.2.2 インストール方法

HPhi++ は次の場所からダウンロードできます。

<https://github.com/QLMS/HPhi/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 HPhi-xxx.tar.gz
```

HPhi++ は cmake を利用してインストールできます。

ちなみに:

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。

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

```
cd $HOME/build/hphi
cmake -DCONFIG=gcc $PathTohphi
make
```

でコンパイルすることができます。全対角化の計算に ScaLAPACK を使用する場合には、`-DUSE_SCALAPACK=ON` のオプションを cmake 時に付けてください。コンパイル後、\$HOME/build/hphi 直下に src フォルダが作成され、実行ファイルである HPhi++ がそのフォルダ内に作成されます。MPI ライブラリがない場合には、MPI 非対応の実行ファイルが作成されます。

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"
- sekirei_acc : 物性研究所システム B "sekirei" (MAGMA ライブラリを使用する場合)
- 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、HPhi++ 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、HPhi++ が src フォルダ内に作成されます。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。また、SSE2 が使用出来る場合には、cmake でのコンパイル時 `-DHAVE_SSE2` を付け加えてください。

5.2.3 ディレクトリ構成

HPhi-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。

```
|--CMakeLists.txt
|--COPYING
|--config/
|   |--fujitsu.cmake
|   |--gcc.cmake
|   |--intel.cmake
|   |--sekirei.cmake
|   |--sekirei_acc.cmake
|--doc/
|   |--en/
|   |--jp/
|   |--fourier/
|   |--sphinx/
|   |--userguide_en.pdf
|   |--userguide_jp.pdf
|--samples/
|--src/
|   |--*.c
|   |--CMakeLists.txt
|   |--include/*.h
|   |--makefile_src
|   |--StdFace/
|--test/
|--tool/
```

5.2.4 基本的な使い方

HPhi++ ではスタンダードモードとエキスパートモードの 2 つのモードが存在します。ここでは、スタンダードモードおよびエキスパートモードでの計算に関して、それぞれ基本的な流れを記載します。

スタンダードモード

スタンダードモードでの動作方法は下記の通りです。

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用入力ファイルの作成

スタンダードモードでは、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) と計算手法 (Lanczos 法、TPQ 法など) を設定します。各ファイルは [How to use HPhi++](#) に従い記載してください。

3. 実行

“-s” (“--standard” でも可) をオプションとして指定の上、1 で作成した入力ファイル名を引数とし、HPhi++ を実行します。

- シリアル/OpenMP 並列 の場合

```
$ Path /HPhi++ -s Input_file_name
```

- MPI 並列/ハイブリッド並列 の場合

```
$ mpiexec -np number_of_processes Path /HPhi++ -s Input_file_name
```

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。プロセス数の指定に関しては計算する系により固定のものに設定する必要があります。詳細は [プロセス数の設定](#) を参照ください。

4. 途中経過

計算実行の経過について output フォルダ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 [出力ファイル](#) 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 フォルダ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 [出力ファイル](#) を参考にしてください。

ちなみに:

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、HPhi++ 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
```

エキスパートモード

エキスパートモードでの動作方法は下記の通りです。

1. 計算用ディレクトリの作成

計算シナリオ名 (名前は任意) を記載したディレクトリを作成します。

2. 詳細入力ファイルの作成

エキスパートモードでは、ハミルトニアンすべての項を記述する詳細入力ファイルと計算条件のファイル、] およびそれらのファイル名のリストファイルを作成します。各ファイルは[エキスパートモード用入力ファイル](#)に従い記載してください。

注釈:

リストファイルの作成はスタンダード用のファイル StdFace.def を用いると容易に作成することができます。

3. 実行

“-e” (“--expert”でも可) をオプションとして指定の上、2 で作成した入力リストファイル名を引数とし、ターミナルから HPhi++ を実行します。

- シリアル/OpenMP 並列

```
$ Path/HPhi++ -e Input_List_file\_name
```

- MPI 並列/ハイブリッド並列

```
$ mpiexec -np number_of_processes Path/HPhi++ -e  
Input_List_file_name
```

プロセス数の指定に関しては計算する系により固定のものに設定する必要があります。

詳細は [プロセス数の設定](#) を参照ください。

4. 途中経過

計算実行の経過について output フォルダ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 [出力ファイル](#) 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 フォルダ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 [出力ファイル](#) を参考にしてください。

プロセス数の設定

MPI 並列/ハイブリッド並列を用いる場合、プロセス数は以下のように設定してください。

1. Standard モード

- 電子系及び近藤格子系

スタンダードモード用入力ファイルで model="Fermion Hubbard", "Kondo Lattice", "Fermion HubbardGC" の場合は、プロセス数が 4^n となるように設定してください。

- スピン系

スタンダードモード用入力ファイルで `model="Spin", "SpinGC"` の場合は、入力ファイルの $2S$ の値に対してプロセス数が $(2S+1)^n$ となるように設定してください (デフォルトは $2S=1$)。

2. Expert モード

- 電子系及び近藤格子系

CalcMod ファイルの **CalcMod** ファイルで、`CalcModel` として fermion Hubbard 模型、近藤模型を選択した場合は、プロセス数が 4^n となるように設定してください。

- スピン系

CalcMod ファイルの **CalcMod** ファイルで、`CalcModel` としてスピン模型を選択した場合は、*LocSpin* 指定ファイルの **LocSpin** ファイルを参考にプロセス数を指定する必要があります。許容されるプロセス数は、サイト数の大きいものから順に局在スピンの状態数 $(2S+1)$ を掛けたもので指定されます。

例えば、**LocSpin** ファイルが

```
=====
NlocalSpin      3
=====
=====i_0IteElc_2S =====
=====
      0      3
      1      2
      2      1
```

で与えられる場合、許容されるプロセス数は $2 = 1 + 1$, $6 = 2 \times (2 + 1)$, $24 = 6 \times (3 + 1)$ となります。

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 HPhi++ 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。

```
$ PATH/HPhi++ -v
```

5.3 チュートリアル

5.3.1 スタンダードモード

Heisenberg 模型

以下のチュートリアルはディレクトリ

```
samples/CG/Heisenberg/
```

内で行います。Heisenberg 模型におけるサンプル入力ファイルは

```
samples/CG/Heisenberg/stan.in
```

にあります。この例では 2 次元正方格子の Heisenberg 模型 (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。

$$\hat{\mathcal{H}} = J \sum_{i,j=1}^4 (\hat{S}_{ij} \cdot \hat{S}_{i+1j} + \hat{S}_{ij} \cdot \hat{S}_{ij+1},)$$

ただし、周期境界条件 ($S_{15} = S_{51} = S_{11}$) を採用します。インプットファイルの中身は次のとおりです。

```
model = "Spin"
method = "CG"
lattice = "square"
W = 4
L = 4
J = 1.0
2Sz = 0
```

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。

Log 出力

標準出力でログが出力されます。また、"output" ディレクトリが自動生成され、そこにも計算経過を示すログが出力されます。例えば、サンプル実行時には以下のファイルが出力されます。

```
CHECK_InterAll.dat      Time_CG_EigenVector.dat  zvo_Lanczos_Step.dat
CHECK_Memory.dat        WarningOnTransfer.dat    zvo_sz_TimeKeeper.dat
CHECK_Sdim.dat          zvo_TimeKeeper.dat
```

ログ出力されるファイルの詳細は [CHECK_Chemi.dat](#) 等をご覧ください。

実行コマンドと標準出力{(MPI 並列/ハイブリッド並列でコンパイルした場合の結果)}は次のとおりです。:

```
$ Path/HPhi++ -s stan.in
```

```
,ammmmmmmmmmmmb,,      Welcome to the
,@@` dm                mb  ==m
,@@` d0000000000000000b Pm,   @@          @@          @@
d@  d000 000 0000000 00000b ~@a  @@          @@          00000000
d@  0000  ^^  0000 m m 000  @,  @@          @@  000  @@  000
@  0000_000_0000mm mm000  @|  00mmmmmmmmmm00  @@  @@  @@
P@  900000000000000000P  @~  00000000000000  @@  @@  @@
```

```

@@      ~~9@@@@@PP~      @P @@      @@  @@@  @@  @@@
~@@b      @@@@@@      ,@@~  @@      @@  @@@@@@
~@@@m, ,@@@@@@@@      ,m@~`  @@      @@      @@
      ~~9@@@@@@@@      ~
      9@P~~~9@P      Version 2.0.3

##### Parallelization Info. #####

OpenMP threads : 1
MPI PEs : 1

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

Open Standard-Mode Inputfile stan.in

KEYWORD : model      | VALUE : Spin
KEYWORD : method      | VALUE : CG
KEYWORD : lattice      | VALUE : square
KEYWORD : w      | VALUE : 4
KEYWORD : l      | VALUE : 4
KEYWORD : j      | VALUE : 1.0
KEYWORD : 2sz      | VALUE : 0

##### Parameter Summary #####

@ Lattice Size & Shape

      a = 1.00000      ##### DEFAULT VALUE IS USED #####
Wlength = 1.00000      ##### DEFAULT VALUE IS USED #####
Llength = 1.00000      ##### DEFAULT VALUE IS USED #####
Wx = 1.00000      ##### DEFAULT VALUE IS USED #####
Wy = 0.00000      ##### DEFAULT VALUE IS USED #####
Lx = 0.00000      ##### DEFAULT VALUE IS USED #####
Ly = 1.00000      ##### DEFAULT VALUE IS USED #####
phase0 = 0.00000      ##### DEFAULT VALUE IS USED #####
phase1 = 0.00000      ##### DEFAULT VALUE IS USED #####

@ Super-Lattice setting

      L = 4
      W = 4
      Height = 1      ##### DEFAULT VALUE IS USED #####
      Number of Cell = 16

@ Hamiltonian

      h = 0.00000      ##### DEFAULT VALUE IS USED #####

```

```

        Gamma = 0.00000      ##### DEFAULT VALUE IS USED #####
        2S = 1                ##### DEFAULT VALUE IS USED #####
        D = 0.00000          ##### DEFAULT VALUE IS USED #####
        J0x = 1.00000
        J0y = 1.00000
        J0z = 1.00000
        J1x = 1.00000
        J1y = 1.00000
        J1z = 1.00000

@ Numerical conditions

        LargeValue = 4.50000      ##### DEFAULT VALUE IS USED #####

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

locspn.def is written.
coulombinter.def is written.
hund.def is written.
exchange.def is written.
CDataFileHead = zvo      ##### DEFAULT VALUE IS USED #####
        Lanczos_max = 2000      ##### DEFAULT VALUE IS USED #####
        initial_iv = -1        ##### DEFAULT VALUE IS USED #####
        exct = 1                ##### DEFAULT VALUE IS USED #####
        LanczosEps = 14        ##### DEFAULT VALUE IS USED #####
LanczosTarget = 2         ##### DEFAULT VALUE IS USED #####
        NumAve = 5              ##### DEFAULT VALUE IS USED #####
ExpecInterval = 20       ##### DEFAULT VALUE IS USED #####
        NOmega = 200           ##### DEFAULT VALUE IS USED #####
        OmegaMax = 72.00000    ##### DEFAULT VALUE IS USED #####
        OmegaMin = -72.00000   ##### DEFAULT VALUE IS USED #####
        OmegaIm = 0.04000      ##### DEFAULT VALUE IS USED #####
        2Sz = 0
modpara.def is written.

@ Spectrum

        SpectrumQW = 0.00000    ##### DEFAULT VALUE IS USED #####
        SpectrumQL = 0.00000    ##### DEFAULT VALUE IS USED #####
        SpectrumQH = 0.00000    ##### DEFAULT VALUE IS USED #####
        SpectrumType = szsz      ##### DEFAULT VALUE IS USED #####
        pair.def is written.

@ CalcMod

        Restart = none          ##### DEFAULT VALUE IS USED #####
InitialVecType = c         ##### DEFAULT VALUE IS USED #####
        EigenVecIO = none       ##### DEFAULT VALUE IS USED #####

```



```

    CalcSpec = none          ##### DEFAULT VALUE IS USED #####
    calcmmod.def is written.

    ioutputmode = 1          ##### DEFAULT VALUE IS USED #####
    greenone.def is written.
    greentwo.def is written.
    namelist.def is written.

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

Read File 'namelist.def'.
Read File 'calcmmod.def' for CalcMod.
Read File 'modpara.def' for ModPara.
Read File 'locspn.def' for LocSpin.
Read File 'coulombinter.def' for CoulombInter.
Read File 'hund.def' for Hund.
Read File 'exchange.def' for Exchange.
Read File 'greenone.def' for OneBodyG.
Read File 'greentwo.def' for TwoBodyG.
Read File 'pair.def' for PairExcitation.

##### Definition files are correct. #####

Read File 'locspn.def'.
Read File 'coulombinter.def'.
Read File 'hund.def'.
Read File 'exchange.def'.
Read File 'greenone.def'.
Read File 'greentwo.def'.
Read File 'pair.def'.

##### Indices and Parameters of Definition files(*.def) are complete. #####

MAX DIMENSION idim_max=12870
APPROXIMATE REQUIRED MEMORY max_mem=0.001647 GB

##### MPI site separation summary #####

INTRA process site
  Site    Bit
    0      2
    1      2
    2      2
    3      2
    4      2
    5      2
    6      2
    7      2

```

```

      8      2
      9      2
     10      2
     11      2
     12      2
     13      2
     14      2
     15      2

INTER process site
  Site      Bit

Process element info
  Process      Dimension      Nup      Ndown      Nelec      Total2Sz      State
        0          12870          8          8          8          0

Total dimension : 12870

#####  LARGE ALLOCATE FINISH !  #####

Start: Calculate HilbertNum for fixed Sz.
End   : Calculate HilbertNum for fixed Sz.

Start: Calculate diagaonal components of Hamiltonian.
End   : Calculate diagaonal components of Hamiltonian.

#####  Eigenvalue with LOBPCG  #####

initial_mode=1 (random): iv = -1 i_max=12870 k_exct =1

  Step      Residual-2-norm      Threshold      Energy
    1      2.44343e+00      1.00000e-07      -5.27456e-01
    2      2.76604e+00      1.87217e-07      -1.87217e+00
    3      2.61923e+00      4.19088e-07      -4.19088e+00
    4      2.57106e+00      5.97098e-07      -5.97098e+00

( snip )

    40      7.39431e-06      1.12285e-06      -1.12285e+01
    41      4.15948e-06      1.12285e-06      -1.12285e+01
    42      2.04898e-06      1.12285e-06      -1.12285e+01
    43      9.92048e-07      1.12285e-06      -1.12285e+01

#####  End   : Calculate Lanczos EigenValue.  #####

#####  End   : Calculate Lanczos EigenVec.  #####

```

```
i=      0 Energy=-11.228483 N= 16.000000 Sz=   0.000000 Doublon=   0.000000
```

この実行では、はじめにハミルトニアンの詳細を記述するファイル (`locspin.def`, `trans.def`, `exchange.def`, `coulombintra.def`, `hund.def`, `namelist.def`, `calcmmod.def`, `modpara.def`) と、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。これらのファイルはエキスパートモードと共通です。

計算結果出力

局所最適ブロック共役勾配 (LOBCG) 法

入力ファイルで `method = "CG"` を選択すると、LOBCG 法での計算が行われます。LOBCG 法での計算が正常終了すると、固有エネルギーおよび一体グリーン関数、二体グリーン関数が計算され、ファイル出力されます。以下に、このサンプルでの出力ファイル例を記載します。(xx には 0 から始まる固有値番号が入ります)。

```
zvo_energy.dat
zvo_cisajsccktalt_eigen_xx.dat  zvo_phys_Nup4_Ndown4.dat
```

スタンダードモードの場合は、`"greenone.def"`、`"greentwo.def"` に基づき、：

```
zvo\_cisajs\_eigen\_xx.dat、zvo\_cisajsccktalt\_eigen\_xx.dat
```

に固有値番号に対応した一体グリーン関数および二体グリーン関数の値が出力されます。

Lanczos 法

Lanczos 法での計算が正常終了すると、固有エネルギーおよび一体グリーン関数、二体グリーン関数が計算され、ファイル出力されます。

```
zvo_energy.dat  zvo_cisajs.dat
zvo_cisajsccktalt.dat
```

スタンダードモードの場合は、`greenone.def`、`greentwo.def` に基づき、一体グリーン関数には $\langle n_{i\sigma} \rangle$ 、二体グリーン関数には $\langle n_{i\sigma} n_{j\sigma'} \rangle$ が自動出力されます。なお、Lanczos 法で求めた固有ベクトルが十分な精度を持つ場合にはその固有ベクトルで計算されます。一方、Lanczos 法で求めた固有ベクトルが十分な精度を持たない場合には、ログ出力に「Accuracy of Lanczos vetor is not enough」が表示され、CG 法で固有ベクトルが求められます。各ファイルの詳細は、セクション [energy.dat](#)、[cisajs.dat](#)、[cisajsccktalt.dat](#) に記載がありますので、ご参照ください。

TPQ 法

入力ファイルで `method = "TPQ"` を選択すると、TPQ 法での計算が行われます。TPQ 法での計算が正常終了すると、以下のファイルが出力されます (%% には run の回数、&& には TPQ のステップ数が入ります)。

```
Norm_rand%.dat SS_rand%.dat
zvo_cisajs_set%step&&.dat
zvo_cisajsckalt_set%step&&.dat
```

Norm_rand%.dat には、逆温度や波動関数の規格前の大きさなどの基礎情報が、各 run 回数に応じステップ数とともに出力されます。また、SS_rand%.dat には、逆温度、エネルギー、ハミルトニアン²の期待値などの物理量が各 run 回数に応じステップ数とともに出力されます。zvo_cisajs_set%step&&.dat と zvo_cisajsckalt_set%step&&.dat には各 run 回数でのステップ数に応じた一体グリーン関数および二体グリーン関数が出力されます。各ファイルの詳細はそれぞれ、セクション [Norm_rand.dat](#), [SS_rand.dat](#), [cisajs.dat](#), [cisajsckalt.dat](#) に記載がありますので、ご参照ください。

全対角化法

入力ファイルで `method = "fulldiag"` を選択すると、全対角化法での計算が行われます。全対角化法での計算が正常終了すると、下記のファイルが出力されます (xx には 0 から始まる固有値番号が入ります)。:

```
Eigenvalue.dat zvo_cisajs_eigen_xx.dat
zvo_cisajsckalt_eigen_xx.dat zvo_phys_Nup4_Ndown4.dat
```

Eigenvalue.dat には固有値番号およびエネルギー固有値が出力されます。また、zvo_cisajs_eigen_xx.dat、zvo_cisajsckalt_eigen_xx.dat には固有値番号に対応した一体グリーン関数および二体グリーン関数の値が出力されます。また、zvo_phys_Nup4_Ndown4.dat, physical quantities には、エネルギーやダブロン³の期待値などの物理量が出力されます。各ファイルの詳細は、それぞれ [Eigenvalue.dat](#) - [cisajsckalt.dat](#) に記載がありますので、ご参照ください。

その他の系でのチュートリアル

`samples/Standard/` 以下にはこの他にも様々なチュートリアルが置いてあります。それぞれのチュートリアルの内容や手順については、各フォルダにある `README.md` をご覧ください。

5.3.2 エキスパートモード

エキスパートモードでは、入力ファイルとして

1. 入力ファイルリスト
2. 基本パラメータ用ファイル
3. Hamiltonian 作成用ファイル
4. 出力結果指定用ファイル

を用意した後、計算を行います。計算開始後に関しては、スタンダードモードと同様です。ここでは前節のスタンダードモードでのチュートリアルを行った後の状況を例に入力ファイルの作成に関する説明を行います。

入力ファイルリストファイル

入力ファイルの種類と名前を指定するファイル `namelist.def` には、下記の内容が記載されています。各入力ファイルリストファイルでは、行毎に Keyword とファイル名を記載し、ファイルの種類の区別を行います。詳細はセクション [入力ファイル指定用ファイル](#) をご覧ください。

ModPara	modpara.def
LocSpin	locspn.def
CoulombInter	coulombinter.def
Hund	hund.def
Exchange	exchange.def
OneBodyG	greenone.def
TwoBodyG	greentwo.def
CalcMod	calcmmod.def
PairExcitation	pair.def
SpectrumVec	zvo_eigenvec_0

基本パラメータの指定

計算モード、計算用パラメータ、局在スピンの位置を以下のファイルで指定します。

計算モードの指定

CalcMod でひも付けられるファイル (ここでは `calcmmod.def`) で計算モードを指定します。ファイルの中身は下記の通りです。計算モード、計算用パラメータ、局在スピンの位置を以下のファイルで指定します。:

```
#CalcType = 0:Lanczos, 1:TPQCalc, 2:FullDiag, 3:CG
#CalcModel = 0:Hubbard, 1:Spin, 2:Kondo, 3:HubbardGC, ..
#Restart = 0:None, 1:Save, 2:Restart&Save, 3:Restart
#CalcSpec = 0:None, 1:Normal, 2:No H*Phi, 3:Save, ...
CalcType    3
CalcModel   1
ReStart     0
CalcSpec    0
CalcEigenVec 0
InitialVecType 0
InputEigenVec 0
```

CalcType で計算手法の選択、CalcModel で対象モデルの選択を行います。ここでは、計算手法として LOBCG 法、対象モデルとしてスピン系 (カノニカル) を選択しています。CalcMod ファイルでは固有ベクトルの入出力機能も指定することができます。CalcMod ファイルの詳細は [CalcMod ファイル](#) をご覧ください。

計算用パラメータの指定

ModPara でひも付けられるファイル (ここでは modpara.def) で計算用パラメータを指定します。ファイルの中身は下記の通りです。:

```
-----
Model_Parameters    0
-----
HPhi_Cal_Parameters
-----
CDataFileHead    zvo
CParaFileHead    zqp
-----
Nsite            16
2Sz              0
Lanczos_max      2000
initial_iv       -1
exct             1
LanczosEps       14
LanczosTarget    2
LargeValue       4.500000000000000e+00
NumAve           5
ExpecInterval    20
NOmega           200
OmegaMax         7.200000000000000e+01    4.000000000000000e-02
OmegaMin         -7.200000000000000e+01    4.000000000000000e-02
OmegaOrg         0.0 0.0
```

このファイルでは、サイト数、伝導電子の数、トータル S_z や Lanczos ステップの最大数などを指定します。ModPara ファイルの詳細はセクション [ModPara ファイル](#) をご覧ください。

局在スピンの位置の指定

LocSpin でひも付けられるファイル (ここでは locspn.def) で局在スピンの位置と S の値を指定します。ファイルの中身は下記の通りです。:

```
=====
NlocalSpin       16
=====
=====i_0LocSpn_1IteElc =====
=====
    0         1
    1         1
    2         1
    3         1
    4         1
    5         1
```

...

LocSpin ファイルの詳細は [LocSpin 指定ファイル](#) をご覧ください。

Hamiltonian の指定

基本パラメータを設定した後は、Hamiltonian を構築するためのファイルを作成します。 HPhi++ では、電子系の表現で計算を行うため、スピン系では以下の関係式

$$\begin{aligned} S_z^{(i)} &= (c_{i\uparrow}^\dagger c_{i\uparrow} - c_{i\downarrow}^\dagger c_{i\downarrow})/2, \\ S_+^{(i)} &= c_{i\uparrow}^\dagger c_{i\downarrow}, \\ S_-^{(i)} &= c_{i\downarrow}^\dagger c_{i\uparrow}. \end{aligned}$$

を用い、電子系の演算子に変換し Hamiltonian の作成をする必要があります。

Transfer 部の指定

Trans でひも付けられるファイル (ここでは zTrans.def) で電子系の Transfer に相当する Hamiltonian

$$\mathcal{H}_+ = - \sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2}.$$

を指定します。ファイルの中身は下記の通りです。:

```
=====
NTransfer      0
=====
=====i_j_s_tijs=====
=====
```

スピン系では外場を掛ける場合などに使用することができます。例えば、サイト 1 に $-0.5S_z^{(1)}$ ($S = 1/2$) の外場を掛けたい場合には、, 電子系の表現 $-0.5/2(c_{1\uparrow}^\dagger c_{1\uparrow} - c_{1\downarrow}^\dagger c_{1\downarrow})$ に書き換えた以下のファイルを作成することで計算することが出来ます。

```
=====
NTransfer      1
=====
=====i_j_s_tijs=====
=====
1 0 1 0 -0.25 0
1 1 1 1 0.25 0
```

Trans ファイルの詳細はセクション [Trans 指定ファイル](#) をご覧ください。

二体相互作用部の指定

InterAll でひも付けられるファイル (ここでは zInterAll.def) で電子系の二体相互作用部に相当する Hamiltonian

$$\mathcal{H} = \sum_{i,j,k,l} \sum_{\sigma_1, \sigma_2, \sigma_3, \sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}.$$

を指定します。ファイルの中身は下記の通りです。

```
=====
NInterAll      96
=====
=====zInterAll=====
=====
  0    0    0    0    1    0    1    0    0.500000  0.000000
  0    0    0    0    1    1    1    1   -0.500000  0.000000
  0    1    0    1    1    0    1    0   -0.500000  0.000000
  0    1    0    1    1    1    1    1    0.500000  0.000000
  0    0    0    1    1    1    1    0    1.000000  0.000000
  0    1    0    0    1    0    1    1    1.000000  0.000000
...
```

ここでは、簡単のためサイト i とサイト $i+1$ 間の相互作用に着目して説明します。 $S = 1/2$ の場合、相互作用の項をフェルミオン演算子で書き換えると、

$$\begin{aligned} \mathcal{H}_{i,i+1} &= J(S_x^{(i)} S_x^{(i+1)} + S_y^{(i)} S_y^{(i+1)} + S_z^{(i)} S_z^{(i+1)}) \\ &= J \left(\frac{1}{2} S_+^{(i)} S_-^{(i+1)} + \frac{1}{2} S_-^{(i)} S_+^{(i+1)} + S_z^{(i)} S_z^{(i+1)} \right) \\ &= J \left[\frac{1}{2} c_{i\uparrow}^\dagger c_{i\downarrow} c_{i+1\downarrow}^\dagger c_{i+1\uparrow} + \frac{1}{2} c_{i\downarrow}^\dagger c_{i\uparrow} c_{i+1\uparrow}^\dagger c_{i+1\downarrow} + \frac{1}{4} (c_{i\uparrow}^\dagger c_{i\uparrow} - c_{i\downarrow}^\dagger c_{i\downarrow}) (c_{i+1\uparrow}^\dagger c_{i+1\uparrow} - c_{i+1\downarrow}^\dagger c_{i+1\downarrow}) \right]. \end{aligned}$$

となります。したがって、 $J = 2$ に対して InterAll ファイルのフォーマットを参考に相互作用を記載すると、 $S_z^{(i)} S_z^{(i+1)}$ の相互作用は

i	0	i	0	i+1	0	i+1	0	0.500000	0.000000
i	0	i	0	i+1	1	i+1	1	-0.500000	0.000000
i	1	i	1	i+1	0	i+1	0	-0.500000	0.000000
i	1	i	1	i+1	1	i+1	1	0.500000	0.000000

となり、それ以外の項は

i	0	i	1	i+1	1	i+1	0	1.000000	0.000000
i	1	i	0	i+1	0	i+1	1	1.000000	0.000000

と表せばよいことがわかります。なお、InterAll 以外にも、Hamiltonian を簡易的に記載するための下記のパイル形式に対応しています。詳細はセクション [InterAll 指定ファイル - PairLift 指定ファイル](#) をご覧ください。

出力ファイルの指定

一体 Green 関数および二体 Green 関数の計算する成分を、それぞれ OneBodyG, TwoBodyG でひも付けられるファイルで指定します。

一体 Green 関数の計算対象の指定

OneBodyG でひも付けられるファイル (ここでは greenone.def) で計算する一体 Green 関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ の成分を指定します。ファイルの中身は下記の通りです

```
=====
NCisAjs          32
=====
===== Green functions =====
=====
  0      0      0      0
  0      1      0      1
  1      0      1      0
  1      1      1      1
  2      0      2      0
  ...
```

一体 Green 関数計算対象成分の指定に関するファイル入力形式の詳細はセクション [OneBodyG 指定ファイル](#) をご覧ください。

二体 Green 関数の計算対象の指定

TwoBodyG でひも付けられるファイル (ここでは greentwo.def) で計算する二体 Green 関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ の成分を指定します。ファイルの中身は下記の通りです。

```
=====
NCisAjsCktAltDC    1024
=====
===== 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
  ...
```

二体 Green 関数計算対象成分の指定に関するファイル入力形式の詳細はセクション [TwoBodyG 指定ファイル](#) をご覧ください。

計算の実行

全ての入力ファイルが準備できた後、計算実行します。実行時はエキスパートモードを指定する "-e" をオプションとして指定の上、入力ファイルリストファイル (ここでは namelist.def) を引数とし、ターミナルから HPhi++ を実行します。

```
$ Path/HPhi++ -e namelist.def
```

計算開始後のプロセスは、スタンダードモードと同様になります。

5.3.3 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。

5.4 ファイルフォーマット

5.4.1 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"
method = "Lanczos"

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
2Sz = 0
```

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表示されます。
- パラメータには三種類あります。
 1. 必ず指定しなければいけないパラメーター (もし存在しない場合には, HPhi++ のエラーメッセージが出力され、プログラムは終了します),
 2. 指定しない場合デフォルト値が使われるパラメーター (もし存在しない場合は、デフォルト値が使用されます),
 3. 使われないパラメーター (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表示されます。)
 例えば、ハイゼンベルグ模型でトランスファー積分 t を指定した場合が相当します。もし “ model=spin” とした場合には、 “ t ” は使用できません。

次に各キーワードの説明をします。

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", "SpinGCCMA" のいずれか)^{*1}

説明 : 計算対象の模型を指定します。上記の文字列はそれぞれカノニカル集団のフェルミ粒子 Hubbard 模型

$$\mathcal{H} = -\mu \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma} - \sum_{i \neq j\sigma} t_{ij} c_{i\sigma}^\dagger c_{j\sigma} + \sum_i U n_{i\uparrow} n_{i\downarrow} + \sum_{i \neq j} V_{ij} n_i n_j, \quad (5.1)$$

同じくカノニカル集団のスピン模型 ($\{\sigma_1, \sigma_2\} = x, y, z$)

^{*1} GC=Grand Canonical

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

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)

$$\mathcal{H} = -\mu \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma} - t \sum_{\langle ij \rangle \sigma} c_{i\sigma}^\dagger c_{j\sigma} + \frac{J}{2} \sum_i \left\{ S_i^+ c_{i\downarrow}^\dagger c_{i\uparrow} + S_i^- c_{i\uparrow}^\dagger c_{i\downarrow} + S_{iz}(n_{i\uparrow} - n_{i\downarrow}) \right\}, \quad (5.2)$$

グランドカノニカル集団のフェルミ粒子 Hubbard 模型 [式 (5.1)], グランドカノニカル集団のスピン模型 [式 (5.2)], グランドカノニカル集団の近藤格子模型 [式 (5.2)] に対応します。

"SpinGCCMA"では"SpinGC"と同じ計算をより速いアルゴリズム^{*2}を用いて行います。ただし、扱うことのできるモデルや MPI 並列数に強い制約があります。以下の"Lattice"の項もご参照ください。

- method

形式: 文字列 ("Lanczos", "TPQ", "Full Diag", "CG", "Time-Evolution"のいずれか)

説明: 実行する計算の種類を指定します。上記の文字列はそれぞれランチョス法による少数固有状態の計算, 熱力学的純粋状態を用いた有限温度計算, 直接法による全固有状態計算, LOBCG 法^{*3*4}による少数固有状態の計算, 実時間発展計算 に対応します。

後述のスペクトル計算において使用される手法もこのパラメーターで指定されます"CG"とした場合には付属している $K\omega$ ライブラリ^{*5} が呼び出され、シードスイッチ^{*6} 付きシフト双共役勾配法^{*7} が適用されます。

- lattice

形式: 文字列 ("Chain Lattice", "Square Lattice", "Triangular Lattice", "Honeycomb Lattice", "Ladder", "Kagome"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 (図 5.1 (a))、2 次元正方格子 (図 5.1 (b))、2 次元三角格子 (図 5.1 (c))、2 次元異方的蜂の巣格子 (図 5.2)、梯子格子 (図 5.4)、カゴメ格子 (図 5.3) に対応します。

method="SpinGCCMA"では、このうち"Chain Lattice", "Honeycomb Lattice", "Ladder", "Kagome"に対応しています。各格子についてのサイズ (L, W) と MPI 並列数 (N_{proc}) の制限は次のとおりです (次節の ‘L’, ‘W’ もご参照ください)。

^{*2} Y. Yamaji *et. al.*, manuscript in preparation.

^{*3} A.V.Knyazev, SIAM Journal on Scientific Computing **23**, 517 (2001).

^{*4} S.Yamada, T.Imamura, M.Machida, The Japan Society for Computational Engineering and Science **2006**, 20060027 (2006).

^{*5} <https://github.com/issp-center-dev/Komega>.

^{*6} S.Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

^{*7} A.Frommer, Computing **70**, 87{109 (2003).

– "Chain Lattice"

$L = 8n$ (ただし: n は $n \geq 1$ の整数), $N_{\text{proc}} \leq 2(L = 8)$, $N_{\text{proc}} \leq 2^{L/2-2}(L > 8)$.

– "Honeycomb Lattice"

$W = 3, L \geq 2$, $N_{\text{proc}} \leq 2(L = 2)$, $N_{\text{proc}} \leq 64(L > 2)$.

– "Ladder"

$W = 2, L = 2n$ (ただし: n は $n \geq 4$ の整数), $N_{\text{proc}} \leq 2^{L-4}$.

– "Kagome"

$W = 3, L \geq 2$, $N_{\text{proc}} \leq 1(L = 2)$, $N_{\text{proc}} \leq 512(L > 2)$.

格子に関するパラメーター

1 次元鎖 [図 5.1 (a)]

- L

形式：自然数

説明：鎖の長さを指定します.

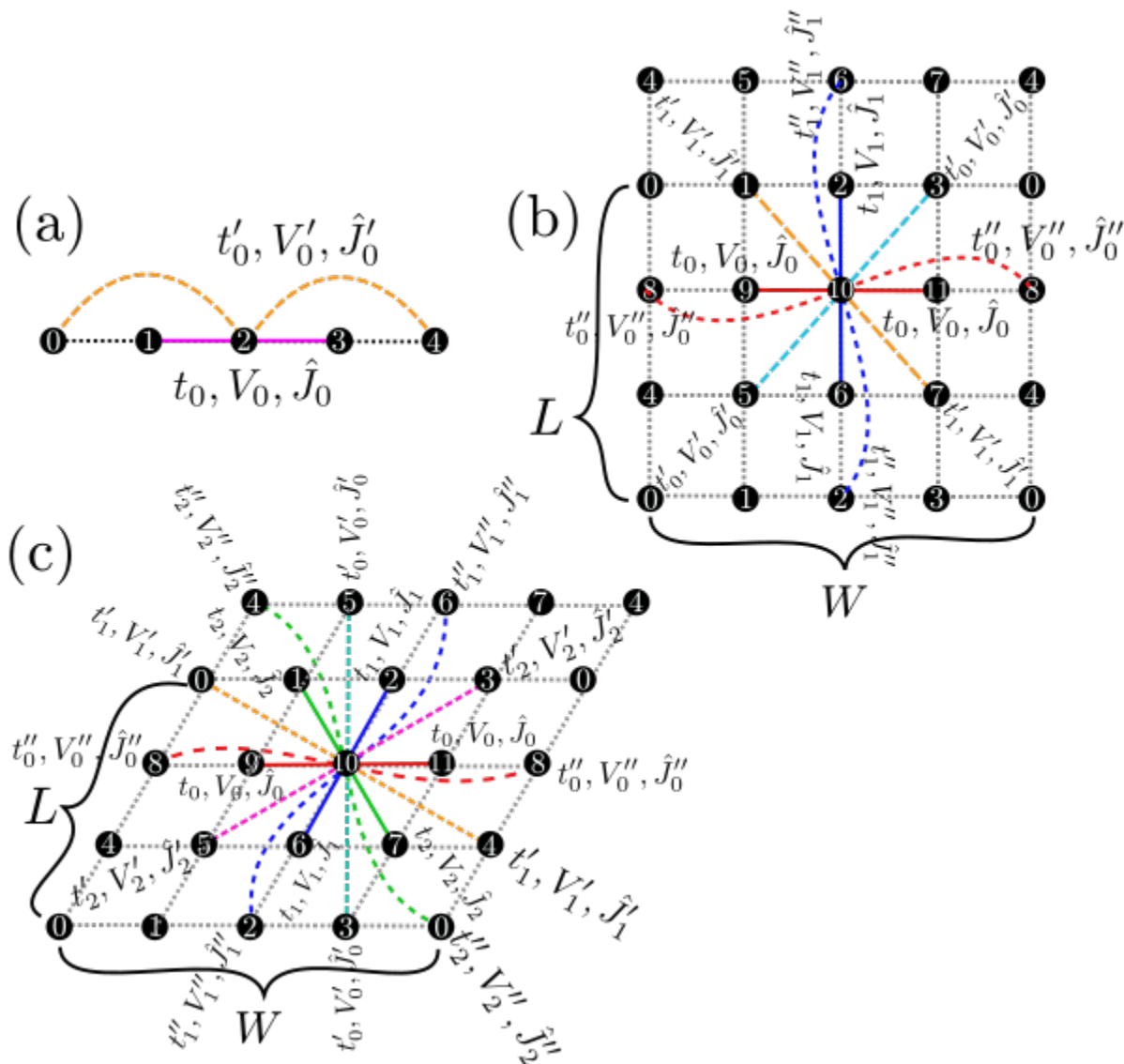


図 5.1 Figure 1: (a)1 次元鎖、(b)2 次元正方格子、(c)2 次元三角格子の模式図。ホッピング積分、オフサイトクーロン積分、スピン結合は、再近接サイト間(マゼンタの実線)ではそれぞれ t, V, J となり、次近接サイト間(緑の破線)ではそれぞれ t', V', J' となります。

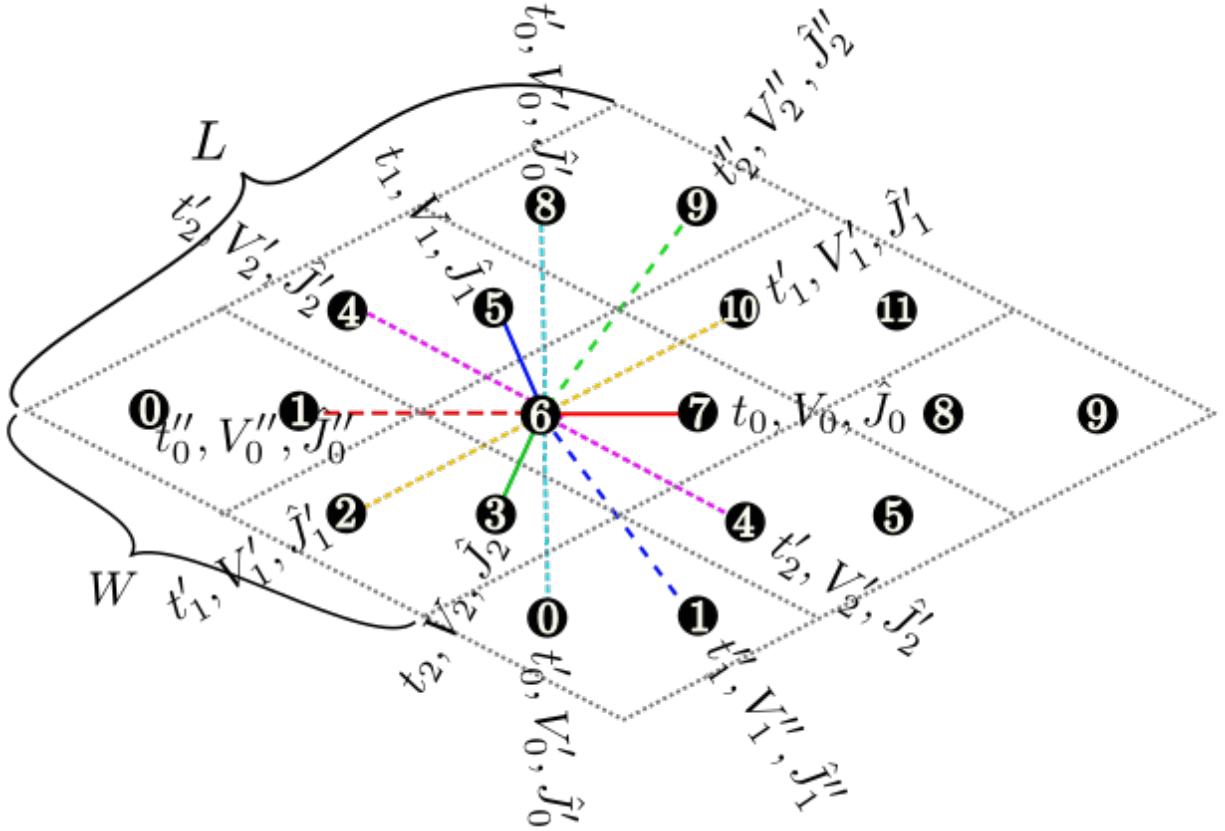


図 5.2 Figure 2: 2次元異方的蜂の巣格子の模式図. ホッピング積分、オフサイトクーロン積分、スピン結合は、ボンドの方向によって異なります。

梯子格子 (図 5.4)

• 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.

矩形格子 [図 5.1 (b)], 三角格子 [図 5.1 (c)], 蜂の巣格子 [図 5.2], カゴメ格子 [図 5.3]

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。ただし、`model=SpinGCCMA` の時には前者のみが使えます。また、両方を指定した場合には HPhi++ を終了します。

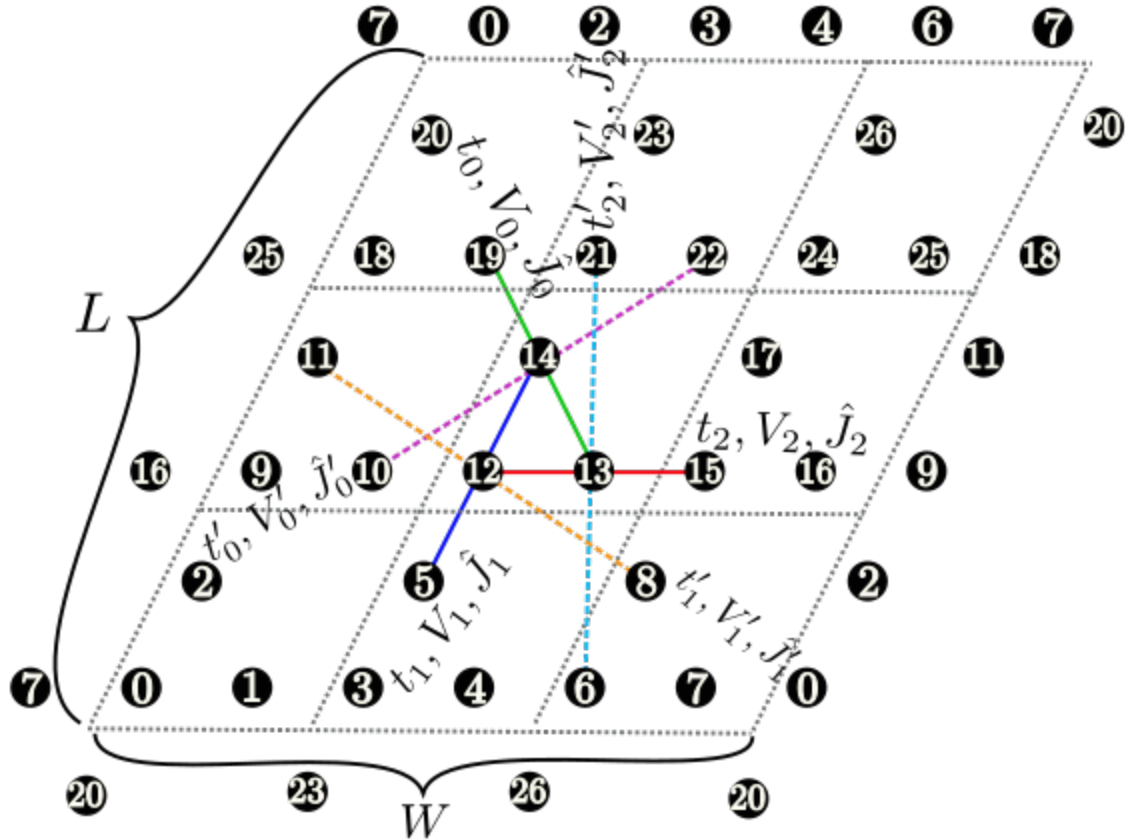


図 5.3 Figure 3: カゴメ格子の模式図.

- 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。

- a0W, a0L, a1W, a1L

形式：自然数

説明：格子を指定する 2 本のベクトル (\vec{a}_0, \vec{a}_1) を指定します (図 5.5)。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` 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の事が出来ます。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
```

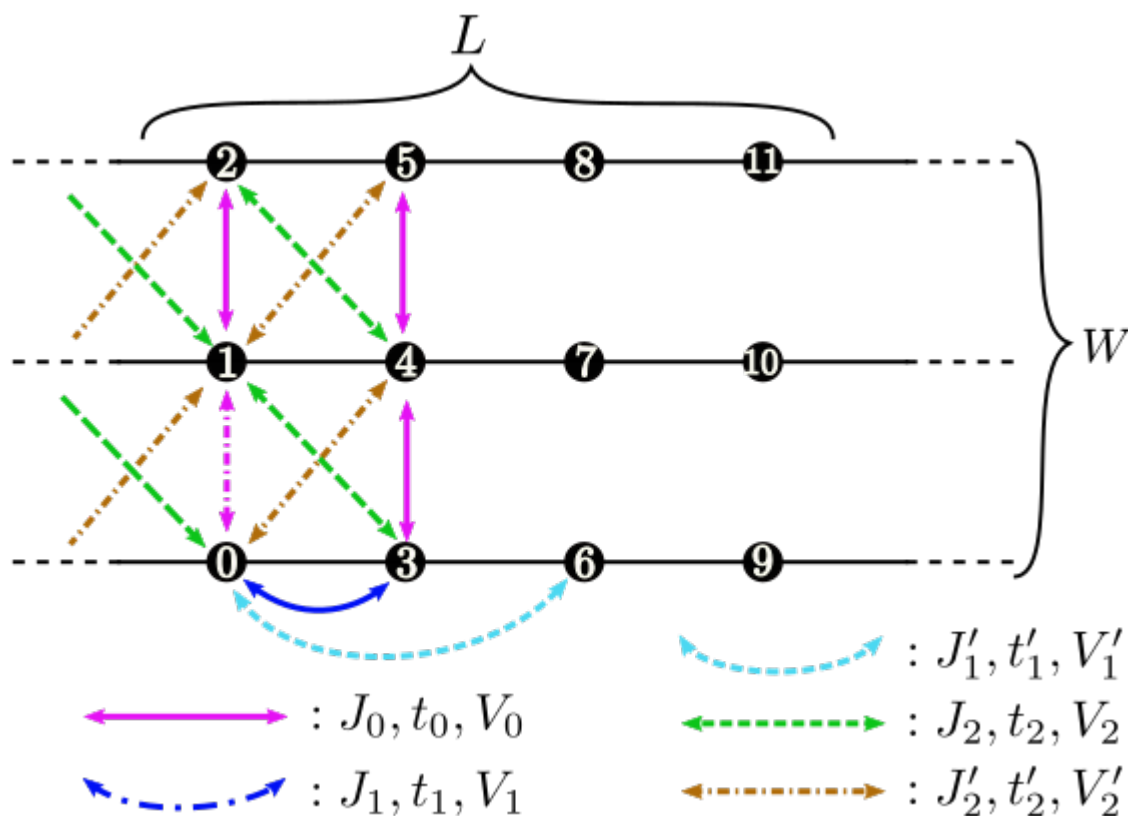



図 5.4 Figure 4: 梯子格子の模式図.

保存量に関するパラメーター

- `nelec`

形式: 整数

説明: 伝導電子数を指定します。model = "Fermion HubbardGC", "Spin", "SpinGC" のときには指定しないでください。

- `2Sz`

形式: 整数

説明: 全スピンの z 成分の 2 倍を指定します。model = "Fermion HubbardGC", "SpinGC" のときには指定しないでください。

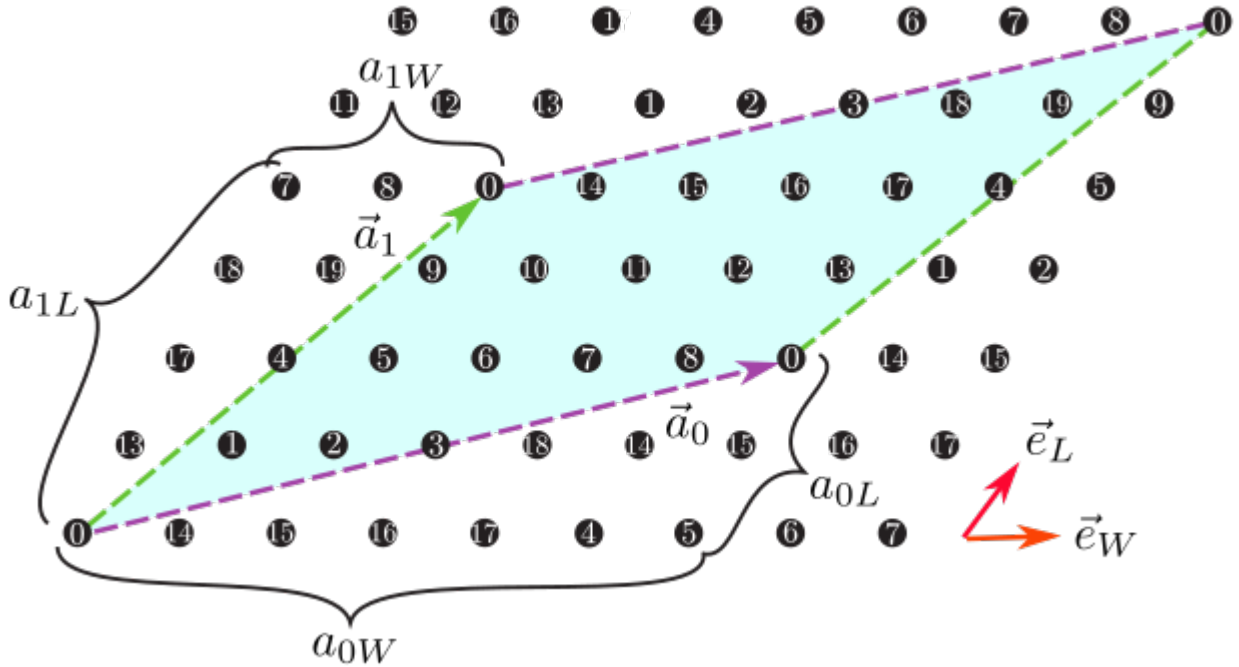


図 5.5 Figure 5: 三角格子において、 $\vec{a}_0 = (6, 2)$, $\vec{a}_1 = (2, 4)$ とした場合のセル形状。 \vec{a}_0 (マゼンタ) および \vec{a}_1 (グリーン) で囲まれた部分 (サイト数は 20) が計算するセルとなる。

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。

- $J_x, J_y, J_z, J_{xy}, J_{yx}, J_{xz}, J_{zx}, J_{yz}, J_{zy}$

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、 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 を指定した場合は HPhi++ を終了します。

- h

形式：実数

説明：縦磁場を指定します。

- Γ, 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。ただし異方性パラメータについては、`model=SpinGCCMA` では使用できません。

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 表 5.1 に表します。

表 5.1 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。

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_0, t_0, V_0	○	○	○	○	○	○
J_1, t_1, V_1	-	○	○	○	○	○
J_2, t_2, V_2	-	-	○	○	○	○
J', t', V' (省略形)	○	○	○	○	○	-
J_0', t_0', V_0'	○	○	○	○	○	-
J_1', t_1', V_1'	-	○	○	○	○	○
J_2', t_2', V_2'	-	-	○	○	○	○
J'', t'', V'' (省略形)	○	○	○	○	-	-
J_0'', t_0'', V_0''	○	○	○	○	-	-
J_1'', t_1'', V_1''	-	○	○	○	-	-
J_2'', t_2'', V_2''	-	-	○	○	-	-

非局所項 [梯子格子 (図 5.4)]

- $t_0, t_1, t_1', t_2, t_2'$

形式：複素数

説明：梯子格子でのホッピング (図 5.4 参照) を指定します。

- $V_0, V_1, V_1', V_2, V_2'$

形式：実数

説明：梯子格子でのオフサイトクーロン積分 (図 5.2 参照) を指定します。

- $J0x, J0y, J0z, J0xy, J0yx, J0xz, J0zx, J0yz, J0zy$
- $J1x, J1y, J1z, J1xy, J1yx, J1xz, J1zx, J1yz, J1zy$
- $J1'x, J1'y, J1'z, J1'xy, J1'yx, J1'xz, J1'zx, J1'yz, J1'zy$
- $J2x, J2y, J2z, J2xy, J2yx, J2xz, J2zx, J2yz, J2zy$
- $J2'x, J2'y, J2'z, J2'xy, J2'yx, J2'xz, J2'zx, J2'yz, J2'zy$.

形式：実数

説明：梯子格子でのスピン相互作用 (図 5.4 参照) を指定します。また対角項について、例えば $J0x, J0y, J0z$ を指定する代わりにパラメータ $J0$ を指定すると $J0x = J0y = J0z = J0$ が代入されます。 $J0$ を指定した上で $J0x, J0y, J0z$ 等も指定した場合は HPhi++ を終了します。 $J1, J1', J2, J2'$ についても同様です。

非局所項 [梯子格子以外 (図 5.1 , 図 5.2 , 図 5.3)]

- $t, t0, t1, t2$

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。また、ホッピングのボンド方向依存性がない場合は $t0, t1, t2$ を別々に指定する代わりにパラメータ t を指定すると、 $t0 = t1 = t2 = t$ が代入されます。 t と $t0$ 等の両方が指定された場合には HPhi++ を終了します。

- $t', t0', t1', t2'$

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。また、ホッピングのボンド方向依存性がない場合は $t0', t1', t2'$ を別々に指定する代わりにパラメータ t' を指定すると、 $t0' = t1' = t2' = t'$ が代入されます。 t' と $t0'$ 等の両方が指定された場合には HPhi++ を終了します。

- $t'', t0'', t1'', t2''$

形式：複素数

説明：Hubbard および近藤格子模型での、第三近接サイト間の各方向のホッピングを指定します。また、ホッピングのボンド方向依存性がない場合は $t0'', t1'', t2''$ を別々に指定する代わりにパラメータ t'' を指定すると、 $t0'' = t1'' = t2'' = t''$ が代入されます。 t'' と $t0''$ 等の両方が指定された場合には HPhi++ を終了します。

- $V, V0, V1, V2$

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、サイト間 Coulomb 積分のボンド方向依存性がない場合は V_0, V_1, V_2 を別々に指定する代わりにパラメータ V を指定すると、 $V_0 = V_1 = V_2 = V$ が代入されます。 V と V_0 等の両方が指定された場合には HPhi++ を終了します。

- V', V_0', V_1', V_2'

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。また、サイト間 Coulomb 積分のボンド方向依存性がない場合は V_0', V_1', V_2' を別々に指定する代わりにパラメータ V' を指定すると、 $V_0' = V_1' = V_2' = V'$ が代入されます。 V' と V_0' 等の両方が指定された場合には HPhi++ を終了します。

- V'', V_0'', V_1'', V_2''

形式 : 実数

説明 : Hubbard および近藤格子模型での、第三近接サイト間の Coulomb 積分を指定します。また、サイト間 Coulomb 積分のボンド方向依存性がない場合は V_0'', V_1'', V_2'' を別々に指定する代わりにパラメータ V'' を指定すると、 $V_0'' = V_1'' = V_2'' = V''$ が代入されます。 V'' と V_0'' 等の両方が指定された場合には HPhi++ 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 HPhi++ を終了します。 J_1, J_2 についても同様です。

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出来ます。 $J_x \sim J_{zy}$ 系列のどれかと $J_{0x} \sim J_{2y}$ 系列のどれかを両方指定した場合には HPhi++ 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

- $J'x, J'y, J'z, J'xy, J'yx, J'xz, J'zx, J'yz, J'zy$
- $J_0'x, J_0'y, J_0'z, J_0'xy, J_0'yx, J_0'xz, J_0'zx, J_0'yz, J_0'zy$
- $J_1'x, J_1'y, J_1'z, J_1'xy, J_1'yx, J_1'xz, J_1'zx, J_1'yz, J_1'zy$
- $J_2'x, J_2'y, J_2'z, J_2'xy, J_2'yx, J_2'xz, J_2'zx, J_2'yz, J_2'zy$

形式: 実数

説明: 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。ただし、蜂の巣格子またはカゴメ格子で `model=SpinGCCMA` の場合にはまだ対応していません。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 HPhi++ を終了します。

- $J''x, J''y, J''z, J''xy, J''yx, J''xz, J''zx, J''yz, J''zy$
- $J_0''x, J_0''y, J_0''z, J_0''xy, J_0''yx, J_0''xz, J_0''zx, J_0''yz, J_0''zy$
- $J_1''x, J_1''y, J_1''z, J_1''xy, J_1''yx, J_1''xz, J_1''zx, J_1''yz, J_1''zy$
- $J_2''x, J_2''y, J_2''z, J_2''xy, J_2''yx, J_2''xz, J_2''zx, J_2''yz, J_2''zy$

形式: 実数

説明: スピン模型での、第三近接サイト間のスピン相互作用を指定します。ただし、蜂の巣格子またはカゴメ格子で `model=SpinGCCMA` の場合にはまだ対応していません。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 HPhi++ 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)。
 \vec{a}_0 方向、 \vec{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。

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

計算条件のパラメーター

- 2S

形式：正の整数 (デフォルト値は 1)

説明：スピン模型での局在スピンの大きさ S の 2 倍を指定します。(例/ 1/2 スピンならば 1)

- Restart

形式：文字列 ("None", "Restart_out", "Restart_in", "Restart")のいずれか。デフォルトは"None")

説明：再計算に関する設定を行う。"None"では再計算に関連するファイル出力をしない。
 "Restart_out"では一から計算を始めて、反復が終了した時点で再計算用のデータをファイル出力する。
 "Restart_in"では再計算用のデータをファイルから受け取り途中から計算を始める。"Restart"では再計算用のデータをファイルから受け取り途中から計算を始め、反復が終了した時点で再計算用のデータをファイル出力する。

- Lanczos_max

形式：整数 (デフォルト値は 2000)

説明：ランチョスステップの上限、LOBCG ステップの上限、TPQ ステップ数、時間発展ステップ数、BiCG ステップの上限を指定します。

- initial_iv

形式：整数 (デフォルト値は-1)

説明：初期条件のベクトルを与えます。

- ランチョス法

- * カノニカル集団かつ $\text{initial_iv} \geq 0$ の場合

- ノンゼロの成分が指定されます。

- * $\text{initial_iv} < 0$ の場合

- 乱数のシードが指定され、全ての成分に対して係数がランダムに与えられます。なお、グランドカノニカルの場合は初期状態として多くの状態を持つよう、こちらの様式が適用されます。

- TPQ 法

- 乱数のシードが指定され、全ての成分に対して係数がランダムに与えられます。

初期ベクトル設定の詳細については、[アルゴリズム](#) を参照ください。

- exct

形式：整数 (デフォルト値は 1)

説明: `method="Lancoz"`ではエネルギーの低いものから数えて、何番目の固有状態を計算するかを指定します。`method="CG"`の時には求める固有状態の本数を指定します。

Note: `nvec >= exct` を満たす必要があります。

- LanczosEps

形式: 整数 (デフォルト値は 14)

説明: ランチョスの収束判定条件を指定します。ひとつ前のステップの固有値との相対誤差が, $10^{-\text{LanczosEps}}$ 以下になったら収束したと判断します。`method="CG"`の時には残差ベクトルの 2-ノルムが $10^{-\text{LanczosEps}/2}$ 以下になったら収束したと判断します。

- LanczosTarget

形式: 整数 (デフォルト値は 2)

説明: エネルギーの低いものから数えて、何番目の固有値でランチョスの収束判定を行うかを指定します。

- LargeValue

形式: 実数 (デフォルト値は下記参照)

説明: (TPQ 法のみで使用) $l - \hat{H}/N_s$ の l 。ハミルトニアン の各項の係数の絶対値の総和をサイト数で割ったものがデフォルト値になります。

- NumAve

形式: 整数 (デフォルト値は 5)

説明: (TPQ 法のみで使用) 独立な run を何回行うかを指定します。

- ExpecInterval

形式: 整数 (デフォルト値は 20)

説明: (TPQ 法のみで使用) 相関関数の計算を何回の TPQ ステップおきに行うかの指定。頻度を上げると計算コストが増大するので注意してください。

- 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 [Bogoliubov 表現](#) をご覧ください。

- InitialVecType

形式：文字 ("C", "R"のいずれか。デフォルトは"C")

説明：固有ベクトルの初期値の型を指定する。C では複素数型、R では実数型とする。

- EigenVecIO

形式：文字列 ("None", "Out", "In"のいずれか。デフォルトは"None")

説明：固有ベクトルの入出力を指定する。"None"では固有ベクトルの入出力を行わない。"Out"では求めた固有ベクトルをファイルに出力する。"In"では固有ベクトルをファイルから取り出し、その後の計算(動的グリーン関数など)を行う。

動的グリーン関数の計算に関するパラメーター

• CalcSpec

形式：文字列 ("None", "Normal", "NoIteration", "Restart_out", "Restart_in", "Restart", "Scratch". デフォルトは"None")

説明：動的グリーン関数の計算に関する設定を行う。"None"では動的グリーン関数を計算しない。"Normal"では一から動的グリーン関数の計算を始める。"NoIteration"では、前回の反復回数と同じところまで反復させる。このとき、ハミルトニアン-ベクトル積演算は行われなため、計算コストは非常に軽い、十分な精度が出せない場合がある。"Restart_out"では一から計算を始めて、反復が終了した時点で再計算用のデータをファイル出力する。"Restart_in"では再計算用のデータをファイルから受け取り途中から計算を始める。"Restart"では再計算用のデータをファイルから受け取り途中から計算を始め、反復が終了した時点で再計算用のデータをファイル出力する。"Scratch"では、まず固有状態の計算が行われたのちに、続けて動的グリーン関数の計算が行われる。この場合は波動関数のファイル入出力は必要なくなる。スペクトル計算において使用される手法はパラメーター `method` で指定されます。(method="CG"とした場合には付属している $K\omega$ ライブラリ^{*1} が呼び出され、シードスイッチ^{*2} 付きシフト双共役勾配法^{*3} が使われます。

• SpectrumType

形式：文字列 ("SzSz", "S+S-", "Density", "up", "down", "SzSz_R", "S+S-_R", "Density_R", "up_R", "down_R" のいずれか。デフォルトは"SzSz")

説明：計算する動的グリーン関数の種類を指定する。逆格子空間での相関関数を直接計算する場合には次の値を使用する。"SzSz"では $\langle \hat{S}_{zq} \hat{S}_{zq} \rangle$ 、"S+S-"では $\langle \hat{S}_q^+ \hat{S}_q^- \rangle$ 、"Density"では $\langle \hat{n}_q \hat{n}_q \rangle$ 、"up"では $\langle \hat{c}_{q\uparrow}^\dagger \hat{c}_{q\uparrow} \rangle$ 、"down"では $\langle \hat{c}_{q\downarrow}^\dagger \hat{c}_{q\downarrow} \rangle$ 。また、実空間での相関関数の計算では次の値を用いる。"SzSz_R"では $\langle \hat{S}_{zR} \hat{S}_{z0} \rangle$ 、"S+S-_R"では $\langle \hat{S}_R^+ \hat{S}_0^- \rangle$ 、"Density_R"では $\langle \hat{n}_R \hat{n}_0 \rangle$ 、"up_R"では $\langle \hat{c}_{R\uparrow}^\dagger \hat{c}_{0\uparrow} \rangle$ 、"down_R"では $\langle \hat{c}_{R\downarrow}^\dagger \hat{c}_{0\downarrow} \rangle$ 、ここで R はすべてのサイト番号にわたる。この実空間での動的相関関数をフーリエ変換して逆格子空間全体での動的相関関数を得る方法については相関関数の *Fourier* 変換ユーティリティーを参照。

• SpectrumQW, SpectrumQL, SpectrumQH

形式：実数 (デフォルトはともに 0.0)

説明：SpectrumType が "SzSz", "S+S-", "Density", "up", "down" のときのみ使用。計算する動的グリーン関数の波数を Fractional coordinate で指定する。逆格子ベクトルは図 5.1, 図 5.2, 図 5.4, 図 5.3 に表されている格子ベクトルと対応するものとなる。

• OmegaMin

形式：実数 (デフォルトは -LargeValue × サイト数)

説明：計算する動的グリーン関数の振動数の実部の下限。

^{*1} <https://github.com/issp-center-dev/Komega>.

^{*2} S. Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

^{*3} A. Frommer, Computing **70**, 87{109 (2003).

- OmegaMax

形式：実数 (デフォルトは LargeValue \times サイト数)

説明：計算する動的グリーン関数の振動数の実部の上限。

- OmegaIm

形式：実数 (デフォルトは $0.01 \times \text{LargeValue}$)

説明：計算する動的グリーン関数の振動数の虚部。

- NOmega

形式：正の整数 (デフォルトは 200)

説明：計算する動的グリーン関数の振動数のグリッド数。

実時間発展計算に関するパラメーター

- dt

形式：正の実数 (デフォルトは 0.1)

説明：時間ステップ幅。

- PumpType

形式：文字列 ("Quench", "Pulse Laser", "AC Laser", "DC Laser" のいずれか。デフォルトは "Quench")

説明：時間依存ハミルトニアンの種類を指定する。"Quench" では 2 体演算子 $U_{\text{quench}} \sum_i n_{i\uparrow} n_{i\downarrow}$ が加えられる。"Pulse Laser", "AC Laser", "DC Laser" では、ホッピング項に $-\sum_{ij\sigma} t_{ij} \exp[-i\mathbf{A}(t) \cdot (\mathbf{R}_i - \mathbf{R}_j)/(2\pi)] c_{i\sigma} c_{j\sigma}$ のように位相因子が付く。ここで $\mathbf{A}(t)$ はベクトルポテンシャルであり、"Pulse Laser" では $\mathbf{A}(t) = \mathbf{A}_0 \exp[-(t-t_0)^2/(2t_{\text{dump}}^2)] \cos[\omega(t-t_0)]$ 、"AC Laser" では $\mathbf{A}(t) = \mathbf{A}_0 \sin[\omega(t-t_0)]$ 、"DC Laser" では $\mathbf{A}(t) = \mathbf{A}_0 t$ となる。

また、各時刻でのベクトルポテンシャルと電場を図示するためのファイル `potential.dat` が出力される。

- Uquench

形式：実数 (デフォルトは 0.0)

説明： U_{quench}

- freq

形式：実数 (デフォルトは 0.1)

説明： ω

- tshift

形式：実数 (デフォルトは 0.0)

説明： t_0

- tdump

形式：実数 (デフォルトは 0.1)

説明： t_{dump}

- VecPotW, VecPotL

形式：実数 (デフォルトはともに 0.0)

説明：時刻 $t = t_0$ でのベクトルポテンシャル \mathbf{A}_0 を逆格子の Fractional coordinate で指定する。逆格子ベクトルは Figs. [図 5.1](#), [図 5.2](#), [図 5.4](#), [図 5.3](#) に表されている格子ベクトルと対応するものとなる。

5.4.2 エキスパートモード用入力ファイル

HPhi++ のエキスパートモードで使用する入力ファイル (*def) に関して説明します。入力ファイルの種別は以下の 4 つで分類されます。

1. List:

キーワード指定なし: 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. Basic parameters:

CalcMod: 計算モードを指定するパラメーターを設定します。

ModPara: 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、Lanczos ステップを何回やるかなど) を設定します。

LocSpin: 局在スピンの位置を設定します (近藤模型でのみ利用)。

3. Hamiltonian:

HPhi++ の Hamiltonian を電子系の表式により指定します。具体的には以下のファイルで指定されます。

Trans: $c_{i\sigma_1}^\dagger c_{j\sigma_2}$ で表される一体項を指定します。

InterAll: $c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$ で表される一般二体相互作用を指定します。

なお、使用頻度の高い相互作用に関しては下記のキーワードで指定することも可能です。

CoulombIntra: $n_{i\uparrow} n_{i\downarrow}$ で表される相互作用を指定します ($n_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$)。

CoulombInter: $n_i n_j$ で表される相互作用を指定します ($n_i = n_{i\uparrow} + n_{i\downarrow}$)。

Hund: $n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}$ で表される相互作用を指定します。

PairHop: $c_{i\uparrow}^\dagger c_{j\uparrow} c_{i\downarrow}^\dagger c_{j\downarrow}$ で表される相互作用を指定します。

Exchange: $S_i^+ S_j^-$ で表される相互作用を指定します。

Ising: $S_i^z S_j^z$ で表される相互作用を指定します。

PairLift: $c_{i\uparrow}^\dagger c_{i\downarrow} c_{j\uparrow}^\dagger c_{j\downarrow}$ で表される相互作用を指定します。

4. Output:

OneBodyG : 出力する一体 Green 関数を指定します。 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ が出力されます。

TwoBodyG : 出力する二体 Green 関数を指定します。 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$

が出力されます。

入力ファイル指定用ファイル

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。

```
CalcMod    calcmmod.def
ModPara    modpara.def
LocSpin    zlocspn.def
Trans      ztransfer.def
InterAll   zinterall.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。

- 必ず指定しなければならないパラメーターは CalcMod, ModPara, LocSpin です。
- 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。
- ファイル読込用キーワードは 表 5.2 により指定します。

表 5.2 定義ファイル一覧

Keywords	指定ファイルの概要
CalcMod	計算モードに関する指定をします。
ModPara	計算で用いるパラメータの指定をします。
LocSpin	各サイトに対して遍歴電子もしくは局在スピンの指定をします。
Trans	一般の一体相互作用に関する設定をします。
InterAll	一般的二体相互作用に関する設定をします。
CoulombIntra	内部クーロン相互作用に関する設定をします。
CoulombInter	サイト間クーロン相互作用に関する設定をします。
Hund	フント結合に関する設定をします。
PairHop	ペアホッピングに関する設定をします。
Exchange	交換相互作用に関する設定をします。
Ising	イジング相互作用に関する設定をします。
PairLift	ペアリフト相互作用に関する設定をします。
OneBodyG	出力する一体グリーン関数 $\langle c_{i\sigma}^\dagger c_{j\sigma} \rangle$ に関する設定をします。
TwoBodyG	出力する二体グリーン関数 $\langle c_{i\sigma}^\dagger c_{j\sigma} c_{k\tau}^\dagger c_{l\tau} \rangle$ に関する設定をします。
SingleExcitation	一体励起状態の生成演算子に関する指定をします。
PairExcitation	二体励起状態の生成演算子に関する指定をします。
SpectrumVec	スペクトル関数を計算するためのリスタート用の入力ベクトルを指定します。

CalcMod ファイル

計算手法、計算モデル、出力モードを指定します。以下のようなフォーマットをしています。

```
CalcType    0
CalcModel   2
CalcEigenVec 0
```

ファイル形式

[string01] [int01]

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。

- [int01]

形式 : int 型

説明 : キーワードに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。

- キーワードを記載後、半角空白を開けた後に整数値を書きます。
- 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。
- キーワード CalcType, CalcModel は省略できません。省略した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。

キーワード

次に各キーワードで指定されるパラメータに関して説明します。

- CalcType

形式 : int 型

説明 : 計算手法の指定を行います。

0: Lanczos 法

1: TPQ を利用した有限温度解析

2: 全対角化

3: LOBCG 法

を選択することができます。

- CalcModel

形式 : int 型

説明 : 計算モデルの指定を行います。

0: fermion Hubbard 模型 (カノニカル : 粒子数・ S_z 保存、粒子数・ S_z 非保存)

1: スピン模型 (カノニカル : S_z 保存)

2: 近藤模型 (カノニカル : 粒子数・ S_z 保存)

3: fermion Hubbard 模型 (グランドカノニカル : 粒子数・ S_z 非保存)

4: スピン模型 (グランドカノニカル : S_z 非保存)

5: 近藤模型 (グランドカノニカル : 粒子数・ S_z 非保存)

を選択することが出来ます。ferimon Hubbard 模型で粒子数保存・ S_z 非保存を選択する場合は、粒子数のみを指定することで計算が可能となります。

- CalcEigenVec

形式 : int 型 (デフォルト値 0)

説明 : 固有ベクトルを計算する際の手法の指定を行います。

0: Lanczos 法 +CG 法 (Lanczos 法での収束が十分でない場合に CG 法での固有ベクトル計算が行われます)

1: Lanczos 法

で選択することが出来ます。

- InitialVecType

形式 : int 型 (デフォルト値 0)

説明 : 固有ベクトルの初期値の型の指定を行います。

0: 複素数

1: 実数

から選択することが出来ます。

- OutputEigenVec

形式 : int 型 (デフォルト値 0)

説明 : 固有ベクトルの出力の指定を行います。0: 出力なし 1: 出力ありから選択することが出来ます。

- InputEigenVec

形式 : int 型 (デフォルト値 0)

説明 : 固有ベクトルの入力指定を行います。0: 入力なし 1: 入力ありから選択することが出来ます。

- ReStart

形式 : int 型 (デフォルト値 0)

説明 : 再計算用のモードを指定します。0: 再計算なし 1: リスタートベクトル入力なし、出力あり 2: リスタートベクトル入力あり、出力あり 3: リスタートベクトル入力あり、出力なしから選択することが出来ます。

- CalcSpec

形式 : int 型 (デフォルト値 0)

説明 : 動的グリーン関数の計算モードを指定します。0: 動的グリーン関数の計算なし 1: 入力ベクトル・励起状態生成ファイルを読み込み計算 2: 三重対角行列の成分を読み込み再計算 3: 三重対角行列の成分と再計算用ベクトルを出力 4: 三重対角行列の成分と再計算用ベクトルを読み込み再計算 5: 三重対角行列の成分と再計算用ベクトルを読み込み再計算・再出力から選択することが出来ます。

- OutputHam

形式 : int 型 (デフォルト値 0)

説明 : (FullDiag) ハミルトニアン出力モードを指定します。0: 出力なし 1: 出力ありから選択することが出来ます。

- InputHam

形式 : int 型 (デフォルト値 0)

説明 : (FullDiag) ハミルトニアンの入力モードを指定します。0: 入力なし 1: 入力ありから選択することが出来ます。

- OutputExcitedVec

形式 : int 型 (デフォルト値 0)

説明 : 励起ベクトルの出力の指定を行います。0: 出力なし 1: 出力ありから選択することが出来ます。

- Scalapack

形式 : int 型 (デフォルト値 0)

説明 : (FullDiag) 全対角化計算での ScaLAPACK ライブラリの使用有無を指定します。

0: ScaLAPACK を使用しない。

1: ScaLAPACK を使用する。

で選択することが出来ます。

- NGPU

形式 : int 型 (デフォルト値 2)

説明 : (FullDiag) 全対角化計算での MAGMA ライブラリを使用する場合の GPU 数を指定します。なお、HPhi++ ではマルチノードでの GPU 計算には対応していません。

ModPara ファイル

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    0
-----
HPhi_Cal_Parameters
-----
CDataFileHead     zvo
CParaFileHead     zqp
-----
Nsite              16
Ncond              16
2Sz                0
Lanczos_max        1000
initial_iv         12
exct               1
LanczosEps         14
LanczosTarget      2
LargeValue         12
NumAve             5
ExpecInterval      20
```

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 (何かが書かれても問題ありません)。
- 6 行: [string01] [string02]
- 7 - 8 行: ヘッダ (何かが書かれても問題ありません)
- 9 行以降: [string01] [int01]

各項目の対応関係は以下の通りです。

- [string01]
形式 : string 型 (固定)
説明 : キーワードの指定を行います。
- [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を指定します。

- [int01]

形式：int 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、半角空白を開けた後に整数値を書きます。
- 行数固定で読み込みを行う為、パラメータの省略はできません。

キーワード

以下、共通パラメータと各手法毎に必要なパラメータを記載します。

共通パラメータ

- CDataFileHead

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_Lanczos_cisajs.dat** として出力されます (xxx に CDataFileHead で指定した文字が記載)。

- Nsite

形式：int 型 (自然数)

説明：サイト数を指定する整数。

- Ncond

形式：int 型 (自然数)

説明：伝導電子数を指定する整数。グランドカノニカルの場合には使用されません。

- 2Sz

形式：int 型 (自然数)

説明： $2S_z$ を指定する整数。グランドカノニカルの場合には使用されません。電子系、近藤格子模型で S_z 保存の計算を行う場合には Ncond を指定する必要があります。

- `initial_iv`

形式 : `int` 型 (整数)

説明 : 初期条件のベクトルを与えます。

- Lanczos 法, CG 法

- * カノニカル集団かつ $\text{initial_iv} \geq 0$ の場合

- ノンゼロの成分が指定されます。

- * $\text{initial_iv} < 0$ の場合

- 乱数のシードが指定され、全ての成分に対して係数がランダムに与えられます。なお、グランドカノニカルの場合は初期状態として多くの状態を持つよう、こちらの様式が適用されます。

- TPQ 法

- 乱数のシードが指定され、全ての成分に対して係数がランダムに与えられます。

初期ベクトル設定の詳細については、[アルゴリズム](#) を参照ください。

- `CalcHS`

形式 : `int` 型 (自然数)

説明 : `CalcHS=1` で量子数を指定したときのヒルベルト空間生成が高速化されます (詳細は <http://issp-center-dep.github.io/HPhi/develop/tips.pdf> を参照)。デフォルトの値は 1 で、高速化アルゴリズムが使用されます。

Lanczos 法で使用するパラメータ

- `exct`

形式 : `int` 型 (自然数)

説明 : Lanczos 法で求める固有ベクトルの番号を指定する整数。例えば、1 なら基底状態のベクトル、2 なら第一励起状態のベクトルを求めます。

- `Lanczos_max`

形式 : `int` 型 (自然数)

説明 : Lanczos ステップを行う回数の最大値を与えます。指定した精度内で収束した場合には、これより短い回数で終了します。この回数以内に収束しない場合はエラー終了します。再計算を行う場合には、`Lanczos_max` の最大数を過去のステップ数よりも大きくする必要があります。

- `LanczosEps`

形式 : int 型 (自然数)

説明 : Lanczos 法の収束判定条件を指定する整数。一つ前のステップの固有値との相対誤差が、 $10^{-\text{LanczosEps}}$ 以下になった場合に収束したと判定します。

- LanczosTarget

形式 : int 型 (自然数)

説明 : 何番目の固有値でランチョスの収束判定を行うかを指定する整数。1 なら基底状態。2 なら第一励起状態を示します。

CG 法で使用するパラメータ

- exct

形式 : int 型 (自然数)

説明 : 求める固有ベクトルの本数を指定します。

- Lanczos_max

形式 : int 型 (自然数)

説明 : CG 法を行う際のステップ数の最大値を与えます。指定した精度内で収束した場合には、これより短い回数で終了します。この回数以内に収束しない場合はエラー終了します。再計算を行う場合には、Lanczos_max の最大数を過去のステップ数よりも大きくする必要があります。

- LanczosEps

形式 : int 型 (自然数)

説明 : 残差ベクトルの 2-ノルムが $10^{-\text{LanczosEps}/2}$ 以下になった場合に収束したと判定します。

TPQ 法で使用するパラメータ

- Lanczos_max

形式 : int 型 (自然数)

説明 : TPQ 法の総ステップ数を与えます。再計算を行う場合には、Lanczos_max の最大数を過去のステップ数よりも大きくする必要があります。

- LargeValue

形式 : double 型 (実数)

説明 : TPQ で使用する $l - \hat{H}/N_s$ の l を指定する整数。

- NumAve

形式 : int 型 (自然数)

説明 : 独立な run を何回行うかを指定する整数。

- ExpecInterval

形式 : int 型 (自然数)

説明 : 相関関数の計算を何回の TPQ ステップおきに行うかの指定する整数。頻度を上げると計算コストが増大するので注意してください。

動的グリーン関数の計算で使用するパラメータ

- OmegaOrg

形式 : complex 型 (複素数)

説明 : 動的グリーン関数を計算する際の振動数 ω の中心点を指定します。スペース区切りで実部、虚部の順に指定し、虚部がない場合は実数として扱います。

- OmegaIm

形式 : double 型 (実数)

説明 : 動的グリーン関数を計算する際の振動数 ω の中心点の虚部を指定します。中心点 OmegaOrg が指定されている場合は OmegaOrg に虚部 OmegaIm を加えます。

- OmegaMax

形式 : complex 型 (複素数)

説明 : 動的グリーン関数を計算する際の振動数 ω の終点を指定します。終点は OmegaOrg+OmegaMax で与えられます。スペース区切りで実部、虚部の順に指定し、虚部がない場合は実数として扱います。

- OmegaMin

形式 : complex 型 (複素数)

説明 : 動的グリーン関数を計算する際の振動数 ω の始点を指定します。終点は OmegaOrg+OmegaMin で与えられます。スペース区切りで実部、虚部の順に指定し、虚部がない場合は実数として扱います。

- NOmega

形式 : int 型 (自然数)

説明 : 動的グリーン関数を計算する際の振動数の刻み幅 $\Delta\omega = (\text{OmegaMax} - \text{OmegaMin})/N_\omega$ を与えるための整数。振動数は $z_n = \text{OmegaOrg} + \text{OmegaMin} + \Delta\omega \times n$ で与えられます。

実時間発展法で使用するパラメータ

- Lanczos_max

形式 : int 型 (自然数)

説明 : 実時間発展法の総ステップ数を与えます。再計算を行う場合には、Lanczos_max の最大数を過去のステップ数よりも大きくする必要があります。

- ExpandCoef

形式 : int 型 (自然数)

説明 : 微小時間 Δt に対して実時間発展を行う際の

$$\exp(-i\hat{\mathcal{H}}\Delta t) = \sum_{n=0}^N \frac{1}{n!} (-i\hat{\mathcal{H}}\Delta t)^n$$

に対する展開次数 N を与えます。

- ExpecInterval

形式 : int 型 (自然数)

説明 : 相関関数の計算を行う際のステップ間隔を指定する整数。頻度を上げると計算コストが増大するので注意してください。

- OutputInterval

形式 : int 型 (自然数)

説明 : 時間発展中の波動関数を出力するステップ間隔を指定する整数。(CalcMod ファイルで OutputEigenVec=1 にすると有効になります)

LocSpin 指定ファイル

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 N_{site} 未満で指定します。

- [int03]

形式 : int 型 (空白不可)

説明 : 局在スピンか遍歴電子かを指定する整数。

0: 遍歴電子 $n > 0$: $2S = n$ の局在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 [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。

Trans 指定ファイル

ここではハミルトニアン

$$H+ = - \sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2}$$

に対する Transfer 積分 $t_{ij\sigma_1\sigma_2}$ を指定します。以下にファイル名を記載します。

```
=====
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：Transfer 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：Transfer 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 N_{site} 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。
- 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。上記の関係が成立しない場合にはエラー終了します。
- 成分が重複して指定された場合にはエラー終了します。

- [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。

InterAll 指定ファイル

ここでは一般二体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。

$$H+ = \sum_{i,j,k,l} \sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$$

なお、スピンに関して計算する場合には、 $i = j, k = l$ となるよう設定してください。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 N_{site} 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。

- Hamiltonian 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。上記の関係が成立しない場合にはエラー終了します。また、エルミート共役の形式は $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$ に対して、 $I_{lkji\sigma_4\sigma_3\sigma_2\sigma_1} c_{l\sigma_4}^\dagger c_{k\sigma_3} c_{j\sigma_2}^\dagger c_{i\sigma_1}$ を満たすように入力してください。
- スピンに関して計算する場合、 $i = j, k = l$ を満たさないペアが存在するとエラー終了します。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

CoulombIntra 指定ファイル

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 ($S = 1/2$ の系でのみ使用可能)。付け加える項は以下で与えられます。

$$H+ = \sum_i U_i n_{i\uparrow} n_{i\downarrow}$$

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 N_{site} 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。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

CoulombInter 指定ファイル

オフ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 ($S = 1/2$ の系でのみ使用可能)。付け加える項は以下で与えられます。

$$H+ = \sum_{i,j} V_{ij} n_i n_j$$

以下にファイル例を記載します。

```
=====
NCoulombInter 6
=====
=====CoulombInter =====
=====
0      1 -0.125000
1      2 -0.125000
2      3 -0.125000
3      4 -0.125000
4      5 -0.125000
5      0 -0.125000
```

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 : オフ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 : オフサイト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 N_{site} 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。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

Hund 指定ファイル

Hund カップリングをハミルトニアンに付け加えます ($S = 1/2$ の系でのみ使用可能)。付け加える項は以下で与えられます。

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

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 N_{site} 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。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

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 ($S = 1/2$ の系でのみ使用可能)。付け加える項は以下で与えられます。

$$H+ = \sum_{i,j} J_{ij}^{\text{Pair}} (c_{i\uparrow}^\dagger c_{j\uparrow} c_{i\downarrow}^\dagger c_{j\downarrow} + h.c.)$$

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 N_{site} 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。

Exchange 指定ファイル

Exchange カップリングをハミルトニアンに付け加えます ($S = 1/2$ の系でのみ使用可能)。電子系の場合には

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

が付け加えられ、スピン系の場合には

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

が付け加えられます。スピン系の $(S_i^+ S_j^- + S_i^- S_j^+)$ を電子系の演算子で書き直すと、 $-(c_{i\uparrow}^\dagger c_{j\uparrow} c_{j\downarrow}^\dagger c_{i\downarrow} + c_{i\downarrow}^\dagger c_{j\downarrow} c_{j\uparrow}^\dagger c_{i\uparrow})$ となることに注意して下さい。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 N_{site} 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。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

Ising 指定ファイル

Ising 相互作用をハミルトニアンに付け加えます ($S = 1/2$ の系でのみ使用可能)。電子系の場合には

$$H+ = \sum_{i,j} J_{ij}^z (n_{i\uparrow} - n_{i\downarrow})(n_{j\uparrow} - n_{j\downarrow})$$

が付け加えられ、スピン系の場合には

$$H+ = \sum_{i,j} J_{ij}^z S_i^z S_j^z$$

が付け加えられます。以下にファイル例を記載します。

```
=====
NIsing 6
=====
=====Ising =====
=====
  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 : Ising 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 : Ising 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 N_{site} 未満で指定します。

- [double01]

形式 : double 型 (空白不可)

説明 : J_{ij}^z 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。
- 成分が重複して指定された場合にはエラー終了します。
- [int01] と定義されている Ising 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。

PairLift 指定ファイル

PairLift カップリングをハミルトニアンに付け加えます ($S = 1/2$ の系でのみ使用可能)。付け加える項は以下で与えられます。

$$H+ = \sum_{i,j} J_{ij}^{\text{PairLift}} (c_{i\uparrow}^\dagger c_{i\downarrow} c_{j\uparrow}^\dagger c_{j\downarrow} + c_{i\downarrow}^\dagger c_{i\uparrow} c_{j\downarrow}^\dagger c_{j\uparrow})$$

以下にファイル例を記載します。

```
=====
NPairLift 6
=====
=====NPairLift =====
=====
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 : PairLift 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 : PairLift 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 N_{site} 未満で指定します。

- [double01]

形式 : double 型 (空白不可)

説明 : J_{ij}^{PairLift} 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。

- スピン系のみで使用可能です。電子系、近藤系で指定した場合は計算に使用されません。
- 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。
- 成分が重複して指定された場合にはエラー終了します。
- [int01] と定義されている PairLift 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。

OneBodyG 指定ファイル

一体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ を計算します。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 N_{site} 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。電子系・近藤格子系の場合は

0: アップスピン、

1: ダウンスピン、

スピン系の場合には、

$0, 1, \dots, 2S + 1$ ($-S - 0.5, -S + 0.5, \dots, S + 0.5$ に対応)

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。
- 成分が重複して指定された場合にはエラー終了します。
- [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。

TwoBodyG 指定ファイル

二体グリーン関数 $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ を計算します。なお、スピンに関して計算する場合には、 $i = j, k = l$ となるよう設定してください。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 N_{site} 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。電子系、近藤格子系の場合には

0: アップスピン、

1: ダウンスピン、

スピン系の場合には、

$0, 1, \dots, 2S + 1$ ($-S - 0.5, -S + 0.5, \dots, S + 0.5$ に対応)

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。
- 成分が重複して指定された場合にはエラー終了します。
- スピンに関して計算する場合、 $i = j, k = l$ を満たさない場合ペアが存在するとエラー終了します。
- [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。

SingleExcitation 指定ファイル

一体励起状態を作成するための演算子 $c_{i\sigma_1}(c_{i\sigma_1}^\dagger)$ を定義します。以下にファイル例を記載します。

```
=====
NSingle          24
=====
===== Single Excitation =====
=====
  0      0      0      1.0      0.0
  0      1      0      1.0      0.0
  1      0      0      1.0      0.0
  (continue...)
11      0      0      1.0      0.0
11      1      0      1.0      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] [double01] [double02]

パラメータ

- [string01]

形式: string 型 (空白不可)

説明: 一体励起演算子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: 一体励起演算子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 N_{site} 未満で指定します。

- [int03]

形式 : int 型 (空白不可)

説明 : スピンを指定する整数。電子系・近藤格子系の場合は

0: アップスピン、

1: ダウンスピン、

スピン系の場合には、

$0, 1, \dots, 2S + 1$ ($-S - 0.5, -S + 0.5, \dots, S + 0.5$ に対応)

を選択することが出来ます。

- [int04]

形式 : int 型 (空白不可)

説明 : 一体励起演算子のタイプを指定する整数。

0: $c_{i\sigma_1}$

1: $c_{i\sigma_1}^\dagger$

を選択することが出来ます。

- [double01], [double02]

形式 : double 型 (空白不可)

説明: $c_{i\sigma_1} (c_{i\sigma_1}^\dagger)$ の実部を [double01]、虚部を [double02] でそれぞれ指定します。

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。
- 成分が重複して指定された場合にはエラー終了します。
- [int01] と定義されている一体励起演算子の総数が異なる場合はエラー終了します。
- [int02]-[int04] 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。

PairExcitation 指定ファイル

二体励起状態を作成するための演算子 $c_{i\sigma_1}c_{j\sigma_2}^\dagger(c_{i\sigma_1}^\dagger c_{j\sigma_2})$ を定義します。なお、 $c_{i\sigma_1}c_{j\sigma_2}^\dagger$ と $c_{i\sigma_1}^\dagger c_{j\sigma_2}$ を混同することは出来ません。また、 S_z 保存の系に対しては $\sigma_1 = \sigma_2$ とする必要があります。以下にファイル例を記載します。

```
=====
NPair          24
=====
===== Pair Excitation =====
=====
  0      0      0      0      0      1.0      0.0
  0      1      0      1      0      1.0      0.0
  1      0      1      0      0      1.0      0.0
  (continue...)
 11      0     11      0      0      1.0      0.0
 11      1     11      1      0      1.0      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] [double01] [double02]

パラメータ

- [string01]

形式: string 型 (空白不可)

説明: 二体励起演算子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: 二体励起演算子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。電子系・近藤格子系の場合は

0: アップスピン、

1: ダウンスピン、

スピン系の場合には、

$0, 1, \dots, 2S + 1$ ($-S - 0.5, -S + 0.5, \dots, S + 0.5$ に対応)

を選択することが出来ます。

- [int06]

形式 : int 型 (空白不可)

説明 : 二体励起演算子のタイプを指定する整数。

0: $c_{i\sigma_1} c_{j\sigma_2}^\dagger$

1: $c_{i\sigma_1}^\dagger c_{j\sigma_2}$

を選択することが出来ます。

- [double01], [double02]

形式 : double 型 (空白不可)

説明 : $c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2})$ の実部を [double01]、虚部を [double02] でそれぞれ指定します。

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。
- 成分が重複して指定された場合にはエラー終了します。
- [int01] と定義されている二体励起演算子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。

SpectrumVec 指定ファイル

スペクトル計算用の入力ファイルのヘッダを指定します。ファイル名およびファイル形式は以下の通りです (ファイル形式は `eigenvec.dat` と同様です)。ファイルデータはバイナリ形式です。

ファイル名

- `##_rank_$.dat`

##は SpectrumVec で指定されるヘッダ、\$はランク番号を表します。また、&&は TPQ 時のサンプリングの番号を表します。

ファイル形式

- 1 行目: `[int01]`
- 2 行目: `[int02]`
- 2 行目-: `[double01] [double02]`

パラメータ

- `[int01]`

形式: int 型

説明: 計算対象のヒルベルト空間数を指定する整数。

- `[int02]`

形式: int 型

説明: 計算に要したステップ数を表します。Lanczos 法では Lanczos ステップ数、TPQ 法ではハミルトニアンを乗算した回数を記載します。

- `[double01], [double02]`

形式: double 型

説明: 入力するベクトルの値を表します。

`[double01]` が実部、`[double02]` が虚部を表します。

OneBodyTE 指定ファイル

各時刻で付加される一体型の演算子 $t_{i\sigma_1 j\sigma_2}(t)c_{i\sigma_1}^\dagger c_{j\sigma_2}$ を定義します。以下にファイル例を記載します。

```
=====
AllTimeStep          100
=====
===== OneBody Time Evolution =====
=====
0.0    4
0  0  1  0      1.0    0.0
1  0  0  0      1.0    0.0
0  1  1  1      1.0    0.0
1  1  0  1      1.0    0.0
0.2    4
(continue...)
```

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 行以降は時刻 t およびその時刻での非ゼロの相互作用の個数を最初に指定し、続けて対応する一体相互作用を指定する形式を取ります。

- m 行: [double01] [int02]
- (m+1) - (m+1+[int02]) 行: [int03] [int04] [int05] [int06] [double02] [double03]

パラメータ

- [string01]

形式: string 型 (空白不可)

説明: ファイル内で定義されている総タイムステップ数を表す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: ファイル内で定義されている総タイムステップ数を指定します。

- [double01]

形式 : double 型 (空白不可)

説明 : 時刻 t を指定します。

- [int02]

形式 : int 型 (空白不可)

説明 : 時刻 t で付加される一体相互作用の個数を指定します。

- [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 N_{site} 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: 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。

- [double02], [double03]

形式 : double 型 (空白不可)

説明 : 時刻 t で付加される一体相互作用の実部を [double02]、虚部を [double03] でそれぞれ指定します。

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。
- ModPara ファイルで指定される LanczosStep が [int02] よりも大きい場合にはエラー終了します。
- [int03]-[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。

TwoBodyTE 指定ファイル

各時刻で付加される二体型の演算子 $I_{i\sigma_1 j\sigma_2 k\sigma_3 l\sigma_4}(t) c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$ を定義します。以下にファイル例を記載します。

```
=====
AllTimeStep          100
=====
===== TwoBody Time Evolution =====
=====
0.0    3
0  1  0  1  0  0  0  0          1.0    0.0
1  0  1  0  0  0  0  0          1.0    0.0
1  0  1  0  2  0  2  0          1.0    0.0
0.2    3
(continue...)
```

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 行以降は時刻 t およびその時刻での非ゼロの相互作用の個数を最初に指定し、続けて対応する二体相互作用を指定する形式を取ります。

- m 行: [double01] [int02]
- $(m+1) - (m+1+[int02])$ 行: [int03] [int04] [int05] [int06] [int07] [int08] [int09] [int10] [double02] [double03]

パラメータ

- [string01]

形式: string 型 (空白不可)

説明: ファイル内で定義されている総タイムステップ数を表す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: ファイル内で定義されている総タイムステップ数を指定します。

- [double01]

形式 : double 型 (空白不可)

説明 : 時刻 t を指定します。

- [int02]

形式 : int 型 (空白不可)

説明 : 時刻 t で付加される二体相互作用の個数を指定します。

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

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], [int06], [int08], [int10]

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。

- [double02], [double03]

形式 : double 型 (空白不可)

説明 : 時刻 t で付加される二体相互作用の実部を [double02]、虚部を [double03] でそれぞれ指定します。

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。
- ModPara ファイルで指定される LanczosStep が [int02] よりも大きい場合にはエラー終了します。
- [int03]-[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。

5.4.3 出力ファイル

出力ファイルに関するファイル形式を記載します。

CHECK_Chemi.dat

Hamiltonian のうち化学ポテンシャル

$$H+ = \sum_{i,\sigma} \mu_{i\sigma} c_{i\sigma}^{\dagger} c_{i\sigma}$$

に関する入力確認を行います。 $\mu_{i\sigma}$ が出力されます。以下にファイル例を記載します。

```
i=0 spin=0 isite1=1 tmp_V=0.000000
i=1 spin=0 isite1=2 tmp_V=0.000000
i=2 spin=0 isite1=3 tmp_V=0.000000
i=3 spin=0 isite1=4 tmp_V=0.000000
i=4 spin=0 isite1=5 tmp_V=0.000000
i=5 spin=0 isite1=6 tmp_V=0.000000
...
```

ファイル形式

以下のようなファイル形式をとります。

- $i=[\text{int01}]$ $\text{spin}=[\text{int02}]$ $\text{isite1}=[\text{int03}]$ $\text{tmp_V}=[\text{double01}]$

パラメータ

- $[\text{int01}]$

形式 : int 型

説明 : 入力番号。

- $[\text{int02}]$

形式 : int 型

説明 : $\mu_{i\sigma}$ のスピン番号 σ を表す整数。

0: アップスピン

1: ダウンスピン

を表します。

- [int03]

形式 : int 型

説明 : $\mu_{i\sigma}$ のサイト番号 i を表す整数。

- [double01]

形式 : double 型

説明 : $\mu_{i\sigma}$ の値。

CHECK_InterAll.dat

Hamiltonian の一般二体相互作用のうち対角成分

$$H+ = \sum_{i,j,\sigma} I_{iijj\sigma_1\sigma_1\sigma_2\sigma_2} c_{i\sigma_1}^\dagger c_{i\sigma_1} c_{i\sigma_2}^\dagger c_{i\sigma_2}$$

に関する入力確認を行います。 $I_{iijj\sigma_1\sigma_1\sigma_2\sigma_2}$ が出力されます。以下にファイル例を記載します。

```
i=0 isite1=1 A_spin=0 isite2=2 B_spin=0 tmp_V=0.500000
i=1 isite1=1 A_spin=0 isite2=2 B_spin=1 tmp_V=-0.500000
i=2 isite1=1 A_spin=1 isite2=2 B_spin=0 tmp_V=-0.500000
i=3 isite1=1 A_spin=1 isite2=2 B_spin=1 tmp_V=0.500000
i=4 isite1=2 A_spin=0 isite2=3 B_spin=0 tmp_V=0.500000
i=5 isite1=2 A_spin=0 isite2=3 B_spin=1 tmp_V=-0.500000
...
```

ファイル形式

以下のようなファイル形式をとります。

- `i=[int01] isite1=[int02] A_spin=[int03] isite2=[int04] B_spin=[int05] tmp_V=[double01]`

パラメータ

- `[int01]`

形式 : int 型

説明 : 入力番号。

- `[int02], [int04]`

形式 : int 型

説明 : $I_{iijj\sigma_1\sigma_1\sigma_2\sigma_2}$ のサイト番号を表す整数。

`[int02]` が i , `[int04]` が j を表します。

- `[int03], [int05]`

形式 : int 型

説明 : $I_{iijj\sigma_1\sigma_1\sigma_2\sigma_2}$ のスピン番号を表す整数。

[int03] が σ_1 , [int05] が σ_2 に対応し、それぞれ

0: アップスピン

1: ダウンスピン

を表します。

- [double01]

形式 : double 型

説明 : $I_{iijj\sigma_1\sigma_2\sigma_2}$ の値。

CHECK_CoulombIntra.dat

Hamiltonian のオンサイトクーロン相互作用

$$H+ = \sum_i U_i n_{i\uparrow} n_{i\downarrow}$$

に関する入力確認を行います。 U_i が出力されます。以下にファイル例を記載します。

```
i=0 isite1=1 tmp_V=4.000000
i=1 isite1=2 tmp_V=4.000000
i=2 isite1=3 tmp_V=4.000000
i=3 isite1=4 tmp_V=4.000000
i=4 isite1=5 tmp_V=4.000000
i=5 isite1=6 tmp_V=4.000000
```

ファイル形式

以下のようなファイル形式をとります。

- $i=[\text{int01}]$ $\text{isite1}=[\text{int02}]$ $\text{tmp_V}=[\text{double01}]$

パラメータ

- $[\text{int01}]$

形式 : int 型

説明 : 入力番号。

- $[\text{int02}]$

形式 : int 型

説明 : U_i のサイト番号 i を表す整数。

- $[\text{double01}]$

形式 : double 型

説明 : U_i の値。

CHECK_Hund.dat

Hamiltonian の Hund カップリング

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

に関する入力確認を行います。 J_{ij}^{Hund} が出力されます。以下にファイル例を記載します。

```
i=0 isite1=1 isite2=2 tmp_V=0.250000
i=1 isite1=2 isite2=3 tmp_V=0.250000
i=2 isite1=3 isite2=4 tmp_V=0.250000
i=3 isite1=4 isite2=5 tmp_V=0.250000
i=4 isite1=5 isite2=6 tmp_V=0.250000
i=5 isite1=6 isite2=1 tmp_V=0.250000
```

ファイル形式

以下のようなファイル形式をとります。

- `i=[int01] isite1=[int02] isite2=[int03] tmp_V=[double01]`

パラメータ

- `[int01]`

形式 : int 型

説明 : 入力番号。

- `[int02], [int03]`

形式 : int 型

説明 : J_{ij}^{Hund} のサイト番号を表す整数。

`[int02]` が i , `[int03]` が j を表します。

- `[double01]`

形式 : double 型

説明 : J_{ij}^{Hund} の値。

CHECK_INTER_U.dat

Hamiltonian のオフサイトクーロン相互作用

$$H+ = \sum_i V_{ij} n_i n_j$$

に関する入力確認を行います。 V_{ij} が出力されます。以下にファイル例を記載します。

```
i=0 isite1=1 isite2=2 tmp_V=-0.125000
i=1 isite1=2 isite2=3 tmp_V=-0.125000
i=2 isite1=3 isite2=4 tmp_V=-0.125000
i=3 isite1=4 isite2=5 tmp_V=-0.125000
i=4 isite1=5 isite2=6 tmp_V=-0.125000
i=5 isite1=6 isite2=1 tmp_V=-0.125000
```

ファイル形式

以下のようなファイル形式をとります。

- $i=[\text{int01}]$ $\text{isite1}=[\text{int02}]$ $\text{isite2}=[\text{int03}]$ $\text{tmp_V}=[\text{double01}]$

パラメータ

- $[\text{int01}]$

形式 : int 型

説明 : 入力番号。

- $[\text{int02}], [\text{int03}]$

形式 : int 型

説明 : V_{ij} のサイト番号を表す整数。

$[\text{int02}]$ が i , $[\text{int03}]$ が j を表します。

- $[\text{double01}]$

形式 : double 型

説明 : V_{ij} の値。

CHECK_Memory.dat

使用するメモリサイズの出力を行います。配列サイズおよび必要なメモリを出力します。以下にファイル例を記載します。

```
MAX DIMENSION idim_max=400
REQUIRED MEMORY max_mem=0.000019 GB
```

ファイル形式

以下のようなファイル形式をとります。

- MAX DIMENSION idim_max=[int01]
- REQUIRED MEMORY max_mem=[double01] GB

パラメータ

- [int01]

形式：int 型

説明：ヒルベルトスペースの数を表します。

- [double01]

形式：double 型

説明：ヒルベルトスペースの確保に必要とするメモリサイズを表します (単位は GB)。

WarningOnTransfer.dat

Transfer の成分が重複している場合に出力されます。以下にファイル例を記載します。

```
double conuntings in transfers: i=0 j=2 spni 0 spnj 0
double conuntings in transfers: i=2 j=0 spni 0 spnj 0
double conuntings in transfers: i=0 j=2 spni 1 spnj 1
double conuntings in transfers: i=2 j=0 spni 1 spnj 1
```

ファイル形式

以下のようなファイル形式をとります。

- double conuntings in transfers: i=[int01] j=[int02] spni [int03] spnj [int04]

パラメータ

- [int01], [int02]

形式 : int 型

説明 : 重複している Transfer のサイト番号を表します。

- [int03], [int04]

形式 : int 型

説明 : 重複している Transfer のスピン番号を表します。

0: アップスピン

1: ダウンスピン

を表します。

CalcTimer.dat

計算処理にかかった時間が出力されます。計算処理、処理番号、計算処理時間 [秒] の順に出力します。以下に TPQ 法の場合の出力例を記載します。

```

All                                [0000]      12.94052
  sz                              [1000]       0.01795
  diagonalcalc                    [2000]       0.00693
  CalcByTPQ                        [3000]      12.90670
    FirstMultiply                 [3100]       0.08416
      rand    in FirstMultiply    [3101]       0.00172
      mltply  in FirstMultiply    [3102]       0.07707
    expec_energy_flct             [3200]       9.06255
      calc flctuation in expec_energy_flct [3201]       1.67779
      mltply  in expec_energy_flct [3202]       7.31207
    expec_onebody                 [3300]       0.11640
    expec_twobody                 [3400]       3.28796
    Multiply                      [3500]       0.14840
    FileIO                        [3600]       0.20493
=====
All mltply                        [0001]       7.38883
  diagonal                       [0100]       0.04153
  Hubbard                        [0300]       7.34636
    trans    in Hubbard          [0310]       7.34595
      double                               [0311]       0.00000
      single                               [0312]       0.00000
      inner                                [0313]       7.34299
    interall in Hubbard          [0320]       0.00008
      interPE                          [0321]       0.00000
      inner                            [0322]       0.00000
    pairhopp in Hubbard          [0330]       0.00006
      interPE                          [0331]       0.00000
      inner                            [0332]       0.00000
    exchange in Hubbard          [0340]       0.00004
      interPE                          [0341]       0.00000
      inner                            [0342]       0.00000
=====

```

TimeKeeper.dat

計算経過情報が出力されます。以下に Lanczos 法の場合の出力例を記載します。

```
diagonal calculation finishes: Wed Sep 16 22:58:49 2015
Lanczos Eigen Value start: Wed Sep 16 22:58:49 2015
1 th Lanczos step: Wed Sep 16 22:58:49 2015
...
122 th Lanczos step: Wed Sep 16 22:58:49 2015
Lanczos Eigenvalue finishes: Wed Sep 16 22:58:49 2015
Lanczos Eigenvector finishes: Wed Sep 16 22:58:49 2015
Lanczos expec energy finishes: Wed Sep 16 22:58:49 2015
CG Eigenvector finishes: Wed Sep 16 22:58:49 2015
CG expec energy finishes: Wed Sep 16 22:58:50 2015
CG expec_cisajs finishes: Wed Sep 16 22:58:50 2015
CG expec_cisajacktalt begins: Wed Sep 16 22:58:50 2015
```

ファイル名

- ##_TimeKeeper.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

sz_TimeKeeper.dat

ヒルベルトスペースを確保する際の経過情報が出力されます。以下に出力例を記載します。

```
initial sz : Wed Sep 16 22:58:49 2015
num_threads==4
omp parallel sz finishes: Wed Sep 16 22:58:49 2015
mid omp parallel sz : Wed Sep 16 22:58:49 2015
omp parallel sz finishes: Wed Sep 16 22:58:49 2015
```

ファイル名

- ##_sz_TimeKeeper.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

Time_CG_EigenVector.dat

(Lanczos 法のみ) CG 法で固有ベクトルを計算する際のログを出力します。

```
allocate succeed !!!
b[4341]=1.000000 bnorm== 1.000000
i_itr=0 itr=5 0.0411202543 0.0000100000
...
i_itr=0 itr=155 0.0000000058 0.0000100000
CG OK: t_itr=155
i_itr=0 itr=155 time=0.000000
fabs(fabs(xb)-1.0)=0.9955114473313577
b[4341]=0.004489 bnorm== 1.000000
i_itr=1 itr=5 13.0033983157 0.0000100000
...
CG OK: t_itr=275
i_itr=1 itr=120 time=0.000000
fabs(fabs(xb)-1.0)=0.0000000000001295
number of iterations in inv1:i_itr=1 itr=120
t_itr=275 0.000000
```

ファイル名

- ##_Time_CG_EigenVector.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

energy.dat

(Lanczos 法、LOBCG 法のみ) Lanczos 法もしくは CG 法で求めた固有ベクトルを用いて計算したエネルギー、ダブルトンと $\langle S_z \rangle$ を出力します。以下にファイル例を記載します。

method="Lanczos"の場合

```
Energy  -7.1043675920
Doublon  0.4164356536
Sz  0.0000000000
```

method="CG"の場合

```
State 0
  Energy  -7.1043675920
  Doublon  0.4164356536
  Sz  0.0000000000

State 1
:
```

ファイル名

- ##_energy.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- Energy [double01]
- Doublon [double02]
- Sz [double03]

パラメータ

- [double01]

形式 : double 型

説明 : Lanczos 法・CG 法で求めた固有ベクトルを用い計算したエネルギー。

- [double02]

形式 : double 型

説明 : Lanczos 法・CG 法で求めた固有ベクトルを用い計算したダブロン $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ (N_s はサイト数)。

- [double03]

形式 : double 型

説明 : 固有ベクトルを用い計算した $\langle S_z \rangle$ 。

Lanczos_Step.dat

(Lanczos 法、LOBCG 法でのみ出力) 固有ベクトルを求めている際のログを出力します。以下にファイル例を示します。

```
LanczosStep  E[1] E[2] E[3] E[4] Target:E[3] E_Max/Nsite
stp = 2 1.2149586211 14.6560471044 xxxxxxxxxxx xxxxxxxxxxx 0.0000000000 xxxxxxxxxxx
stp=4 -5.6626980051 3.1523174817 12.4860778911 21.2322666770 12.4860778911 2.6540333346
stp=6 -8.5113374325 -2.3219709559 4.3459108959 11.5079386600 4.3459108959 3.0307814358
stp=8 -9.5061025854 -5.2757708534 -0.1734100333 5.2236216333 -0.1734100333 3.2049774861
stp=10 -9.7541889139 -6.6054773893 -2.9493235242 1.2364826532 -2.9493235242 3.
↳2686702753
...
stp=84 -10.4543987874 -9.8960493865 -9.7550111859 -9.7407358084 -9.7550111859 3.
↳3731105157
stp=86 -10.4543987874 -9.8960493865 -9.7550111859 -9.7407358084 -9.7550111859 3.
↳3731105157
stp=88 -10.4543987874 -9.8960493865 -9.7550111859 -9.7407358084 -9.7550111859 3.
↳3731105157
```

method="CG"の場合

Step	Residual-2-norm	Threshold	Energy		
1	6.79819e+00	8.19743e-07	7.86586e+00	8.19743e+00	8.02804e+00
2	7.47402e+00	3.69905e-07	3.35827e+00	3.63546e+00	3.69905e+00
3	5.30943e+00	2.44472e-07	-2.44472e+00	-2.23296e+00	-1.95487e+00
4	4.52737e+00	5.10297e-07	-5.10297e+00	-4.92390e+00	-4.58682e+00
5	3.66168e+00	7.14105e-07	-7.14105e+00	-6.91226e+00	-6.44532e+00
6	3.12717e+00	8.27201e-07	-8.27201e+00	-7.93262e+00	-7.44680e+00
...					
152	1.05602e-06	1.04544e-06	-1.04544e+01	-9.89605e+00	-9.89605e+00
153	1.07401e-06	1.04544e-06	-1.04544e+01	-9.89605e+00	-9.89605e+00
154	9.45018e-07	1.04544e-06	-1.04544e+01	-9.89605e+00	-9.89605e+00

ファイル名

- ##_Lanczos_Step.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- method="Lanczos"の場合

stp= [int01] [double01] [double02] [double03] [double04] [double-a] [double-b]

- method="CG"の場合

[int01] [double-c] [double-d] [double01] [double02] ...

パラメータ

- [int01]

形式 : int 型

説明 : Lanczos 法・LOBCG 法での計算ステップ数。

- [double01], [double02], [double03], [double04] ...

形式 : double 型

説明 : Lanczos 法・LOBCG 法で計算時に求められた固有値を昇順に並べたもの。Lanczos 法では 4 個分、LOBCG 法では `exct` で指定した個数が出力されます (上記の例では ‘`exct=3`’)。Lanczos 法では縮退している固有値はまとめて 1 個と数えられますが、LOBCG 法では縮退している個数に応じて固有値が出力されます。上記の例では第一励起状態が 2 重以上に縮退していることが分かります。

- [double-a]

形式 : double 型

説明 : (Lanczos 法のみ) `LanczosTarget` で指定した、収束判定に用いられる固有値が出力されます (上記の例では ‘`LanczosTarget=3`’)。

- [double-b]

形式 : double 型

説明 : (Lanczos 法のみ) 最大固有値をサイト数で割ったもの。TPQ 法での `LargeValue` の下限値となります。

- [double-c]

形式 : double 型

説明 : (LOBCG 法のみ) 計算した固有ベクトルの残差の 2-ノルムの中で最大のもの。収束判定に用いられます。

- [double-d]

形式 : double 型

説明 : (LOBCG 法のみ) 収束判定のしきい値. `LanczosEps` で指定した値に基底状態のエネルギーの絶対値をかけています.

Time_TPQ_Step.dat

(TPQ 法でのみ出力) TPQ 法でのステップ毎の開始時間を出力します。再計算の場合は値が追記されます。以下にファイル例を示します。

```
set 0 step 1:TPQ begins: Wed Jul 13 07:59:20 2016
set 0 step 2:TPQ begins: Wed Jul 13 07:59:20 2016
set 0 step 3:TPQ begins: Wed Jul 13 07:59:20 2016
...
set 4 step 1997:TPQ begins: Wed Jul 13 07:59:32 2016
set 4 step 1998:TPQ begins: Wed Jul 13 07:59:32 2016
set 4 step 1999:TPQ begins: Wed Jul 13 07:59:32 2016
```

ファイル名

- ##_TPQ_Step.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- set [int01] stp [int02]: TPQ begins: [string01]

パラメータ

- [int01]

形式 : int 型

説明 : TPQ 法でのシード数。

- [int02]

形式 : int 型

説明 : TPQ 法でのステップ数。

- [string01]

形式 : string 型

説明 : 計算開始時間。

Norm_rand.dat

(TPQ 法でのみ出力) TPQ 法での有限温度計算時のログを出力します。再計算の場合は値が追記されます。以下にファイル例を示します。

```
# inv_temp, global_norm, global_1st_norm, step_i
0.017471 19.046586 11.288975 1
0.034863 19.089752 11.288975 2
...
31.999572 20.802362 11.288975 1997
32.015596 20.802362 11.288975 1998
32.031620 20.802362 11.288975 1999
```

ファイル名

- Norm_rand???.dat

???は TPQ 法計算時の run の番号を表します。

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 逆温度 $1/k_B T$ 。

- [double02]

形式 : double 型

説明 : TPQ 法で計算される規格化前の波動関数 (ベクトル) のノルム: $\langle \tilde{\psi}_k | \tilde{\psi}_k \rangle$ 。ただし、 $|\tilde{\psi}_k\rangle \equiv (I - \hat{H}/N_s)|\psi_{k-1}\rangle$ 。

- [double03]

形式 : double 型

説明：規格化前の初期波動関数 (ランダムベクトル) のノルム： $\langle \tilde{\psi}_0 | \tilde{\psi}_0 \rangle$ 。ただし、 $|\tilde{\psi}_0\rangle$ は規格化前の初期波動関数。

- [int01]

形式：int 型

説明：初期ランダムベクトルに $(l - \hat{H}/N_s)$ (l は ModPara ファイルの LargeValue、 N_s はサイト数) を作
用させた回数。

SS_rand.dat

(TPQ 法でのみ出力)TPQ 法での有限温度計算結果を出力します。再計算の場合は値が追記されます。以下にファイル例を示します。

```
# inv_tmp, energy, phys_var, phys_doublon, phys_num, step_i
0.017471  5.526334 45.390269 1.464589 6.000000 1
0.034863  5.266718 42.655559 1.434679 6.000000 2
...
31.999572  -4.814170 23.176231 0.590568 6.000000 1997
32.015596  -4.814170 23.176231 0.590568 6.000000 1998
32.031620  -4.814170 23.176231 0.590568 6.000000 1999
```

ファイル名

- SS_rand???.dat

???は TPQ 法計算時の run の番号を表します。

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [double04] [double05] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 逆温度 $1/k_B T$ (ただし, $k_B = 1$)。

- [double02]

形式 : double 型

説明 : エネルギーの期待値 $\langle H \rangle$ 。

- [double03]

形式 : double 型

説明 : ハミルトニアン の 2 乗の期待値 $\langle H^2 \rangle$ 。

- [double04]

形式 : double 型

説明 : ダブロンダブロン期待値 $\sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ 。

- [double05]

形式 : double 型

説明 : 粒子数 $\langle \hat{n} \rangle$ 。

- [int01]

形式 : int 型

説明 : 初期ランダムベクトルに $(l - \hat{H}/N_s)$ (l は ModPara ファイルの LargeValue、 N_s はサイト数) を作用させた回数。

Flct_rand.dat

(TPQ 法でのみ出力) TPQ 法での揺らぎに関する有限温度計算結果を出力します。再計算の場合は値が追記されます。以下にファイル例を示します。

```
# inv_temp, N, N^2, D, D^2, Sz, Sz^2, step_i
0.0826564 12.00 144.00 0.00 0.00 0.0009345626081113 0.2500 1
0.1639935 12.00 144.00 0.00 0.00 0.0023147006319775 0.2500 2
0.2440168 12.00 144.00 0.00 0.00 0.0037424057659867 0.2500 3
...
135.97669 12.00 144.00 0.00 0.00 -0.0000000000167368 0.2500 1998
136.04474 12.00 144.00 0.00 0.00 -0.0000000000165344 0.2500 1999
```

ファイル名

- Flct_rand???.dat

???は TPQ 法計算時の run の番号を表します。

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [double04] [double05] [double06] [double07] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 逆温度 $1/k_B T$ 。

- [double02]

形式 : double 型

説明 : 全粒子数 $\sum_i \langle \hat{n}_i \rangle$ 。

- [double03]

形式 : double 型

説明 : 粒子数の 2 乗の期待値: $\langle (\sum_i \hat{n}_i)^2 \rangle$ 。

- [double04]

形式 : double 型

説明 : ダブロン $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ の期待値 (ただし N_s はサイト数)。

- [double05]

形式 : double 型

説明 : ダブロン の二乗 $\frac{1}{N_s} \langle (\sum_i n_{i\uparrow} n_{i\downarrow})^2 \rangle$ の期待値 (ただし N_s はサイト数)。

- [double06]

形式 : double 型

説明 : スピンの S_z 成分 $\frac{1}{N_s} \sum_i \langle \hat{S}_i^z \rangle$ の期待値 (ただし N_s はサイト数)。

- [double07]

形式 : double 型

説明 : スピンの S_z 成分 の二乗 $\frac{1}{N_s} \langle (\sum_i \hat{S}_i^z)^2 \rangle$ の期待値 (ただし N_s はサイト数)。

- [int01]

形式 : int 型

説明 : 初期ランダムベクトルに $(l - \hat{H}/N_s)$ (l は ModPara ファイルの LargeValue、 N_s はサイト数) を作
用させた回数。

Time_TE_Step.dat

(実時間発展法でのみ出力) 実時間発展法でのステップ毎の開始時間を出力します。以下にファイル例を示します。

```
step 1:TE begins: Wed Jul 13 07:59:20 2017
step 2:TE begins: Wed Jul 13 07:59:20 2017
step 3:TE begins: Wed Jul 13 07:59:20 2017
...
step 1997:TE begins: Wed Jul 13 07:59:32 2017
step 1998:TE begins: Wed Jul 13 07:59:32 2017
step 1999:TE begins: Wed Jul 13 07:59:32 2017
```

ファイル名

- ##_TE_Step.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- stp [int01]: TE begins: [string01]

パラメータ

- [int01]

形式 : int 型

説明 : TE 法でのステップ数。

- [string01]

形式 : string 型

説明 : 計算開始時間。

Norm.dat

(実時間発展法でのみ出力) 実時間発展法でのタイムステップ毎のログを出力します。以下にファイル例を示します。

```
# time, norm, step_i
0.0000000000000000 0.9999999999999994 0
0.0100000000000000 1.0000233421898765 1
0.0200000000000000 1.0000211100654208 2
0.0300000000000000 1.0000182214014706 3
0.0400000000000000 1.0000148460317946 4
0.0500000000000000 1.0000111372562455 5
0.0600000000000000 1.0000072252313270 6
0.0700000000000000 1.0000032174168609 7
0.0800000000000000 0.999992048382456 8
0.0900000000000000 0.9999952720176869 9
0.1000000000000000 0.9999915078951970 10
```

ファイル名

- Norm.dat

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 時間 t 。

- [double02]

形式 : double 型

説明 : 実時間発展法で計算される時刻 t での波動関数 (ベクトル) のノルム。

- [int01]

形式 : int 型

説明：タイムステップ数。

SS.dat

(実時間発展法でのみ出力) 実時間発展法での各ステップ毎の計算結果を出力します。以下にファイル例を示します。

```
# time, energy, phys_var, phys_doublon, phys_num, step_i
0.0000000000000000 -6.0412438187293001 38.8635272290786489 ...
0.0100000000000000 -5.9310482979751606 37.9593669819686923 ...
0.0200000000000000 -5.8287182777288828 37.1390062697724801 ...
0.0300000000000000 -5.7384311863736031 36.4282319427381651 ...
0.0400000000000000 -5.6636677030535481 35.8466140292489897 ...
0.0500000000000000 -5.6070659264425551 35.4081795274108799 ...
0.0600000000000000 -5.5703150294725914 35.1222606981659666 ...
0.0700000000000000 -5.5540895348193438 34.9942503380419154 ...
0.0800000000000000 -5.5580244678717312 35.0260574979670665 ...
0.0900000000000000 -5.5807312981978212 35.2161499389042660 ...
0.1000000000000000 -5.6198544688797947 35.5591788356216298 ...
...
```

ファイル名

- SS.dat

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [double04] [double05] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 時間 t 。

- [double02]

形式 : double 型

説明 : エネルギーの期待値 $\langle H \rangle$ 。

- [double03]

形式 : double 型

説明：ハミルトニアン²の期待値 $\langle H^2 \rangle$ 。

- [double04]

形式：double 型

説明：ダブロン $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ (ただし N_s はサイト数)。

- [double05]

形式：double 型

説明：粒子数 $\langle \hat{n} \rangle$ 。

- [int01]

形式：int 型

説明：タイムステップ数。

Ftct.dat

(実時間発展法でのみ出力) 実時間発展法での揺らぎに関する各時刻での計算結果を出力します。以下にファイル例を示します。

```
# time, N, N^2, D, D^2, Sz, Sz^2, step_i
0.0000000000000000 7.999999999999999 63.9999999999999929 ...
0.0100000000000000 8.000000000000000 64.0000000000004832 ...
0.0200000000000000 8.000000000000000 64.000000000000142 ...
0.0300000000000000 8.000000000000000 64.0000000000008100 ...
0.0400000000000000 7.999999999999999 63.9999999999993463 ...
0.0500000000000000 7.999999999999999 63.9999999999996163 ...
0.0600000000000000 7.999999999999999 63.9999999999997016 ...
0.0700000000000000 8.000000000000000 64.0000000000006679 ...
0.0800000000000000 8.000000000000000 64.0000000000007390 ...
0.0900000000000000 7.999999999999999 63.9999999999996803 ...
0.1000000000000000 7.999999999999999 63.9999999999992539 ...
...
```

ファイル名

- Ftct.dat

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [double04] [double05] [double06] [double07] [int01]

パラメータ

- [double01]

形式 : double 型

説明 : 時間 t 。

- [double02]

形式 : double 型

説明 : 全粒子数 $\sum_i \langle \hat{n}_i \rangle$ 。

- [double03]

形式 : double 型

説明：粒子数の 2 乗の期待値 $\langle (\sum_i \hat{n}_i)^2 \rangle$ 。

- [double04]

形式：double 型

説明：ダブロン $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ の期待値 (ただし N_s はサイト数)。

- [double05]

形式：double 型

説明：ダブロンの二乗 $\frac{1}{N_s} \langle (\sum_i n_{i\uparrow} n_{i\downarrow})^2 \rangle$ の期待値 (ただし N_s はサイト数)。

- [double06]

形式：double 型

説明：スピンの S_z 成分 $\frac{1}{N_s} \sum_i \langle \hat{S}_i^z \rangle$ の期待値 (ただし N_s はサイト数)。

- [double07]

形式：double 型

説明：スピンの S_z 成分の二乗 $\frac{1}{N_s} \langle (\sum_i \hat{S}_i^z)^2 \rangle$ の期待値 (ただし N_s はサイト数)。

- [int01]

形式：int 型

説明：タイムステップ数。

Eigenvalue.dat

(FullDiag でのみ出力) 全対角法で計算したエネルギーを出力します。以下にファイル例を記載します。

```
0 -4.8141698096
1 -3.7968502453
2 -3.2462822372
...
397 13.9898305290
398 14.4896221034
399 14.8525199079
```

ファイル形式

以下のファイル形式で記載されます。

- [int01] [double01]

パラメータ

- [int01]

形式 : int 型

説明 : 固有値の index。基底エネルギーを 0 として、エネルギーの低い順から採番する。

- [double01]

形式 : double 型

説明 : エネルギー $\langle H \rangle$ 。

phys.dat

(FullDiag でのみ出力) 全対角法で計算したエネルギーと物理量を出力します。エネルギーの低い基底エネルギーから順に出力されます。以下にファイル例を記載します。

<H>	<N>	<Sz>	<S2>	<D>
-4.814170	0.000000	0.000000	-0.000000	0.590568
-3.796850	0.000000	0.000000	1.333333	0.423804
...				
14.489622	0.000000	0.000000	0.000000	2.550240
14.852520	0.000000	0.000000	0.000000	2.329157

ファイル名

- カノニカル: `##_phys_Nup_$$Ndown%%.dat`
- グランドカノニカル: `##_phys.dat`

##は ModPara ファイル内の [string02] で指定されるヘッダ、\$\$は Nup、%% は Ndown を表します。

ファイル形式

1 行目はヘッダで、2 行目以降は以下のファイル形式で記載されます。

- [double01] [double02] [double03] [double04] [double05]

パラメータ

- [double01]

形式 : double 型

説明 : エネルギーの期待値 $\langle H \rangle$ 。

- [double02]

形式 : double 型

説明 : 粒子数の期待値 $\langle \hat{n} \rangle$ 。

- [double03]

形式 : double 型

説明 : スピンの z 成分の期待値 $\langle S_z \rangle$ 。

- [double04]

形式 : double 型

説明 : スピンの 2 乗の期待値 $\langle S^2 \rangle$ 。

- [double05]

形式 : double 型

説明 : ダブロン $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ (ただし:math:N_s はサイト数)。

ham.dat

(FullDiag でのみ出力) CalcMod ファイルで OutputHam=1 の場合に、HΦ 内部で計算されたハミルトニアンを MatrixMarket 形式で出力します。CalcMod ファイルで InputHam=1 とすると、定義ファイル一式と本ファイルを読み込み、再計算することができます。以下にファイル例を記載します。

```

%%%MatrixMarket matrix coordinate complex hermitian
28 28 56
1 1 1.000000 0.000000
2 1 0.500000 0.000000
3 2 0.500000 0.000000
4 3 0.500000 0.000000
5 4 0.500000 0.000000
6 5 0.500000 0.000000
7 6 0.500000 0.000000
7 7 1.000000 0.000000
...

```

ファイル名

- ##_ham.dat

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- 1 行：ヘッダ
- 2 行：[int01] [int02] [int03]
- 3 行-：[int04] [int05] [double01] [double02]

パラメータ

- [int01]

形式：int 型

説明：Hamiltonian の行数。

- [int02]

形式：int 型

説明：Hamiltonian の列数。

- [int03]

形式 : int 型

説明 : Hamiltonian の非零の要素数。

- [double01], [double02]

形式 : double 型

説明 : Hamiltonian の値を表します。

[double01] が実部、[double02] が虚部を表します。

cisajs.dat

OneBodyG 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。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
...
```

ファイル名

Lanczos 法: ##_cisajs.dat

TPQ 法: ##_cisajs_set??step%%.dat

実時間発展法: ##_cisajs_step%%.dat

全対角化法、LOBCG 法: ##_cisajs_eigen&&.dat

##は ModPara ファイル内の [string02] で指定されるヘッダ、??は TPQ 法計算時の run の番号、%% は TPQ 法でのステップ数、&&は固有値の番号を表します。

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。

cisajscalt.dat

TwoBodyG 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。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
...
```

ファイル名

Lanczos 法: `##_cisajscalt.dat`

TPQ 法: `##_cisajscalt_set??step%%.dat`

実時間発展法: `##_cisajscalt_step%%.dat`

全対角化法、LOBCG 法: `##_cisajscalt_eigen&&.dat`

##_は ModPara ファイル内の [string02] で指定されるヘッダ、??は TPQ 法計算時の run の番号、%% は TPQ 法でのステップ数、&&は固有値の番号を表します。

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。

eigenvec.dat

CalcMod ファイルの OutputEigenVec=1 の場合に、Lanczos 法で計算された固有ベクトルを出力します。InputEigenVec=1 の場合には、出力されたファイルの読み込みを行います。ファイルはバイナリ形式で出力されます。ファイル名およびファイル形式は以下の通りです。

ファイル名

- ##_eigenvec_&&_rank_\$.dat

##は ModPara ファイル内の [string02] で指定されるヘッダ、&&は固有値の番号、\$\$はランク番号を表します。

ファイル形式

Note: eigen_vector の一番最初の成分に計算に使用しない値が入っています。

tmpvec.dat

CalcMod ファイルの ReStart=1, 2 の場合に、計算途中のベクトルを出力します。ファイルはバイナリ形式で出力されます。ファイル名およびファイル形式は以下の通りです (ファイル形式は eigenvec.dat と同様です)。

ファイル名

- Lanczos 法 : ##_tmpvec_rank_\$.dat
- TPQ 法, LOBPCG 法 : ##_tmpvec_set_&&_rank_\$.dat

##は ModPara ファイル内の [string02] で指定されるヘッダ、\$はランク番号を表します。また、&&は TPQ 時のサンプリングの番号もしくは LOBPCG 法での固有値の番号を表します。

ファイル形式

次のようなソースコードを用いて、バイナリファイルとして出力されます (実際の HPhi++ のソースコードとは変数名等が異なります)。

Lanczos

```
fp = fopen("zvo_tmpvec_rank_0.dat", "wb");
fwrite(&number_of_interations, sizeof(int), 1, fp);
fwrite(&local_size, sizeof(unsigned long int), 1, fp);
fwrite(&last_vector[0], sizeof(complex double), local_size+1, fp);
fwrite(&second_last_vector[0], sizeof(complex double), local_size+1, fp);
fclose(fp);
```

TPQ および LOBPCG

```
fp = fopen("zvo_tmpvec_set_0_rank_0.dat", "wb");
fwrite(&number_of_interations, sizeof(int), 1, fp);
fwrite(&local_size, sizeof(unsigned long int), 1, fp);
fwrite(&last_vector[0], sizeof(complex double), local_size+1, fp);
fclose(fp);
```

ただし、number_of_interations は反復回数、local_size は固有ベクトルのサイズ (MPI 並列を使っている場合は全ヒルベルト次元とは異なります)、last_vector は最後の反復でのベクトル、second_last_vector は最後から 2 番目の反復でのベクトルをそれぞれ表します。

`last_vector`、`second_last_vector` の一番最初の成分に計算に使用しない値が入っています。

DynamicalGreen.dat

動的グリーン関数の計算結果を出力します。ファイル名およびファイル形式は以下の通りです。

ファイル名

- `##_DynamicalGreen.dat`

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- 1 行目-: [double01] [double02] [double03] [double04]

パラメータ

- [double01], [double02]

形式 : double 型

説明 : 振動数の値を表します。

[double01] が実部、[double02] が虚部を表します。

- [double03], [double04]

形式 : double 型

説明 : 動的グリーン関数の値を表します。

[double03] が実部、[double04] が虚部を表します。

recalcvec.dat

Lanczso 法を用いた動的グリーン関数の再計算に必要な 2 つのベクトルが出力されます。ファイルはバイナリ形式で出力されます。ファイル名およびファイル形式は以下の通りです。

ファイル名

- `##_recalcvec_rank_$.dat`

##は ModPara ファイル内の [string02] で指定されるヘッダ、\$\$はランク番号を表します。

ファイル形式

- 1 行目 : [int01]
- 2 行目 : [int02]
- 3 行目 - [int02]+3 行: [double01] [double02]
- 4 行目 - $2 \times [\text{int02}] + 4$ 行: [double03] [double04]

パラメータ

- [int01]

形式 : int 型

説明 : 動的グリーン関数の計算に要したステップ数を表します。

- [int02]

形式 : Long int 型

説明 : 計算対象のヒルベルト空間数を指定する整数。

- [double01], [double02]

形式 : double

説明 : Lanczos 法での \vec{v}_{k+1} を出力します。[double01] が実部、[double02] が虚部を表します。一番最初の成分に計算に使用しない値が入っています。

- [double03], [double04]

形式 : double

説明 : Lanczos 法での \vec{v}_{k+1} を出力します。[double01] が実部、[double02] が虚部を表します。一番最初の成分に計算に使用しない値が入っています。

TMcomponents.dat

Lanczos 法を用いた動的グリーン関数の再計算に必要な三重対角行列の要素と励起状態のノルムが出力されます。ファイル名およびファイル形式は以下の通りです。

ファイル名

- `##_TMcomponents.dat`

##は ModPara ファイル内の [string02] で指定されるヘッダを表します。

ファイル形式

- 1 行目 : [int01]
- 2 行目 : [double01]
- 3 行目 - [int02]+3 行: [double02] [double03]

パラメータ

- [int01]

形式 : int 型

説明 : 動的グリーン関数の計算に要したステップ数 N_d を表します。

- [double01]

形式 : double 型

説明 : 励起状態のノルムを表します。

- [double02], [double03]

形式 : double 型

説明 : Lanczos 法を用いた動的グリーン関数の再計算に必要な三重対角行列の要素 α_i, β_i ($i = 1, \dots, N_d$) の値を表します。一番最初の成分に計算に使用しない値が入っています。[double02] が α_i 、[double03] が β_i に対応します。

excitedvec.dat

CalcMod ファイルの OutputExcitedVec=1 の場合に、pair.def と入力ベクトルから計算された励起ベクトルを出力します。ファイルはテキスト形式で出力されます。以下にファイル例を記載します。

```
4096
0.0009135367, -0.0004231751
-0.0009712430, 0.0293999045
-0.0294644178, 0.0210086435
0.0214247977, -0.0107587147
0.0272643022, -0.0404634256
0.0034322694, -0.0184446640
0.0019911098, 0.0004403706
0.0114685735, -0.0114381935
0.0092888239, 0.0088235535
...
```

ファイル名

- ##_excitedvec_&&_rank_\$.dat

##は ModPara ファイル内の [string02] で指定されるヘッダ、&&は固有値の番号、\$はランク番号を表します。

ファイル形式

- Line 1: [int01]
- Lines2-: [double01] [double02]

パラメータ

- [int01]

形式 : long int 型

説明 : ベクトルの次元数。

- [double01], [double02]

形式 : double 型

説明 : 励起ベクトルの値を表します。

[double01] が実部、[double02] が虚部を表します。

5.4.4 エラーメッセージ一覧

- ERROR ! Unsupported Keyword !

存在しないキーワードを指定した場合に表示され、プログラムは停止します。

- "ERROR ! Keyword is duplicated !

同じキーワードを 2 回指定した場合に表示され、プログラムは停止します。

- ERROR ! Unsupported Solver : *solver*

- ERROR ! Unsupported Model : *model*

- Sorry, this system is unsupported in the STANDARD MODE...

Please use the EXPART MODE, or write a NEW FUNCTION and post it us.

method, model, lattice のどれかまたは複数にサポートしていないパラメーターを入れた場合、プログラムは停止します。

- ERROR ! $\text{abs}(2 * \text{Sz}) > \text{n site}$ in Hubbard model !

- ERROR ! $\text{Nelec} > 2 * \text{n site}$ in Hubbard model !

- ERROR ! $(\text{nelec} + 2 * \text{Sz}) \% 2 \neq 0$ in Hubbard model !

- ERROR ! $\text{nelec} \leq \text{n site} \ \&\& \ 2 * |\text{Sz}| > \text{nelec}$ in Hubbard model !

- ERROR ! $\text{nelec} > \text{n site} \ \&\& \ 2 * |\text{Sz}| > 2 * \text{n site} - \text{nelec}$ in Hubbard model !

- ERROR ! $\text{abs}(2 * \text{Sz}) > \text{n site}$ in Spin model !

- ERROR ! $(\text{n site} + 2 * \text{Sz}) \% 2 \neq 0$ in Spin model !

- ERROR ! $\text{abs}(2 * \text{Sz}) > \text{n site}$ in Hubbard model !

- ERROR ! $\text{Nelec_cond} / 2 + \text{Nelec_loc} > \text{n site}$ in Kondo model !

- ERROR ! $(\text{nelec_cond} + \text{nelec_loc} + 2 * \text{Sz}) \% 2 \neq 0$ in Kondo model !

- ERROR ! $\text{nelec_cond} \leq \text{n site} / 2 \ \&\& \ 2 * |\text{Sz}| > \text{nelec_cond} + \text{nelec_loc} \dots$

- ERROR ! $\text{nelec_cond} > \text{n site} / 2 \ \&\& \ \text{abs}(\text{Sz}^2) > \text{n site} / 2 * 3 - \text{nelec} \dots$

カノニカル集団の計算において、入力されたサイト数、電子数、全スピン z 成分が実現できない組み合わせである場合 (例えば、電子数がサイト数の 2 倍よりも大きい、など) プログラムは停止します。

- Check ! `` `` is SPECIFIED but will NOT be USED.

“ Please COMMENT-OUT this line “

“ or check this input is REALLY APPROPRIATE for your purpose ! “

使われないパラメーターを指定した時には、ユーザーに入力ファイルの確認を促しプログラムは停止します。実際に必要のないパラメーターの場合は該当する行を削除もしくはコメントアウトしてください。

- ERROR ! ``\ `` is NOT specified !

必ず指定しなければならないキーワードが指定されていない場合にはプログラムは停止します。

- ##### DEFAULT VALUE IS USED #####

これはエラーメッセージではありません。入力ファイルで指定がなかったためにデフォルト値が使われたことを知らせるメッセージです。

5.5 アルゴリズム

5.5.1 Lanczos 法

手法概要

Lanczos 法の解説については TITPACK^{*1} のマニュアルと 線形計算の数理^{*2} を参考にしています。

Lanczos 法はある初期ベクトルにハミルトニアンを作用させて最大・最小近傍の固有値・固有ベクトルを求める方法です。Lanczos 法で固有値を求める際には Hilbert 空間の次元の大きさの波動関数を表すベクトルが 2 つ^{*3} あれば原理的には実行可能なため、大規模疎行列の対角化手法として有用であることが知られています。後述するように、固有ベクトルを求める際には Hilbert 空間の次元の大きさのベクトルがもう 1 本必要です。

Lanczos 法の 原理はべき乗法に基づいています。べき乗法ではある任意のベクトル \vec{x}_0 に Hamiltonian を逐次的に作用させて、 $\mathcal{H}^n \vec{x}_0$ を作成します。このとき、生成される空間 $\mathcal{K}_{n+1}(\mathcal{H}, \vec{x}_0) = \{\vec{x}_0, \mathcal{H}^1 \vec{x}_0, \dots, \mathcal{H}^n \vec{x}_0\}$ は Krylov 部分空間といわれます。初期ベクトルを \mathcal{H} の固有ベクトル \vec{e}_i (対応する固有値を E_i とする) の重ね合わせで表すと

$$\vec{x}_0 = \sum_i a_i \vec{e}_i$$

となります。ここで、 E_0 を絶対値最大の固有値としました。またハミルトニアンはエルミートであるため、固有値は全て実数であることに注意する必要があります。これに Hamiltonian の \mathcal{H}^n を作用させると、

$$\mathcal{H}^n \vec{x}_0 = E_0^n \left[a_0 \vec{e}_0 + \sum_{i \neq 0} \left(\frac{E_i}{E_0} \right)^n a_i \vec{e}_i \right]$$

となり、絶対値最大固有値 E_0 に対応する固有ベクトルが支配的になります。適切な変換を行って、この固有ベクトルの成分を抽出するのが Lanczos 法です。

^{*1} http://www.stat.phys.titech.ac.jp/~nishimori/titpack2_new/index-e.html

^{*2} M. Sugihara, K. Murota, Theoretical Numerical Linear Algebra, Iwanami Studies in Advanced Mathematics, Iwanami Shoten, Publishers, 2009.

^{*3} 高速化のために、In HPhi++ ではハミルトニアンの対角成分を表すベクトル 1 本と、スピン z 成分 S_z 保存、粒子数保存の場合はその状態を指定するベクトル 1 本を余計に確保しています。いずれのベクトルの大きさも Hilbert 空間の次元です。

Lanczos 法では、 $\mathcal{K}_n(\mathcal{H}, \vec{x}_0)$ から正規直交ベクトル $\vec{v}_0, \dots, \vec{v}_{n-1}$ を次の手続きにしたがって順次生成していきます。初期条件を $\vec{v}_0 = \vec{x}_0/|\vec{x}_0|, \beta_0 = 0, \vec{x}_{-1} = 0$ とすると、正規直交基底は次の手続きによって逐次的に生成することができます。

$$\begin{aligned}\alpha_k &= (\mathcal{H}\vec{v}_k, \vec{v}_k) \\ \vec{w} &= \mathcal{H}\vec{v}_k - \beta_k \vec{v}_{k-1} - \alpha_k \vec{v}_k \\ \beta_{k+1} &= |\vec{w}| \\ \vec{v}_{k+1} &= \frac{\vec{w}}{|\vec{w}|}\end{aligned}$$

この定義から、 α_k, β_k ともに実数であることに注意する必要があります。

これらの正規直交基底が成す Krylov 部分空間の中でもとのハミルトニアンに対する固有値問題は、

$$T_n = V_n^\dagger \mathcal{H} V_n$$

と変形されます。ここで、 V_n を $\vec{v}_i (i = 0, 1, \dots, n-1)$ を並べた行列です。 T_n は三重対角行列であり、その対角成分は α_i 、副対角成分は β_i で与えられます。この三重対角行列 T_n の固有値はもとのハミルトニアンの固有値の近似値となっています ($V^\dagger V = I, I$ は単位行列であることに注意)。 T_n の固有ベクトルを $\tilde{\vec{e}}_i$ とするともとのハミルトニアンの固有ベクトルとの関係は $\vec{e}_i = V \tilde{\vec{e}}_i$ で与えられます。 V を覚えていれば、Lanczos 法を行うと同時に固有ベクトルを求めることができますが、実際の場合は (Hilbert 空間の次元 \times Lanczos 法の反復回数) の大きさの行列を保持することは不可能です。そこで、Lanczos 法で固有値を求めた後、 $\tilde{\vec{e}}_i$ を保存しておき^{*4}、再び同じ Lanczos 法の計算を行い求めた V から元の固有ベクトルを再現するようにしています。

Lanczos 法では、元の Hilbert 空間の次元より十分小さい反復回数で最大及び最小に近い固有値を精度よく求めることができることが知られています。すなわち T_n の次元 n は数百-数千程度ですみます。定量的には、最大及び最小固有値の評価の誤差は Lanczos の反復回数に対して指数関数的に減少することが示されています (詳細は Ref.^{*2} を参照して下さい)。

逆反復法

固有値の近似値がわかっているときは適当なベクトル \vec{y}_0 に対して $(\mathcal{H} - E_n)^{-1}$ を逐次的に作用させれば、固有値 E_n に対応する固有ベクトルの成分が支配的になり、固有ベクトルを精度良く求めることができます。

$(\mathcal{H} - E_n)^{-1}$ を作用させる方程式を書き換えると以下の連立方程式が得られます。

$$\vec{y}_k = (\mathcal{H} - E_n) \vec{y}_{k+1}$$

この連立方程式を共役勾配法 (CG 法) を用いて解くことで、固有ベクトルを求めることができます。その固有ベクトルから固有値およびその他の相関関数を求めることができます。ただし、CG 法の実行にはヒルベルト空間の次元のベクトルを 4 本確保する必要があり、大規模系の計算を実行する際にはメモリが足りなくなる恐れがあるので注意が必要です。

^{*4} $\tilde{\vec{e}}_i$ の次元は高々 Lanczos 法の反復回数であることに注意。

実際の実装について

初期ベクトルの設定について

Lanczos 法では、スタンダードモード用の入力ファイル、もしくはエキスパートモードでは ModPara で指定する `initial_iv`($\equiv r_s$) により初期ベクトルの設定方法を指定します。また、初期ベクトルは ModPara で指定される入力ファイルの `InitialVecType` を用い、実数もしくは複素数の指定をすることができます。

- カノニカル集団かつ `initial_iv` ≥ 0 の場合

ヒルベルト空間のうち、

$$(N_{\text{dim}}/2 + r_s) \% N_{\text{dim}}$$

の成分が与えられます。ここで、 N_{dim} は対象となるヒルベルト空間の総数で、 $N_{\text{dim}}/2$ はデフォルト値 `initial_iv` = 1 で特殊なヒルベルト空間の選択をさけるために加えられています。なお、選択された成分の係数は実数の場合は 1、複素数の場合には $(1 + i)/\sqrt{2}$ が与えられます。

- グランドカノニカル集団 もしくは `initial_iv` < 0 の場合

初期ベクトルはランダムベクトルとして与えられます。乱数のシードは

$$123432 + |r_s|$$

で指定します。ここで、 n_{run} は run の回数であり、run の最大回数はスタンダードモード入力ファイル、もしくは ModPara で指定される入力ファイルの `NumAve` で指定します。`initial_iv` は入力ファイルで指定します。乱数は SIMD-oriented Fast Mersenne Twister (dSFMT) を用い発生させています^{*5}。

収束判定について

HPhi++ では、 T_n の対角化に lapack のルーチン `dsyev` を使用しており、 T_n の基底状態の次の固有値（第一励起状態のエネルギー）を収束判定条件に用いています。デフォルトの設定では、最初の 5 回の Lanczos ステップの後に、2 回毎に T_n の対角化を行い、前の Lanczos ステップの第一励起状態のエネルギーと指定した精度以内で一致すれば、収束したと判定しています。なお、収束する際の精度は `CDataFileHead` (エキスパートモードでは ModPara ファイル内) で指定することが可能です。

その後、Lanczos 法を再度行い、逐次 V を求めて、指定した準位の固有ベクトルを求めます。得られた固有ベクトル $|n\rangle$ を用い、エネルギーの期待値 $E_n = \langle n | \mathcal{H} | n \rangle$ およびバリエーション $\Delta = \langle n | \mathcal{H}^2 | n \rangle - (\langle n | \mathcal{H} | n \rangle)^2$ を求めて、 E_n が Lanczos 法で求めた固有値と指定した精度で一致しているか、バリエーションが指定した精度以下になっているかをチェックしています。指定した精度に達していれば、対角化を終了しています。

指定した精度に達していない場合には逆反復法を用いて再度固有ベクトルを求め直します。逆反復法の初期ベクトルとして Lanczos 法で求めた固有ベクトルをとった方が一般に収束が早いので、標準の設定ではそのように取っています。

^{*5} <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/index.html>

5.5.2 完全対角化

手法概要

ハミルトニアン \mathcal{H} を実空間配置 $|\psi_j\rangle (j = 1 \cdots N)$ を用いて作成します: $H_{ij} = \langle \psi_i | \hat{H} | \psi_j \rangle$ 。この行列を対角化することで、固有値 E_i 、固有ベクトル $|\Phi_i\rangle$ を求めることができます ($i = 1 \cdots N$)。なお、対角化では lapack の **dsyev** また **zheev** を用いています。また、有限温度計算用に各固有エネルギー状態の期待値 $\langle A_i \rangle \equiv \langle \Phi_i | \hat{A} | \Phi_i \rangle$ を計算・出力するようにしています。

有限温度物理量の計算

完全対角化で求めた $\langle A_i \rangle \equiv \langle \Phi_i | \hat{A} | \Phi_i \rangle$ を用い、

$$\langle \hat{A} \rangle = \frac{\sum_{i=1}^N \langle A_i \rangle e^{-\beta E_i}}{\sum_{i=1}^N e^{-\beta E_i}}$$

の関係から有限温度の物理量を計算します。実際の計算処理としてはポスト処理により計算を行います。

5.5.3 熱的純粋量子状態による有限温度計算

杉浦・清水によって、少数個（サイズが大きい場合はほぼ一つ）の波動関数から有限温度の物理量を計算する方法が提案されました^{*1}。その状態は熱的純粋量子状態 (TPQ) と呼ばれています。TPQ はハミルトニアンを波動関数に順次作用させて得られるので、Lanczos 法の技術がそのまま使うことができます。ここでは、とくに計算が簡単な, micro canonical TPQ(mTPQ) の概要を述べます。

$|\psi_0\rangle$ をあるランダムベクトルとします。これに $(l - \mathcal{H}/N_s)(l$ はある定数、 N_s はサイト数) を k 回作用させた（規格化された）ベクトルは次のように与えられます。

$$|\psi_k\rangle \equiv \frac{(l - \mathcal{H}/N_s)|\psi_{k-1}\rangle}{|(l - \mathcal{H}/N_s)|\psi_{k-1}\rangle|}.$$

この $|\psi_k\rangle$ が mTPQ 状態で、この mTPQ 状態に対応する逆温度 β_k は以下のように内部エネルギー u_k から求めることができます。

$$\beta_k \sim \frac{2k/N_s}{l - u_k}, \quad u_k = \langle \psi_k | \mathcal{H} | \psi_k \rangle / N_s.$$

そして、任意^{*2} の物理量 \hat{A} の β_k での平均値は

$$\langle \hat{A} \rangle_{\beta_k} = \langle \psi_k | \hat{A} | \psi_k \rangle / N_s$$

となります。有限系では最初の乱数ベクトルによる誤差がありますので、いくつか独立な計算を行って、 $|\psi_0\rangle$ に関する平均値および標準偏差を見積もっています。

^{*1} S. Sugiura, A. Shimizu, Phys. Rev. Lett. **108**, 240401 (2012).

^{*2} 局所的にという条件がつきます。

実際の実装について

初期ベクトルの設定について

熱的純粋量子状態による有限温度計算では、初期ベクトルは全ての成分に対してランダムな係数を与えます。初期ベクトルの係数の型は ModPara で指定される入力ファイルの InitialVecType を用い、実数もしくは複素数の指定をすることができます。乱数のシードは initial_iv ($\equiv r_s$) により

$$123432 + (n_{\text{run}} + 1) \times |r_s| + k_{\text{Thread}} + N_{\text{Thread}} \times k_{\text{Process}}$$

で与えられます。ここで、 n_{run} は run の回数であり、run の最大回数はスタンダードモード用入力ファイル、もしくは ModPara で指定される入力ファイルの NumAve で指定します。initial_iv はスタンダードモード用の入力ファイル、もしくはエキスパートモードでは ModPara で指定される入力ファイルで指定します。乱数は SIMD-oriented Fast Mersenne Twister(dSFMT) を用い発生させています^{*3}。また、 k_{Thread} , N_{Thread} , k_{Process} はそれぞれスレッド番号、スレッド数、プロセス番号を表します。したがって同じ initial_iv を用いても、並列数が異なる場合には別の初期波動関数が生成されます。

5.5.4 動的グリーン関数

$\mathcal{H}\Phi$ では励起状態 $|\Phi'\rangle = \hat{O}|\Phi_0\rangle$ に対する動的関数

$$I(z) = \langle \Phi' | \frac{1}{\mathcal{H} - z\hat{I}} | \Phi' \rangle$$

を計算することができます。演算子 \hat{O} はシングル励起状態

$$\sum_{i,\sigma_1} A_{i\sigma_1} c_{i\sigma_1} (c_{i\sigma_1}^\dagger)$$

およびペア励起状態

$$\sum_{i,j,\sigma_1,\sigma_2} A_{i\sigma_1 j\sigma_2} c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2})$$

として、それぞれ定義することが出来ます。例えば、動的スピン感受率を計算する場合はペア励起演算子を用い

$$\hat{O} = \hat{S}(\mathbf{k}) = \sum_j \hat{S}_j^z e^{i\mathbf{k}\cdot\mathbf{r}_j} = \sum_j \frac{1}{2} (c_{j\uparrow}^\dagger c_{j\uparrow} - c_{j\downarrow}^\dagger c_{j\downarrow}) e^{i\mathbf{k}\cdot\mathbf{r}_j}$$

のように定義することで計算することができます。なお、動的グリーン関数の計算には、Lanczos 法を用いた連分数展開による解法 $\hat{A}\tilde{\alpha}^{*1}$ とシフト型クリロフ理論による解法 $\hat{A}\tilde{\alpha}^{*2}$ の 2 つが実装されています。詳細については各文献を参照してください。

^{*3} <http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/SFMT>.

^{*1} E. daggerotto, Rev. Mod. Phys. **66**, 763-840 (1994).

^{*2} S.Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

5.5.5 実時間発展

$\mathcal{H}\Phi$ では初期波動関数を $|\Phi(t_0)\rangle$ として、

$$|\Phi(t_n)\rangle = e^{-i\mathcal{H}\Delta t_n}|\Phi(t_{n-1})\rangle$$

の係数を用いて逐次時間発展の計算をしています。ここで時刻 $t_n = \sum_{j=1}^n \Delta t_j$ です。実際の計算では $e^{-i\mathcal{H}\Delta t_n}$ を

$$e^{-i\mathcal{H}\Delta t_n} = \sum_{l=0}^m \frac{1}{l!} (-i\mathcal{H}\Delta t_n)^l$$

と近似しています。展開次数の上限 m は `ModPara` ファイルの `ExpandCoef` で指定することができます。展開次数が十分かどうかは、ノルム $\langle \Phi(t_n) | \Phi(t_n) \rangle = 1$ が成立しているかで検証することができます。

5.5.6 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。スピンの演算子は次のように生成・消滅演算子で書き換えられます。

$$\begin{aligned} S_{iz} &= \sum_{\sigma=-S}^S \sigma c_{i\sigma}^\dagger c_{i\sigma} \\ S_i^+ &= \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma+1}^\dagger c_{i\sigma} \\ S_i^- &= \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma}^\dagger c_{i\sigma+1} \end{aligned}$$

5.6 Calculation of physical quantities

In this chapter, we explain post procedures to calculate physical quantities. All scripts used in this chapter are included in `tool` folder.

5.6.1 Calculation of specific heat

5.6.2 Calculation of spectrum function

5.6.3 Calculation of magnetization curve

.1 相関関数の Fourier 変換ユーティリティー

.1.1 概要

本資料は, mVMC および HPhi++ で計算されたサイト表示の静的相関関数を Fourier 変換し, 出力するユーティリティと HPhi++ で計算されたサイト表示の動的相関関数を Fourier 変換し, 出力するユーティリティに関するマニュアルである.

要件

本ユーティリティの使用要件は mVMC および HPhi++ と同じである.

対応する量

本ユーティリティは以下の相関関数の Fourier 変換に対応している.

1 体相関

$$\langle \hat{c}_{\mathbf{k}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{k}\beta\uparrow} \rangle \equiv \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{c}_{\mathbf{0}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{R}\beta\uparrow} \rangle \quad (3)$$

$$\langle \hat{c}_{\mathbf{k}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{k}\beta\downarrow} \rangle \equiv \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{c}_{\mathbf{0}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{R}\beta\downarrow} \rangle \quad (4)$$

密度-密度相関

$$\langle \hat{\rho}_{\mathbf{k}\alpha} \hat{\rho}_{\mathbf{k}\beta} \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle (\hat{\rho}_{\mathbf{0}\alpha} - \langle \hat{\rho}_{\mathbf{0}\alpha} \rangle) (\hat{\rho}_{\mathbf{R}\beta} - \langle \hat{\rho}_{\mathbf{R}\beta} \rangle) \rangle \quad (5)$$

スピン-スピン相関

$$\langle \hat{S}_{\mathbf{k}\alpha}^z \hat{S}_{\mathbf{k}\beta}^z \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{S}_{\mathbf{0}\alpha}^z \hat{S}_{\mathbf{R}\beta}^z \rangle \quad (6)$$

$$\langle \hat{S}_{\mathbf{k}\alpha}^+ \hat{S}_{\mathbf{k}\beta}^- \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{S}_{\mathbf{0}\alpha}^+ \hat{S}_{\mathbf{R}\beta}^- \rangle \quad (7)$$

$$\langle \hat{\mathbf{S}}_{\mathbf{k}\alpha} \cdot \hat{\mathbf{S}}_{\mathbf{k}\beta} \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{\mathbf{S}}_{\mathbf{0}\alpha} \cdot \hat{\mathbf{S}}_{\mathbf{R}\beta} \rangle \quad (8)$$

動的相関

$$\begin{aligned} & \text{begin}\{\text{align}\} \text{ langle } \{\text{hat X}\}_{\{\{\text{bf k}\} \text{ alpha uparrow}\}^{\{\text{dagger}\}}} \{\text{hat X}\}_{\{\{\text{bf k}\} \text{ beta uparrow}\}} \text{rangle } (\omega) \\ & \equiv \sum_{\{\{\text{bf R}\}^{\{N_{\{\{\text{bf R}\}\}}\}}\}} e^{\{-i \{\{\text{bf k}\} \cdot \{\{\text{bf R}\}\}\}} \text{ langle } \{\text{hat X}\}_{\{\{\text{bf R}\} \text{ alpha uparrow}\}^{\{\text{dagger}\}}} \\ & (\omega - \{\{\text{hat H}\}\}^{\{-1\}} \{\{\text{hat X}\}_{\{\{\text{bf 0}\} \text{ beta uparrow}\}} \text{rangle } \text{end}\{\text{align}\} \end{aligned}$$

励起演算子 \hat{X} は任意のものを指定できる。スタンダードモードでは上記の 1 体相関、2 体相関の励起演算子を自動的に生成できる。

1.2 静的相関関数についてのチュートリアル

このチュートリアルでは、正方格子ハバードモデル (8 サイト) を例にとり説明する。

HPhi++/vmc.out の実行

- HPhi++ の場合

基底状態および相関関数の計算を行う。入力ファイルは次の通り。

```
a0w = 2
a0l = 2
a1w = -2
a1l = 2
model="Hubbard"
method="CG"
lattice="square"
t=1.0
U=8.0
nelec = 8
2Sz=0
```

```
$ HPhi++ -s input
```

- mVMC の場合

まず変分波動関数の最適化を行う。入力ファイルは次の通り。

```
a0w = 2
a0l = 2
a1w = -2
a1l = 2
model="Hubbard"
lattice="square"
t=1.0
U=8.0
nelec = 8
2Sz=0
```

```
$ vmc.out -s input
```

相関関数を計算するために、入力ファイルに以下の行を付け加える。

```
NVMCCalMode = 1
```

相関関数を計算する。

```
$ vmc.out -s input output/zqp_opt.dat
```

これにより、カレントディレクトリの `output/` 以下に 1 体および 2 体の相関関数が出力される。

関連するファイル

- StdFace.def (mVMC/HPhi++ のマニュアル参照)
- zqp_opt.dat (mVMC のマニュアル参照)
- greenone.def (計算する相関関数のインデックスの指定)
- greentwo.def (計算する相関関数のインデックスの指定)

相関関数のフーリエ変換

ユーティリティプログラム `greenr2k` を使って、相関関数をフーリエ変換する。

```
$ echo "4 20
G 0 0 0
X 0.5 0 0
M 0.5 0.5 0
G 0 0 0
16 16 1" >> geometry.dat
$ greenr2k namelist.def geometry.dat
```

これにより、カレントディレクトリの `output/` 以下にフーリエ変換された相関関数が出力される。

関連するファイル

- output/zvo_cisajs_001.dat (サイト表示の 1 体および 2 体相関関数の計算結果)
- output/zvo_cisajs.dat (サイト表示の 1 体および 2 体相関関数の計算結果)
- output/zvo_cisajsktalt_001.dat (サイト表示の 1 体および 2 体相関関数の計算結果)
- output/zvo_cisajsktalt.dat (サイト表示の 1 体および 2 体相関関数の計算結果)
- geometry.dat (サイトの位置と軌道のインデックス, k 点)
- output/zvo_corr*.dat (k パス上での相関関数)

相関関数のプロット

gnuplot を使って、相関関数を k 空間でプロットする.

```
load "kpath.gp"
plot "output/zvo_corr_eigen0.dat" u 1:12 w l
```

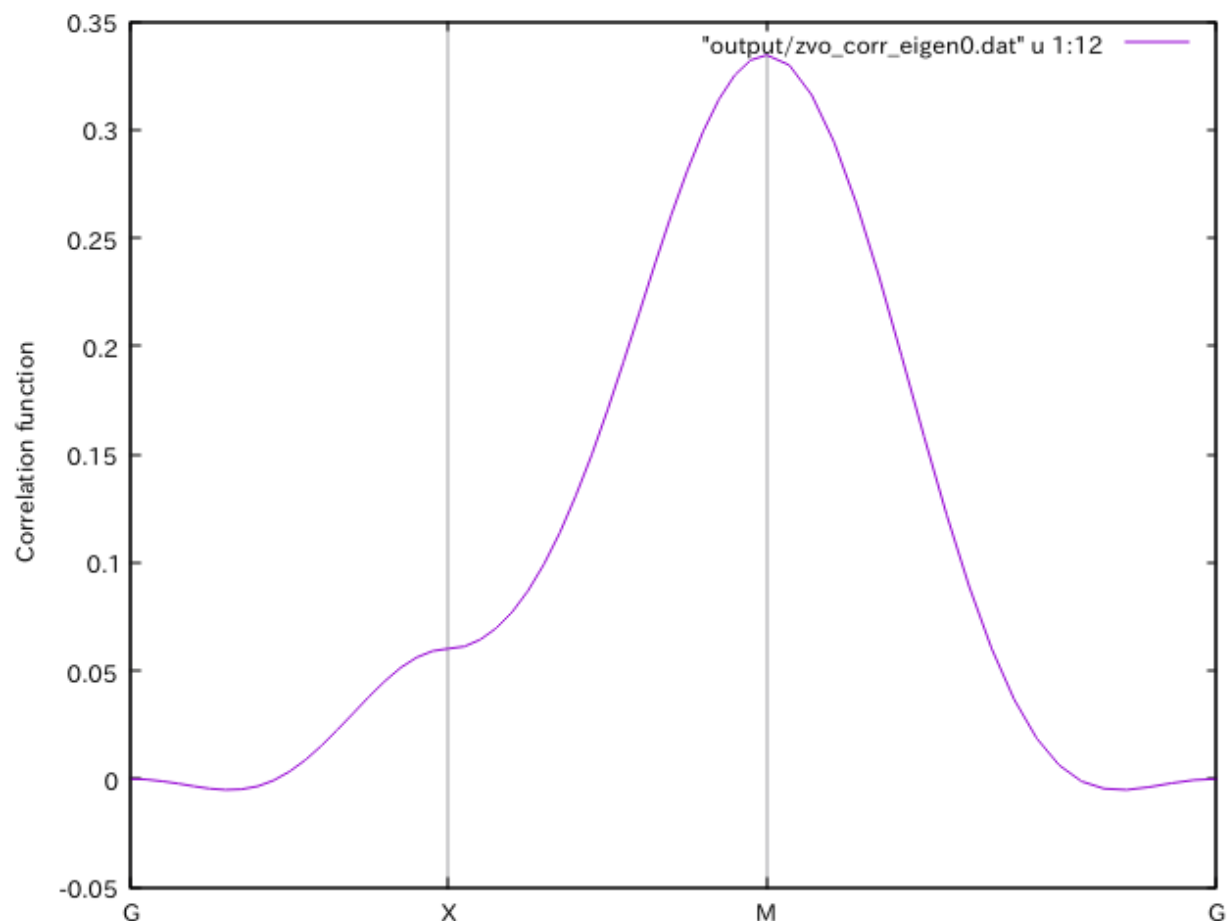


図 6 Figure 6: 相関関数 $\langle S_k \cdot S_k \rangle$ (12 列目) をプロットした図.

関連するファイル

- kpath.gp (*gnuplot* スクリプト)
- output/zvo_corr*.dat (k パス上での相関関数)

.1.3 動的相関関数についてのチュートリアル

このチュートリアルでは, 1 次元ハイゼンベルグ模型モデル (12 サイト) を例にとり説明する.

HPhi++ の実行

基底状態および相関関数の計算を行う。入力ファイルは次の通り。

```
model = Spin
lattice = Chain
method = CG
L = 12
2Sz = 0
J = 1.0
CalcSpec = Scratch
SpectrumType = SzSz_r
OmegaIm = 0.1
OmegaMin = -6.0
OmegaMax = -2.0
```

```
$ HPhi++ -s input
```

これにより、カレントディレクトリの `output/` 以下に動的相関関数が出力される。

関連するファイル

- `stan.in` (mVMC/HPhi++ のマニュアル参照)

相関関数のフーリエ変換

ユーティリティプログラム `dynamicalr2k` を使って、相関関数をフーリエ変換する。

```
$ echo "4 20
G 0 0 0
X 0.5 0 0
M 0.5 0.5 0
G 0 0 0" >> geometry.dat
$ dynamicalr2k namelist.def geometry.dat
```

これにより、カレントディレクトリの `output/` 以下にフーリエ変換された相関関数が出力される。

関連するファイル

- `output/zvo_DynamicalGreen.dat`
- `geometry.dat` (サイトの位置と軌道のインデックス, k 点)
- `output/zvo_dyn.dat`

相関関数のプロット

`gnuplot` を使って、相関関数を k 空間でプロットする。

```
load "kpath.gp"
splot "output/zvo_dyn.dat" u 1:2:(-$4) w l
```

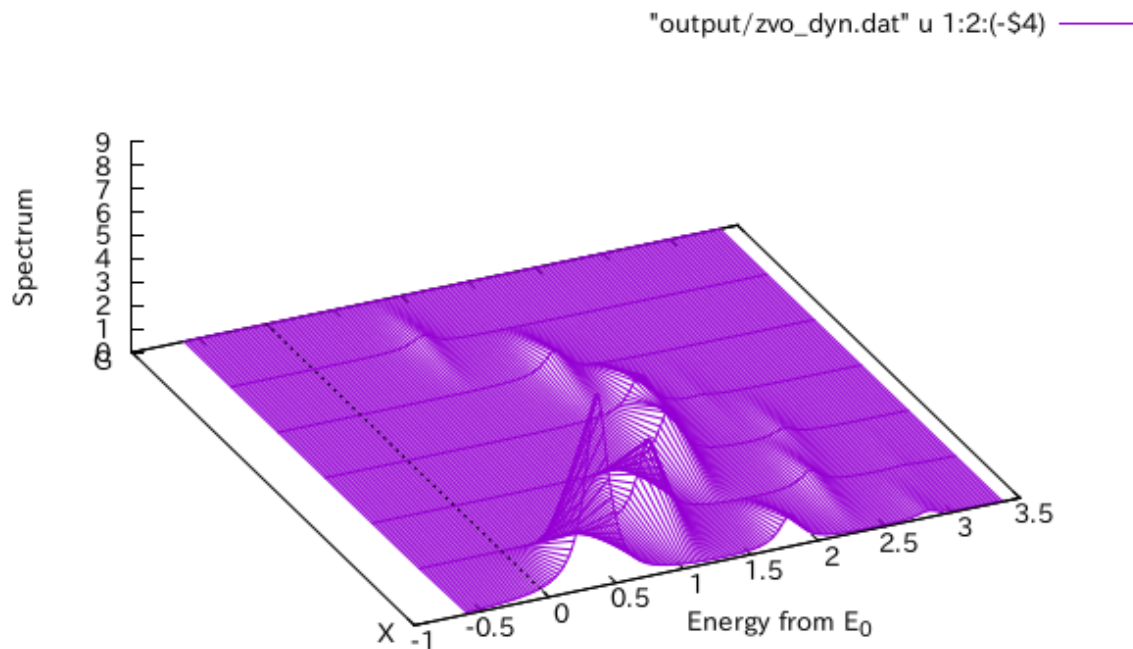


図 7 Figure 7: 相関関数 $\langle S_k \cdot S_k \rangle(\omega)$ の虚部 (4 列目) をプロットした図.

関連するファイル

- kpath.gp (*gnuplot* スクリプト)
- output/zvo_dyn.dat

.1.4 ファイルフォーマット

サイトの位置と軌道のインデックス, k 点

静的相関関数についてのチュートリアルでのファイル名は `geometry.dat`. 各サイトの位置と軌道の情報は mVMC/HPhi++ のスタンダードモードを用いた場合には自動的に生成される.

```
1.0000000000000000e+00    0.0000000000000000e+00    0.0000000000000000e+00 (1)
0.0000000000000000e+00    1.0000000000000000e+00    0.0000000000000000e+00 (1)
0.0000000000000000e+00    0.0000000000000000e+00    1.0000000000000000e+00 (1)
0.0000000000000000e+00    0.0000000000000000e+00    0.0000000000000000e+00 (2)
2 2 0          (3)
-2 2 0         (3)
```

```

0 0 1      (3)
0 0 0 0    (4)
-1 1 0 0   (4)
0 1 0 0    (4)
1 1 0 0    (4)
-1 2 0 0   (4)
0 2 0 0    (4)
1 2 0 0    (4)
0 3 0 0    (4)
4 20       (5)
G 0 0 0    (6)
X 0.5 0 0  (6)
M 0.5 0.5 0 (6)
G 0 0 0    (6)
16 16 1    (7)

```

1. 単位格子ベクトル. 任意の単位 (スタンダードモードで自動生成).
2. 1 体項がシミュレーションセルの境界を跨いだときに付く位相 (単位 degree) (スタンダードモードで自動生成)
3. シミュレーションセルの形状を指定する三本の整数ベクトル. スタンダードモードの入力パラメーター `a0W`, `a0L`, `a0H`, `a1W...` に対応する (スタンダードモードで自動生成).
4. 各サイトの座標結晶並進ベクトル (指数) および内部座標 (軌道) のインデックス (スタンダードモードで自動生成).
5. k バスのノード (対称性の高い点) の数と, ノード間の k 点の分割数.
6. k ノードのラベルとフラクショナル座標
7. 運動量分布関数の FermiSurfer ファイルを作成する時の k グリッド

サイト表示の 1 体および 2 体相関関数

計算する相関関数のインデックスの指定

mVMC/HPhi++ で計算する相関関数を指定する. スタンダードモードを使った場合には自動的に生成される. 総合的な説明は mVMC/HPhi++ のマニュアルを参照. [静的相関関数についてのチュートリアル](#) でのファイル名は `greenone.def` (1 体) および `greentwo.def` (2 体) である.

対応する量にある相関関数を計算するためには, 以下のようにインデックスを指定する必要がある.

- $\langle \hat{c}_{\mathbf{k}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{k}\beta\uparrow} \rangle$
 $\langle \hat{c}_{0\alpha\uparrow}^\dagger \hat{c}_{\mathbf{R}\beta\uparrow} \rangle$ に対して, \mathbf{R} が全ての単位胞, α および β がそれぞれ全ての軌道を網羅するようにする.
- $\langle \hat{c}_{\mathbf{k}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{k}\beta\downarrow} \rangle$

$\langle \hat{c}_{0\alpha\downarrow}^\dagger \hat{c}_{\mathbf{R}\beta\downarrow} \rangle$ に対して, \mathbf{R} が全ての単位胞, α および β がそれぞれ全ての軌道を網羅するようにする.

- $\langle \hat{\rho}_{\mathbf{k}\alpha} \hat{\rho}_{\mathbf{k}\beta} \rangle$ および $\langle \hat{S}_{\mathbf{k}\alpha}^z \hat{S}_{\mathbf{k}\beta}^z \rangle$

$\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{0\alpha\sigma} \hat{c}_{\mathbf{R}\beta\sigma'}^\dagger \hat{c}_{\mathbf{R}\beta\sigma'} \rangle$ に対して, \mathbf{R} が全ての単位胞, α および β がそれぞれ全ての軌道を網羅し, σ および σ' が \uparrow, \downarrow を網羅するようにする.

- $\langle \hat{S}_{\mathbf{k}\alpha}^+ \hat{S}_{\mathbf{k}\beta}^- \rangle$ および $\langle \hat{S}_{\mathbf{k}\alpha} \cdot \hat{S}_{\mathbf{k}\beta} \rangle$

HPhi++ の場合は $\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{0\alpha-\sigma} \hat{c}_{\mathbf{R}\beta-\sigma}^\dagger \hat{c}_{\mathbf{R}\beta\sigma} \rangle$ に対して, \mathbf{R} が全ての単位胞, α および β がそれぞれ全ての軌道を網羅し, σ が \uparrow, \downarrow を網羅するようにする. mVMC の場合は $\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{\mathbf{R}\beta\sigma} \hat{c}_{\mathbf{R}\beta-\sigma}^\dagger \hat{c}_{0\alpha-\sigma} \rangle$ に対して, \mathbf{R} が全ての単位胞, α および β がそれぞれ全ての軌道を網羅し, σ が \uparrow, \downarrow を網羅するようにする. いずれの場合も演算子の順番に注意にすること.

スタンダードモードのデフォルト (outputmode="corr") では, 自動的に上記のインデックスが指定されるため, 特に気にする必要はない.

サイト表示の 1 体および 2 体相関関数の計算結果

計算する相関関数のインデックスの指定 で指定したインデックスを持つ相関関数が mVMC/HPhi++ によって計算され, ファイルに出力される. 総合的な説明は mVMC/HPhi++ のマニュアルを参照. 静的相関関数についてのチュートリアル でのファイル名は output/zvo_cisajs_001.dat および output/zvo_cisajsktalt_001.dat (mVMC), output/zvo_cisajs.dat および output/zvo_cisajsktalt.dat (HPhi++).

greenr2k ユーティリティはこのファイルを読み込んで計算を行う. この時, (スタンダードモードを使わず自分でインデックスを指定するなどにより) 計算する相関関数のインデックスの指定 で挙げたインデックスの相関関数のなかで欠けているものがある場合, それを 0 として扱う.

k パス上での相関関数

Fourier 変換された相関関数 (波数表示) が入っている. ユーティリティ greenr2k によって生成される. 静的相関関数についてのチュートリアル でのファイル名は output/zvo_corr_eigen0.dat である.

```
# k-length[1]
# Orbital 1 to Orbital 1
# UpUp[ 2, 3] (Re. Im.) DownDown[ 4, 5]
# Density[ 6, 7] SzSz[ 8, 9] S+S-[ 10, 11] S.S[ 12, 13]
0.00000E+00 0.88211E+00 -0.50000E-09 0.88211E+00 0.40000E-09 ...
0.25000E-01 0.87976E+00 -0.46625E-09 0.87976E+00 0.42882E-09 ...
0.50000E-01 0.87276E+00 -0.42841E-09 0.87276E+00 0.45201E-09 ...
:
```

はじめに各カラムに出力されている量の説明がコメントとして書かれ, それに続いて k 点の距離とそれぞれの相関関数の実部と虚部が書かれている.

gnuplot スクリプト

greenr2k にて作成される. gnuplot でこれを読み込むことでグラフ中に k 点のラベルを表示する. ファイル名は kpath.gp である.

```
set xtics ('G'      0.00000, 'X'      0.50000, 'M'      1.00000, 'G'      1.70711)
set ylabel 'Correlation function'
set grid xtics lt 1 lc 0
```

運動量分布関数の等値面をプロットするための FermiSurfer ファイル

greenr2k にて作成される. ファイル名は output/zvo_corr_eigen0.dat.frmsf

.1.5 greenr2k ユーティリティの動作について

このユーティリティは、次のようにして使う.

```
$ ${PATH}/greenr2k ${NAMELIST} ${GEOMETRY}
```

ここで, $\${PATH}$ は greenr2k ユーティリティのバイナリのあるディレクトリのパス, $\${NAMELIST}$ は HPhi++/mVMC の NameList インプットファイル名, $\${GEOMETRY}$ は サイトの位置と軌道のインデックス, k 点ファイルへのパスである.

HPhi++ の各モード (Lanczos, TPQ, 全対角化, LOBCG) および mVMC のどの計算で得られた相関関数の Fourier 変換を行うかによって、動作が若干異なる. 以下では ModPara インプットファイルの CDataFileHead が "zvo" (デフォルト値) であるとする.

HPhi++-TPQ

この場合に HPhi++ は、各試行/TPQ ステップ毎に zvo_cisajs_run*step*.dat (1 体), zvo_cisajsckalt_run*step*.dat (2 体) というファイルを output/ ディレクトリに出力する. greenr2k ユーティリティは、各試行/TPQ ステップ毎に 1 体および 2 体の相関関数を読み込み Fourier 変換を行った後、zvo_corr_run*step*.dat という名前のファイルとして output/ ディレクトリに出力する.

HPhi++-全対角化および LOBCG

この場合に HPhi++ は、各波動関数ごとに zvo_cisajs_eigen*.dat (1 体), zvo_cisajsckalt_eigen*.dat (2 体) というファイルを output/ ディレクトリに出力する. greenr2k ユーティリティは、各波動関数ごとに 1 体および 2 体の相関関数を読み込み Fourier 変換を行った後、zvo_corr_eigen*.dat という名前のファイルとして output/ ディレクトリに出力する.

mVMC

この場合に `vmc.out` は, `ModPara` インプットファイルで指定された `NDataIdxStart` および `NDataQtySmp` というパラメーターに応じて試行を行いインデックスをつけられた `zvo_cisajs_???.dat` (1 体), `zvo_cisajscktalt_???.dat` (2 体) というファイルを `output/` ディレクトリに出力する. `greenr2k` ユーティリティはそれらのファイルを読み込み, 各試行に対して Fourier 変換を行った後, それらの実部, 虚部ごとに平均値

$$\langle A \rangle = \frac{1}{N_{\text{Try}}} \sum_{i=1}^{N_{\text{Try}}} A_i \quad (9)$$

および標準誤差

$$\delta A = \frac{1}{N_{\text{Try}} - 1} \sqrt{\frac{1}{N_{\text{Try}}} \sum_{i=1}^{N_{\text{Try}}} (A_i - \langle A \rangle)^2} \quad (10)$$

を計算し, 平均値と誤差を含んだ単一のファイル `zvo_corr_eigen*.dat` を `output/` ディレクトリに出力する.

.1.6 dynamicalr2k ユーティリティの動作について

このユーティリティは, 次のようにして使う.

```
$ ${PATH}/dynamicalr2k ${NAMELIST} ${GEOMETRY}
```

ここで, `${PATH}` は `dynamicalr2k` ユーティリティのバイナリのあるディレクトリのパス, `${NAMELIST}` は HPhi++/mVMC の NameList インプットファイル名, `${GEOMETRY}` は サイトの位置と軌道のインデックス, k 点ファイルへのパスである. この場合に HPhi++ が `output/` ディレクトリに出力するサイト表示の動的相関関数は, `zvo_DynamicalGreen.dat` である. `greenr2k` ユーティリティは, これらを読み込み Fourier 変換を行った後, 単一のファイル `zvo_corr.dat` を `output/` ディレクトリに出力する.

.1.7 Contact

このユーティリティについてのご意見, ご質問, バグ報告等ありましたら下記までお問い合わせください.

河村光晶

```
mkawamura_at_issp.u-tokyo.ac.jp
```

`_at_` を `@` に変えてください.

.2 Wannier 関数を用いたダウンフォールディング

.2.1 概要

本資料では, [RESPACK](#) と mVMC および HPhi++ を用いて, ダウンフォールディングをした格子モデルを計算する機能について説明する.

$$\begin{aligned}
 \mathcal{H} = & \sum_{i,j,\alpha,\beta,\sigma} t_{i\alpha j\beta} c_{i\alpha\sigma}^\dagger c_{j\beta\sigma} \\
 & + \sum_{i,\alpha} U_{i\alpha i\alpha} n_{i\alpha\uparrow} n_{i\alpha\downarrow} + \sum_{(i,\alpha) < (j,\beta)} U_{i\alpha j\beta} n_{i\alpha} n_{j\beta} - \sum_{(i,\alpha) < (j,\beta)} J_{i\alpha j\beta} (n_{i\alpha\uparrow} n_{j\beta\uparrow} + n_{i\alpha\downarrow} n_{j\beta\downarrow}) \\
 & + \sum_{(i,\alpha) < (j,\beta)} J_{i\alpha j\beta} (c_{i\alpha\uparrow}^\dagger c_{j\beta\downarrow}^\dagger c_{i\alpha\downarrow} c_{j\beta\uparrow} + c_{j\beta\uparrow}^\dagger c_{i\alpha\downarrow}^\dagger c_{j\beta\downarrow} c_{i\alpha\uparrow}) \\
 & + \sum_{(i,\alpha) < (j,\beta)} J_{i\alpha j\beta} (c_{i\alpha\uparrow}^\dagger c_{i\alpha\downarrow}^\dagger c_{j\beta\downarrow} c_{j\beta\uparrow} + c_{j\beta\uparrow}^\dagger c_{j\beta\downarrow}^\dagger c_{i\alpha\downarrow} c_{i\alpha\uparrow})
 \end{aligned}$$

要件

[QuantumESPRESSO](#) もしくは [xTAPP](#) を用いて Kohn-Sham 軌道を用いたのちに, [RESPACK](#) で Wannier 関数, 誘電関数, 有効相互作用を計算し, それらを用いて構成した格子モデルを mVMC もしくは HPhi++ で計算する. したがってそれらのプログラムが使用可能である必要がある.

.2.2 チュートリアル

このチュートリアルでは Sr_2VO_4 を 2 次元 3 軌道 Hubbard モデルにダウンフォールドして, それを HPhi++/mVMC で計算する. DFT 計算は [QuantumESPRESSO](#) で行う.

電荷密度の SCF 計算

まず, DFT による電荷密度の SCF 計算を行う.

scf.in

```

&CONTROL
  calculation = 'scf'
  pseudo_dir = '../pseudo/'
  prefix = 'sr2vo4'
/
&SYSTEM
  ibrav = 0
  nat = 7
  ntyp = 3
  ecutwfc = 45.000000
  ecutrho = 360.000000

```



```

occupations = 'tetrahedra_opt'
/
&ELECTRONS
mixing_beta = 0.3
/
CELL_PARAMETERS angstrom
-1.917800 1.917800 6.280100
1.917800 -1.917800 6.280100
1.917800 1.917800 -6.280100
ATOMIC_SPECIES
O 15.999400 O_pbe_v1.2.uspp.F.UPF
Sr 87.620000 Sr.pbe-spn-rrkjus_psl.1.0.0.UPF
V 50.941500 V_pbe_v1.uspp.F.UPF
ATOMIC_POSITIONS crystal
Sr 0.644600 0.644600 1.000000
Sr 0.355400 0.355400 0.000000
V 0.000000 0.000000 0.000000
O 0.500000 0.000000 0.500000
O 0.000000 0.500000 0.500000
O 0.842000 0.842000 1.000000
O 0.158000 0.158000 0.000000
K_POINTS automatic
4 4 4 0 0 0

```

擬ポテンシャル (UPF ファイル) は The SG15 Optimized Norm-Conserving Vanderbilt (ONCV) pseudopotentials のものを使う。

http://www.quantum-simulation.org/potentials/sg15_oncv/sg15_oncv_upf_2015-10-07.tar.gz

QuantumESPRESSO 内のプログラム `pw.x` を使う。

```
$ pw.x -in scf.in
```

(Optional) バンド計算と描画

`band.in`

```

&CONTROL
calculation = 'bands'
pseudo_dir = '../pseudo/'
prefix = 'sr2vo4'
/
&SYSTEM
ibrav = 0
nat = 7
ntyp = 3
ecutwfc = 45.000000

```

```

ecutrho = 360.000000
nbnd = 57

/
&ELECTRONS
/
CELL_PARAMETERS angstrom
-1.917800 1.917800 6.280100
1.917800 -1.917800 6.280100
1.917800 1.917800 -6.280100
ATOMIC_SPECIES
O 15.999400 O_pbe_v1.2.uspp.F.UPF
Sr 87.620000 Sr.pbe-spn-rrkjus_psl.1.0.0.UPF
V 50.941500 V_pbe_v1.uspp.F.UPF
ATOMIC_POSITIONS crystal
Sr 0.644600 0.644600 1.000000
Sr 0.355400 0.355400 0.000000
V 0.000000 0.000000 0.000000
O 0.500000 0.000000 0.500000
O 0.000000 0.500000 0.500000
O 0.842000 0.842000 1.000000
O 0.158000 0.158000 0.000000
K_POINTS crystal
49
0.5000000000 0.5000000000 -0.5000000000 1.0
0.4433287500 0.5566712500 -0.5000000000 1.0
0.3866575000 0.6133425000 -0.5000000000 1.0
0.3299862500 0.6700137500 -0.5000000000 1.0
0.2733150000 0.7266850000 -0.5000000000 1.0
0.2166437500 0.7833562500 -0.5000000000 1.0
0.1599725000 0.8400275000 -0.5000000000 1.0
0.1033012500 0.8966987500 -0.5000000000 1.0
0.0466300000 0.9533700000 -0.5000000000 1.0
0.0919660000 0.9080340000 -0.4546620000 1.0
0.1373020000 0.8626980000 -0.4093240000 1.0
0.1826380000 0.8173620000 -0.3639860000 1.0
0.2279740000 0.7720260000 -0.3186480000 1.0
0.2733100000 0.7266900000 -0.2733100000 1.0
0.3186480000 0.6813520000 -0.3186480000 1.0
0.3639860000 0.6360140000 -0.3639860000 1.0
0.4093240000 0.5906760000 -0.4093240000 1.0
0.4546620000 0.5453380000 -0.4546620000 1.0
0.5000000000 0.5000000000 -0.5000000000 1.0
0.3750000000 0.3750000000 -0.3750000000 1.0
0.2500000000 0.2500000000 -0.2500000000 1.0
0.1250000000 0.1250000000 -0.1250000000 1.0
0.0000000000 0.0000000000 0.0000000000 1.0
-0.0625000000 0.0625000000 0.0000000000 1.0
-0.1250000000 0.1250000000 0.0000000000 1.0
-0.1875000000 0.1875000000 0.0000000000 1.0

```

-0.2500000000	0.2500000000	0.0000000000	1.0
-0.3125000000	0.3125000000	0.0000000000	1.0
-0.3750000000	0.3750000000	0.0000000000	1.0
-0.4375000000	0.4375000000	0.0000000000	1.0
-0.5000000000	0.5000000000	0.0000000000	1.0
-0.5000000000	0.5000000000	0.0466300000	1.0
-0.4546620000	0.4546620000	0.0919660000	1.0
-0.4093240000	0.4093240000	0.1373020000	1.0
-0.3639860000	0.3639860000	0.1826380000	1.0
-0.3186480000	0.3186480000	0.2279740000	1.0
-0.2733100000	0.2733100000	0.2733100000	1.0
-0.2277583333	0.2277583333	0.2277583333	1.0
-0.1822066667	0.1822066667	0.1822066667	1.0
-0.1366550000	0.1366550000	0.1366550000	1.0
-0.0911033333	0.0911033333	0.0911033333	1.0
-0.0455516667	0.0455516667	0.0455516667	1.0
0.0000000000	0.0000000000	0.0000000000	1.0
0.0000000000	0.0833333333	0.0000000000	1.0
0.0000000000	0.1666666667	0.0000000000	1.0
0.0000000000	0.2500000000	0.0000000000	1.0
0.0000000000	0.3333333333	0.0000000000	1.0
0.0000000000	0.4166666667	0.0000000000	1.0
0.0000000000	0.5000000000	-0.0000000000	1.0

ここでも `pw.x` を使う.

```
$ pw.x -in band.in
```

`bands.in`

```
&BANDS
  prefix = 'sr2vo4'
  lsym = .false.
/
```

QuantumESPRESSO の `bands.x` を使う.

```
$ bands.x -in bands.in
```

出力された `bands.out.gnu` を GnuPlot などで読み込んでバンドを描く.

Kohn-Sham 軌道の計算

`nscf.in`

```
&CONTROL
  calculation = 'nscf'
  pseudo_dir = '../pseudo/'
  wf_collect = .true.
  prefix = 'sr2vo4'
/
&SYSTEM
  ibrav = 0
  nat = 7
  ntyp = 3
  ecutwfc = 45.000000
  ecutrho = 360.000000
  occupations = 'tetrahedra_opt'
  nbnd = 57
/
&ELECTRONS
/
CELL_PARAMETERS angstrom
-1.917800 1.917800 6.280100
1.917800 -1.917800 6.280100
1.917800 1.917800 -6.280100
ATOMIC_SPECIES
O 15.999400 O_pbe_v1.2.uspp.F.UPF
Sr 87.620000 Sr.pbe-spn-rrkjus_psl.1.0.0.UPF
V 50.941500 V_pbe_v1.uspp.F.UPF
ATOMIC_POSITIONS crystal
Sr 0.644600 0.644600 1.000000
Sr 0.355400 0.355400 0.000000
V 0.000000 0.000000 0.000000
O 0.500000 0.000000 0.500000
O 0.000000 0.500000 0.500000
O 0.842000 0.842000 1.000000
O 0.158000 0.158000 0.000000
K_POINTS automatic
4 4 4 0 0 0
```

ここでも `pw.x` を使う.

```
$ pw.x -in nscf.in
```

次に RESPACK に付属のユーティリティ `qe2respack.sh` を使う. 引数は `[prefix].save` ディレクトリ名.

```
$ qe2respack.sh sr2cuo3.save
```

Wannier 関数, 誘電関数, 有効相互作用の計算

`respack.in`

```

&PARAM_CHIQW
Num_freq_grid = 1
!Ecut_for_eps =
flg_cRPA = 1
MPI_num_proc_per_qcomm = 1
MPI_num_qcomm = 4
!flg_calc_type = 2
!n_calc_q = 8
/
&PARAM_WANNIER
N_wannier = 3
Lower_energy_window = 9.865
Upper_energy_window = 13.35
N_initial_guess = 3
/
dxy 0.2 0.0 0.0 0.0
dyz 0.2 0.0 0.0 0.0
dzx 0.2 0.0 0.0 0.0
&PARAM_INTERPOLATION
N_sym_points = 10
dense = 12, 12, 12
/
0.50000 0.50000 -0.50000
0.04663 0.95337 -0.50000
0.27331 0.72669 -0.27331
0.50000 0.50000 -0.50000
0.00000 0.00000 0.00000
-0.50000 0.50000 0.00000
-0.50000 0.50000 0.04663
-0.27331 0.27331 0.27331
0.00000 0.00000 0.00000
0.00000 0.50000 0.00000
&PARAM_VISUALIZATION
flg_vis_wannier = 1,
ix_vis_min = -1,
ix_vis_max = 2,
iy_vis_min = -1,
iy_vis_max = 2,
iz_vis_min = -1,
iz_vis_max = 2
/
&PARAM_CALC_INT
calc_ifreq = 1
ix_intJ_min = 0
ix_intJ_max = 0
iy_intJ_min = 0
iy_intJ_max = 0
iz_intJ_min = 0
iz_intJ_max = 0

```

```
/
```

RESPACK のプログラム `calc_wannier`, `calc_chiqw`, `calc_j3d`, `calc_w3d` を使う.

```
$ calc_wannier < respack.in
$ calc_chiqw < respack.in
$ calc_w3d < respack.in
$ calc_j3d < respack.in
```

HPhi++/mVMC によるモデル計算

まず, REPACK によって出力されたホッピング等のファイルを Wannier90 の形式に変換する. これには HPhi++/mVMC に付属のユーティリティー `respack2wan90.py` を使う. 引数は HPhi++/mVMC のスタンダードモードのパラメーター `CDataFileHead` と同じにする. 引数を指定しない場合は `zvo` (HPhi++/mVMC の `CDataFileHead` のデフォルト) が指定されたものとする.

```
$ respack2wan90.py zvo
```

これで, HPhi++/mVMC で実行する準備が出来たので, スタンダードモードを用いて計算する.

`respack.in`

```
model = "Hubbard"
lattice = "wannier90"
a0w = 2
a0l = 0
a0h = 2
a1w = 0
a1l = 2
a1h = 2
a2w = 1
a2l = 0
a2h = 0
method = "CG"
2Sz = 0
nelec = 4
exct = 1
cutoff_t = 0.2
cutoff_u = 0.4
cutoff_j = 0.1
```

```
$ vmc.out -s stan.in
```

.2.3 スタンドモードの入力パラメーター

以下に入力ファイルの例を示す.

respack.in

```
model = "Hubbard"
lattice = "wannier90"
a0w = 2
a0l = 0
a0h = 2
a1w = 0
a1l = 2
a1h = 2
a2w = 1
a2l = 0
a2h = 0
method = "CG"
2Sz = 0
nelec = 4
exct = 1
cutoff_t = 0.2
cutoff_u = 0.4
cutoff_j = 0.1
```

Wannier 関数を用いたダウンフォールディングに特有のパラメーター設定は次の通りである.

- lattice = "wannier90"
- cutoff_t, cutoff_u, cutoff_j

形式: 実数

デフォルト値: $1.0e-8$

ホッピング, Coulomb 積分, 交換積分に対して, これより小さい値を無視する.

- cutoff_tW, cutoff_tL, cutoff_tH
- cutoff_UW, cutoff_UL, cutoff_UH
- cutoff_JW, cutoff_JL, cutoff_JH

形式: 整数

デフォルト値: すべてのレンジの項を含む

ホッピング, Coulomb 積分, 交換積分に対して, これらの値を越える並進ベクトル \mathbf{R} を持つものを無視するようにする.

- cutoff_length_t, cutoff_length_U, cutoff_length_J

形式 : 実数

デフォルト値 : -1.0 (すべてのレンジの項を含む)

ホッピング, Coulomb 積分, 交換積分に対して, この距離を超えるものを無視する. 距離はワニエ関数の中心座標と単位格子ベクトルから求める.

- W, L, Height
- a0W, a0L, a0H, a1W, a1L, a1H, a2W, a2L, a2H
- Wsub, Lsub, Hsub
- a0Wsub, a0Lsub, a0Hsub, a1Wsub, a1Lsub, a1Hsub, a2Wsub, a2Lsub, a2Hsub

このように 3 番目の次元が現れる.

.2.4 ファイルフォーマット

HPhi++/mVMC のスタンダードモードでは以下のファイルを読み込む. これらのファイルは RESPACK のプログラムを実行したディレクトリで, HPhi++/mVMC に付属のユーティリティ `respack2wan90.py` を使うと得られる.

ジオメトリー

ファイル名は `[CDataFileHeat]_geom.dat`. このファイルを必要に応じて修正することで取り扱う軌道数を変更できる.

```
-1.917800 1.917800 6.280100
1.917800 -1.917800 6.280100
1.917800 1.917800 -6.280100
3
0.000000 -0.000000 -0.000000
-0.000000 -0.000000 -0.000000
0.000000 0.000000 0.000000
```

- 1 - 3 行目

結晶並進ベクトル. デカルト座標系で任意単位.

- 4 行目

mVMC/HPhi++ で取り入れる, ユニットセルあたりの軌道数. `respack2wan90.py` によって生成すると, Wannier 関数の数と同じになる. Wannier 関数の数よりも少なくすることも可能であり, その場合には先頭からこの数の分だけの軌道を取り入れたモデルが構築される.

- 5 行目以降

各軌道の Wannier 中心 (フラクショナル座標). Fourier 変換ツールで使われる.

ホッピング, **Coulomb** 積分, 交換積分

ファイル名は [CDataFileHeat]_hr.dat, [CDataFileHeat]_ur.dat, [CDataFileHeat]_jr.dat.
Wannier90 のホッピング積分の形式に従う.

.2.5 Contact

このユーティリティについてのご意見, ご質問, バグ報告等ありましたら下記までお問い合わせください.

河村光晶

`mkawamura_at_issp.u-tokyo.ac.jp`

`_at_` を `@` に変えてください.

.3 謝辞

HPhi++ の開発に当たり、西森秀稔教授が開発された先駆的な数値対角化パッケージ TITPACK の実装を参考にしました。特にランチョス法の部分は TITPACK の fortran コードを C 言語に移植したものがもとになっています。また、HPhi++ のエキスパートモードにおけるユーザー・インターフェースの設計の際、田原大資氏が開発された mVMC の柔軟なユーザー・インターフェースが礎となっています。インタフェースに関するコードの一部分は mVMC のものを使用しています。この場を借りて、お二人に感謝します。本山裕一博士と加藤康之博士には、バグを見つけて報告していただいたことに感謝します。星健夫准教授と曽我部知広准教授には、Shifted Krylov 法についてのレクチャーや御相談に対して、またライブラリ $K\omega$ の開発にご協力いただいたことに感謝します。

HPhi++ ver. 0.1 - ver. 3.1.0 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 (2015 年度, 2016 年度, 2017 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。