



# **HΦ Documentation**

***Release 3.5.1***

**University of Tokyo**

**Jun 14, 2022**



# CONTENTS

<b>1</b>	<b>What is <math>\mathcal{H}\Phi</math>?</b>	<b>1</b>
1.1	What is $\mathcal{H}\Phi$ ?	1
1.1.1	License	2
1.1.2	Copyright	2
1.1.3	Contributors	2
1.2	Operating environment	4
<b>2</b>	<b>How to use <math>\mathcal{H}\Phi</math></b>	<b>5</b>
2.1	Prerequisite	5
2.2	Installation	5
2.3	Directory structure	6
2.4	Basic usage	7
2.4.1	<i>Standard</i> mode	7
2.4.2	<i>Expert</i> mode	8
2.4.3	Creating input files for <i>Expert</i> mode	9
2.4.4	Setting the process number for MPI/hybrid parallelization	9
2.4.5	Printing version ID	10
<b>3</b>	<b>Tutorial</b>	<b>11</b>
3.1	Quick guide to <i>Standard</i> mode	11
3.1.1	Heisenberg model	11
3.1.2	Other tutorials	17
3.2	Quick guide to <i>Expert</i> mode	17
3.2.1	File list for input files	18
3.2.2	Files for basic parameters	18
3.2.3	Files for constructing Hamiltonian	20
3.2.4	Setting output components	21
3.2.5	Running	22
3.3	Fourier transformation of correlation functions	22
<b>4</b>	<b>File specification</b>	<b>23</b>
4.1	Input files for <i>Standard</i> mode	23
4.1.1	Parameters for the type of calculation	24
4.1.2	Parameters for the lattice	26
4.1.3	Parameters for conserved quantities	32
4.1.4	Parameters for the Hamiltonian	33
4.1.5	Parameters for the numerical condition	38
4.1.6	Parameters for the dynamical Green's function	41
4.1.7	Parameters for time-evolution	42
4.2	Input files for <i>Expert</i> mode	43

4.2.1	List file for the input files . . . . .	44
4.2.2	CalcMod file . . . . .	46
4.2.3	ModPara file . . . . .	50
4.2.4	LocSpin file . . . . .	55
4.2.5	Trans file . . . . .	57
4.2.6	InterAll file . . . . .	59
4.2.7	CoulombIntra file . . . . .	61
4.2.8	CoulombInter file . . . . .	63
4.2.9	Hund file . . . . .	65
4.2.10	PairHop file . . . . .	67
4.2.11	Exchange file . . . . .	69
4.2.12	Ising file . . . . .	71
4.2.13	PairLift file . . . . .	73
4.2.14	OneBodyG file . . . . .	75
4.2.15	TwoBodyG file . . . . .	77
4.2.16	SingleExcitation file . . . . .	79
4.2.17	PairExcitation file . . . . .	81
4.2.18	SpectrumVec File . . . . .	83
4.2.19	OneBodyTE File . . . . .	84
4.2.20	TwoBodyTE File . . . . .	86
4.3	Output files . . . . .	88
4.3.1	CHECK_Chemi.dat . . . . .	88
4.3.2	CHECK_InterAll.dat . . . . .	89
4.3.3	CHECK_CoulombIntra.dat . . . . .	90
4.3.4	CHECK_Hund.dat . . . . .	91
4.3.5	CHECK_INTER_U.dat . . . . .	92
4.3.6	CHECK_Memory.dat . . . . .	93
4.3.7	WarningOnTransfer.dat . . . . .	94
4.3.8	CalcTimer.dat . . . . .	95
4.3.9	TimeKeeper.dat . . . . .	96
4.3.10	sz_TimeKeeper.dat . . . . .	97
4.3.11	Time_CG_EigenVector.dat . . . . .	98
4.3.12	energy.dat . . . . .	99
4.3.13	Lanczos_Step.dat . . . . .	100
4.3.14	Time_TPQ_Step.dat . . . . .	102
4.3.15	Norm_rand.dat . . . . .	103
4.3.16	SS_rand.dat . . . . .	104
4.3.17	Flct_rand.dat . . . . .	106
4.3.18	Time_TE_Step.dat . . . . .	108
4.3.19	Norm.dat . . . . .	109
4.3.20	SS.dat . . . . .	110
4.3.21	Flct.dat . . . . .	112
4.3.22	Eigenvalue.dat . . . . .	114
4.3.23	phys.dat . . . . .	115
4.3.24	ham.dat . . . . .	116
4.3.25	cisajs.dat . . . . .	118
4.3.26	cisajscalt.dat . . . . .	120
4.3.27	eigenvec.dat . . . . .	122
4.3.28	tmpvec.dat . . . . .	123
4.3.29	DynamicalGreen.dat . . . . .	124
4.3.30	recalcvec.dat . . . . .	125
4.3.31	TMcomponents.dat . . . . .	126
4.3.32	eigenvec.dat . . . . .	127
4.3.33	residual.dat . . . . .	128

4.4	Error messages . . . . .	129
<b>5</b>	<b>Algorithm</b>	<b>131</b>
5.1	Lanczos method . . . . .	131
5.1.1	Details of Lanczos method . . . . .	131
5.1.2	Inverse iteration method . . . . .	132
5.1.3	Details of implementation . . . . .	132
5.2	Full Diagonalization method . . . . .	133
5.2.1	Overview . . . . .	133
5.2.2	Finite-temperature calculations . . . . .	133
5.3	Finite-temperature calculations by the TPQ method . . . . .	134
5.3.1	Details of implementation . . . . .	134
5.4	Dynamical Green's function . . . . .	135
5.5	Real time evolution method . . . . .	135
5.6	Bogoliubov representation . . . . .	136
<b>6</b>	<b>Acknowledgement</b>	<b>137</b>
<b>A</b>	<b>HPhi/mVMC Fourier-Transformation utility</b>	<b>139</b>
A.1	Overview . . . . .	139
A.1.1	Prerequisite . . . . .	139
A.1.2	Supported quantities . . . . .	139
A.2	Tutorial . . . . .	140
A.2.1	Run HPhi/vmc.out . . . . .	140
A.2.2	Fourier transformation of correlation functions . . . . .	141
A.2.3	Display correlation functions . . . . .	141
A.3	File format . . . . .	141
A.3.1	Geometry . . . . .	141
A.3.2	One- and Two-body correlation function in the site representation . . . . .	143
A.3.3	Correlation functions on the $k$ path . . . . .	144
A.3.4	gnuplot script . . . . .	144
A.3.5	FermiSurfer file to display the isosurface of the momentum distribution . . . . .	145
A.4	Behavior of greenr2k utility . . . . .	145
A.4.1	HPhi-Lanczos . . . . .	145
A.4.2	HPhi-TPQ . . . . .	145
A.4.3	HPhi-Full diagonalization and LOBCG . . . . .	145
A.4.4	mVMC . . . . .	145
A.5	Contact . . . . .	146
<b>B</b>	<b>Downfolding with Wannier functions</b>	<b>147</b>
B.1	Overview . . . . .	147
B.1.1	Prerequisite . . . . .	148
B.2	Tutorial . . . . .	148
B.2.1	SCF calculation of charge density . . . . .	148
B.2.2	(Optional) Band structure . . . . .	149
B.2.3	Kohn-Sham orbitals for Wannier . . . . .	151
B.2.4	Wannier function, dielectric function, effective interaction . . . . .	152
B.2.5	Quantum lattice mode for HPhi/mVMC . . . . .	153
B.3	Input parameters for Standard mode . . . . .	154
B.4	File format . . . . .	157
B.4.1	Geometry . . . . .	157
B.4.2	Hopping, Coulomb, exchange integrals, charge densities . . . . .	157
B.5	Contact . . . . .	158



WHAT IS  $\mathcal{H}\Phi$ ?1.1 What is  $\mathcal{H}\Phi$ ?

Comparison between experimental observation and theoretical analysis is a crucial step in condensed-matter physics research. The temperature dependence of specific heat and magnetic susceptibility, for example, has been studied to extract the nature of low energy excitations of and magnetic interactions between electrons, respectively, through comparison with theories such as Landau's Fermi liquid theory and the Curie-Weiss law.

For the flexible and quantitative comparison of theoretical and experimental data, the exact diagonalization approach<sup>1</sup> is one of the most reliable numerical tools that requires no approximation or inspiration of genius. For the last few decades, a numerical diagonalization package for quantum spin Hamiltonians, TITPACK, developed by Prof. Hidetoshi Nishimori of Tokyo Institute of Technology, has been widely used in the condensed-matter physics community. Nevertheless, limited computational resources have hindered the ability of non-expert users to apply the package to quantum systems with a large number of electrons or spins.

In contrast, the recent and rapid development of a parallel computing infrastructure has opened up new avenues for user-friendly larger scale diagonalizations up to 18-site Hubbard clusters or  $36 S = 1/2$  quantum spins. In addition, recent advances in quantum statistical mechanics<sup>2345</sup> allow the finite temperature properties of quantum many-body systems to be calculated at computational costs similar to those of the calculations of ground state properties, which also allows theoretical results for the temperature dependence of, for example, specific heat and magnetic susceptibility, to be compared with experimental results quantitatively<sup>6</sup>. To utilize the parallel computing infrastructure with narrow bandwidth and distributed-memory architectures, efficient, user-friendly, and highly parallelized diagonalization packages are highly desirable.

$\mathcal{H}\Phi$ , a flexible diagonalization package for solving quantum lattice Hamiltonians, has been developed as a descendant of the pioneering package TITPACK. The Lanczos method for calculations of the ground state and a few excited states properties, as well as finite temperature calculations based on thermal pure quantum states<sup>7</sup>, are implemented in the  $\mathcal{H}\Phi$  package, with an easy-to-use and flexible user interface. By using  $\mathcal{H}\Phi$ , you can analyze a wide range of quantum lattice Hamiltonians including simple Hubbard and Heisenberg models, multi-band extensions of the Hubbard model, exchange couplings that break the SU(2) symmetry of quantum spins, such as Dzyaloshinskii-Moriya and Kitaev interactions, and Kondo lattice models describing itinerant electrons coupled with quantum spins.  $\mathcal{H}\Phi$  calculates a variety of physical quantities, such as internal energy at zero temperature or finite temperatures, temperature dependence of specific heat, and charge/spin structure factors. A broad spectrum of users including experimental scientists is cordially welcome.

<sup>1</sup> E. Dagotto, Rev. Mod. Phys. **66**, 763-840 (1994).

<sup>2</sup> M. Imada, M. Takahashi, Journal of the Physical Society of Japan **55**, 3354-3361 (1986).

<sup>3</sup> J. Jaklić, P. Prelovšek, Phys. Rev. B **49**, 5065-5068 (1994).

<sup>4</sup> A. Hams, H. De Raedt, Phys. Rev. E **62**, 4365-4377 (2000).

<sup>5</sup> S. Sugiura, A. Shimizu, Phys. Rev. Lett. **108**, 240401 (2012).

<sup>6</sup> Y. Yamaji, Y. Nomura, M. Kurita, R. Arita, M. Imada, Phys. Rev. Lett. **113**, 107201 (2014).

### 1.1.1 License

The distribution of the program package and the source codes for  $\mathcal{H}\Phi$  follow GNU General Public License version 3 (GPL v3) or later. We hope that you cite the reference, [Comp. Phys. Commun. 217 \(2017\) 180-192](#), when you publish the results using  $\mathcal{H}\Phi$  (hphi).

### 1.1.2 Copyright

© 2015- The University of Tokyo. All rights reserved. This software was developed with the support of "Project for advancement of software usability in materials science" of The Institute for Solid State Physics, The University of Tokyo.

### 1.1.3 Contributors

This software was developed by the following contributors.

- ver.3.5.1 (released on 2022/6/14)
- ver.3.5 (released on 2021/9/29)
- ver.3.4 (released on 2020/6/2)
  - Developers
    - \* Kota Ido  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Mitsuaki Kawamura  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Yuichi Motoyama  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Kazuyoshi Yoshimi  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Takahiro Misawa  
(Waseda Research Institute for Science and Engineering (RISE), Waseda University)
    - \* Youhei Yamaji  
(Department of Applied Physics, The University of Tokyo)
    - \* Synge Todo  
(Department of Physics, The University of Tokyo)
    - \* Yusuke Konishi  
(Academeia Co., Ltd.)
  - Project coordinator
    - \* Naoki Kawashima  
(The Institute for Solid State Physics, The University of Tokyo)
- ver.3.3 (released on 2019/7/19)
- ver.3.2 (released on 2019/4/27)
- ver.3.1 (released on 2018/9/3)
- ver.3.0 (released on 2017/12/22)



- Developers
  - \* Takahiro Misawa  
(The Institute for Solid State Physics, The University of Tokyo)
  - \* Kazuyoshi Yoshimi  
(The Institute for Solid State Physics, The University of Tokyo)
  - \* Mitsuaki Kawamura  
(The Institute for Solid State Physics, The University of Tokyo)
  - \* Kota Ido  
(Department of Applied Physics, The University of Tokyo)
  - \* Youhei Yamaji  
(Department of Applied Physics, The University of Tokyo)
  - \* Synge Todo  
(Department of Physics, The University of Tokyo)
- Project coordinator
  - \* Naoki Kawashima  
(The Institute for Solid State Physics, The University of Tokyo)
- ver.2.0 (released on 2017/4/11)
- ver.1.2 (released on 2016/11/14)
- ver.1.1 (released on 2016/5/13)
- ver.1.0 (released on 2016/4/5)
  - Developers
    - \* Takahiro Misawa  
(Department of Applied Physics, The University of Tokyo)
    - \* Kazuyoshi Yoshimi  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Mitsuaki Kawamura  
(The Institute for Solid State Physics, The University of Tokyo)
    - \* Youhei Yamaji  
(Department of Applied Physics, The University of Tokyo)
    - \* Synge Todo  
(Department of Physics, The University of Tokyo)
  - Project coordinator
    - \* Naoki Kawashima  
(The Institute for Solid State Physics, The University of Tokyo)

## 1.2 Operating environment

H $\Phi$  was tested on the following platforms

- The supercomputer system-B "ohtaka" in ISSP
- Linux PC + Intel compiler
- Linux PC + GCC.
- Mac + GCC.

## HOW TO USE $\mathcal{H}\Phi$

### 2.1 Prerequisite

$\mathcal{H}\Phi$  requires the following packages:

- C/fortran compiler (Intel, Fujitsu, GNU, etc. )
- BLAS/LAPACK library (Intel MKL, Fujitsu, ATLAS, etc.)
- MPI library (if you do not use MPI, this is not required).

---

**Tip:**

**E.g. /Settings of Intel compiler**

When you use the Intel compiler, you can easily use the scripts attached to the compiler.

In the case of the bash in the 64-bit OS, write the following in your `~/ .bashrc`:

```
source /opt/intel/bin/compilervars.sh intel64
or
source /opt/intel/bin/iccvars.sh intel64
source /opt/intel/mkl/bin/mklvars.sh
```

Please read the manuals of your compiler/library for more information.

---

### 2.2 Installation

You can download  $\mathcal{H}\Phi$  at the following location.

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

You can obtain the  $\mathcal{H}\Phi$  directory by typing

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

$\mathcal{H}\Phi$  can be installed by using cmake. We can compile  $\mathcal{H}\Phi$  as:

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

To use ScaLAPACK library for full diagonalization, the cmake option `-DUSE_SCALAPACK=ON` is needed. Here, we set a path to  $\mathcal{H}\Phi$  as `$PathTohphi` and to a build directory as `$HOME/build/hphi`. After compilation, `src` folder is constructed below a `$HOME/build/hphi` folder and we obtain an executable `HPhi` in `src/` directory. When no MPI library exists in the system, an executable `HPhi` is automatically compiled without an MPI library.

In the above example, we compile  $\mathcal{H}\Phi$  by using a gcc compiler. We can select a compiler by using the following options:

- `sekirei` : ISSP system-B "sekirei"
- `sekirei_acc`: ISSP system-B "sekirei" (for using MAGMA library)
- `fujitsu` : Fujitsu compiler
- `intel` : Intel compiler + Linux PC
- `gcc` : GCC compiler + Linux PC.

An example of compiling  $\mathcal{H}\Phi$  by using the Intel compiler is shown as follows:

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

After compilation, `src` folder is created below the `build` folder and an execute  $\mathcal{H}\Phi$  in the `src` folder. Please note that we must delete the `build` folder and repeat the above operations when we change the compiler.

---

**Tip:**

CMake automatically tries to download StdFace package (<https://github.com/issp-center-dev/StdFace>, parser for standard mode),

but this may fail due to network problem (e.g., IP unreachable)

In such a case, please run `sh src/StdFace/download.sh` to download manually.

If you want to use StdFace downloaded into another directory,  
pass `-DSTDFACE_DIR=<path_to_stdface>` to cmake command.

---

## 2.3 Directory structure

When `HPhi-xxx.tar.gz` is unzipped, the following directory structure is composed.

```
|--CMakeLists.txt
|--COPYING
|--config/
|   |--fujitsu.cmake
|   |--gcc.cmake
|   |--intel.cmake
|   |--sekirei.cmake
|   |--sekirei_acc.cmake
|--doc/
|   |--en/
|   |--ja/
|   |--tutorial/
```

(continues on next page)

(continued from previous page)

```
|      |--tutorial_HPhi_en.pdf
|      |--userguide_en.pdf
|      |--userguide_jp.pdf
|--README.md
|--samples/
|--src/
|      |--*.c
|      |--CMakeLists.txt
|      |--include/*.h
|      |--makefile_src
|      |--StdFace/
|--test/
|--tool/
```

## 2.4 Basic usage

H $\Phi$  has two modes: Standard mode and Expert mode. Here, the basic flows of calculations of the Standard and expert modes are shown.

### 2.4.1 *Standard mode*

The procedure of calculation through the standard mode is as follows:

1. Create a directory for a calculation scenario

First, you create a working directory for the calculation.

2. Create input files for Standard mode

In Standard mode, you can choose a model (the Heisenberg model, Hubbard model, etc.) and a lattice (the square lattice, triangular lattice, etc.) from those provided; you can specify some parameters (such as the first/second nearest neighbor hopping integrals and the on-site Coulomb integral) for them. Finally, you have to specify the numerical method (such as the Lanczos method) employed in this calculation. The input file format is described in [How to use HPhi](#).

3. Run

Run an executable HPhi in the terminal by setting option "-s" (or "--standard") and the name of the input file written in the previous step.

- Serial/OpenMP parallel

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

- MPI parallel/ Hybrid parallel

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

When you use a queuing system in workstations or super computers, sometimes the number of processes is specified as an argument for the job-submitting command. If you need more information, please refer to your system manuals. The number of processes depends on the target system of the models. The details of setting the number of processes are shown in [Creating input files for Expert mode](#).

4. Watch calculation logs

Log files are outputted in the "output" folder, which is automatically created in the directory for a calculation scenario. The details of the output files are shown in *Output files*.

#### 5. Results

If the calculation is completed normally, the result files are outputted in the "output" folder. The details of the output files are shown in *Output files*

---

**Tip:****The number of threads for OpenMP**

If you specify the number of OpenMP threads for  $\mathcal{H}\Phi$ , you should set it as follows (in the case of 16 threads) before running:

```
export OMP_NUM_THREADS=16
```

---

## 2.4.2 Expert mode

The calculation procedure for Expert mode is as follows.

1. Create a directory for a calculation scenario

First, you create a directory and give it the name of a calculation scenario (you can attach an arbitrary name to a directory).

2. Create input files for Expert mode

For Expert mode, you should create input files for constructing Hamiltonian operators, calculation conditions, and a list file for the filenames of the input files (see the file formats shown in *Input files for Expert mode*).

---

**Note:**

A list file can be easily created by using Standard mode.

---

3. Run

Run  $\mathcal{H}\Phi$  in the terminal by setting option "-e" (or "--expert") and the file name of a list file.

- Serial/OpenMP

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

- MPI/Hybrid

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

A number of processes depend on a target of system for models. The details of setting a number of processes are shown in *Creating input files for Expert mode*.

4. While running

Log files are outputted in the "output" folder which is automatically created in the directory for a calculation scenario. The details of the output files are shown in *Output files*.

5. Results

If the calculation is finished normally, the result files are outputted in the "output" folder. The details of the output files are shown in *Output files*.

### 2.4.3 Creating input files for *Expert* mode

This mode is for creating input files for *Expert* mode. A set of input files created using this mode gives a model provided in *Standard* mode. The usage is shown as follows.

1. Create an input file for *Standard* mode.
2. Setting an option "-sdry" and an input file (in this example, StdFace.def), run  $\mathcal{H}\Phi$ .

```
$ Path/HPhi -sdry StdFace.def
```

In this case, you should not use MPI parallelization (mpirun, mpiexec, etc.).

3. The following files are created as the input files for *Expert* mode in the current working directory.

```
calcm0d.def  greentwo.def  namelist.def  zTrans.def
greenone.def modpara.def  zInterAll.def zlocspn.def
```

### 2.4.4 Setting the process number for MPI/hybrid parallelization

For using MPI/hybrid parallelization, the process number must be set as follows.

1. Standard mode

- Hubbard/Kondo model

When model in the input file for Standard mode is set as "Fermion Hubbard", "Kondo Lattice", or "Fermion HubbardGC", the process number must be equal to  $4^n$ .

- Spin model

When model in the input file for Standard mode is set as "Spin" or "SpinGC", the process number must be equal to  $(2S + 1)^n$ , where  $2S$  is set in the input file (the default value is 1).

2. Expert mode

- Hubbard/Kondo model

When the model is selected as the Fermion Hubbard model or Kondo model by setting CalcModel in a **CalcMod** file, the process number must be equal to  $4^n$ . See [CalcMod file](#) for details of the CalcModel file.

- Spin model

When the model is selected as the spin model by setting CalcModel in a **CalcMod** file, the process number is fixed by a **LocSpin** file. The process number must be equal to the number calculated by multiplying the state number of the localized spin ( $2S + 1$ ) in descending order by the site number. See [LocSpin file](#) for details of the **LocSpin** file.

For example, when a **LocSpin** file is given as follows, the process number must be equal to  $2 = 1 + 1$ ,  $6 = 2 \times (2 + 1)$ ,  $24 = 6 \times (3 + 1)$ .

```
=====
NlocalSpin      3
=====
=====i_0IteElc_2S =====
```

(continues on next page)

(continued from previous page)

=====	
0	3
1	2
2	1

### 2.4.5 Printing version ID

By using the `-v` option as follows, you can check which version of  $\mathcal{H}\Phi$  you are using.

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



## 3.1 Quick guide to *Standard* mode

### 3.1.1 Heisenberg model

This tutorial should be performed in

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

The input file is provided as follows:

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

In this case, we treat the two-dimensional antiferromagnetic Heisenberg model that has a nearest neighbor spin coupling.

$$\mathcal{H} = J \sum_{i,j=1}^4 (\mathbf{S}_{ij} \cdot \mathbf{S}_{i+1j} + \mathbf{S}_{ij} \cdot \mathbf{S}_{ij+1}), \quad (3.1)$$

where we use the periodic boundary condition ( $\mathbf{S}_{15} = \mathbf{S}_{51} = \mathbf{S}_{11}$ ).

The input file is as follows:

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

In this tutorial, J and the number of sites are set to 1 (arbitrary unit) and 16, respectively.

## Log output

Log messages are outputted to the standard output. Log files for the calculation procedure are created in the "output" directory which is automatically created. In this example, the following files are outputted.

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	

The details of the outputted files are shown in *CHECK\_Chemi.dat* etc. We execute

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

and obtain the following standard outputs (the compilation mode is MPI parallel/hybrid parallel)

```

,anmmmmmmmmmmmmmb,,      Welcome to the
,@` dm      mb ==m
,@` d@@@@@@@@@@@@@@@@@b Pm,  @@      @@      @@
d@ d@@@ @@@ @@@@@@ @@@@b ~@a  @@      @@      @@@@@@@@
d@ @@@@ ^^^ @@@@ m m @@@  @,  @@      @@      @@@ @@@
@ @@@@ @@@ @@@@mm mm@@@  @|  @mmmmmmmmmm@@ @@@ @@@ @@@
P@ 9@@@@@@@@@@@@@@@@@P  @~ @@@@@@@@@@@@@@@@@ @@@ @@@ @@@
@@ ~9@@@@@@@@PPP~  @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 #####

```

(continues on next page)

(continued from previous page)

```

      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.

```

(continues on next page)

(continued from previous page)

```

@ 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 #####

```

(continues on next page)

(continued from previous page)

## 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

(continues on next page)

(continued from previous page)

```

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

```

In the beginning of this run, files describing the details of the considered Hamiltonian (`locspin.def`, `trans.def`, `exchange.def`, `coulombintra.def`, `hund.def`, `namelist.def`, `calcmmod.def`, `modpara.def`) and files specifying the elements of the correlation functions that will be calculated (`greenone.def`, `greentwo.def`) are generated.

## Outputs for calculation results

### Locally Optimal Block Conjugate Gradient (LOBCG) method

When a calculation by the LOBCG method is finished normally, eigenenergies, one-body Green's functions, and two-body Green's functions are calculated and outputted to the files, respectively. In this sample, the following files are outputted.

```

zvo_energy.dat
zvo_cisajsckalt_eigen_xx.dat  zvo_phys_Nup4_Ndown4.dat

```

where `xx` is the number of the eigenstate counting from 0.

### Lanczos method

When a calculation by the Lanczos method is completed normally, eigenenergies, one-body Green's functions, and two-body Green's functions are calculated and outputted to the files, respectively. In this sample, the following files are outputted.

```

zvo_energy.dat zvo_cisajs.dat
zvo_cisajsckalt.dat

```

For Standard mode, all pairs of  $\langle n_{i\sigma} \rangle$  are calculated as one-body Green's functions and those of  $\langle n_{i\sigma} n_{j\sigma'} \rangle$  are calculated as two-body Green's functions on the basis of the definition files, `greenone.def` and `greentwo.def`. When the accuracy of the Lanczos vectors is sufficient, one-body and two-body Green's functions are calculated by the eigenvectors obtained by the Lanczos method. When the accuracy of the Lanczos vectors is *not* sufficient, a message "Accuracy of Lanczos vector is not enough" is outputted to the standard output and the one-body and two-body Green's functions are calculated by the eigenvectors obtained by CG method. The details of output files are shown in *energy.dat*, *cisajs.dat*, *cisajsckalt.dat*.

## TPQ method

When `method="TPQ"` is selected in an input file, a calculation by the TPQ method is started. After the calculation is completed normally, the following files are outputted, where `%%` is the number of runs and `&&` is the number of steps for the TPQ method.

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

In `Norm_rand%%.dat`, basic information such as the inverse of temperature and the norm of the wave function before normalization is outputted with a TPQ step for each number of runs. In `SS_rand%%.dat`, physical quantities such as the inverse of temperature, energy, and expected value of the square of the Hamiltonian are outputted with a TPQ step for each number of runs. In `zvo_cisajs_set%%step&&.dat` and `zvo_cisajsktalt_set%%step&&.dat`, one-body and two-body Green's functions are outputted for each number of a TPQ steps and runs. The details of these files are shown in *Norm\_rand.dat*, *SS\_rand.dat*, *cisajs.dat*, *cisajsktalt.dat*.

## Full diagonalization method

When `method = "fulldiag"` is selected in an input file, a calculation by the full diagonalization method is started. After the calculation is completed normally, the following files are outputted, where `xx` is the number of the eigenstate counting from 0.

```
Eigenvalue.dat zvo_cisajs_eigen_xx.dat
zvo_cisajsktalt_eigen_xx.dat zvo_phys_Nup4_Ndown4.dat
```

In `Eigenvalue.dat`, an eigennumber and an eigenvalue are outputted for each line. In `zvo_cisajs_eigen_xx.dat` and `zvo_cisajsktalt_eigen_xx.dat`, one-body Green's functions and two-body Green's functions are outputted for each eigennumber. In `zvo_phys_Nup4_Ndown4.dat`, physical quantities, such as the expected values of energy and the doublon are outputted. The details of these files are shown in *Eigenvalue.dat* - *cisajsktalt.dat*.

## 3.1.2 Other tutorials

There are many tutorials in `samples`. For more details, please see `README.md` at each directory or **the manual for HPhi-tutorial**<https://issp-center-dev.github.io/HPhi/manual/develop/tutorial/en/html/index.html> .

## 3.2 Quick guide to Expert mode

For Expert mode, the following input files are needed.

1. A file list for input files
2. Files for basic parameters
3. Files for constructing Hamiltonian
4. Files for setting output components.

The process after calculation is the same as in Standard mode. In this section, we demonstrate Expert mode in the directory where the tutorial at the previous section was performed.

### 3.2.1 File list for input files

In namelist.def, the types of input files and filenames are defined as shown below. By writing the keyword and filenames at each line, the types of files are distinguished. The details of namelist.def are shown in [List file for the input files](#).

```

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

```

### 3.2.2 Files for basic parameters

In this subsection, we show how to set a calculation mode, the parameters for the calculation, and the positions of the localized spins.

#### Setting a calculation mode

The calculation mode is set in a CalcMod file (in this sample file, calcmmod.def). The contents of the files are as follows.

```

#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

```

We select a calculation method in CalcType and a target model in CalcModel. In this sample, we set the Lanczos method as a calculation method and the target model as the spin system (canonical ensemble). The details of a CalcMod file are shown in [CalcMod file](#).

#### Setting parameters for calculation

The parameters for the calculation are set in a ModPara file(in this sample, modpara.def). The contents of this file are as follows.

```

-----
Model_Parameters 0
-----
HPhi_Cal_Parameters
-----

```

(continues on next page)



(continued from previous page)

```

CDataFileHead  zvo
CParaFileHead  zqp
-----
Nsite          16
2Sz            0
Lanczos_max    2000
initial_iv     -1
exct           1
LanczosEps     14
LanczosTarget  2
LargeValue     4.5000000000000000e+00
NumAve         5
ExpecInterval  20
NOmega         200
OmegaMax       7.2000000000000000e+01    4.0000000000000000e-02
OmegaMin       -7.2000000000000000e+01    4.0000000000000000e-02
OmegaOrg       0.0 0.0

```

In this file, we set the parameters for the calculation, such as the site number, the total number of conduction electrons, the total  $S_z$  and the number of Lanczos steps. The details of the ModPara file are shown in [ModPara file](#).

### Setting positions of localized spins

The positions  $S$  of the localized spins are defined by a LocSpin file (in this sample, locspn.def). The contents of the files are as follows.

```

=====
NlocalSpin     16
=====
=====i_0IteElc_1LocSpn =====
=====
0      1
1      1
2      1
3      1
4      1
5      1
...

```

When CalcModel in a CalcMod file is set as the spin system, all the sites are automatically treated as localized spins. The details of a LocSpin file are shown in [LocSpin file](#).

### 3.2.3 Files for constructing Hamiltonian

After setting the basic parameters, we create input files for constructing the Hamiltonian. Since the calculations are performed by using the representation of the fermion operators in  $\mathcal{H}\Phi$ , we must rewrite the spin operator. For example, in the case of  $S = 1/2$ , we rewrite the equation by using the relation

$$\begin{aligned} S_i^z &= (c_{i\uparrow}^\dagger c_{i\uparrow} - c_{i\downarrow}^\dagger c_{i\downarrow})/2, \\ S_i^+ &= c_{i\uparrow}^\dagger c_{i\downarrow}, \\ S_i^- &= c_{i\downarrow}^\dagger c_{i\uparrow}. \end{aligned} \quad (3.2)$$

#### Setting transfer integrals

In a Trans file (in this sample, zTrans.def), we set the transfer part of the Hamiltonian,

$$\mathcal{H}_+ = - \sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2} c_{i\sigma_1}^\dagger c_{j\sigma_2}. \quad (3.3)$$

The contents of the files are as follows.

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

We can use this term when an electric magnetic field is added in the spin system. For example, when a magnetic field is added at a site 1 such as  $-0.5S_z^{(1)}$  for  $S = 1/2$ , this term can be rewritten as  $-0.5/2(c_{1\uparrow}^\dagger c_{1\uparrow} - c_{1\downarrow}^\dagger c_{1\downarrow})$ . Thus, the input file becomes as follows.

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

The details for a Trans file are shown in *Trans file*.

#### Setting general two-body interactions

In an InterAll file (in this sample, zInterall.def), we set the general two-body interaction part of the Hamiltonian,

$$\mathcal{H}_+ = \sum_{i,j,k,l} \sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}. \quad (3.4)$$

The contents of the files are as follows.

```
=====
NInterAll      96
=====
=====zInterAll=====
```

(continues on next page)

(continued from previous page)

=====									
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
...									

Here, we explain the interaction between site  $i$  and site  $j$  in the case of  $S = 1/2$ , for simplicity. Using fermion operators, the interaction terms for the spin operators can be rewritten as

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

Thus, the interaction  $S_i^z S_{i+1}^z$  for  $J = 2$  can be written as

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

in the format of an InterAll file. The other terms can be written as follows.

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

There are other file formats for constructing the Hamiltonian. The details of the input formats of two-body interactions are shown in *InterAll file - PairLift file*.

### 3.2.4 Setting output components

In OneBodyG and TwoBodyG files, the indices of one-body and two-body Green's functions are defined, respectively.

#### Setting indices of one-body Green's functions

In a OneBodyG file (in this sample, greenone.def), the indices of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$  are defined. The contents of files are as follows.

=====			
NCisAjs	32		
=====			
===== Green functions =====			
=====			
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

(continues on next page)

(continued from previous page)

```

    2      0      2      0
...

```

The details of the input formats of a OneBodyG file are shown in *OneBodyG file*.

### Setting indices of two-body Green's functions

In the TwoBodyG file (in this sample, greentwo.def), the indices of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$  are defined. The contents of this file are as follows.

```

=====
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
...

```

The details of the input formats of the TwoBodyG file are shown in *TwoBodyG file*.

### 3.2.5 Running

After creating all the input files above, we are ready to run a program. For Expert mode, we must set an option "-e" and a file name list (in this sample, namelist.def) as arguments to run  $\mathcal{H}\Phi$ .

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

The process after the calculation is the same as that of Standard mode.

## 3.3 Fourier transformation of correlation functions

This package has a utility which performs the Fourier transformation of the correlation function and plots that function. For details, see Appendix A.

## FILE SPECIFICATION

### 4.1 Input files for *Standard* mode

An example of an input file for the standard mode is as follows:

```
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
```

#### Basic rules for input files

- In each line, there is a set of a keyword (before an “=”) and a parameter(after an “=”); they are separated by “=”.
- You can describe keywords in a random order.
- Empty lines and lines beginning with a “//”(comment outs) are skipped.
- Upper- and lowercase are not distinguished. Double quotes and blanks are ignored.
- There are three types of parameters.
  1. Parameters that must be specified (if not,  $\mathcal{H}\Phi$  will stop with error messages),
  2. Parameters that it is not necessary to specified (if not specified, default values are used),
  3. Parameters that must not be specified (if specified,  $\mathcal{H}\Phi$  will stop with error messages).An example of type 3 is the transfer  $t$  parameter for the Heisenberg spin system. If you choose “model=spin”, you should not specify “ $t$ ”.

We explain each keyword as follows.

### 4.1.1 Parameters for the type of calculation

- **model**

**Type :** String (choose from "Fermion Hubbard", "Spin", "Kondo Lattice", "Fermion HubbardGC", "SpinGC", "Kondo LatticeGC", "SpinGCCMA")<sup>1</sup>

**Description :** The target model is specified with this parameter; the expressions above denote the canonical ensemble of the Fermion in the Hubbard model

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

the canonical ensemble in the Spin model( $\{\alpha, \beta\} = \{x, y, z\}$ )

$$\begin{aligned} \mathcal{H} = & -h \sum_i S_i^z - \Gamma \sum_i S_i^x + D \sum_i S_i^z S_i^z \\ & + \sum_{ij, \alpha} J_{ij\alpha} S_i^\alpha S_j^\alpha + \sum_{ij, \alpha \neq \beta} J_{ij\alpha\beta} S_i^\alpha S_j^\beta, \end{aligned}$$

the canonical ensemble in the Kondo lattice model

$$\mathcal{H} = -\mu \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma} - t \sum_{\langle ij \rangle \sigma} c_{i\sigma}^\dagger c_{j\sigma} + \frac{J}{2} \sum_i \left\{ S_i^+ c_{i\downarrow}^\dagger c_{i\uparrow} + S_i^- c_{i\uparrow}^\dagger c_{i\downarrow} + S_i^z (n_{i\uparrow} - n_{i\downarrow}) \right\}, \quad (4.2)$$

the grand canonical ensemble of the Fermion in the Hubbard model [Eqn. (4.1)], the grand canonical ensemble in the Spin model [Eqn. (4.2)], and the grand canonical ensemble in the Kondo lattice model [Eqn. (4.2)], respectively.

When model="SpinGCCMA", by using a more efficient algorithm<sup>2</sup>, HΦ calculates a system that is the same as "SpinGC". However, supported models and MPI processes are highly limited. See "Lattice" section.

- **method**

**Type :** String (choose from "Lanczos", "TPQ", "Full Diag", "CG", "Time Evolution")

**Description :** The calculation type is specified with this parameter; the above expressions above denote the single eigenstate calculation by using the Lanczos method, at the finite-temperature by using the thermally pure quantum state, the full diagonalization method, the multiple eigenstates calculation by using the LOBCG method<sup>3,4</sup>, and the simulation of real-time evolution, respectively.

The scheme employed for the spectrum calculation is also specified with this parameter. If "CG" is chosen, the shifted bi-conjugate gradient method<sup>5</sup> together with the seed-switch technique<sup>6</sup> is employed with the help of the Kω library<sup>7</sup>.

- **lattice**

**Type :** String (choose from "Chain Lattice", "Square Lattice", "Triangular Lattice", "Honeycomb Lattice", "Kagome", "Ladder")

**Description :** The lattice shape is specified with this parameter; the expressions above denote the one-dimensional chain lattice ( Fig. 4.1 (a)), the two-dimensional square lattice ( Fig. 4.1 (b)), the two-dimensional triangular lattice ( Fig. 4.1 (c)), the two-dimensional anisotropic honeycomb lattice ( Fig. 4.2 ), the Kagome Lattice( Fig. 4.3 ), and the ladder lattice ( Fig. 4.4 ) respectively.

<sup>1</sup> GC=Grand Canonical

<sup>2</sup> Y. Yamaji *et. al.*, manuscript in preparation.

<sup>3</sup> A.V.Knyazev, SIAM Journal on Scientific Computing **23**, 517 (2001).

<sup>4</sup> S.Yamada, T.Imamura, M.Machida, The Japan Society for Computational Engineering and Science **2006**, 20060027 (2006).

<sup>5</sup> A.Frommer, Computing **70**, 87-109 (2003).

<sup>6</sup> S.Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

<sup>7</sup> <https://github.com/issp-center-dev/Komega>.

In `method="SpinGCCMA"`, only "Chain Lattice", "Honeycomb Lattice", "Kagome", and "Ladder" are supported. The limits of  $L$ ,  $W$ , and the number of MPI processes ( $N_{\text{proc}}$ ) are as follows:

- "Chain Lattice"

$L = 8n$  (where  $n$  is an integer number under the condition  $n \geq 1$ ),  $N_{\text{proc}} \leq 2(L = 8)$ ,  $N_{\text{proc}} \leq 2^{L/2-2}(L > 8)$ .

- "Honeycomb Lattice"

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

- "Kagome"

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

- "Ladder"

$W = 2, L = 2n$  (where  $n$  is an integer number under the condition  $n \geq 4$ ),  $N_{\text{proc}} \leq 2^{L-4}$ .

### 4.1.2 Parameters for the lattice

#### Chain [ Fig. 4.1 (a)]

- $L$

**Type :** Integer

**Description :** The length of the chain is specified with this parameter.

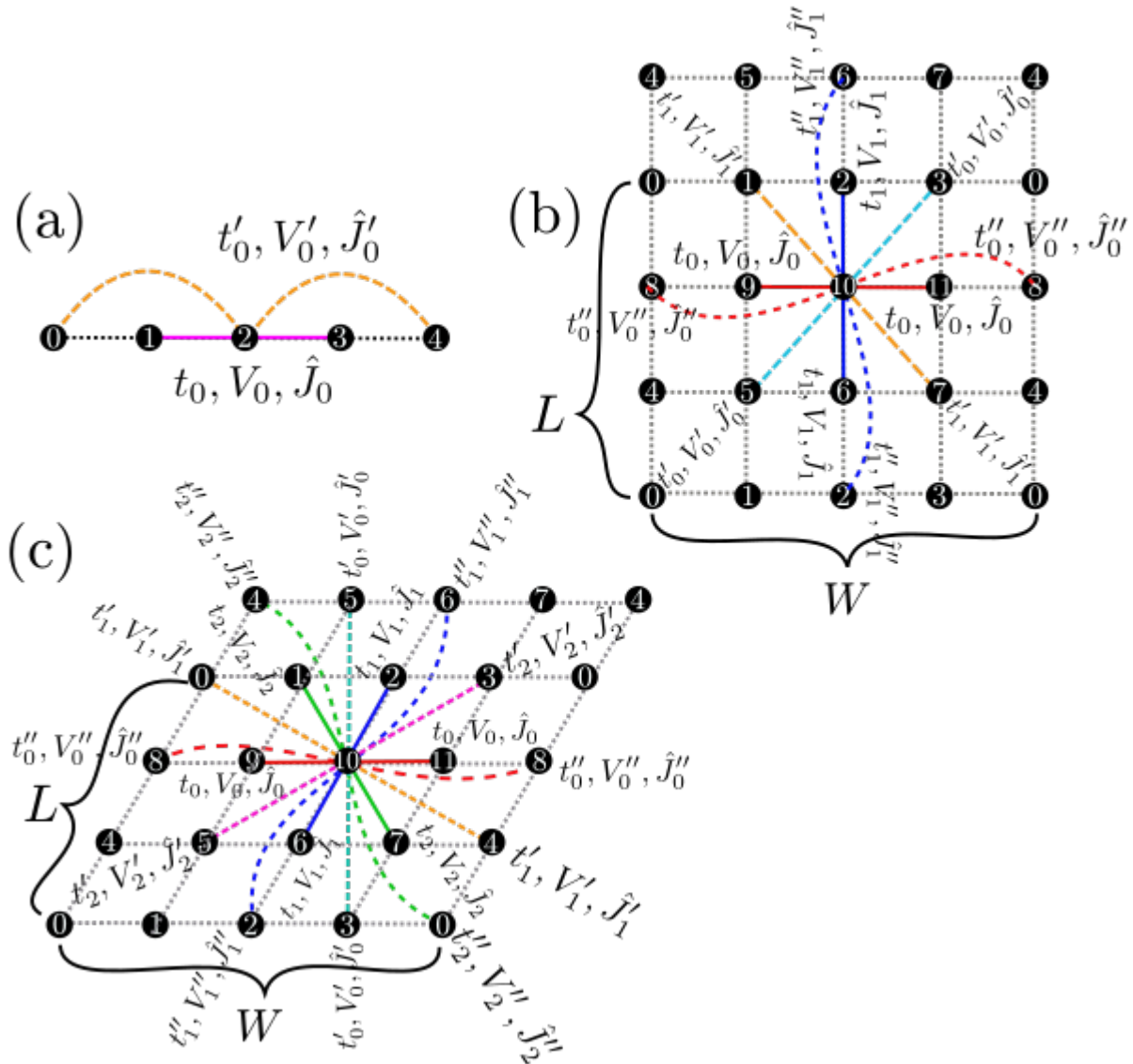


Fig. 4.1: Schematic illustration of (a) one-dimensional chain lattice, (b) two-dimensional square lattice, and (c) two-dimensional triangular lattice. They have  $t$ ,  $V$ , and  $J$  as the nearest neighbor hopping, an offsite Coulomb integral, and a spin-coupling constant, respectively (magenta solid lines); they also have  $t'$ ,  $V'$ , and  $J'$  as the next nearest neighbor hopping, offsite Coulomb integral, and spin-coupling constant, respectively (green dashed line).



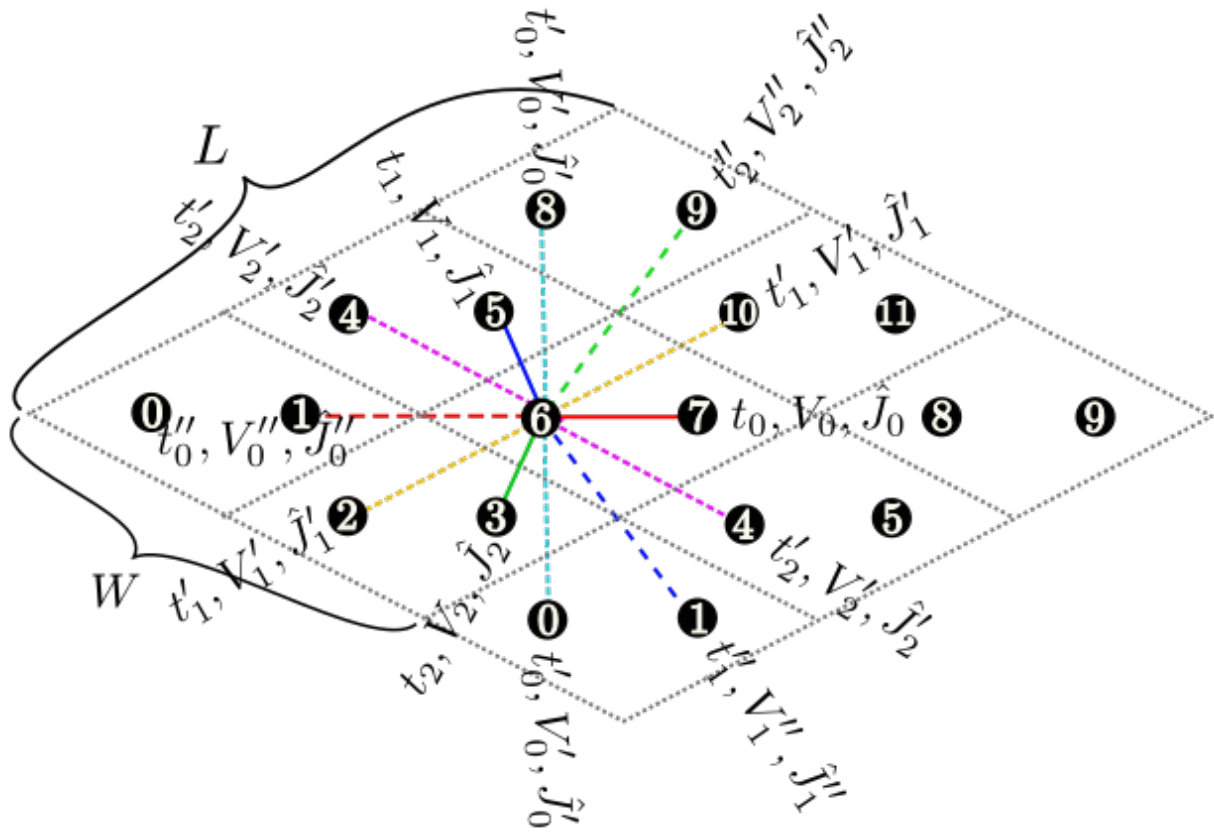


Fig. 4.2: Schematic illustration of the anisotropic honeycomb lattice. The first/second/third nearest neighbor hopping integral, spin coupling, and offsite Coulomb integral depend on the bond direction.

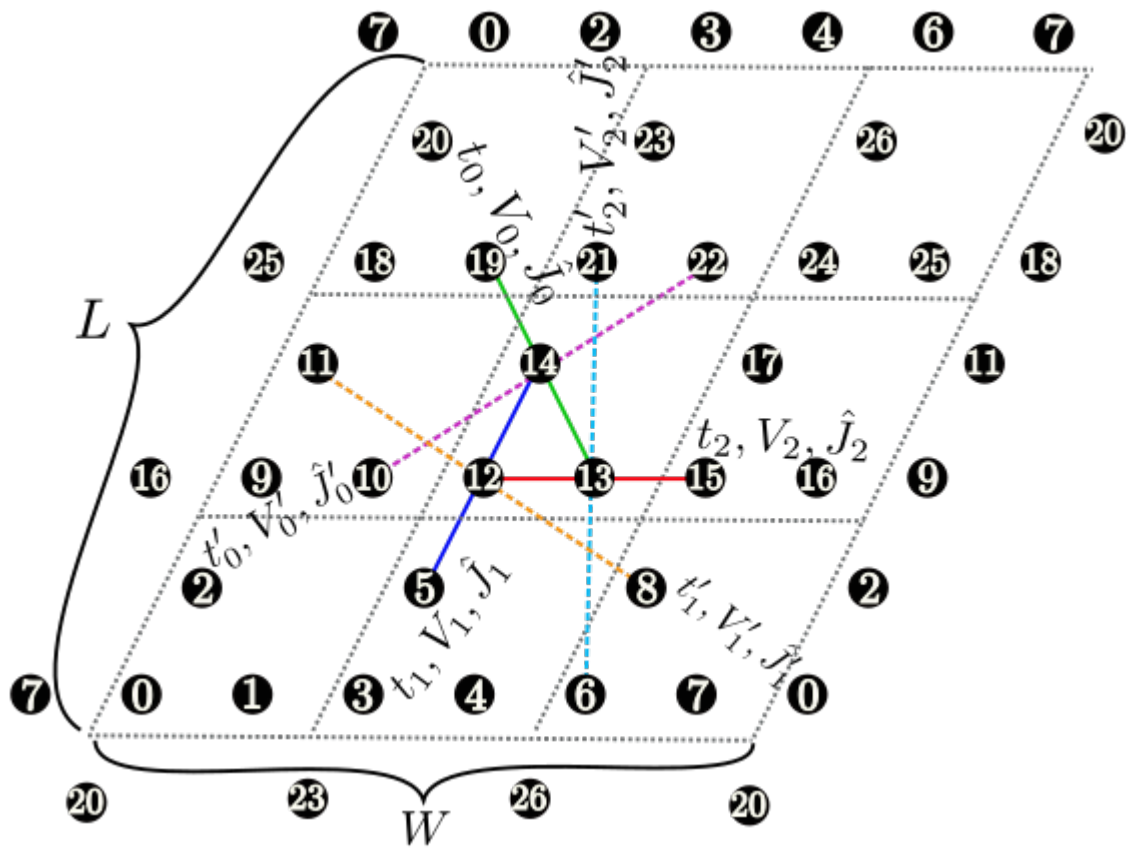


Fig. 4.3: Schematic illustration of the Kagome lattice.

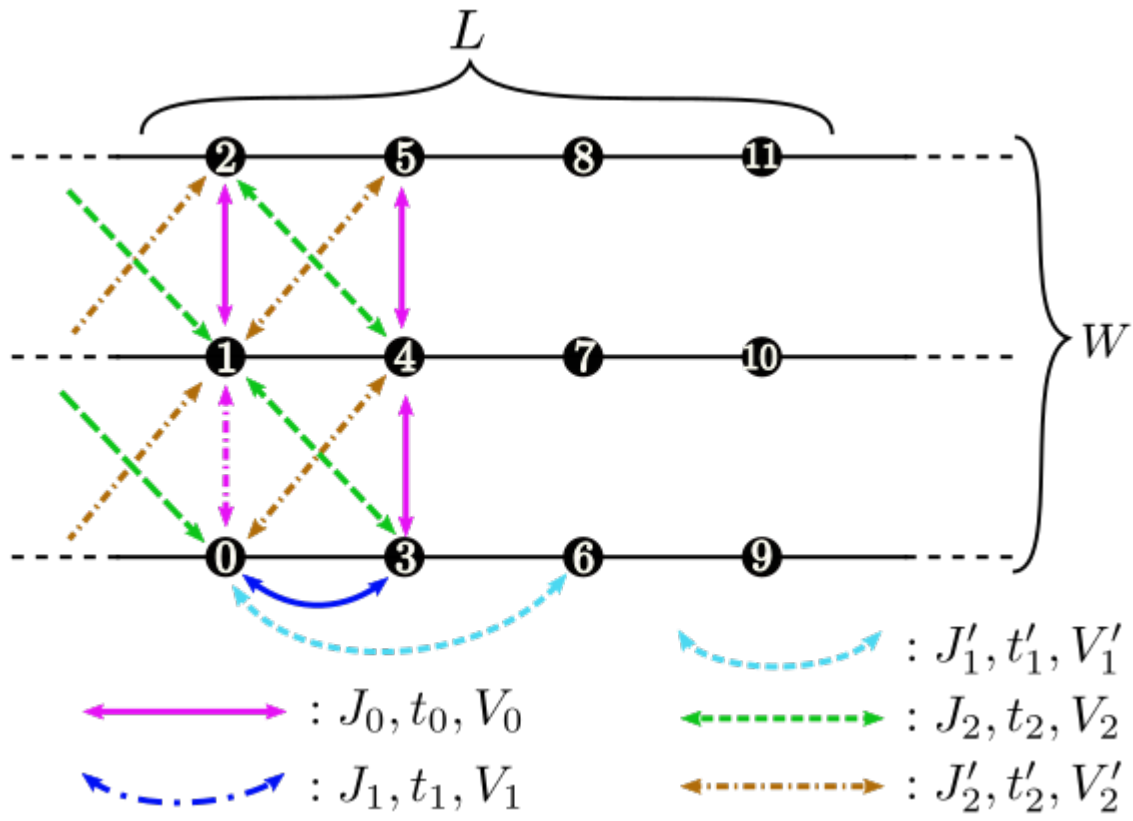


Fig. 4.4: Schematic illustration of the ladder lattice.

**Ladder ( Fig. 4.4 )**

- L

**Type :** Integer

**Description :** The length of the ladder is specified with this parameter.

- W

**Type :** Integer

**Description :** The number of the ladder is specified with this parameter.

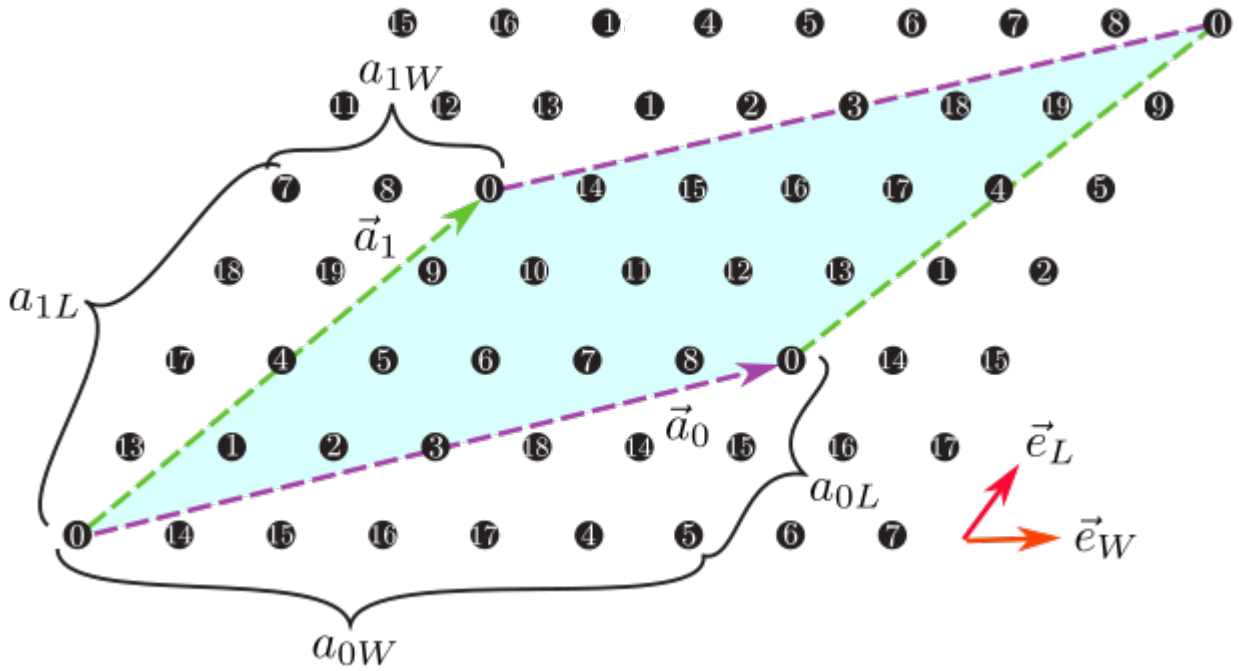


Fig. 4.5: Shape of the numerical cell when  $\mathbf{a}_0 = (6, 2)$ ,  $\mathbf{a}_1 = (2, 4)$  in the triangular lattice. The region surrounded by  $\mathbf{a}_0$  (magenta dashed arrow) and  $\mathbf{a}_1$  (green dashed arrow) becomes the cell to be calculated (20 sites).

**Tetragonal lattice** [ Fig. 4.1 (b) ], triangular lattice [ Fig. 4.1 (c) ], honeycomb lattice [ Fig. 4.2 ], Kagome lattice [ Fig. 4.3 ]

In these lattices, we can specify the shape of the numerical cell by using the following two methods.

- W, L

**Type :** Integer

**Description :** The alignment of the original unit cells (dashed black lines in Fig. 4.1 - Fig. 4.3 ) is specified with this parameter.

- a0W, a0L, a1W, a1L

**Type :** Integer

**Description :** We can specify two vectors ( $\mathbf{a}_0, \mathbf{a}_1$ ) that surround the numerical cell (Fig. 4.5 ). These vectors should be specified in the fractional coordinate.

If we use both these methods,  $\mathcal{H}\Phi$  stops. When `model=SpinGCCMA`, we can use only the former.

We can check the shape of the numerical cell by using a file `lattice.gp` which is written in Standard mode. This file can be read by `gnuplot` as follows:

```
$ gnuplot lattice.gp
```

### 4.1.3 Parameters for conserved quantities

- nelec

**Type :** Positive integer

**Description :** The number of valence electrons is specified with this parameter. When model = "Fermion HubbardGC", "Spin", or "SpinGC", it should not be specified.

- 2Sz

**Type :** Integer

**Description :** The  $z$  component of the twofold total spin is specified with this parameter. When model = "Fermion HubbardGC" or "SpinGC", it should not be specified.

#### 4.1.4 Parameters for the Hamiltonian

A default value is 0 unless a specific value is written in the description. Table 4.1 shows the parameters for each models. In the case of a complex type, a file format is “*a real part, an imaginary part*” while in the case of a real type, only “*a real part*”.

##### Local terms

- mu

**Type :** Real

**Description :** It is available only for the Hubbard and Kondo lattice model. The chemical potential  $\mu$  (including the site potential) is specified with this parameter.

- U

**Type :** Real

**Description :** It is available only for the Hubbard and Kondo lattice model. The onsite Coulomb integral  $U$  is specified with this parameter.

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

**Type :** Real

**Description :** It is available only for the Kondo lattice model. The spin-coupling constant between the valence and the local electrons is specified with this parameter. If the exchange coupling  $J$  is specified in the input file, instead of Jx, Jy, Jz, the diagonal exchange couplings, Jx, Jy, Jz, are set as  $Jx = Jy = Jz = J$ . When both the set of exchange couplings (Jx, Jy, Jz) and the exchange coupling  $J$  are specified in the input file, HΦ will stop.

- h

**Type :** Real

**Description :** The longitudinal magnetic field is specified with this parameter.

- Gamma, D

**Type :** Real

**Description :** (Spin model) The transverse magnetic field, and the single-site anisotropy parameter are specified with these parameters. The single-site anisotropy parameter is not available for model=SpinGCCMA.

The non-local terms described below should be specified differently according to the lattice structure: For lattice=Ladder, the non-local terms are specified differently from those for the lattice=Chain Lattice, Square Lattice, Triangular Lattice, Honeycomb Lattice, Kagome. Below, the available parameters for each lattice are shown in Table 4.1 .

Table 4.1: Interactions for each model defined in an input file. We can define spin couplings as a matrix format.

Interactions	1D chain	2D square	2D triangular	Honeycomb	Kagome	Ladder
J, t, V(simplified)	○	○	○	○	○	-
J0, t0, V0	○	○	○	○	○	○
J1, t1, V1	-	○	○	○	○	○
J2, t2, V2	-	-	○	○	○	○
J', t', V'(simplified)	○	○	○	○	○	-
J0', t0', V0'	○	○	○	○	○	-
J1', t1', V1'	-	○	○	○	○	○
J2', t2', V2'	-	-	○	○	○	○
J'', t'', V''(simplified)	○	○	○	○	-	-
J0'', t0'', V0''	○	○	○	○	-	-
J1'', t1'', V1''	-	○	○	○	-	-
J2'', t2'', V2''	-	-	○	○	-	-

### Non-local terms[ for Ladder ( Fig. 4.4 )]

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

**Type :** Complex

**Description :** (Hubbard and Kondo lattice model) Hopping integrals in the ladder lattice (see Fig. 4.4 ) are specified with this parameter.

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

**Type :** Real

**Description :** (Hubbard and Kondo lattice model) Offsite Coulomb integrals on the ladder lattice ( Fig. 4.2 ) are specified with these parameters.

- 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.

**Type :** Real

**Description :** (Spin model) Spin-coupling constants in the ladder lattice (see Fig. 4.4 ) are specified with these parameters. If the simplified parameter J0 is specified in the input file instead of the diagonal couplings, J0x, J0y, J0z, these diagonal couplings are set as J0x = J0y = J0z = J0. If both J0 and the set of the couplings (J0x, J0y, J0z) are specified, HΦ will stop. The above rules are also valid for the simplified parameters, J1, J1', J2, and J2'.



## Non-local terms [other than Ladder ( Fig. 4.1 , Fig. 4.2 , Fig. 4.3 )]

- $t, t_0, t_1, t_2$

**Type :** Complex

**Description :** (Hubbard and Kondo lattice model) The nearest neighbor hoppings for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with these parameters. If there is no bond dependence of the hoppings, the simplified parameter  $t$  is available to specify  $t_0, t_1$ , and  $t_2$  as  $t_0 = t_1 = t_2 = t$ . If both  $t$  and the set of the hoppings ( $t_0, t_1, t_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- $t', t'_0, t'_1, t'_2$

**Type :** Complex

**Description :** (Hubbard and Kondo lattice model) The second nearest neighbor hoppings for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with these parameter. If there is no bond dependence of the hoppings, the simplified parameter  $t'$  is available to specify  $t'_0, t'_1$ , and  $t'_2$  as  $t'_0 = t'_1 = t'_2 = t'$ . If both  $t'$  and the set of the hoppings ( $t'_0, t'_1, t'_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- $t'', t''_0, t''_1, t''_2$

**Type :** Complex

**Description :** (Hubbard and Kondo lattice model) The third nearest neighbor hoppings for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with these parameter. If there is no bond dependence of the hoppings, the simplified parameter  $t''$  is available to specify  $t''_0, t''_1$ , and  $t''_2$  as  $t''_0 = t''_1 = t''_2 = t''$ . If both  $t''$  and the set of the hoppings ( $t''_0, t''_1, t''_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- $V, V_0, V_1, V_2$

**Type :** Real

**Description :** (Hubbard and Kondo lattice model) The nearest neighbor offsite Coulomb integrals  $V$  for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with these parameters. If there is no bond dependence of the offsite Coulomb integrals, the simplified parameter  $V$  is available to specify  $V_0, V_1$ , and  $V_2$  as  $V_0 = V_1 = V_2 = V$ . If both  $V$  and the set of the Coulomb integrals ( $V_0, V_1, V_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- $V', V'_0, V'_1, V'_2$

**Type :** Real

**Description :** (Hubbard and Kondo lattice model) The second nearest neighbor-offsite Coulomb integrals  $V'$  for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with this parameter. If there is no bond dependence of the offsite Coulomb integrals, the simplified parameter  $V'$  is available to specify  $V'_0, V'_1$ , and  $V'_2$  as  $V'_0 = V'_1 = V'_2 = V'$ . If both  $V'$  and the set of the Coulomb integrals ( $V'_0, V'_1, V'_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- $V'', V''_0, V''_1, V''_2$

**Type :** Real

**Description :** (Hubbard and Kondo lattice model) The third nearest neighbor-offsite Coulomb integrals  $V''$  for each direction (see Fig. 4.1 - Fig. 4.3 ) are specified with this parameter. If there is no bond dependence of the offsite Coulomb integrals, the simplified parameter  $V''$  is available to specify  $V''_0, V''_1$ , and  $V''_2$  as  $V''_0 = V''_1 = V''_2 = V''$ . If both  $V''$  and the set of the Coulomb integrals ( $V''_0, V''_1, V''_2$ ) are specified,  $\mathcal{H}\Phi$  will stop.

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

**Type :** Real

**Description :** (Spin model) The nearest neighbor exchange couplings for each direction are specified with these parameters. If the simplified parameter  $J_0$  is specified, instead of  $J_{0x}$ ,  $J_{0y}$ ,  $J_{0z}$ , the exchange couplings,  $J_{0x}$ ,  $J_{0y}$ ,  $J_{0z}$ , are set as  $J_{0x} = J_{0y} = J_{0z} = J_0$ . If both  $J_0$  and the set of the exchange couplings ( $J_{0x}$ ,  $J_{0y}$ ,  $J_{0z}$ ) are specified,  $\mathcal{H}\Phi$  will stop. The above rules are valid for  $J_1$  and  $J_2$ .

If there is no bond dependence of the exchange couplings, the simplified parameters,  $J_x$ ,  $J_y$ ,  $J_z$ ,  $J_{xy}$ ,  $J_{yx}$ ,  $J_{xz}$ ,  $J_{zx}$ ,  $J_{yz}$ ,  $J_{zy}$ , are available to specify the exchange couplings for every bond as  $J_{0x} = J_{1x} = J_{2x} = J_x$ . If any simplified parameter ( $J_x$ - $J_{zy}$ ) is specified in addition to its counterparts ( $J_{0x}$ - $J_{2zy}$ ),  $\mathcal{H}\Phi$  will stop. Below, examples of parameter sets for nearest neighbor exchange couplings are shown.

- If there are no bond-dependent, and no anisotropic and offdiagonal exchange couplings (such as  $J_{xy}$ ), please specify  $J$  in the input file.
  - If there are no bond-dependent and offdiagonal exchange couplings but there are anisotropic couplings, please specify the non-zero couplings in the diagonal parameters,  $J_x$ ,  $J_y$ ,  $J_z$ .
  - If there are no bond-dependent exchange couplings but there are anisotropic and offdiagonal exchange couplings, please specify the non-zero couplings in the nine parameters,  $J_x$ ,  $J_y$ ,  $J_z$ ,  $J_{xy}$ ,  $J_{yz}$ ,  $J_{xz}$ ,  $J_{yx}$ ,  $J_{zy}$ ,  $J_{zx}$ .
  - If there are no anisotropic and offdiagonal exchange couplings, but there are bond-dependent couplings, please specify the non-zero couplings in the three parameters,  $J_0$ ,  $J_1$ ,  $J_2$ .
  - If there are no anisotropic exchange couplings, but are bond-dependent and offdiagonal couplings, please specify the non-zero couplings in the nine parameters,  $J_{0x}$ ,  $J_{0y}$ ,  $J_{0z}$ ,  $J_{1x}$ ,  $J_{1y}$ ,  $J_{1z}$ ,  $J_{2x}$ ,  $J_{2y}$ ,  $J_{2z}$ .
  - If there are bond-dependent, anisotropic, and offdiagonal exchange couplings, please specify the non-zero couplings in the twenty-seven parameters from  $J_{0x}$  to  $J_{2zy}$ .
- $J'x, J'y, J'z, J'xy, J'yx, J'xz, J'zx, J'yz, J'zy$
  - $J_0'x, J_0'y, J_0'z, J_0'xy, J_0'yx, J_0'xz, J_0'zx, J_0'yz, J_0'zy$
  - $J_1'x, J_1'y, J_1'z, J_1'xy, J_1'yx, J_1'xz, J_1'zx, J_1'yz, J_1'zy$
  - $J_2'x, J_2'y, J_2'z, J_2'xy, J_2'yx, J_2'xz, J_2'zx, J_2'yz, J_2'zy$

**Type :** Real

**Description :** (Spin model) The second nearest neighbor exchange couplings are specified. However, for lattice = Honeycomb Lattice and lattice = Kagome with model=SpinGCCMA, the second nearest neighbor exchange couplings are not available in the *Standard* mode. If the simplified parameter  $J''$  is specified, instead of  $J''x$ ,  $J''y$ ,  $J''z$ , the exchange couplings are set as  $J''x = J''y = J''z = J''$ . If both  $J''$  and the set of the couplings ( $J''x$ ,  $J''y$ ,  $J''z$ ) are specified,  $\mathcal{H}\Phi$  will stop.

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

**Type :** Real

**Description :** (Spin model) The third nearest neighbor exchange couplings are specified. However, for lattice = Honeycomb Lattice and lattice = Kagome with model=SpinGCCMA, the third nearest neighbor exchange couplings are not available in the *Standard* mode. If the simplified parameter  $J'''$  is specified, instead of  $J'''x$ ,  $J'''y$ ,  $J'''z$ , the exchange couplings are set as  $J'''x = J'''y = J'''z = J'''$ . If both  $J'''$  and the set of the couplings ( $J'''x$ ,  $J'''y$ ,  $J'''z$ ) are specified,  $\mathcal{H}\Phi$  will stop.

- phase0, phase1

**Type :** Double (0.0 as defaults)

**Description :** We can specify the phase for the hopping through the cell boundary with these parameter (unit: degree). These factors for the  $a_0$  direction and the  $a_1$  direction can be specified independently. For the one-dimensional system, only `phase0` can be used. For example, a hopping from  $i$ -th site to  $j$ -th site through the cell boundary with the positive direction becomes as

$$\exp(i \times \text{phase0} \times \pi/180) \times t c_{j\sigma}^\dagger c_{i\sigma} + \exp(-i \times \text{phase0} \times \pi/180) \times t^* c_{i\sigma}^\dagger c_{j\sigma} \quad (4.3)$$

### 4.1.5 Parameters for the numerical condition

- 2S

**Type :** Positive integer (1 as a default)

**Description :** The  $2S$  at each site in the localized spin system is specified. (E.g. 1 for the  $1/2$  system)

- Restart

**Type :** String (choose from "None", "Restart\_out", "Restart\_in", "Restart". "None" as a default)

**Description :** The condition of the restart is specified. "None" for omitting file IOs for the restart, "Restart\_out" for starting calculation from scratch and generating a restart-file after the calculation finishes, "Restart\_in" for starting calculation with the restart-file generated in the previous run, "Restart" for "Restart\_out" + "Restart\_in".

- Lanczos\_max

**Type :** Positive integer (default value: 2000)

**Description :** The upper limit of the Lanczos/LOBCG/BiCG step and the number of steps for TPQ/Time=evolution are specified with this parameter.

- initial\_iv

**Type :** Integer (default value: -1)

**Description :** An initial vector is specified with this parameter.

- Lanczos method

- \* For the canonical ensemble and  $\text{initial\_iv} \geq 0$

The non-zero components of an initial vector are specified with this parameter.

- \* For the grand canonical ensemble or  $\text{initial\_iv} < 0$

The seed of the random generator is given by this parameter and the random vector is used as the initial vector.

- TPQ method

The seed of the random generator is given by this parameter and the random vector is used as the initial vector.

See [Algorithm](#) for details of setting an initial vector.

- exct

**Type :** Positive integer (default value: 1)

**Description :** The number of eigenvectors obtained from the ground energy by the Lanczos method are specified.

When  $\text{exct}=2$ , the eigenvector of the first-excited state is obtained. When  $\text{method}=\text{"CG"}$ , the number of states to be calculated is specified.

**Note:** The condition  $\text{nvec} \geq \text{exct}$  must be satisfied.

- LanczosEps

**Type :** Positive integer (default value: 14)

**Description :** The convergence criterion for the Lanczos method is specified with this parameter. If the difference between the old and the new target eigenvalue falls below  $10^{-\text{LanczosEps}}$ , the Lanczos step will finish. For `method="CG"`, we assume the calculation is converged when the 2-norm of the residual vector becomes smaller than  $10^{-\text{LanczosEps}/2}$ .

- LanczosTarget

**Type :** Positive integer (default value: 2)

**Description :** The target eigenenergy for the convergence criterion is specified. If it is set to 1, the target eigenenergy becomes the ground state.

- LargeValue

**Type :** Double (the default value is provided below)

**Description :** (Only for TPQ)  $l$  as  $(l - \mathcal{H}/N_s)$  is used in the mTPQ calculation. Usually, the largest eigenvalue of the Hamiltonian is used as  $l$ . Thus, the default value of  $l$  is taken as the summation of the absolute values of each coefficient in the Hamiltonian divided by the number of sites. In the cTPQ calculation, `LargeValue` is used as  $\Delta\tau = 1/\text{LargeValue}$ . The definition of  $\Delta\tau$  is explained in [Algorithm](#).

- NumAve

**Type :** Positive integer (default value: 5)

**Description :** (Only for TPQ) The number of independent runs for the TPQ method is specified with this parameter.

- ExpecInterval

**Type :** Positive integer (default value: 20)

**Description :** (Only for TPQ) The interval of calculating correlation functions in the TPQ iteration is specified.

**Note:** A small interval increases the time cost of calculations.

- OutputMode

**Type :** Choose from "none", "correlation", and "full" (correlation as default)

**Description :** Indices of correlation functions are specified with this keyword. "none" indicates correlation functions will not be calculated. When `outputmode="correlation"`, the correlation function supported by the utility `fourier` is computed. For more details, see the document in `doc/fourier/`. If "full" is selected,  $\langle c_{i\sigma}^\dagger c_{j\sigma'} \rangle$  is computed at all  $i, j, \sigma, \sigma'$ , and  $\langle c_{i_1\sigma_1}^\dagger c_{i_2\sigma_2} c_{i_3\sigma_3}^\dagger c_{i_4\sigma_4} \rangle$  is computed at all  $i_1, i_2, i_3, i_4, \sigma_1, \sigma_2, \sigma_3, \sigma_4$ .

In a spin system, the indices are specified as those of the Bogoliubov representation (see [Bogoliubov representation](#)).

- InitialVecType

**Type :** Character (choose from "C", "R". "C" as a default)

**Description :** The type of the initial eigenvector is specified. C for the complex number, and R for the real number.

- EigenVecIO

**Type :** String (choose from "None", "Out", "In". "None" as a default)

**Description :** The I/O of the eigenvector is specified. "None" for omitting the IO of the eigenvector, "Out" for writing the eigenvector to a file, "In" for reading the eigenvector from a file and using it in the subsequent calculation (such as the Green's function).

- HamIO

**Type :** String (choose from "None", "Out", "In". "None" as a default)

**Description :** (Only used in Full Diag mode)

The I/O of the Hamiltonian is specified.

None: not output Hamiltonian.

Out: output Hamiltonian.

In: Input Hamiltonian.

- OutputExcitedVec

**Type :** String (choose from "None" or "Out". "None" as a default)

**Description :** (Only used in the mode to calculate dynamical green's functions)

The mode to output the excited vector is specified.

None: not output the excited vector.

Out: output the excited vector.

### 4.1.6 Parameters for the dynamical Green's function

- CalcSpec

**Type :** String (choose from "None", "Normal", "NoIteration", "Restart\_out", "Restart\_in", "Restart", "None" as default.)

**Description :** The condition for the calculation of the dynamical Green's function is specified. "None" for omitting the calculation of the dynamical Green's function. "Normal" for calculating that function from scratch, "NoIteration" for calculating that function with the same iteration in the previous run (In this case, the Hamiltonian-vector product is not performed. Although the numerical cost is very small, the convergence is not guaranteed), "Restart\_out" for calculating that function from scratch and writing the restart-file at the end, "Restart\_in" for starting the calculation with the previously written restart-file, "Restart" for "Restart\_out" + "Restart\_in".

The scheme for the spectrum calculation is specified by using the parameter method. If method="CG" is chosen, the shifted bi-conjugate gradient method<sup>1</sup> together with the seed-switch technique<sup>2</sup> is employed with the help of the  $K\omega$  library<sup>3</sup>.

- SpectrumType

**Type :** String (choose from "SzSz", "S+S-", "Density", "up", "down". "SzSz" as default.)

**Description :** The type of the dynamical Green's function to be computed is specified. "SzSz" for  $\langle S_{-q}^z S_q^z \rangle$ , "S+S-" for  $\langle S_{-q}^+ S_q^- \rangle$ , "Density" for  $\langle n_{-q} n_q \rangle$ , "up" for  $\langle c_{q\uparrow}^\dagger c_{q\uparrow} \rangle$ , "down" for  $\langle c_{q\downarrow}^\dagger c_{q\downarrow} \rangle$ .

- SpectrumQW, SpectrumQL

**Type :** Double (default value: 0.0)

**Description :** The wave number (Fractional coordinate) of the dynamical Green's function is specified. The reciprocal lattice vector is computed from the direct lattice vector shown in Fig. 4.1 , Fig. 4.2 , Fig. 4.3 , Fig. 4.4 .

- OmegaOrg

**Type :** Double (0.0 as default.)

**Description :** The origin value of the frequency.

- OmegaMin

**Type :** Double (-LargeValue times the number of sites as default.)

**Description :** The lower limit of the real part of the frequency measured from OmegaOrg.

- OmegaMax

**Type :** Double (LargeValue times the number of sites as default.)

**Description :** The upper limit of the real part of the frequency measured from OmegaOrg.

- OmegaIm

**Type :** Double (0.01\*LargeValue as a default.)

**Description :** The imaginary part of the frequency.

- NOmega

**Type :** Positive integer (200 as a default.)

**Description :** The number of frequencies.

<sup>1</sup> A. Frommer, Computing **70**, 87-109 (2003).

<sup>2</sup> S. Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

<sup>3</sup> <https://github.com/issp-center-dev/Komega>.

### 4.1.7 Parameters for time-evolution

- dt

**Type :** Positive Double(Default value : 0.1)

**Description :** The width of time steps.

- PumpType

**Type :** String (Chosen from "Quench", "Pulse Laser", "AC Laser", and "DC Laser".Default value : "Quench")

**Description :** The type of time-dependent Hamiltonian. For "Quench", two body operator  $U_{\text{quench}} \sum_i n_{i\uparrow} n_{i\downarrow}$  is added. For "Pulse Laser", "AC Laser", and "DC Laser", the hopping term is modulated as  $-\sum_{ij\sigma} t_{ij} \exp[-\mathbf{A}(t) \cdot (\mathbf{R}_i - \mathbf{R}_j)/(2\pi)] c_{i\sigma}^\dagger c_{j\sigma}$ , where  $\mathbf{A}(t)$  is the vector potential which is defined as  $\mathbf{A}(t) = \mathbf{A}_0 \exp[-(t - t_0)^2/(2t_{\text{dump}}^2)] \cos[\omega(t - t_0)]$ ,  $\mathbf{A}(t) = \mathbf{A}_0 \sin[\omega(t - t_0)]$ , and  $\mathbf{A}(t) = \mathbf{A}_0 t$  for "Pulse Laser", "AC Laser", and "DC Laser", respectively.

potential.dat file is written for displaying the vector potential and the electrical field at each time step.

- Uquench

**Type :** Double(Default value : 0.0)

**Description :**  $U_{\text{quench}}$

- freq

**Type :** Double(Default value : 0.1)

**Description :**  $\omega$

- tshift

**Type :** Double(Default value : 0.0)

**Description :**  $t_0$

- tdump

**Type :** Double (Default value : 0.1)

**Description :**  $t_{\text{dump}}$

- VecPotW, VecPotL

**Type :** Double (Default value : 0.0)

**Description :** The vector potential (fractional coordinate of the reciprocal space) at  $t = t_0$ . The reciprocal lattice vector is computed from the direct lattice vector shown in Fig. 4.1 , Fig. 4.2 , Fig. 4.3, Fig. 4.4 .



## 4.2 Input files for *Expert* mode

In this section, the details of the input files for the expert mode are explained. The input files are categorized according to the following four parts.

(1) **List:** This file is a list of the input file names with the keywords. Each keyword is fixed, but the file names can be determined freely.

(2) **Basic parameters:**

The following input files give the basic parameters. The types of input files are determined by the keywords.

**CalcMod:** Set the parameters for the calculation modes.

**ModPara:** Set the parameters for the basic parameters, such as site number, electron number, and Lanczos step.

**LocSpin:** Set the location of the local spin (used only in the Kondo model).

(3) **Hamiltonian:**

The Hamiltonian for  $\mathcal{H}\Phi$  is denoted by the format of the interactions for the electron system. The types of interaction are determined by the following keywords.

**Trans:** The one-body part,  $c_{i\sigma_1}^\dagger c_{j\sigma_2}$

**InterAll:** The general two-body interactions,  $c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$ .

We can set the interactions that are frequently used by the following keywords.

**CoulombIntra:** On-site Coulomb interactions,  $n_{i\uparrow}n_{i\downarrow}$  ( $n_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$ )

**CoulombInter:** Off-site Coulomb interactions,  $n_i n_j$  ( $n_i = n_{i\uparrow} + n_{i\downarrow}$ )

**Hund:** Hund couplings,  $n_{i\uparrow}n_{j\uparrow} + n_{i\downarrow}n_{j\downarrow}$

**PairHop:** Pair hopping couplings,  $c_{i\uparrow}^\dagger c_{j\uparrow} c_{i\downarrow}^\dagger c_{j\downarrow}$

**Exchange:** Exchange couplings,  $c_{i\uparrow}^\dagger c_{j\uparrow} c_{j\downarrow}^\dagger c_{i\downarrow}$

**Ising:** Ising interactions,  $S_i^z S_j^z$

**PairLift:** PairLift couplings,  $c_{i\uparrow}^\dagger c_{i\downarrow} c_{j\uparrow}^\dagger c_{j\downarrow}$ .

(4) **Output:**

The target for the output is determined.

**OneBodyG :** One-body Green's functions,  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$

**TwoBodyG :** Two-body Green's functions,  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ .

List file for the input files

### 4.2.1 List file for the input files

This file determines the input filenames, which are correlated with the keywords. The file format is as follows.

```
CalcMod  calcmmod.def
ModPara  modpara.def
LocSpin  zlocspn.def
Trans    ztransfer.def
InterAll zinterall.def
OneBodyG zcisajs.def
TwoBodyG zcisajscktaltdc.def
```

#### File format

[string01][string02]

#### Parameters

- [string01]  
**Type :** String  
**Description :** Select a word from keywords.
- [string02]  
**Type :** String  
**Description :** An input filename that is correlated with the keywords.

#### Use rules

- After setting keywords at [string01], the half-width state is needed for writing a filename. You can set the filename freely.
- Keywords for input files are shown in [Table 4.2](#)
- Essential keywords are “CalcMod”, “ModPara”, and “LocSpin”.
- Keywords can be set in random order.
- If the keywords or filenames are incorrect, the program is terminated.
- When the head of a line is “#”, the line is skipped.

Table 4.2: List of the definition files

Keywords	Details of corresponding files
CalcMod	Parameters for modes of calculation
ModPara	Parameters for calculation
LocSpin	Configurations of the local spins for Hamiltonian
Trans	Transfer and chemical potential for Hamiltonian
InterAll	Two-body interactions for Hamiltonian
CoulombIntra	CoulombIntra interactions
CoulombInter	CoulombInter interactions
Hund	Hund couplings
PairHop	Pair hopping couplings
Exchange	Exchange couplings
Ising	Ising interactions
PairLift	Pair lift couplings
OneBodyG	Output components for one-body Green's functions $\langle c_{i\sigma}^\dagger c_{j\sigma} \rangle$
TwoBodyG	Output components for two-body Green's functions $\langle c_{i\sigma}^\dagger c_{j\sigma} c_{k\tau}^\dagger c_{l\tau} \rangle$
SingleExcitation	Operators for generating a single excited state
PairExcitation	Operators for generating a pair excited state
SpectrumVec	An input vector to calculate a restart vector
OneBodyTE	Time-dependent transfer integrals
TwoBodyTE	Time-dependent two-body interactions

### 4.2.2 CalcMod file

This file determines the parameters for the calculation method, model, and output mode. The file format is as follows.

CalcType	0
CalcModel	2
CalcEigenVec	0

#### File format

[string01] [int01]

#### Parameters

- [string01]

**Type :** String

**Description :** Select a word from keywords.

- [int01]

**Type :** Int

**Description :** A parameter that is correlated with a keyword.

#### Use rules

- After setting the keywords at [string 01], a half-width blank is needed for setting a parameter.
- Keywords can be set in random order.
- If the keywords or filenames are incorrect, the program is terminated.
- The keywords “CalcType” and “CalcModel” are essential.
- When a head of line is “#”, the line is skipped.

#### Keywords and parameters

The parameters correlated with the keywords are as follows.

- CalcType

**Type :** Int

**Description :** Select the method for calculation from the following list:

- 0: Lanczos method
- 1: mTPQ method
- 2: Full diagonalization method
- 3: LOBCG for the ground state
- 4: Time-evolution
- 5: cTPQ method

- CalcModel

**Type :** Int

**Description :** Select the model from the following list:

- 0: Fermion Hubbard model (canonical ensemble: conservation of particles or conservation of particles and the component of  $S_z$ )
- 1: Spin model (canonical ensemble: conservation of the component of  $S_z$ )
- 2: Kondo lattice model (canonical ensemble: conservation of particles, the component of  $S_z$ )
- 3: Fermion Hubbard model (grand canonical ensemble)
- 4: Spin model (grand canonical ensemble)
- 5: Kondo lattice model (grand canonical ensemble).

For the fermion Hubbard model, you can select the model under the conservation of the particles by setting NCond in the ModPara file. When you want to select the model under the conservation of particles and the component of  $S_z$ , set both NCond and 2Sz in the ModPara file.

- CalcEigenVec

**Type :** Int (default value: 0)

**Description :** Select the method to calculate the eigenvectors:

- 0: Lanczos+CG methods (when the convergence of eigenvectors is not sufficient for using the Lanczos method, the CG method is applied to calculate eigenvectors).
- 1: Lanczos method.

- InitialVecType

**Type :** Int (default value: 0)

**Description :** Select the type of an initial vector ( $v0$ ):

- 1: Real part ( $\text{Re}[v0]$ ) and imaginary part ( $\text{Im}[v0]$ ) of the initial vector are give as the normally distributed random numbers. Thus, the normalized initial vectors are uniformly distributed on the  $N_H$  dimensional super sphere ( $N_H$  is the dimension of the Hilbert space).
- 0: Complex type ( $\text{Re}[v0] \in [-1 : 1]$ ,  $\text{Im}[v0] \in [-1 : 1]$  ).
- 1: Real type ( $\text{Re}[v0] \in [-1 : 1]$ ,  $\text{Im}[v0] = 0$ ).

- OutputEigenVec

**Type :** Int (default value: 0)

**Description :** Select the mode of outputting an eigenvector:

- 0: Not output an eigenvector
- 1: Output an eigenvector.

- InputEigenVec

**Type :** Int (default value: 0)

**Description :** Select the mode of inputting an eigenvector:

- 0: Not input an eigenvector
- 1: Input an eigenvector.

- ReStart

**Type :** Int (default value: 0)

**Description :** Select the mode of inputting a restart vector:

- 0: Not restart calculation
- 1: Output a restart vector
- 2: Input a restart vector and output a new restart vector
- 3: Input a restart vector.

- CalcSpec

**Type :** Int (default value: 0)

**Description :** Select the mode of calculating dynamical Green's functions:

- 0: Not calculate dynamical Green's functions
- 1: (not restart) Input an initial vector and files for generating single excited or pair excited states
- 2: Input components of triangular diagonal matrix
- 3: Output both components of triangular diagonal matrix and a restart vector
- 4: Input both components of triangular diagonal matrix and a restart vector
- 5: Input and output both components of triangular diagonal matrix and a restart vector.

- OutputHam

**Type :** Int (default value: 0)

**Description :** Full Diag)Select the mode of outputting Hamiltonian:

- 0: not output Hamiltonian.
- 1: output Hamiltonian.

- InputHam

**Type :** Int (default value: 0)

**Description :** (Full Diag)Select the mode of inputting Hamiltonian:

- 0: not input Hamiltonian.
- 1: input Hamiltonian.

- OutputExcitedVec

**Type :** Int (default value: 0)

**Description :** Select the mode of outputting an excited vector:

0: Not output an eigenvector

1: Output an eigenvector.

- Scalapack

**Type :** Int (default value: 0)

**Description :** (Full Diag)Select to use ScaLAPACK library for full diagonalization:

0: not to use ScaLAPACK.

1: use ScaLAPACK.

- NGPU

**Type :** Int (default value: 2)

**Description :** (Full Diag)Select the number of GPU devices for full diagonalization:

H $\Phi$  does not support to use GPU devices at multi-nodes.

### 4.2.3 ModPara file

This file determines the parameters for calculation. The file format is as follows.

```
-----  
Model_Parameters    0  
-----  
VMC_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
```

#### File format

- Lines 1-4: Header
- Line 6: [string01] [string02]
- Lines 7-8: Header
- Lines 9- : [string01] [int01].

#### Parameters

- [string01]  
**Type :** String  
**Description :** Select a word from keywords.
- [string02]  
**Type :** String (a blank parameter is not allowed)  
**Description :** Set a header for output files.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** A parameter that is correlated with a keyword.



## Use rules

- From Line 9: After setting keywords at [string 01], a half-width blank is needed for setting a parameter
- All the parameters are needed and the order for the parameters is fixed

## Keywords and parameters

In the following, common parameters and parameters for each method are shown.

### Common parameters

- CDataFileHead

**Type :** String (a blank parameter is not allowed)

**Description :** A header for output files. For example, the output filename for one-body Green's function becomes "xxx\_Lanczos\_cisajs.dat" (xxx are the characters set by CDataFileHead).

- Nsite

**Type :** Int (positive integer)

**Description :** The number of sites.

- Ncond

**Type :** Int (positive integer)

**Description :** The number of conduction electrons (not used in grand canonical ensemble).

- 2Sz

**Type :** Int (positive integer)

**Description :** The total value of  $2S_z$  (not used in grand canonical ensemble). For conservation of  $S_z$  in the case of CalcModel = 0 (fermion Hubbard model) or 2 (Kondo lattice model), set Ncond.

- initial\_iv

**Type :** Int

**Description :** An initial vector is specified with this parameter.

- Lanczos method

- \* For canonical ensemble and  $\text{initial\_iv} \geq 0$

The non-zero components of an initial vector are specified with this parameter.

- \* For grand canonical ensemble or  $\text{initial\_iv} < 0$

The seed of the random generator is given by this parameter and the random vector is used as the initial vector.

- TPQ method

The seed of the random generator is given by this parameter and the random vector is used as the initial vector.

See [Algorithm](#) for details of setting an initial vector.

- CalcHS

**Type :** Int (positive integer)

**Description :** If CalcHS=1, an efficient algorithm for generating the restricted Hilbert space with the specified quantum number is used (Details of algorithm is shown in <http://www.pasums.issp.u-tokyo.ac.jp/wp-content/themes/HPhi/media/develop/tips.pdf> [in Japanese]). Default value is 1 and the efficient algorithm is used.

## Lanczos method

- Lanczos\_max

**Type :** Int (positive integer)

**Description :** The maximum number of Lanczos steps in the calculation. When the convergence within the specified accuracy is satisfied, the calculation is completed before a step reaches Lanczos\_max. In the case of restart calculation, Lanczos\_max must be larger than that of the previous calculation.

- exct

**Type :** Int (positive integer)

**Description :** An integer for setting the number of eigenvectors obtained from the ground energy by the Lanczos method.

- LanczosEps

**Type :** Int (positive integer)

**Description :** An integer for judging the convergence of the Lanczos method. The convergence is determined by whether the condition is satisfied that the relative error between an eigenvalue and an eigenvalue at the Lanczos step of the one step before is less than  $10^{-\text{LanczosEps}}$ .

- LanczosTarget

**Type :** Int (positive integer)

**Description :** An integer giving the target of the eigenvalue for judging the convergence of the Lanczos method. For example, the target becomes a ground state when LanczosTarget is equal to one, and a first excited state when LanczosTarget is equal to two.

## CG method

- exct

**Type :** Int (positive integer)

**Description :** The number of eigenvectors is specified.

- Lanczos\_max

**Type :** Int (positive integer)

**Description :** The maximum number of iteration steps in the calculation. When the convergence within the specified accuracy is satisfied, the calculation is completed before a step reaches Lanczos\_max. In the case of restart calculation, Lanczos\_max must be larger than that of the previous calculation.

- LanczosEps

**Type :** Int (positive integer)

**Description :** For method="CG", the calculation finishes when the 2-norm of the residual vector becomes smaller than  $10^{-\text{LanczosEps}/2}$ .

## TPQ (mTPQ/cTPQ) method

- Lanczos\_max

**Type :** Int (positive integer)

**Description :** The total number of TPQ steps is specified with this parameter. In the case of restart calculation, Lanczos\_max must be larger than that of the previous calculation.

- LargeValue

**Type :** Double

**Description :** An integer giving  $l$  of  $l - \mathcal{H}/N_s$  used in the mTPQ method. In the cTPQ method,  $l$  is used as the width of the imaginary-time evolution, i.e.,  $\Delta\tau = 1/l$ . See [Algorithm](#) for the details of the cTPQ method.

- NumAve

**Type :** Int

**Description :** An integer giving the number of independent runs for the TPQ method.

- ExpecInterval

**Type :** Int

**Description :** An integer giving the interval steps of calculating the correlation functions in the TPQ method.

**Note:** A small interval increases the time cost of calculations.

- ExpandCoef

**Type :** Int (positive integer)

**Description :** An integer giving the expansion order  $n_{\max}$  for cTPQ method;

$$U(\Delta\tau) = \sum_{n=0}^{n_{\max}} \frac{1}{n!} \left( -\frac{\Delta\tau}{2} \mathcal{H} \right)^n. \quad (4.4)$$

See [Algorithm](#) for the details of the cTPQ method.

## Calculating dynamical Green's functions

- OmegaOrg

**Type :** Complex

**Description :** The center value of the frequency. Specify the real and imaginary parts in that order separated by a space, and if there is no imaginary part, the real part of the frequency is only given.

- OmegaIm

**Type :** Double

**Description :** The imaginary part of the frequency. When OmegaOrg is defined in a modpara file, OmegaIm is added to the imaginary value of OmegaOrg.

- OmegaMin

**Type :** Complex

**Description :** The lower limit of the frequency from OmegaOrg. Specify the real and imaginary parts in that order separated by a space, and if there is no imaginary part, the real part of the frequency is only given.

- OmegaMax

**Type :** Complex

**Description :** The upper limit of the frequency from OmegaOrg. Specify the real and imaginary parts in that order separated by a space, and if there is no imaginary part, a real part of the frequency is only given.

- NOmega

**Type :** Int

**Description :** The integer for defining the step size of the frequency  $\Delta\omega = (\text{OmegaMax} - \text{OmegaMin})/N_\omega$ . The frequency is given by  $z_n = \text{OmegaOrg} + \text{OmegaMin} + \Delta\omega \times n$ .

## Real time evolution method

- Lanczos\_max

**Type :** Int (positive integer)

**Description :** The total number of real time evolution steps is specified with this parameter. In the case of restart calculation, Lanczos\_max must be larger than that of the previous calculation.

- ExpandCoef

**Type :** Int (positive integer)

**Description :** An integer giving the expansion order  $n$  for real-time evolution method;

$$\exp(-i\mathcal{H}\Delta t) = \sum_{i=0}^N \frac{1}{n!} (-i\mathcal{H}\Delta t)^n. \quad (4.5)$$

- ExpecInterval

**Type :** Int (positive integer)

**Description :** An integer giving the interval steps of calculating the correlation functions.

**Note:** A small interval increases the time cost of calculations.

- OutputInterval

**Type :** Int (positive integer)

**Description :** An integer giving the interval steps of output the wave function.

The wave vector is output when OutputEigenVec=1 in CalcMod file.

## 4.2.4 LocSpin file

This file determines sites with localized spins. The file format is as follows.

```
=====
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
```

### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03].

### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of localized spins. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of localized spins.
- [int02]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}] < N_{\text{site}}$ ).
- [int03]  
**Type :** Int (a blank parameter is not allowed)

**Description :** An integer for selecting an electron state whether the electron state is a localized spin or an itinerant electron state:

0: Itinerant electron state  $n > 0$ : Localized spin state with  $2S = n$ .

### Use rules

- Headers cannot be omitted.
- A program is terminated when [int01] is different from the total number of localized spins indicated by [int03].
- A program is terminated, when [int02] is different from the total number of sites.
- A program is terminated under the condition [int02] < 0 or Nsite ≤ [int02].

## 4.2.5 Trans file

This file determines the values of the transfer integrals  $t_{ij\sigma_1\sigma_2}$ ,

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

An example of the file format is as follows.

```
=====
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
...

```

### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [int05] [double01] [double02].

### Parameters

- [string01]  
**Type** : String (a blank parameter is not allowed)  
**Description** : A keyword for the total number of transfer integrals. You can freely give a name to the keyword.
- [int01]  
**Type** : Int (a blank parameter is not allowed)  
**Description** : An integer giving the total number of transfer integrals.
- [int02], [int04]  
**Type** : Int (a blank parameter is not allowed)  
**Description** : An integer giving a site index ( $0 \leq [\text{int02}], [\text{int04}] < \text{Nsite}$ ).

- [int03], [int05]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

0: Up-spin

1: Down-spin.

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for a real part of  $t_{ij\sigma_1\sigma_2}$ .

- [double02]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for an imaginary part of  $t_{ij\sigma_1\sigma_2}$ .

### Use rules

- Headers cannot be omitted.
- Since the Hamiltonian must be Hermitian, the relation  $t_{ij\sigma_1\sigma_2} = t_{ji\sigma_2\sigma_1}^\dagger$  must be satisfied. A program is terminated when this relation is broken.
- A program is terminated when the components of the on-site interactions are double counted.
- A program is terminated when [int01] is different from the total number of transfer integrals defined in this file.
- A program is terminated when [int02]-[int05] are outside the range of the defined values.



### 4.2.6 InterAll file

This file determines the values of generalized two-body interactions integrals  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ ,

$$\mathcal{H}+ = \sum_{i,j,k,l} \sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}. \quad (4.7)$$

For the off-diagonal component, be sure to provide a pair of i1 sigma1 i2 sigma2 i3 sigma3 i4 sigma4 and i4 sigma4 i3 sigma3 i2 sigma2 i1 sigma1. For spin, the conditions  $i = j$  and  $k = l$  must be satisfied. An example of the file format is as follows.

```
=====
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
...
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [int05] [int06] [int07] [int08] [int09] [double01] [double02].

## Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of generalized two-body interactions. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of generalized two-body interactions.

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

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int04}], [\text{int06}], [\text{int08}] < \text{Nsite}$ ).

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

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

0: Up-spin,

1: Down-spin.

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for a real part of  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ .

- [double02]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for an imaginary part of  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ .

## Use rules

- Headers cannot be omitted.
- Since the Hamiltonian must be Hermitian, the relation  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} = I_{lkji\sigma_4\sigma_3\sigma_2\sigma_1}^\dagger$  must be satisfied. A program is terminated when this relation is broken. It is noted that the term of the Hermitian conjugate for  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$  should be inputted as  $I_{lkji\sigma_4\sigma_3\sigma_2\sigma_1} c_{l\sigma_4}^\dagger c_{k\sigma_3} c_{j\sigma_2}^\dagger c_{i\sigma_1}$ .
- A program is terminated when the conditions  $i = j$  and  $k = l$  are not satisfied for the spin model.
- A program is terminated when the components of the on-site interactions are double counted.
- A program is terminated when [int01] is different from the total number of generalized two-body interactions defined in this file.
- A program is terminated when [int02]-[int09] are outside the range of the defined values.

### 4.2.7 CoulombIntra file

This file determines the values of the on-site interactions  $U_i$  (for  $S = 1/2$  system),

$$\mathcal{H}+ = \sum_i U_i n_{i\uparrow} n_{i\downarrow}. \quad (4.8)$$

An example of the file format is as follows.

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

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [double01].

#### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of on-site interactions. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of on-site interactions.
- [int02]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}] < \text{Nsite}$ ).
- [double01]  
**Type :** Double (a blank parameter is not allowed)  
**Description :** A value for  $U_i$ .

### **Use rules**

- Headers cannot be omitted.
- A program is terminated when the components of on-site interactions are double counted.
- A program is terminated when [int01] is different from the total number of on-site interactions defined in this file.
- A program is terminated when [int02] is outside the range of the defined values.

### 4.2.8 CoulombInter file

This file determines the values of off-site interactions  $V_{ij}$  (for  $S = 1/2$  system),

$$\mathcal{H}+ = \sum_{i,j} V_{ij} n_i n_j. \quad (4.9)$$

An example of the file format is as follows.

```
=====
NCoulombInter 6
=====
=====CoulombInter =====
=====
  0      1  1.0000
  1      2  1.0000
  2      3  1.0000
  3      4  1.0000
  4      5  1.0000
  5      0  1.0000
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01].

#### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of off-site interactions. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of off-site interactions.
- [int02], [int03]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).
- [double01]  
**Type :** Double (a blank parameter is not allowed)  
**Description :** A value for  $V_{ij}$ .

### Use rules

- Headers cannot be omitted.
- A program is terminated when the components of off-site interactions are double counted.
- A program is terminated when [int01] is different from the total number of off-site interactions defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.

## 4.2.9 Hund file

This file determines the values of Hund couplings  $J_{ij}^{\text{Hund}}$  (for  $S = 1/2$  system),

$$\mathcal{H}+ = - \sum_{i,j} J_{ij}^{\text{Hund}} (n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}). \quad (4.10)$$

An example of the file format is as follows.

```
=====
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
```

### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01].

### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of Hund couplings. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of Hund couplings.
- [int02], [int03]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).
- [double01]  
**Type :** Double (a blank parameter is not allowed)  
**Description :** A value for  $J_{ij}^{\text{Hund}}$ .

### **Use rules**

- Headers cannot be omitted.
- A program is terminated when the components of the Hund couplings are double counted.
- A program is terminated when [int01] is different from the total number of Hund couplings defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.



### 4.2.10 PairHop file

This file determines the values of PairHop couplings  $J_{ij}^{\text{Pair}}$  (for  $S = 1/2$  system),

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

An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01].

#### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of PairHop couplings. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of PairHop couplings.
- [int02], [int03]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).
- [double01]  
**Type :** Double (a blank parameter is not allowed)  
**Description :** A value for  $J_{ij}^{\text{Pair}}$ .

### **Use rules**

- Headers cannot be omitted.
- A program is terminated when [int01] is different from the total number of the PairHop couplings defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.

### 4.2.11 Exchange file

This file determines the values of Exchange couplings  $J_{ij}^{\text{Ex}}$  (for  $S = 1/2$  system). For the fermion electronic system, the exchange terms are given as

$$\mathcal{H}+ = \sum_{i,j} J_{ij}^{\text{Ex}} (c_{i\uparrow}^\dagger c_{j\uparrow} c_{j\downarrow}^\dagger c_{i\downarrow} + c_{i\downarrow}^\dagger c_{j\downarrow} c_{j\uparrow}^\dagger c_{i\uparrow}), \quad (4.12)$$

while for the spin system, they are given as

$$\mathcal{H}+ = \sum_{i,j} J_{ij}^{\text{Ex}} (S_i^+ S_j^- + S_i^- S_j^+). \quad (4.13)$$

We note that  $(S_i^+ S_j^- + S_i^- S_j^+)$  in the spin system is written by the operators for electrons as  $-(c_{i\uparrow}^\dagger c_{j\uparrow} c_{j\downarrow}^\dagger c_{i\downarrow} + c_{i\downarrow}^\dagger c_{j\downarrow} c_{j\uparrow}^\dagger c_{i\uparrow})$ . An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01].

#### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of Exchange couplings. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of Exchange couplings.
- [int02], [int03]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for  $J_{ij}^{\text{Ex}}$ .

### Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the exchange couplings are double counted.
- A program is terminated when [int01] is different from the total number of exchange couplings defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.

### 4.2.12 Ising file

This file determines the values of Ising interactions  $J_{ij}^z$  (for  $S = 1/2$  system). For the fermion electronic system, the Ising terms are given as

$$\mathcal{H}+ = \sum_{i,j} J_{ij}^z (n_{i\uparrow} - n_{i\downarrow})(n_{j\uparrow} - n_{j\downarrow}). \quad (4.14)$$

For the spin system, they are given as

$$\mathcal{H}+ = \sum_{i,j} J_{ij}^z S_i^z S_j^z. \quad (4.15)$$

An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01].

#### Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of Ising interactions. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of Ising interactions.

- [int02], [int03]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for  $J_{ij}^z$ .

### Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the Ising interactions are double counted.
- A program is terminated when [int01] is different from the total number of Ising interactions defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.

### 4.2.13 PairLift file

This file determines the values of PairLift couplings  $J_{ij}^{\text{PairLift}}$  (for  $S = 1/2$  system),

$$\mathcal{H}+ = \sum_{i,j} J_{ij}^{\text{PairLift}} (c_{i\uparrow}^\dagger c_{i\downarrow} c_{j\uparrow}^\dagger c_{j\downarrow} + c_{i\downarrow}^\dagger c_{i\uparrow} c_{j\downarrow}^\dagger c_{j\uparrow}). \quad (4.16)$$

An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [double01] .

#### Parameters

- [string01]  
**Type :** String (a blank parameter is not allowed)  
**Description :** A keyword for the total number of PairLift couplings. You can freely give a name to the keyword.
- [int01]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving the total number of PairLift couplings.
- [int02], [int03]  
**Type :** Int (a blank parameter is not allowed)  
**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int03}] < \text{Nsite}$ ).
- [double01]  
**Type :** Double (a blank parameter is not allowed)  
**Description :** A value for  $J_{ij}^{\text{PairLift}}$ .

### **Use rules**

- Headers cannot be omitted.
- A program is terminated when the components of the PairLift couplings are double counted.
- A program is terminated when [int01] is different from the total number of PairLift couplings defined in this file.
- A program is terminated when either [int02] or [int03] is outside the range of the defined values.



#### 4.2.14 OneBodyG file

This file determines the target components of the one-body Green's function  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ . An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [int05].

## Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of one-body Green's functions. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of one-body Green's functions.

- [int02], [int04]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int04}] < \text{Nsite}$ ).

- [int03], [int05]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

For Hubbard or Kondo system, the index can be selected from

0: Up-spin,

1: Down-spin.

For Spin system, the index can be selected from

$0, 1, \dots, 2S + 1$  (corresponding to  $-S - 0.5, -S + 0.5, \dots, S + 0.5$ ).

## Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the one-body Green's functions are double counted.
- A program is terminated when [int01] is different from the total number of one-body Green's functions defined in this file.
- A program is terminated when [int02]-[int05] are outside the range of the defined values.

### 4.2.15 TwoBodyG file

This file determines the target components of the two-body Green's function  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ . For the spin system, the conditions  $i = j$  and  $k = l$  must be satisfied. An example of the file format is as follows.

```
=====
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
  ...
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [int05] [int06] [int07] [int08] [int09].

## Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of two-body Green's functions. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of two-body Green's functions.

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

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int04}], [\text{int06}], [\text{int08}] < \text{Nsite}$ ).

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

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

For Hubbard or Kondo system, the index can be selected from

0: Up-spin,

1: Down-spin.

For Spin system, the index can be selected from

$0, 1, \dots, 2S + 1$  (corresponding to  $-S - 0.5, -S + 0.5, \dots, S + 0.5$ ).

## Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the two-body Green's functions are double counted.
- A program is terminated when the conditions  $i = j$  and  $k = l$  are not satisfied for the spin model.
- A program is terminated when [int01] is different from the total number of two-body Green's functions defined in this file.
- A program is terminated, when [int02]-[int09] are outside the range of the defined values.

## 4.2.16 SingleExcitation file

The operators to generate the single excited state  $c_{i\sigma_1}(c_{i\sigma_1}^\dagger)$  are defined. An example of the file format is as follows.

```
=====
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
```

### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [double01] [double02].

### Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of single excitation operators. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of total number of single excitation operators.

- [int02]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}] < \text{Nsite}$ ).

- [int03]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

For Hubbard or Kondo system, the index can be selected from

0: Up-spin,

1: Down-spin.

For Spin system, the index can be selected from

$0, 1, \dots, 2S + 1$  (corresponding to  $-S - 0.5, -S + 0.5, \dots, S + 0.5$ ).

- [int04]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a type of single excitation operators:

0:  $c_{i\sigma_1}$

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

- [double01], [double02]

**Type :** Double (a blank parameter is not allowed)

**Description :** [double01] gives the real part of  $c_{i\sigma_1}(c_{i\sigma_1}^\dagger)$ , while [double02] gives the imaginary part of  $c_{i\sigma_1}(c_{i\sigma_1}^\dagger)$ .

## Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the single excitation operators are double counted.
- A program is terminated when the conditions  $i = j$  and  $k = l$  are not satisfied for spin model.
- A program is terminated when [int01] is different from the total number of two-body Green's functions defined in this file.
- A program is terminated, when [int02]-[int04] are outside the range of the defined values.

### 4.2.17 PairExcitation file

The operators to generate the pair excited state  $c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2})$  are defined. The type of pair excitation operators ( $c_{i\sigma_1} c_{j\sigma_2}^\dagger$  or  $c_{i\sigma_1}^\dagger c_{j\sigma_2}$ ) must be same in the input file. In the  $S_z$  conserved system,  $\sigma_1$  must be equal to  $\sigma_2$ . An example of the file format is as follows.

```
=====
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
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header
- Lines 6-: [int02] [int03] [int04] [int05] [int06] [double01] [double02].

#### Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of the pair excitation operators. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of pair excitation operators.

- [int02], [int04]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int02}], [\text{int04}] < \text{Nsite}$ ).

- [int03], [int05]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

For Hubbard or Kondo system, the index can be selected from

0: Up-spin,

1: Down-spin.

For Spin system, the index can be selected from

$0, 1, \dots, 2S + 1$  (corresponding to  $-S - 0.5, -S + 0.5, \dots, S + 0.5$ ).

- [int06]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a type of pair excitation operators:

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

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

- [double01], [double02]

**Type :** Double (a blank parameter is not allowed)

**Description :** [double01] gives the real part of  $c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2})$ , while [double02] gives the imaginary part of  $c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2})$ .

## Use rules

- Headers cannot be omitted.
- A program is terminated when the components of the pair excitation operators are double counted.
- A program is terminated when the conditions  $i = j$  and  $k = l$  are not satisfied for the spin model.
- A program is terminated when [int01] is different from the total number of two-body Green's functions defined in this file.
- A program is terminated when [int02]-[int06] are outside the range of the defined values.



### 4.2.18 SpectrumVec File

The header of the input file for the initial vector to calculate the dynamical Green's functions is defined. The file name and the file format (binary type) are as follows.

#### File name

- `##_rank_$.dat`

## is the name of the head indicated by the key word `SpectrumVec` in the `CalcMod` file, and \$ is the number of the rank.

#### File format

- Line 1: [int01]
- Line 2: [int02]
- Lines 3 -: [double01] [double02].

#### Parameters

- [int01]

**Type :** Int

**Description :** The total number of the targets of the Hilbert spaces.

- [int02]

**Type :** Int

**Description :** The number of Lanczos or TPQ steps.

- [double01], [double02]

**Type :** double

**Description :** The coefficient value of the input vector.

[double01] is a real part and [double02] is an imaginary part.

### 4.2.19 OneBodyTE File

This file determines the values of the transfer integrals  $t_{ij\sigma_1\sigma_2}(t)$  at each time  $t$ ,

$$\mathcal{H}(t) = - \sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2}(t) c_{i\sigma_1}^\dagger c_{j\sigma_2}. \quad (4.17)$$

An example of the file format is as follows.

```
=====
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...)
```

#### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header

From line 6, time  $t$  and total number of transfer integrals  $N(t)$  are first defined and next the transfer integrals  $t_{ij\sigma_1\sigma_2}(t)$  are defined.

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

#### Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of transfer integrals. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total time steps defined in this file.

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** Time  $t$ .

- [int02]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of transfer integrals at time  $t$ .

- [int03], [int05]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int03}], [\text{int05}] < \text{Nsite}$ ).

- [int04], [int06]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

0: Up-spin

1: Down-spin.

- [double02], [double03]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for real and imaginary part of  $t_{ij\sigma_1\sigma_2}(t)$  at time  $t$  is defined [double02] and [double03], respectively.

### Use rules

- Headers cannot be omitted.
- A program is terminated when LanczosStep defined in ModPara is greater than [int02].
- A program is terminated when [int03]-[int06] are outside the range of the defined values.

## 4.2.20 TwoBodyTE File

This file determines the values of the two-body interactions  $I_{i\sigma_1 j\sigma_2 k\sigma_3 l\sigma_4}(t) c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}$  at each time  $t$ ,

$$\mathcal{H}(t) = \sum_{ijkl} \sum_{\sigma_1 \sigma_2 \sigma_3 \sigma_4} I_{i\sigma_1 j\sigma_2 k\sigma_3 l\sigma_4}(t) c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4}. \quad (4.18)$$

An example of the file format is as follows.

```
=====
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...)
```

### File format

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3-5: Header

From line 6, time  $t$  and total number of transfer integrals  $N(t)$  are first defined and next the two-body interactions  $I_{i\sigma_1 j\sigma_2 k\sigma_3 l\sigma_4}(t)$  are defined.

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

### Parameters

- [string01]

**Type :** String (a blank parameter is not allowed)

**Description :** A keyword for the total number of generalized two-body interactions. You can freely give a name to the keyword.

- [int01]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total time steps defined in this file.

- [double01]

**Type :** Double (a blank parameter is not allowed)

**Description :** Time  $t$ .

- [int02]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving the total number of two-body interactions at time  $t$ .

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

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a site index ( $0 \leq [\text{int03}], [\text{int05}], [\text{int07}], [\text{int09}] < \text{Nsite}$ ).

- [int04], [int06], [int08], [int10]

**Type :** Int (a blank parameter is not allowed)

**Description :** An integer giving a spin index:

0: Up-spin

1: Down-spin.

- [double02], [double03]

**Type :** Double (a blank parameter is not allowed)

**Description :** A value for real and imaginary part of  $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}(t)$  at time  $t$  is defined [double02] and [double03], respectively.

### Use rules

- Headers cannot be omitted.
- A program is terminated when LanczosStep defined in ModPara is greater than [int02].
- A program is terminated when [int03]-[int10] are outside the range of the defined values.

## 4.3 Output files

In this section, the details of the output files of the expert mode are explained.

### 4.3.1 CHECK\_Chemi.dat

This file is outputted to check the input of chemical potential  $\mu_{i\sigma}$ ,

$$\sum_{i,\sigma} \mu_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma}. \quad (4.19)$$

An example of the file format is as follows.

```
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
...
```

#### File format

- i=[int01] spin=[int02] isite1=[int03] tmp\_V=[double01]

#### Parameters

- [int01]

**Type :** Int

**Description :** The counted number of inputting terms.

- [int02]

**Type :** Int

**Description :** An integer for showing the spin index of  $\mu_{i\sigma}$ :

0: Up-spin,

1: Down-spin.

- [int03]

**Type :** Int

**Description :** An integer for showing the site index of  $\mu_{i\sigma}$ .

- [double01]

**Type :** Double

**Description :** A value for  $\mu_{i\sigma}$ .

### 4.3.2 CHECK\_InterAll.dat

This file is outputted to check the input of the diagonal components of general two-body interactions,

$$\sum_{i,j,\sigma_1,\sigma_2} I_{ijj\sigma_1\sigma_1\sigma_2\sigma_2} c_{i\sigma_1}^\dagger c_{i\sigma_1} c_{i\sigma_2}^\dagger c_{i\sigma_2}. \quad (4.20)$$

An example of the file format is as follows.

```
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
...
```

#### File format

- i=[int01] isite1=[int02] A\_spin=[int03] isite2=[int04] B\_spin=[int05] tmp\_V=[double01]

#### Parameters

- [int01]

**Type :** Int

**Description :** The counted number of inputting terms.

- [int02], [int04]

**Type :** Int

**Description :** An integer for showing the site index of  $I_{ijj\sigma_1\sigma_1\sigma_2\sigma_2}$ .  
[int02] and [int04] correspond to  $i$  and  $j$ , respectively.

- [int03], [int05]

**Type :** Int

**Description :** An integer for showing the spin index of  $I_{ijj\sigma_1\sigma_1\sigma_2\sigma_2}$ :  
0: Up-spin  
1: Down-spin.  
[int03] and [int05] correspond to  $\sigma_1$  and  $\sigma_2$ , respectively.

- [double01]

**Type :** Double

**Description :** A value for  $I_{ijj\sigma_1\sigma_1\sigma_2\sigma_2}$ .

### 4.3.3 CHECK\_CoulombIntra.dat

This file is outputted to check the input of the on-site interactions  $U_i$ ,

$$\sum_i U_i n_{i\uparrow} n_{i\downarrow}. \quad (4.21)$$

An example of the file format is as follows.

```
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
```

#### File format

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

#### Parameters

- `[int01]`  
**Type :** Int  
**Description :** The counted number of inputting terms.
- `[int02]`  
**Type :** Int  
**Description :** An integer for showing the site index of  $U_i$ .
- `[double01]`  
**Type :** Double  
**Description :** A value for  $U_i$ .



### 4.3.4 CHECK\_Hund.dat

This file is outputted to check the input of the Hund couplings  $J_{ij}^{\text{Hund}}$ ,

$$- \sum_{i,j} J_{ij}^{\text{Hund}} (n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}). \quad (4.22)$$

An example of the file format is as follows.

```
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
```

#### File format

- i=[int01] isite1=[int02] isite2=[int03] tmp\_V=[double01]

#### Parameters

- [int01]

**Type :** Int

**Description :** The counted number of inputting terms.

- [int02], [int03]

**Type :** Int

**Description :** An integer for showing the site index of  $J_{ij}^{\text{Hund}}$ .  
[int02] and [int03] correspond to  $i$  and  $j$ , respectively.

- [double01]

**Type :** Double

**Description :** A value for  $J_{ij}^{\text{Hund}}$ .

### 4.3.5 CHECK\_INTER\_U.dat

This file is outputted to check the input of the diagonal components of the off-site interactions  $V_{ij}$ ,

$$\sum_{i,j} V_{ij} n_i n_j \quad (4.23)$$

An example of the file format is as follows.

```
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
```

#### File format

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

#### Parameters

- `[int01]`

**Type :** Int

**Description :** The counted number of inputting terms.

- `[int02], [int03]`

**Type :** Int

**Description :** An integer giving the site index of  $V_{ij}$ .

`[int02]` and `[int03]` correspond to  $i$  and  $j$ , respectively.

- `[double01]`

**Type :** Double

**Description :** A value for  $V_{ij}$ .

### 4.3.6 CHECK\_Memory.dat

This file shows the size of the memory used in the calculation. An example of the file format is as follows.

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

#### File format

- MAX DIMENSION idim\_max=[int01]
- REQUIRED MEMORY max\_mem =[double01] GB

#### Parameters

- [int01]

**Type :** Int

**Description :** An integer to show the total numbers of the Hilbert space under a calculation.

- [double01]

**Type :** Double

**Description :** The size of memory to store the Hilbert space in a calculation (GB unit).

### 4.3.7 WarningOnTransfer.dat

This file shows the double counted components of transfer integrals. An example of the file format is as follows.

```
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
```

#### File format

- Double countings in transfers: i=[int01] j=[int02] spni [int03] spnj [int04]

#### Parameters

- [int01], [int02]

**Type :** Int

**Description :** The integer of the site number where the transfer integrals are double counted.

- [int03], [int04]

**Type :** Int

**Description :** The integer of the spin index of a transfer integral:

0: Up-spin

1: Down-spin.

### 4.3.8 CalcTimer.dat

The name of the calculation process, the process number, and the calculation process time are outputted in order at each line in the CalcTime.dat file. An example of the file format for the TPQ method is as follows.

```

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
=====

```

### 4.3.9 TimeKeeper.dat

This file is outputted to show the calculation process information. An example of the file format for the Lanczos method is as follows.

```
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_cisajackalt begins: Wed Sep 16 22:58:50 2015
```

#### File name

- `##_TimeKeeper.dat`

## indicates a header defined by [string02] in a ModPara file.

#### 4.3.10 sz\_TimeKeeper.dat

This file is outputted to show the process information to obtain the Hilbert space needed for the calculation. An example of the file format is as follows.

```
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
```

##### File name

- ##\_sz\_TimeKeeper.dat

## indicates a header defined by [string02] in a ModPara file.

### 4.3.11 Time\_CG\_EigenVector.dat

(For the Lanczos method) The process for calculating the eigenvector by the CG method is outputted. An example of the file format is as follows.

```
allocate succeed !!!
b[4341]=1.000000 bnorm== 1.000000
i_itr=0 itr=5 0.0411202543 0.0000100000
...
i_itr=0 itr=155 0.00000000058 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 invl:i_itr=1 itr=120
t_itr=275 0.000000
```

#### File name

- ##\_Time\_CG\_EigenVector.dat

## indicates a header defined by [string02] in a ModPara file.



### 4.3.12 energy.dat

(For the Lanczos method) The values of the energy, doublon, and  $\langle S_z \rangle$  calculated by using the eigenvector obtained by the Lanczos or CG method are outputted. An example of the file format is as follows.

For method="Lanczos"

```
Energy  -7.1043675920
Doublon  0.4164356536
Sz  0.000000000000
```

For method="CG"

```
State 0
  Energy  -7.1043675920
  Doublon  0.4164356536
  Sz  0.000000000000

State 1
:
```

#### File name

- ##\_energy.dat

## indicates a header defined by [string02] in a ModPara file.

#### File format

- Line 1: Energy [double01]
- Line 2: Doublon [double02]
- Line 3: Sz [double02].

#### Parameters

- [double01]

**Type :** Double

**Description :** The value of the energy calculated by the eigenvector obtained by the Lanczos or CG method.

- [double02]

**Type :** Double

**Description :** The value of the doublon calculated by the eigenvector obtained by the Lanczos or CG method,  $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$  ( $N_s$  is the total number of sites).

- [double03]

**Type :** Double

**Description :** The value of  $S_z$  calculated by the eigenvector obtained by the Lanczos or CG method.

### 4.3.13 Lanczos\_Step.dat

(For the Lanczos method) This file is outputted to show the process information for calculating the eigenvector by Lanczos method. An example of the file format for the Lanczos method is shown as follows.

For method="Lanczos"

```
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
```

For method="CG"

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

#### File name

- ##\_Lanczos\_Step.dat

## indicates a header defined by [string02] in a ModPara file.

#### File format

- For ``method="Lanczos"``

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

- For method="CG"

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

## Parameters

- [int01]

**Type :** Int

**Description :** The iteration number of the Lanczos and LOBCG method.

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

**Type :** Double

**Description :** Eigenvalues computed with the Lanczos or LOBCG method (ascending order). Four and `exct` eigenvalues are printed for the Lanczos and LOBCG method, respectively (in the above case, `exct=3`). While the degenerate eigenstates are printed as a single state in the Lanczos method, they are printed separately in LOBCG method. In the above case, we can find there is a degeneracy in the first excited state.

- [double-a]

**Type :** Double

**Description :** (Only for the Lanczos method) The eigenvalue used for the convergence check. It was specified by `LanczosTarget` (in the above case, `LanczosTarget=3`).

- [double-b]

**Type :** Double

**Description :** (Only for the Lanczos method) The maximum eigenvalue divided by the number of sites. It is the lower limit of `LargeValue` in TPQ method.

- [double-c]

**Type :** Double

**Description :** (Only for LOBCG method) The maximum of the 2-norm of each residual vector. It is used for the convergence check.

- [double-d]

**Type :** Double

**Description :** (Only for LOBCG method) The convergence threshold. This is obtained by the value specified with `LanczosEps` times the absolute value of the energy of the ground state.

### 4.3.14 Time\_TPQ\_Step.dat

(For the TPQ method) This file is outputted to show the time for starting the calculation of the TPQ method at each seed and step. In the restart calculation, the values are added to the previous file. An example is as follows.

```
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
```

#### File name

- `##_TPQ_Step.dat`

## indicates a header defined by [string02] in a ModPara file.

#### File format

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

#### Parameters

- [int01]

**Type :** Int

**Description :** The seed number in the calculation of the TPQ method.

- [int02]

**Type :** Int

**Description :** The step number in the calculation of the TPQ method.

- [string01]

**Type :** String

**Description :** The time for starting the calculation of the TPQ method at each seed and step.

### 4.3.15 Norm\_rand.dat

(For the TPQ method) This file is outputted to show the calculation process information for the TPQ method. In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# 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
```

#### File name

- Norm\_rand???.dat

?? indicates the number of runs in the calculation of the TPQ method.

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [int01].

#### Parameters

- [double01]

**Type :** Double

**Description :** Inverse temperature  $1/k_B T$ .

- [double02]

**Type :** Double

**Description :** The norm of a wave function before normalization given by  $\langle \tilde{\psi}_k | \tilde{\psi}_k \rangle$ , where  $|\tilde{\psi}_k\rangle \equiv (I - \mathcal{H}/N_s)|\psi_{k=1}\rangle$ .

- [double03]

**Type :** Double

**Description :** The norm of an initial wave function before normalization given by  $\langle \tilde{\psi}_0 | \tilde{\psi}_0 \rangle$ , where  $|\tilde{\psi}_0\rangle$  is an initial random vector.

- [int01]

**Type :** Int

**Description :** The number of operations of  $(I - \mathcal{H}/N_s)$  for an initial wave function, where  $l$  is LargeValue defined in a ModPara file and  $N_s$  is the total number of sites.

### 4.3.16 SS\_rand.dat

(For the TPQ method) This file is outputted to show the calculation results for the TPQ method. In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# 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
```

#### File name

- SS\_rand??.dat

?? indicates the number of runs in the calculation of the TPQ method.

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [double04] [double05] [int01].

#### Parameters

- [double01]

**Type :** Double

**Description :** Inverse temperature  $1/k_{\text{B}}T(k_{\text{B}} = 1)$ .

- [double02]

**Type :** Double

**Description :** The expected value of the Hamiltonian (energy)  $\langle \mathcal{H} \rangle$ .

- [double03]

**Type :** Double

**Description :** The expected value of the square of the Hamiltonian  $\langle \mathcal{H}^2 \rangle$ .

- [double03]

**Type :** Double

**Description :** The expected value of the doublon,  $\sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$ .

- [double05]

**Type :** Double

**Description :** The total number of particles  $\langle \hat{n} \rangle$ .

- [int01]

**Type :** Int

**Description :** The number of operations of  $(l - \mathcal{H}/N_s)$  for an initial wave function, where  $l$  is LargeValue defined in a ModPara file and  $N_s$  is the total number of sites.

### 4.3.17 Flct\_rand.dat

(For the TPQ method) This file is outputted to show the calculation results of the fluctuation of the particle number, doublon, and  $S_z$  for the TPQ method. In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# 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 -0.00000000000167368 0.2500 1998
136.04474 12.00 144.00 0.00 0.00 -0.00000000000165344 0.2500 1999
```

#### File name

- Flct\_rand??.dat

?? indicates the number of runs in the calculation of the TPQ method.

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [double04] [double05] [double06] [double07] [int01].

#### Parameters

- [double01]  
**Type :** Double  
**Description :** Inverse temperature  $1/k_B T$ .
- [double02]  
**Type :** Double  
**Description :** A total particle number  $\sum_i \langle n_i \rangle$ .
- [double03]  
**Type :** Double  
**Description :** The expected value of the square of the particle number  $\langle (\sum_i n_i)^2 \rangle$ .
- [double04]  
**Type :** Double  
**Description :** The expected value of doublon  $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$  ( $N_s$  is the total number of sites).
- [double05]  
**Type :** Double  
**Description :** The expected value of the square of doublon  $\frac{1}{N_s} \langle (\sum_i n_{i\uparrow} n_{i\downarrow})^2 \rangle$  ( $N_s$  is the total number of sites).



- [double06]

**Type :** Double

**Description :** The expected value of  $S_z$ ,  $\frac{1}{N_s} \sum_i \langle S_i^z \rangle$  ( $N_s$  is the total number of sites).

- [double07]

**Type :** Double

**Description :** The expected value of the square of  $S_z$ ,  $\frac{1}{N_s} \langle (\sum_i S_i^z)^2 \rangle$  ( $N_s$  is the total number of sites).

- [int01]

**Type :** Int

**Description :** The number of operations of  $(l - \mathcal{H}/N_s)$  for an initial wave function, where  $l$  is LargeValue defined in a ModPara file and  $N_s$  is the total number of sites.

### 4.3.18 Time\_TE\_Step.dat

(For the real time evolution method) This file is outputted to show the time for starting the calculation of the real time evolution method at each step. In the restart calculation, the values are added to the previous file. An example is as follows.

```
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
```

#### File name

- ##\_TE\_Step.dat

## indicates a header defined by [string02] in a ModPara file.

#### File format

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

#### Parameters

- [int01]

**Type :** Int

**Description :** The step number in the calculation.

- [string01]

**Type :** String

**Description :** The time for starting the calculation at each step.

### 4.3.19 Norm.dat

(For the real time evolution method) This file is outputted to show the calculation process information for the real time evolution method. In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# time, global_norm, global_1st_norm, step_i
0.0000000000000000 0.9999999999999994 0.0000000000000000 0
0.0100000000000000 1.0000233421898765 0.0000000000000000 1
0.0200000000000000 1.0000211100654208 0.0000000000000000 2
0.0300000000000000 1.0000182214014706 0.0000000000000000 3
0.0400000000000000 1.0000148460317946 0.0000000000000000 4
0.0500000000000000 1.0000111372562455 0.0000000000000000 5
0.0600000000000000 1.0000072252313270 0.0000000000000000 6
0.0700000000000000 1.0000032174168609 0.0000000000000000 7
0.0800000000000000 0.9999992048382456 0.0000000000000000 8
0.0900000000000000 0.9999952720176869 0.0000000000000000 9
0.1000000000000000 0.9999915078951970 0.0000000000000000 10
```

#### File name

- Norm.dat

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [int01].

#### Parameters

- [double01]

**Type :** Double

**Description :** Time  $t$

- [double02]

**Type :** Double

**Description :** The norm of a wave function at  $t$ .

- [int01]

**Type :** Int

**Description :** Time step.

### 4.3.20 SS.dat

(For the real time evolution method) This file is outputted to show the calculation results. In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# 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 ...
...
```

#### File name

- SS.dat

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [double04] [double05] [int01].

#### Parameters

- [double01]  
**Type :** Double  
**Description :** Time  $t$ .
- [double02]  
**Type :** Double  
**Description :** The expected value of the Hamiltonian (energy)  $\langle \mathcal{H} \rangle$ .
- [double03]  
**Type :** Double  
**Description :** The expected value of the square of the Hamiltonian  $\langle \mathcal{H}^2 \rangle$ .
- [double04]  
**Type :** Double  
**Description :** The expected value of the doublon,  $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$  ( $N_s$  is the total number of sites).

- [double05]

**Type :** Double

**Description :** The total number of particles  $\langle \hat{n} \rangle$ .

- [int01]

**Type :** Int

**Description :** Time step.

### 4.3.21 Flct.dat

(For the real time evolution method) This file is outputted to show the calculation results of the fluctuation of the particle number, doublon, and  $S_z$ . In the restart calculation, the values are added to the previous file. An example of the file format is as follows.

```
# time, N, N^2, D, D^2, Sz, Sz^2, step_i
0.0000000000000000 7.999999999999991 63.99999999999929 ...
0.0100000000000000 8.0000000000000604 64.0000000000004832 ...
0.0200000000000000 8.0000000000000018 64.000000000000142 ...
0.0300000000000000 8.00000000000001013 64.0000000000008100 ...
0.0400000000000000 7.999999999999183 63.999999999993463 ...
0.0500000000000000 7.999999999999520 63.999999999996163 ...
0.0600000000000000 7.999999999999627 63.999999999997016 ...
0.0700000000000000 8.0000000000000835 64.0000000000006679 ...
0.0800000000000000 8.0000000000000924 64.0000000000007390 ...
0.0900000000000000 7.999999999999600 63.999999999996803 ...
0.1000000000000000 7.999999999999067 63.999999999992539 ...
...
```

#### File name

- Flct.dat

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [double04] [double05] [double06] [double07] [int01].

#### Parameters

- [double01]

**Type :** Double

**Description :** Time  $t$

- [double02]

**Type :** Double

**Description :** A total particle number  $\sum_i \langle n_i \rangle$ .

- [double03]

**Type :** Double

**Description :** The expected value of the square of the particle number  $\langle (\sum_i n_i)^2 \rangle$ .

- [double04]

**Type :** Double

**Description :** The expected value of doublon  $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$  ( $N_s$  is the total number of sites).

- [double05]

**Type :** Double

**Description :** The expected value of the square of doublon  $\frac{1}{N_s} \langle (\sum_i n_{i\uparrow} n_{i\downarrow})^2 \rangle$  ( $N_s$  is the total number of sites).

- [double06]

**Type :** Double

**Description :** The expected value of  $S_z$ ,  $\frac{1}{N_s} \sum_i \langle S_i^z \rangle$  ( $N_s$  is the total number of sites).

- [double07]

**Type :** Double

**Description :** The expected value of the square of  $S_z$ ,  $\frac{1}{N_s} \langle (\sum_i S_i^z)^2 \rangle$  ( $N_s$  is the total number of sites).

- [int01]

**Type :** Int

**Description :** Time step.

### 4.3.22 Eigenvalue.dat

(For the FullDiag method) This file is outputted to show the energies calculated by the FullDiag method. An example of the file format is as follows.

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

#### File format

- [int01] [double01]

#### Parameters

- [int01]

**Type :** Int

**Description :** The index of eigenvalues. The index 0 is for the energy of the ground state and the indexes are labeled in descending order for energies.

- [double01]

**Type :** Double

**Description :** The expected value of the Hamiltonian (energy)  $\langle \mathcal{H} \rangle$ .



### 4.3.23 phys.dat

(For the FullDiag method) This file is outputted to show the physical values calculated by the FullDiag method. The data are outputted in descending order for energies. An example of the file format is as follows.

<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

#### File name

- Canonical ensemble: ##\_phys\_Nup\_\$\$Ndown%%.dat
- Grand canonical ensemble: ##\_phys.dat.

##, \$\$, and %% indicate [string02], Nup, and Ndown defined in a ModPara file, respectively.

#### File format

- Line 1: Header
- Lines 2-: [double01] [double02] [double03] [double04] [double05].

#### Parameters

- [double01]

**Type :** Double

**Description :** The energy  $\langle \mathcal{H} \rangle$ .

- [double02]

**Type :** Double

**Description :** The total number of particles  $\langle n \rangle$ .

- [double03]

**Type :** Double

**Description :** The expected value of  $S_z$ ,  $\langle S_z \rangle$ .

- [double04]

**Type :** Double

**Description :** The expected value of  $S^2$ ,  $\langle S^2 \rangle$ .

- [double05]

**Type :** Double

**Description :** The expected value of doublon,  $\frac{1}{N_s} \sum_i \langle n_{i\uparrow} n_{i\downarrow} \rangle$  ( $N_s$  is the total number of sites).

### 4.3.24 ham.dat

(For the FullDiag method) When OutputHam=1 in the CalcMod file, the Hamiltonian calculated by  $\mathcal{H}\Phi$  is outputted by the MatrixMarket format. The recalculation by using this file can be down when InputHam=1 in the CalcMod file. An example of the file format is as follows.

```
%%%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
...
```

#### File name

- ##\_ham.dat

## indicates [string02] in a ModPara file.

#### File format

- Line 1: Header
- Line 2: [int01] [int02] [int03]
- Lines3-:[int04] [int05] [double01] [double02]

#### Parameters

- [int01]  
**Type :** Int  
**Description :** Number of rows of Hamiltonian.
- [int02]  
**Type :** Int  
**Description :** Number of columns of Hamiltonian.
- [int03]  
**Type :** Int  
**Description :** Number of nonzero elements of Hamiltonian.
- [double01], [double02]  
**Type :** Double

**Description :** The value of Hamiltonian; [double01] and [double02] represent real and imaginary part of the Hamiltonian, respectively.

### 4.3.25 cisajs.dat

This file is the outputted files for one-body Green's function  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ . The target components are set in the input file with the keyword "OneBodyG". An example of the file format is as follows.

```
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
...
```

#### File name

- Lanczos method: ##\_cisajs.dat
- TPQ method: ##\_cisajs\_set??step%%.dat
- Full diagonalization method, LOBCG method: ##\_cisajs\_eigen&&.dat.
- Real time evolution method: ## cisajs step%%.dat

##, ??, %%, and && indicate [string02] in a ModPara file, the number of runs in calculation in the TPQ method, the number of steps in the TPQ method, and the index of the eigenvalues, respectively.

#### File format

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

#### Parameters

- [int01], [int03]

**Type :** Int

**Description :** The integer of the site number. [int01] and [int03] show the  $i$  and  $j$  site numbers, respectively.

- [int02], [int04]

**Type :** Int

**Description :** The integer of the spin index:

0: Up-spin

1: Down-spin.

[int02] and [int04] show  $\sigma_1$  and  $\sigma_2$ , respectively.

- [double01], [double02]

**Type :** Double

**Description :** The value of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ .

[double01] and [double02] show the real and imaginary part of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} \rangle$ , respectively.

### 4.3.26 cisajscktalt.dat

This file is the outputted files for the two-body Green's function  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ . The target components are set in the input file with the keyword "TwoBodyG". An example of the file format is as follows.

```
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
...
```

#### File name

- Lanczos method: ##\_cisajscktalt.dat
- TPQ method: ##\_cisajscktalt\_set??step%%.dat
- Full diagonalization method, LOBCG method: ##\_cisajscktalt\_eigen&&.dat
- Real time evolution method: ## cisajscktalt step%%.dat

##, ??, %%, and && indicate [string02] in a ModPara file, the number of runs in calculation in the TPQ method, the number of steps in the TPQ method, and the index of the eigenvalues, respectively.

#### File format

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

#### Parameters

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

**Type :** Int

**Description :** The integer of the site number. [int01], [int03], [int05], and [int07] show the  $i$ ,  $j$ ,  $k$ , and  $l$  site numbers, respectively.

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

**Type :** Int

**Description :** The integer of the spin index:

0: Up-spin

1: Down-spin.

[int02], [int04], [int06], and [int08] show  $\sigma_1$ ,  $\sigma_2$ ,  $\sigma_3$ , and  $\sigma_4$ , respectively.

- [double01], [double02]

**Type :** Double

**Description :** The value of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ .

[double01] and [double02] show the real and imaginary part of  $\langle c_{i\sigma_1}^\dagger c_{j\sigma_2} c_{k\sigma_3}^\dagger c_{l\sigma_4} \rangle$ , respectively.

### 4.3.27 eigenvec.dat

When OutputEigenVec=1 in a CalcMod file, the eigenvectors calculated by the Lanczos method are outputted. When InputEigenVec=1 in a CalcMod file, eigenvectors are inputted by this outputted file. The file format is of the binary type.

#### File name

- `##_eigenvec_&&_rank_$.dat`

## indicates [string02] in a ModPara file, && is the number of eigenvalues, and \$\$ is a number of rank.

#### File format

This file is written through the following source code (a little different from the actual  $\mathcal{H}\Phi$  source).

```
fp = fopen("zvo_eigenvec_0_rank_0.dat", "wb");
fwrite(&number_of_interations, sizeof(int), 1,fp);
fwrite(&local_size, sizeof(unsigned long int),1,fp);
fwrite(&eigen_vector[0], sizeof(complex double),local_size+1, fp);
fclose(fp);
```

where `number_of_interations` is the number of iterations, `local_size` is the size of eigenvector (if MPI is used, it differs from the dimension of the Hilbert space), `eigen_vector` is the (complex) eigenvector.

**Note:** The fist component of `eigen_vector` (`eigen_vector[0]`) is not used for calculation.



### 4.3.28 tmpvec.dat

When ReStart=1, 2 in a CalcMod file, vectors after the calculation stops at an indicated step are outputted. The file format is of the binary type. An example of the file format is as follows.

#### File name

- Lanczos method: ##\_tmpvec\_rank\_\$.dat
- TPQ and LOBPCG method: ##\_tmpvec\_set\_&&\_rank\_\$.dat .

## indicates [string02] in a ModPara file, and \$\$ is the number of rank. && is the sampling number for the TPQ calculation.

#### File format

This file is written through the following source code (a little different from the actual  $\mathcal{H}\Phi$  source).

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 and 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);
```

where `number_of_interations` is the number of iterations, `local_size` is the size of eigenvector (if MPI is used, it differs from the dimension of the Hilbert space), `last_vector` is the vector at the last iteration and `second_last_vector` is the vector at the second last iteration.

**Note:** The fist component of `last_vector` and `second_last_vector` (`last_vector[0]` and `second_last_vector[0]`) are not used for calculation.

### 4.3.29 DynamicalGreen.dat

This file is the outputted file for calculating the dynamical Green's function. An example of the file format is as follows.

#### File name

- `##_DynamicalGreen.dat`

`##` indicates [string02] in a ModPara file.

#### File format

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

#### Parameters

- [double01], [double02]

**Type :** Double

**Description :** The value of the frequency  $z$ .

[double01] and [double02] are a real and an imaginary part of  $z$ , respectively.

- [double03], [double04]

**Type :** Double

**Description:** The value of dynamical Green's functions  $G(z)$ .

[double03] and [double04] are a real and an imaginary part of  $G(z)$ , respectively.

### 4.3.30 recalcvec.dat

This file is the outputted file for two vectors to recalculate the dynamical Green's function by the Lanczos method. The file format is of the binary type. An example of the file format is as follows.

#### File name

- `##_recalcvec_rank_$.dat`

## indicates [string02] in a ModPara file and \$\$ is the number of rank.

#### File format

- Line 1: [int01]
- Line 2: [int02]
- Lines 3 - 3+ [int02]: [double01] [double02]
- Lines 4+ [int02] - 4+2 × [int02]: [double03] [double04].

#### Parameters

- [int01]

**Type :** Int

**Description :** The step for calculating dynamical Green's functions by the Lanczos method  $N_d$ .

- [int02]

**Type :** Long Int

**Description :** The total number of targets of the Hilbert spaces.

- [double01], [double02]

**Type :** Double

**Description :** The value of the vector  $v_{k+1}$  for recalculating dynamical Green's functions by the Lanczos method.

[double01] and [double02] are a real part and an imaginary part of  $v_{k+1}$ , respectively. The first component is not used for calculation.

- [double03], [double04]

**Type :** Double

**Description :** The value of the vector  $v_k$  for recalculating dynamical Green's functions by the Lanczos method. [double03] and [double04] are a real part and an imaginary part of  $v_k$ , respectively. The first component is not used for calculation.

### 4.3.31 TMcomponents.dat

This file is the outputted files for the components of the tridiagonal matrix and the norm of the excited state to recalculate the dynamical Green's function by the Lanczos method. The file format is of the binary type. An example of the file format is as follows.

#### File name

- `##_TMcomponents.dat`

## indicates [string02] in a ModPara file and \$\$ is the number of rank.

#### File format

- Line 1: [int01]
- Line 2: [double01]
- Lines 3-: [double02] [double03].

#### Parameters

- [int01]

**Type :** Int

**Description :** The step for calculating dynamical Green's functions by the Lanczos method  $N_d$ .

- [double01]

**Type :** Double

**Description :** The value of the norm of the excited state.

- [double02], [double03]

**Type :** Double

**Description :** The value of the components of the tridiagonal matrix to recalculate dynamical Green's functions by the Lanczos method  $\alpha_i, \beta_i (i = 1, \dots, N_d)$ . [double02] is  $\alpha_i$  and [double03] is  $\beta_i$ .

### 4.3.32 eigenvec.dat

When OutputExcitedVec=1 in a CalcMod file, the excited vector calculated by using the input vector and excited operators defined in parr.def is outputted. An example of the file format is as follows.

```
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
...
```

#### File name

- ##\_excitedvec\_&&\_rank\_\$.dat

## indicates [string02] in a ModPara file, && is the number of eigenvalues, and \$\$ is a number of rank.

#### File format

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

#### Parameters

- [int01]

**Type :** Int

**Description :** Dimensions of the excited vector.

- [double01], [double02]

**Type :** Double

**Description :** The value of the excited vector; [double01] and [double02] represent real and imaginary part of the excited vector, respectively.

### 4.3.33 residual.dat

This file is outputted to show the calculation process information for the dynamical Green's function by using the LOBCG method (CalcType=3 in CalcMod file). An example of the file format is as follows.

#### File name

- residual.dat

#### File format

- Lines 1-NOmega:[int01] [double01] [double02] [double03] [double04]
- Line NOmega+1 :empty
- Repeat Lines 1-NOmega+1.

#### Parameters

- [int01]

**Type :** Int

**Description :** Number of Iteration Steps. Only multiples of 10 are printed.

- [double01]

**Type :** Double

**Description :** The value of frequencies.

- [double02], [double03]

**Type :** Double

**Description :** The value of dynamical Green's functions  $G(z)$  at [int01] iteration step. [double02] and [double03] are a real and an imaginary part of  $G(z)$ , respectively.

- [double04]

**Type :** Double

**Description :** The norm of the residual vector at [int01] iteration step.

## 4.4 Error messages

- ERROR ! Unsupported Keyword !

The program stops because unsupported keyword is specified.

- "ERROR ! Keyword *keyword* is duplicated !

The program stops because a parameter is specified twice.

- ERROR ! Unsupported Solver : *solver* The program stops because unsupported parameter for method is specified.

- ERROR ! Unsupported Model : *model*

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

Sorry, specified combination,

MODEL : *model*

LATTICE : *lattice*

is unsupported in the STANDARD MODE...

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

The program stops because unsupported parameter for model and/or lattice is specified.

- ERROR !  $\text{abs}(2 * \text{Sz}) > \text{nsite}$  in Hubbard model !
- ERROR !  $\text{Nelec} > 2 * \text{nsite}$  in Hubbard model !
- ERROR !  $(\text{nelec} + 2 * \text{Sz}) \% 2 \neq 0$  in Hubbard model !
- ERROR !  $\text{nelec} \leq \text{nsite} \ \&\& \ 2 * |\text{Sz}| > \text{nelec}$  in Hubbard model !
- ERROR !  $\text{nelec} > \text{nsite} \ \&\& \ 2 * |\text{Sz}| > 2 * \text{nsite} - \text{nelec}$  in Hubbard model !
- ERROR !  $\text{abs}(2 * \text{Sz}) > \text{nsite}$  in Spin model !
- ERROR !  $(\text{nsite} + 2 * \text{Sz}) \% 2 \neq 0$  in Spin model !
- ERROR !  $\text{abs}(2 * \text{Sz}) > \text{nsite}$  in Hubbard model !
- ERROR !  $\text{Nelec\_cond} / 2 + \text{Nelec\_loc} > \text{nsite}$  in Kondo model !
- ERROR !  $(\text{nelec\_cond} + \text{nelec\_loc} + 2 * \text{Sz}) \% 2 \neq 0$  in Kondo model !
- ERROR !  $\text{nelec\_cond} \leq \text{nsite} / 2 \ \&\& \ 2 * |\text{Sz}| > \text{nelec\_cond} + \text{nelec\_loc} \dots$
- ERROR !  $\text{nelec\_cond} > \text{nsite} / 2 \ \&\& \ \text{abs}(\text{Sz}^2) > \text{nsite} / 2 * 3 - \text{nelec} \dots$

In the calculation of the canonical ensemble, there are some irrelevant combinations of the number of electrons, the number of sites, and the total spin moment ( the number of electrons is larger twice than the number of sites); If these situations are detected, the program will stop.

- Check !*keyword*is SPECIFIED but will NOT be USED.

Please COMMENT-OUT this line

or check this input is REALLY APPROPRIATE for your purpose !

Because an unnecessary parameter is specified, the program suggests checking the input file. If that parameter is actually unnecessary, please delete or comment out this line.

- ERROR !*keyword*is NOT specified !

The program stops because a prerequisite keyword is not specified.

- *keyword = value* ##### DEFAULT VALUE IS USED #####

This is not an error message. The program states that the default value is used because this keyword is not specified.



## ALGORITHM

### 5.1 Lanczos method

#### 5.1.1 Details of Lanczos method

Some parts of this section are based on the manual of TITPACK<sup>1</sup> and the textbook published by M. Sugihara and K. Murota<sup>2</sup> (these references are written in Japanese).

In the Lanczos method, by successively operating the Hamiltonian to the initial vector, we obtain the accurate eigenvalues around the maximum and minimum eigenvalues and associated eigenvectors. Because we can perform the Lanczos method by using only two vectors, the dimensions of which are the dimensions of the total Hilbert space<sup>3</sup>, the Lanczos method is frequently used for the diagonalization of the large matrices. As explained in detail below, one additional vector is necessary for obtaining the eigenvector.

The principle of the Lanczos method is based on the power method. In the power method, by successively operating the Hamiltonian  $\hat{\mathcal{H}}$  to the arbitrary vector  $\mathbf{x}_0$ , we generate  $\hat{\mathcal{H}}^n \mathbf{x}_0$ . The obtained space  $\mathcal{K}_{n+1}(\hat{\mathcal{H}}, \mathbf{x}_0) = \{\mathbf{x}_0, \hat{\mathcal{H}} \mathbf{x}_0, \dots, \hat{\mathcal{H}}^n \mathbf{x}_0\}$  is called the Krylov subspace. The initial vector is represented by the superposition of the eigenvectors  $\mathbf{e}_i$  (the corresponding eigenvalues are  $E_i$ ) of  $\hat{\mathcal{H}}$  as

$$\mathbf{x}_0 = \sum_i a_i \mathbf{e}_i. \quad (5.1)$$

Here,  $E_0$  denotes the maximum absolute values of the eigenvalues. We note that all the eigenvalues are real numbers because the Hamiltonian is Hermitian. By operating  $\hat{\mathcal{H}}^n$  to the initial vector, we obtain the relation as

$$\hat{\mathcal{H}}^n \mathbf{x}_0 = E_0^n \left[ a_0 \mathbf{e}_0 + \sum_{i \neq 0} \left( \frac{E_i}{E_0} \right)^n a_i \mathbf{e}_i \right]. \quad (5.2)$$

This relation indicates that the eigenvector of  $E_0$  becomes dominant for sufficiently large  $n$ . In the Lanczos method, we obtain the eigenvalues and eigenvectors by performing the appropriate transformation for the obtained Krylov subspace.

In the Lanczos method, we successively generate the normalized orthogonal basis  $\mathbf{v}_0, \dots, \mathbf{v}_{n-1}$  from the Krylov subspace  $\mathcal{K}_n(\hat{\mathcal{H}}, \mathbf{x}_0)$ . We define an initial vector and associated components as  $\mathbf{v}_0 = \mathbf{x}_0/|\mathbf{x}_0|$ ,  $\beta_0 = 0$ ,  $\mathbf{x}_{-1} = 0$ . From this initial condition, we can obtain the normalized orthogonal basis:

$$\begin{aligned} \alpha_k &= (\hat{\mathcal{H}} \mathbf{v}_k, \mathbf{v}_k), \\ \mathbf{w} &= \hat{\mathcal{H}} \mathbf{v}_k - \beta_k \mathbf{v}_{k-1} - \alpha_k \mathbf{v}_k, \\ \beta_{k+1} &= |\mathbf{w}|, \\ \mathbf{v}_{k+1} &= \frac{\mathbf{w}}{|\mathbf{w}|}. \end{aligned} \quad (5.3)$$

---

<sup>1</sup> [http://www.stat.phys.titech.ac.jp/~nishimori/titpack2\\_new/index-e.html](http://www.stat.phys.titech.ac.jp/~nishimori/titpack2_new/index-e.html)

<sup>2</sup> M. Sugihara, K. Murota, Theoretical Numerical Linear Algebra, Iwanami Studies in Advanced Mathematics, Iwanami Shoten, Publishers, 2009.

<sup>3</sup> In  $\mathcal{H}\Phi$ , to reduce the numerical cost, we use some additional vectors; a vector for accumulating the real-space diagonal elements of the Hamiltonian and a vector for specifying the given  $S_z$  space and given particle space. The dimension of these vectors is that of the Hilbert space.

From these definitions, it is obvious that  $\alpha_k, \beta_k$  are real numbers.

In the subspace spanned by these normalized orthogonal basis, the Hamiltonian is transformed as

$$T_n = V_n^\dagger \hat{\mathcal{H}} V_n. \quad (5.4)$$

Here,  $V_n$  is a matrix whose column vectors are  $\mathbf{v}_i (i = 0, 1, \dots, n-1)$ .  $T_n$  is a tridiagonal matrix and its diagonal elements are  $\alpha_i$  and subdiagonal elements are  $\beta_i$ . It is known that the eigenvalues of  $\hat{\mathcal{H}}$  are well approximated by the eigenvalues of  $T_n$  for sufficiently large  $n$ . (We note that  $V^\dagger V = I$ ,  $I$  is an identity matrix). The original eigenvectors of  $\hat{\mathcal{H}}$  are obtained by  $\mathbf{e}_i = V \tilde{\mathbf{e}}_i$ , where  $\tilde{\mathbf{e}}_i$  denotes the eigenvectors of  $T_n$ . From  $V$ , we can obtain the eigenvectors of  $\hat{\mathcal{H}}$  by performing the Lanczos method. However, in the actual calculations, it is difficult to keep  $V$ , because its dimension is large [dimension of  $V = (\text{dimension of the total Hilbert space}) \times (\text{the number of Lanczos iterations})$ ]. Thus, to obtain the eigenvectors, we again perform the same Lanczos calculations after we obtain the eigenvalues from the Lanczos methods. In the first Lanczos calculation, we keep  $\tilde{\mathbf{e}}_i$ , because its dimension is small<sup>4</sup>. From this procedure, we obtain the eigenvectors from  $V$ .

In the Lanczos method, within a few hundred or thousand Lanczos iterations, we obtain accurate eigenvalues near the maximum and minimum eigenvalues. The necessary number of iterations is sufficiently small as compared to the dimensions of the total Hilbert space.

We note that it is shown that the errors of the maximum and minimum eigenvalues become exponentially small as a function of Lanczos iterations (for details, see Ref.<sup>?</sup>).

### 5.1.2 Inverse iteration method

From the approximate value of the eigenvalues ( $E_n$ ), by successively operating  $(\hat{\mathcal{H}} - E_n)^{-1}$  to the initial vector  $\mathbf{y}_0$ , we can obtain the accurate eigenvector for  $E_n$ .

From  $(\hat{\mathcal{H}} - E_n)^{-1} \mathbf{y}_0$ , we obtain linear simultaneous equations such as

$$\mathbf{y}_k = (\hat{\mathcal{H}} - E_n) \mathbf{y}_{k+1}. \quad (5.5)$$

By solving this equation using the conjugate gradient method (CG method), we obtain the eigenvector. From the obtained eigenvector, we can calculate the eigenvalues and correlation functions. We note that four additional vectors are necessary to perform the CG method. For a large system size, it may be impossible to allocate memory to the additional vectors.

### 5.1.3 Details of implementation

#### Initial vector

For the Lanczos method, an initial vector is specified with `initial_iv` ( $\equiv r_s$ ) defined in an input file for Standard mode or a ModPara file for Expert mode. The type of initial vector can be selected as a real number or complex number by using `InitialVecType` in a ModPara file.

- For canonical ensemble and `initial_iv`  $\geq 0$ , a component of a target of the Hilbert space is given by

$$(N_{\text{dim}}/2 + r_s) \% N_{\text{dim}}, \quad (5.6)$$

where  $N_{\text{dim}}$  is the total number of the Hilbert spaces and  $N_{\text{dim}}/2$  is added to avoid selecting a special Hilbert space for a default value `initial_iv` = 1. When the type of initial vector is selected as a real number, the coefficient value is given by 1, while when it is selected as a complex number, the value is given by  $(1 + i)/\sqrt{2}$ .

---

<sup>4</sup> Upper bound of the dimensions of  $\tilde{\mathbf{e}}_i$  is # of Lanczos iterations.

- For a grand canonical ensemble or `initial_iv < 0`, the initial vector is given by using a random generator, i.e., the coefficients of all the components for the initial vector are given by random numbers. The seed is calculated as

$$123432 + |r_s|, \quad (5.7)$$

where  $r_s$  is the number given by an input file and  $n_{\text{run}}$  is the number of runs. The maximum value of  $n_{\text{run}}$  is defined by `NumAve` in an input file for Standard mode or a `ModPara` file for Expert mode. Random numbers are generated by using SIMD-oriented Fast Mersenne Twister (dSFMT)<sup>5</sup>.

## Convergence condition

In  $\mathcal{H}\Phi$ , we use `dsyev` (routine of LAPACK) for diagonalization of  $T_n$ . We use the energy of the first excited state of  $T_n$  as the criterion of convergence. In the standard setting, after five Lanczos steps, we diagonalize  $T_n$  every two Lanczos steps. If the energy of the first excited states coincides with the previous energy within the specified accuracy, the Lanczos iteration finishes. The accuracy of the convergence can be specified by `CDataFileHead` (`ModPara` file in the expert mode).

After obtaining the eigenvalues, we again perform the Lanczos iteration to obtain the eigenvector. From the eigenvectors  $|n\rangle$ , we calculate energy  $E_n = \langle n|\hat{\mathcal{H}}|n\rangle$  and variance  $\Delta = \langle n|\hat{\mathcal{H}}^2|n\rangle - (\langle n|\hat{\mathcal{H}}|n\rangle)^2$ . If  $E_n$  coincides with the eigenvalues obtained by the Lanczos iteration and  $\Delta$  is smaller than the specified value, we finish diagonalization.

If the accuracy of the Lanczos method is not sufficient, we perform the CG method to obtain the eigenvector. As an initial vector of the CG method, we use the eigenvectors obtained by the Lanczos method in the standard setting. This frequently accelerates the convergence.

## 5.2 Full Diagonalization method

### 5.2.1 Overview

We generate the matrix of  $\hat{\mathcal{H}}$  by using the real space configuration  $|\psi_j\rangle (j = 1 \cdots d_{\text{H}}, \text{ where } d_{\text{H}} \text{ is the dimension of the Hilbert space})$ :  $\mathcal{H}_{ij} = \langle \psi_i|\hat{\mathcal{H}}|\psi_j\rangle$ . By diagonalizing this matrix, we can obtain all the eigenvalues  $E_i$  and eigenvectors  $|\Phi_i\rangle (i = 1 \cdots d_{\text{H}})$ . In the diagonalization, we use a LAPACK routine, such as `dsyev` or `zheev`. We also calculate and output the expectation values  $A_i \equiv \langle \Phi_i|\hat{A}|\Phi_i\rangle$ . These values are used for the finite-temperature calculations.

### 5.2.2 Finite-temperature calculations

From  $A_i \equiv \langle \Phi_i|\hat{A}|\Phi_i\rangle$ , we calculate the finite-temperature properties by using the relation

$$\langle \hat{A} \rangle = \frac{\sum_{i=1}^N A_i e^{-\beta E_i}}{\sum_{i=1}^N e^{-\beta E_i}}. \quad (5.8)$$

The calculation should be performed by using the own postscripts.

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

## 5.3 Finite-temperature calculations by the TPQ method

Sugiura and Shimizu showed that it is possible to calculate the finite-temperature properties from a few wavefunctions (in the thermodynamic limit, only one wave function is necessary)<sup>1,2</sup>. The wavefunction is called the thermal pure quantum (TPQ) state. Because the TPQ state can be generated by operating the Hamiltonian to the random initial wavefunction, we directly use the routine Lanczos method for the TPQ calculations. The TPQ state is given by

$$|\Phi(\beta)\rangle \equiv \exp\left(-\frac{\beta\hat{\mathcal{H}}}{2}\right)|\Phi_{\text{rand}}\rangle. \quad (5.9)$$

Here,  $\beta$  represents inverse temperature and  $|\Phi_{\text{rand}}\rangle$  is a random initial vector. It is shown that it is possible to calculate physical quantities at inverse temperature  $\beta$  as expectation values with respect to  $|\Phi(\beta)\rangle$ .

### 5.3.1 Details of implementation

#### Construction of the micro canonical TPQ (mTPQ) state

Here, we explain how to construct the micro canonical TPQ (mTPQ) state<sup>3</sup>, which offers the simplest method for calculating finite-temperature properties.

Let  $|\Phi_{\text{rand}}\rangle$  be a random initial vector. By operating  $(l - \hat{\mathcal{H}}/N_s)^k$  ( $l$  is constant and  $N_s$  represents the number of sites) to  $|\Phi_{\text{rand}}\rangle$ , we obtain the  $k$ th TPQ states as

$$|\Phi_k\rangle \equiv \frac{(l - \hat{\mathcal{H}}/N_s)|\Phi_{k-1}\rangle}{|(l - \hat{\mathcal{H}}/N_s)|\Phi_{k-1}\rangle}. \quad (5.10)$$

From  $|\Phi_k\rangle$ , we estimate the corresponding inverse temperature  $\beta_k$  as

$$\beta_k \sim \frac{2k/N_s}{l - u_k}, u_k = \langle \Phi_k | \hat{\mathcal{H}} | \Phi_k \rangle / N_s, \quad (5.11)$$

where  $u_k$  is the internal energy. The arbitrary local physical properties at  $\beta_k$  are also estimated as

$$\langle \hat{A} \rangle_{\beta_k} = \langle \Phi_k | \hat{A} | \Phi_k \rangle. \quad (5.12)$$

In a finite-size system, error is caused by the choice of the initial random vector. To estimate the average value and error of the physical properties, we perform some independent calculations by changing  $|\Phi_{\text{rand}}\rangle$ . Since the temperature

#### Construction of the canonical TPQ (cTPQ) state

Here, we explain how to construct the canonical TPQ (cTPQ) state<sup>2</sup>, which is another way to construct the TPQ state. In the cTPQ method,  $\exp[-\beta\hat{\mathcal{H}}/2]$  is approximated as follows:

$$\begin{aligned} \exp\left(-\frac{\beta\hat{\mathcal{H}}}{2}\right)|\Phi_{\text{rand}}\rangle &\sim |\Phi_k\rangle = U(\Delta\tau)^k |\Phi_{\text{rand}}\rangle \\ U(\Delta\tau) &= \sum_{n=0}^{n_{\text{max}}} \frac{1}{n!} \left(-\frac{\Delta\tau}{2} \hat{\mathcal{H}}\right)^n \\ \beta_k &= k\Delta\tau \end{aligned} \quad (5.13)$$

As in the mTPQ method, physical quantities are calculated as

$$\langle \hat{A} \rangle_{\beta_k} = \frac{\langle \Phi_k | \hat{A} | \Phi_k \rangle}{\langle \Phi_k | \Phi_k \rangle}. \quad (5.14)$$

<sup>1</sup> S. Sugiura and A. Shimizu, Phys. Rev. Lett. **108**, 240401 (2012).

<sup>2</sup> S. Sugiura and A. Shimizu, Phys. Rev. Lett. **111**, 010401 (2013).

## Initial vector

For the TPQ method, the initial vector is given by using a random generator, i.e., the coefficients of all the components for the initial vector are given by random numbers. The seed is calculated as

$$123432 + (n_{\text{run}} + 1) \times |r_s| + k_{\text{Thread}} + N_{\text{Thread}} \times k_{\text{Process}}, \quad (5.15)$$

where  $r_s$  is the number given by an input file and  $n_{\text{run}}$  is the number of runs.  $r_s$  and the maximum value of  $n_{\text{run}}$  are defined by `initial_iv` and `NumAve` in an input file for Standard mode or a `ModPara` file for Expert mode, respectively. Random numbers are generated by using SIMD-oriented Fast Mersenne Twister (dSFMT)<sup>3</sup>. We can select the type of initial vector as a real number or complex number by using `InitialVecType` in a `ModPara` file.  $k_{\text{Thread}}$ ,  $N_{\text{Thread}}$ , and  $k_{\text{Process}}$  indicate the thread ID, number of threads, process ID, respectively; the initial vector depends both on `initial_iv` and the number of parallelisms.

## 5.4 Dynamical Green's function

Using  $\mathcal{H}\Phi$ , we can calculate a dynamical Green's function

$$I(z) = \langle \Phi' | \frac{1}{\mathcal{H} - z\hat{I}} | \Phi' \rangle, \quad (5.16)$$

where  $|\Phi'\rangle = \hat{O}|\Phi_0\rangle$  is an excited state and  $\hat{O}$  is an excitation operator defined as a single excitation operator

$$\sum_{i,\sigma_1} A_{i\sigma_1} c_{i\sigma_1} (c_{i\sigma_1}^\dagger) \quad (5.17)$$

or a pair excitation operator

$$\sum_{i,j,\sigma_1,\sigma_2} A_{i\sigma_1 j\sigma_2} c_{i\sigma_1} c_{j\sigma_2}^\dagger (c_{i\sigma_1}^\dagger c_{j\sigma_2}). \quad (5.18)$$

For example, the dynamical spin susceptibilities can be calculated by defining  $\hat{O}$  as

$$\hat{O} = \hat{S}(\mathbf{k}) = \sum_j \hat{S}_j^z e^{i\mathbf{k}\cdot\mathbf{r}_j} = \sum_j \frac{1}{2} (c_{j\uparrow}^\dagger c_{j\uparrow} - c_{j\downarrow}^\dagger c_{j\downarrow}) e^{i\mathbf{k}\cdot\mathbf{r}_j}. \quad (5.19)$$

There are two modes implemented in  $\mathcal{H}\Phi$ . One is the continued fraction expansion method by using Lanczos method<sup>1</sup> and the other is the shifted Krylov method<sup>2</sup>. See the reference for the details of each algorithm.

## 5.5 Real time evolution method

In  $\mathcal{H}\Phi$ , real time evolution calculation is done by using the following relation

$$|\Phi(t_n)\rangle = \exp(-i\mathcal{H}\Delta t_n) |\Phi(t_{n-1})\rangle, \quad (5.20)$$

where  $|\Phi(t_0)\rangle$  is an initial wave function and  $t_n = \sum_{j=1}^n \Delta t_j$ . In calculation, we approximate  $\exp(-i\mathcal{H}\Delta t_n)$  as

$$\exp(-i\mathcal{H}\Delta t_n) = \sum_{l=0}^m \frac{1}{l!} (-i\mathcal{H}\Delta t_n)^l. \quad (5.21)$$

Here, the cut-off integer  $m$  can be set by `ExpandCoef` in `ModPara`. We can judge whether the expansion order is enough or not by checking the norm conservation  $\langle \Phi(t_n) | \Phi(t_n) \rangle = 1$  and energy conservation  $\langle \Phi(t_n) | \hat{\mathcal{H}} | \Phi(t_n) \rangle = E$ .

<sup>3</sup> <http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/SFMT>.

<sup>1</sup> E. Dagotto, Rev. Mod. Phys. **66**, 763-840 (1994).

<sup>2</sup> S. Yamamoto, T. Sogabe, T. Hoshi, S.-L. Zhang, T. Fujiwara, Journal of the Physical Society of Japan **77**, 114713 (2008).

## 5.6 Bogoliubov representation

In the spin system, the spin indices in the input files of `transfer`, `InterAll`, and correlation functions are specified as those of the Bogoliubov representation. The spin operators are written by using creation/annihilation operators:

$$\begin{aligned} S_{iz} &= \sum_{\sigma=-S}^S \sigma c_{i\sigma}^\dagger c_{i\sigma} \\ S_i^+ &= \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma+1}^\dagger c_{i\sigma} \\ S_i^- &= \sum_{\sigma=-S}^{S-1} \sqrt{S(S+1) - \sigma(\sigma+1)} c_{i\sigma}^\dagger c_{i\sigma+1} \end{aligned} \tag{5.22}$$

In HPhi, the index of the highest- $\sigma$  state is 0.

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Prof. Hidetoshi Nishimori and Mr. Daisuke Tahara. Implementation of the Lanczos algorithm in  $\mathcal{H}\Phi$  written in C is based on the pioneering diagonalization package TITPACK written in Fortran by Prof. Nishimori. For developing the user interface of  $\mathcal{H}\Phi$ , we followed the design concept of the user interface in the program for variational Monte Carlo developed by Mr. Tahara. A part of the user interface in  $\mathcal{H}\Phi$  is based on his original codes. We would also like to express our thanks for the support of the "*Project for advancement of software usability in materials science*" of The Institute for Solid State Physics, The University of Tokyo, for the development of  $\mathcal{H}\Phi$  ver. 0.1 - ver. 2.0.4

Dr. Yuichi Motoyama and Dr. Yasuyuki Kato found and reported bugs. We acknowledge their kindness and carefulness.

We thank Prof. Takeo Hoshi and Prof. Tomohiro Sogabe for the fruitful discussions on the shifted Krylov method and the development of the  $K\omega$  library.





## HPHI/MVMC FOURIER-TRANSFORMATION UTILITY

### A.1 Overview

This document is the manual for the utility to perform the Fourier transformation of the correlation function in the site representation generated by mVMC or  $\mathcal{H}\Phi$ .

#### A.1.1 Prerequisite

The prerequisite of this utility is the same as that of mVMC or  $\mathcal{H}\Phi$ .

#### A.1.2 Supported quantities

This utility supports the Fourier transformation of the following quantities:

One-body correlations

$$\langle \hat{c}_{\mathbf{k}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{k}\beta\uparrow} \rangle \equiv \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{c}_{\mathbf{0}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{R}\beta\uparrow} \rangle \quad (1.1)$$

$$\langle \hat{c}_{\mathbf{k}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{k}\beta\downarrow} \rangle \equiv \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{c}_{\mathbf{0}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{R}\beta\downarrow} \rangle \quad (1.2)$$

Density-density correlation

$$\langle \hat{\rho}_{\mathbf{k}\alpha} \hat{\rho}_{\mathbf{k}\beta} \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle (\hat{\rho}_{\mathbf{0}\alpha} - \langle \hat{\rho}_{\mathbf{0}\alpha} \rangle) (\hat{\rho}_{\mathbf{R}\beta} - \langle \hat{\rho}_{\mathbf{R}\beta} \rangle) \rangle \quad (1.3)$$

Spin-Spin correlations

$$\langle \hat{S}_{\mathbf{k}\alpha}^z \hat{S}_{\mathbf{k}\beta}^z \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{S}_{\mathbf{0}\alpha}^z \hat{S}_{\mathbf{R}\beta}^z \rangle \quad (1.3)$$

$$\langle \hat{S}_{\mathbf{k}\alpha}^+ \hat{S}_{\mathbf{k}\beta}^- \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{S}_{\mathbf{0}\alpha}^+ \hat{S}_{\mathbf{R}\beta}^- \rangle \quad (1.4)$$

$$\langle \hat{\mathbf{S}}_{\mathbf{k}\alpha} \cdot \hat{\mathbf{S}}_{\mathbf{k}\beta} \rangle \equiv \frac{1}{N_{\mathbf{R}}} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} \langle \hat{\mathbf{S}}_{\mathbf{0}\alpha} \cdot \hat{\mathbf{S}}_{\mathbf{R}\beta} \rangle \quad (1.5)$$

## A.2 Tutorial

In this tutorial, we explain through a sample calculation of the 8-site Hubbard model on the square lattice.

### A.2.1 Run HPhi/vmc.out

- For  $\mathcal{H}\Phi$

We calculate the ground state and the correlation function with the following input file

```
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
```

- For mVMC

First, we optimize the trial wavefunction with the following input

```
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
```

We add the following line to the input file to compute the correlation function.

```
NVMCCalMode = 1
```

Compute the correlation function.

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

Then the one- and two-body correlation function are written to files in the output/ directory.

Related files

- StdFace.def (See the manuals for mVMC/ $\mathcal{H}\Phi$ )

- zqp\_opt.dat (See the manual for mVMC)
- greenone.def (*Specify the index of correlation function to be computed*)
- greentwo.def (*Specify the index of correlation function to be computed*)

## A.2.2 Fourier transformation of correlation functions

Perform the Fourier transformation of the correlation function by using the utility `greenr2k`.

```
$ 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
```

Then the Fourier-transformed correlation functions are written to a file in `output/`.

Related files

- `output/zvo_cisajs_001.dat` (*Results of correlation function in the site representation*)
- `output/zvo_cisajs.dat` (*Results of correlation function in the site representation*)
- `output/zvo_cisajsktalt_001.dat` (*Results of correlation function in the site representation*)
- `output/zvo_cisajsktalt.dat` (*Results of correlation function in the site representation*)
- `geometry.dat` (*Geometry*)
- `output/zvo_corr.dat` (*Correlation functions on the  $k$  path*)

## A.2.3 Display correlation functions

Plot the correlation function in the  $k$  space by using `gnuplot`.

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

Related files

- `kpath.gp` (*gnuplot script*)
- `output/zvo_corr.dat` (*Correlation functions on the  $k$  path*)

## A.3 File format

### A.3.1 Geometry

The file name in the *Tutorial* is `geometry.dat`. When we use Standard mode of mVMC/H $\Phi$ , the information of the cell and geometry is generated automatically.

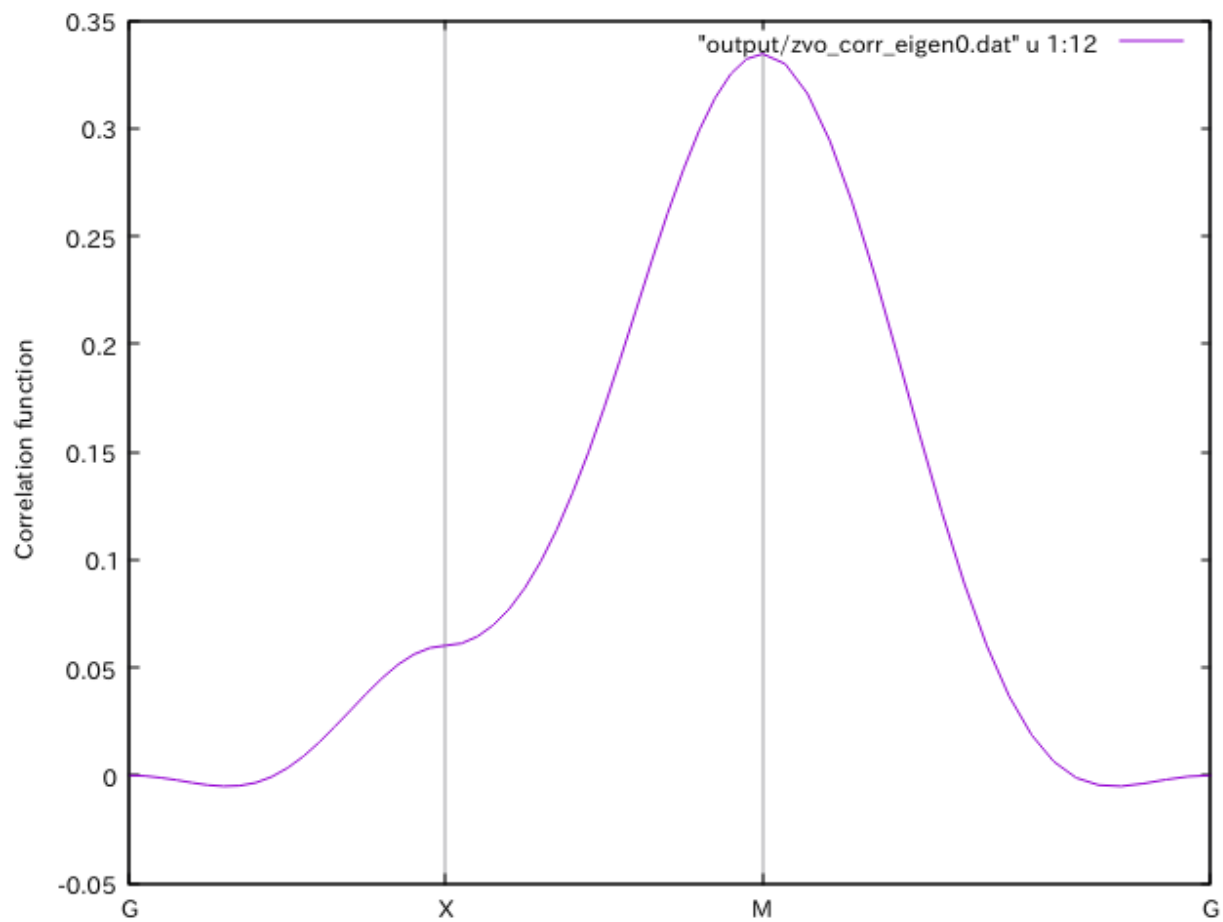


Fig. 1.1: The spin-spin correlation  $\langle \mathbf{S}_k \cdot \mathbf{S}_k \rangle$  (Column 12).

```

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. The unit lattice vectors. Arbitrary unit (Generated by Standard mode).
2. The phase for the one-body term across boundaries of the simulation cell (degree unit, Generated by Standard mode).
3. Three integer vector specifying the shape of the simulation cell. They are the same as the input parameters `a0W`, `a0L`, `a0H`, `a1W...` in Standard mode (Generated by standard mode).
4. The index of the lattice vector and the orbital each site (Generated by standard mode).
5. The number of  $k$  node (high-symmetry point) and the number of  $k$  along high symmetry line.
6. Fractional coordinate of  $k$  nodes.
7. The  $k$  grid to plot the isosurface of the momentum distribution function.

### A.3.2 One- and Two-body correlation function in the site representation

#### Specify the index of correlation function to be computed

Specify the index of correlation functions computed with mVMC/HΦ. When we use the standard mode, this file is generated automatically. The general description is written in the manuals for mVMC/HΦ. The file names in [Tutorial](#) are `greennone.def` (one body) and `greentwo.def` (two body).

For calculating correlation functions in [Supported quantities](#), indices must be specified as follows:

- $\langle \hat{c}_{\mathbf{k}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{k}\beta\uparrow} \rangle$   
 $\langle \hat{c}_{\mathbf{0}\alpha\uparrow}^\dagger \hat{c}_{\mathbf{R}\beta\uparrow} \rangle$  with  $\mathbf{R}$  ranging on the all unit cell, and  $(\alpha, \beta)$  ranging on the all orbitals in the unit cell.
- $\langle \hat{c}_{\mathbf{k}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{k}\beta\downarrow} \rangle$   
 $\langle \hat{c}_{\mathbf{0}\alpha\downarrow}^\dagger \hat{c}_{\mathbf{R}\beta\downarrow} \rangle$  with  $\mathbf{R}$  ranging on the all unit cell, and  $(\alpha, \beta)$  ranging on the all orbitals in the unit cell.
- $\langle \hat{\rho}_{\mathbf{k}\alpha} \hat{\rho}_{\mathbf{k}\beta} \rangle$  and  $\langle \hat{S}_{\mathbf{k}\alpha}^z \hat{S}_{\mathbf{k}\beta}^z \rangle$

$\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{0\alpha\sigma} \hat{c}_{\mathbf{R}\beta\sigma'}^\dagger \hat{c}_{\mathbf{R}\beta\sigma'} \rangle$  with  $\mathbf{R}$  ranging on the all unit cell,  $(\alpha, \beta)$  ranging on the all orbitals in the unit cell, and  $(\sigma, \sigma')$  ranging from  $\uparrow$  to  $\downarrow$ .

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

For  $\mathcal{H}\Phi$ ,  $\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{0\alpha-\sigma} \hat{c}_{\mathbf{R}\beta-\sigma}^\dagger \hat{c}_{\mathbf{R}\beta\sigma} \rangle$  with  $\mathbf{R}$  ranging on the all unit cell,  $(\alpha, \beta)$  ranging on the all orbitals in the unit cell, and  $\sigma$  ranging from  $\uparrow$  to  $\downarrow$ . For mVMC,  $\langle \hat{c}_{0\alpha\sigma}^\dagger \hat{c}_{\mathbf{R}\beta\sigma} \hat{c}_{\mathbf{R}\beta-\sigma}^\dagger \hat{c}_{0\alpha-\sigma} \rangle$  with  $\mathbf{R}$  ranging on the all unit cell,  $(\alpha, \beta)$  ranging on the all orbitals in the unit cell, and  $\sigma$  ranging from  $\uparrow$  to  $\downarrow$ . In the both cases, please care the order of operators.

In the default settings of Standard mode (outputmode="corr"), the above indices are specified automatically. Therefore we do not have to care it.

## Results of correlation function in the site representation

The correlation functions having the indices specified in *Specify the index of correlation function to be computed* are computed by mVMC/ $\mathcal{H}\Phi$ , and written to files. The general description of this file is written in the manuals of mVMC/ $\mathcal{H}\Phi$ . File names in the *Tutorial* are output/zvo\_cisajs\_001.dat and output/zvo\_cisajsckalt\_001.dat (mVMC), or output/zvo\_cisajs.dat and output/zvo\_cisajsckalt.dat ( $\mathcal{H}\Phi$ ).

The utility `fourier` reads these files before the calculation. If some of the correlation functions with indices written in *Specify the index of correlation function to be computed* are lacking (for example, because Standard mode was not used), this utility assume them as 0.

### A.3.3 Correlation functions on the $k$ path

This file contains the Fourier-transformed correlation function and generated by the utility `fourier`. The file name in the *Tutorial* is output/zvo\_corr.dat.

```
# 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 ...
: :
```

First, the information of the quantities at each column is written, and then the  $k$  coordinate along the path and the real- and imaginary- part of the correlation function are written.

### A.3.4 gnuplot script

This file is generated by `greenr2k`. This script is used for displaying the  $k$  labels in gnuplot. The file name is `kpath.gp`.

```
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
```

### A.3.5 FermiSurfer file to display the isosurface of the momentum distribution

This file is generated by `greenr2k`. The file name in the tutorial is `output/zvo_corr_eigen0.dat.frmsf`.

## A.4 Behavior of `greenr2k` utility

This utility is used as follows:

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

where `${PATH}` is the path to the directory where the executable `fourier` exists, `${NAMELIST}` is the NameList input-file name of  $\mathcal{H}\Phi$ /mVMC, and `${GEOMETRY}` is the path to the *Geometry* file.

The behavior of this utility is slightly different between the correlation functions from each mode of  $\mathcal{H}\Phi$  (Lanczos, TPQ, Full diagonalization, LOBCG) and mVMC. In the following cases, we assume that `CDataFileHead` in the `ModPara` input file is "zvo" (default).

### A.4.1 HPhi-Lanczos

In this case, HPhi writes correlation functions to the files `zvo_cisajs.dat` (one body) and `zvo_cisajsckalt.dat` (two body) in `output/` directory. `fourier` utility reads this files, performs the Fourier transformation, and generate single file `zvo_corr.dat` in `output/` directory.

### A.4.2 HPhi-TPQ

HPhi writes correlation functions to files `zvo_cisajs_run*step*.dat` (one body), `zvo_cisajsckalt_run*step*.dat` (two body) at each trial and TPQ step to the `output/` directory. `fourier` utility reads the one- and the two-body correlation function at each trial/TPQ-step, and performs Fourier transformation, and write to a file `zvo_corr_run*step*.dat` in `output/` directory.

### A.4.3 HPhi-Full diagonalization and LOBCG

HPhi writes correlation functions to files `zvo_cisajs_eigen*.dat` (one body) and `zvo_cisajsckalt_eigen*.dat` (two body) for each wavefunction to the `output/` directory. `fourier` utility reads the one- and the two-body correlation function at each state and performs Fourier transformation, and write to a file `zvo_corr_eigen*.dat` in `output/`.

### A.4.4 mVMC

`vmc.out` performs calculations according to the input parameters `NDataIdxStart` and `NDataQtySmp` in `ModPara` file, and it generates `zvo_cisajs_???.dat` (one body) and `zvo_cisajsckalt_???.dat` (two body) in `output/` directory. `fourier` utility reads all of these files, performs Fourier transformation, computes the average

$$\langle A \rangle = \frac{1}{N_{\text{Try}}} \sum_{i=1}^{N_{\text{Try}}} A_i \quad (1.6)$$

and the standard error

$$\delta A = \frac{1}{N_{\text{Try}} - 1} \sqrt{\frac{1}{N_{\text{Try}}} \sum_{i=1}^{N_{\text{Try}}} (A_i - \langle A \rangle)^2} \quad (1.6)$$

of the real- and imaginary-part of each correlation function, and writes them to a file `zvo_corr_eigen*.dat` in `output/` directory.

## A.5 Contact

If you have any comments, questions, bug reports etc. about this utility, please contact to the main developer (Mitsuaki Kawamura) by sending the e-mail (the address is shown below).

<code>mkawamura_at_issp.u-tokyo.ac.jp</code>
--

Please change `_at_` into `@`, when you will send the e-mail.



## DOWNFOLDING WITH WANNIER FUNCTIONS

### B.1 Overview

In this document, we introduce how we compute downfolded models with mVMC or  $\mathcal{H}\Phi$  in conjunction to [RESPACK](#). In [RESPACK](#), the screened direct integrals  $U_{mn}(\mathbf{R}, \omega)$  and the screened exchanged integrals  $J_{mn}(\mathbf{R}, \omega)$  are given as follows:

$$U_{mn}(\mathbf{R}, \omega) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{m\mathbf{0}}^*(\mathbf{r}) w_{m\mathbf{0}}(\mathbf{r}) W(\mathbf{r}, \mathbf{r}', \omega) w_{n\mathbf{R}}^*(\mathbf{r}') w_{n\mathbf{R}}(\mathbf{r}'),$$

$$J_{mn}(\mathbf{R}, \omega) = \int_V d\mathbf{r} \int_V d\mathbf{r}' w_{m\mathbf{0}}^*(\mathbf{r}) w_{n\mathbf{R}}(\mathbf{r}) W(\mathbf{r}, \mathbf{r}', \omega) w_{n\mathbf{R}}^*(\mathbf{r}') w_{m\mathbf{0}}(\mathbf{r}').$$

Here,  $V$  is the volume of the crystal,  $w_{i\mathbf{R}}(\mathbf{r})$  is the  $i$ -th wannier function at  $\mathbf{R}$ -th cell,  $W(\mathbf{r}, \mathbf{r}', \omega)$  is the screened Coulomb interactions, respectively. In the following, the components at  $\omega = 0$  are only considered. Then, the Hamiltonian of the two-body interactions are given as follows:

$$\mathcal{H}_{\text{int}} = \frac{1}{2} \sum_{\sigma\rho} \sum_{ij} \sum_{nm} \left[ U_{mn}(\mathbf{R}_{ij}) c_{im,\sigma}^\dagger c_{jn,\rho}^\dagger c_{jn,\rho} c_{im,\sigma} \right. \\ \left. + J_{mn}(\mathbf{R}_{ij}) (c_{im,\sigma}^\dagger c_{jn,\rho}^\dagger c_{im,\rho} c_{jn,\sigma} + c_{im,\sigma}^\dagger c_{im,\rho}^\dagger c_{jn,\rho} c_{jn,\sigma}) \right],$$

where  $\mathbf{R}_{ij} \equiv \mathbf{R}_i - \mathbf{R}_j$ . Since mVMC and  $\mathcal{H}\Phi$  cannot directly treat the following type of interactions  $c_{i,\sigma}^\dagger c_{j,\rho}^\dagger c_{k,\rho'} c_{l,\sigma'}$ , the Hamiltonian must be rewritten as follows:

$$\mathcal{H}_{\text{int}} = \sum_{i,m} U_{mm}(\mathbf{0}) n_{im,\uparrow} n_{im,\downarrow} + \sum_{(i,m) < (j,n)} U_{mn}(\mathbf{R}_{ij}) n_{im} n_{jn} \\ - \sum_{(i,m) < (j,n)} J_{mn}(\mathbf{R}_{ij}) (n_{im,\uparrow} n_{jn,\uparrow} + n_{im,\downarrow} n_{jn,\downarrow}) \\ + \sum_{(i,m) < (j,n)} J_{mn}(\mathbf{R}_{ij}) (c_{im,\uparrow}^\dagger c_{jn,\downarrow}^\dagger c_{im,\downarrow} c_{jn,\uparrow} + \text{h.c.}) \\ + \sum_{(i,m) < (j,n)} J_{mn}(\mathbf{R}_{ij}) (c_{im,\uparrow}^\dagger c_{im,\downarrow}^\dagger c_{jn,\downarrow} c_{jn,\uparrow} + \text{h.c.}).$$

The lattice model is defined by the following Hamiltonian:

$$\mathcal{H} = \sum_{m,n,i,j,\sigma} [t_{mn}(\mathbf{R}_{ij}) - t_{mn}^{\text{DC}}(\mathbf{R}_{ij})] c_{im\sigma}^\dagger c_{jn\sigma} + \mathcal{H}_{\text{int}}, \quad (2.1)$$

where  $t_{mn}^{\text{DC}}(\mathbf{R}_{ij})$  is the one-body correction term given as:

$$\begin{aligned}
 t_{mm}^{\text{DC}}(\mathbf{0}) &\equiv \alpha U_{mm}(\mathbf{0}) D_{mm}(\mathbf{0}) + \sum_{(\mathbf{R},n) \neq (\mathbf{0},m)} U_{mn}(\mathbf{R}) D_{nn}(\mathbf{0}) \\
 &\quad - (1 - \alpha) \sum_{(\mathbf{R},n) \neq (\mathbf{0},0)} J_{mn}(\mathbf{R}) D_{nn}(\mathbf{R}), \\
 t_{mn}^{\text{DC}}(\mathbf{R}_{ij}) &\equiv \frac{1}{2} J_{mn}(\mathbf{R}_{ij}) (D_{nm}(\mathbf{R}_{ji}) + 2\text{Re}[D_{nm}(\mathbf{R}_{ji})]) \\
 &\quad - \frac{1}{2} U_{mn}(\mathbf{R}_{ij}) D_{nm}(\mathbf{R}_{ji}), \quad (\mathbf{R}_{ij}, m) \neq (\mathbf{0}, n), \\
 D_{mn}(\mathbf{R}_{ij}) &\equiv \sum_{\sigma} \left\langle c_{im\sigma}^{\dagger} c_{jn\sigma} \right\rangle_{\text{KS}},
 \end{aligned} \tag{2.2}$$

Here,  $t_{mm}^{\text{DC}}(\mathbf{0})$  is the term to correct the chemical potential,  $t_{mn}^{\text{DC}}(\mathbf{R}_{ij})$  is term to correct transfer integrals. These terms are introduced to avoid double counting in analyzing the lattice model. To adopt these corrections or not can be selected by the option `doublecounting` in the input file. The strength of  $U_{Rij}$  and  $J_{Rij}$  can be controlled by multiplying tuning parameters  $\lambda_U, \lambda_J$ . For details, see `Input parameters for Standard mode`.

### B.1.1 Prerequisite

We compute the Kohn-Sham orbitals with `QuantumESPRESSO` or `xTAPP`, and obtain the Wannier function, the dielectric function, the effective interaction with `RESPACK`, and simulate quantum lattice models with `mVMC` or `HΦ`. Therefore, these programs must be available in our machine.

## B.2 Tutorial

In this tutorial, we downfold  $\text{Sr}_2\text{CuO}_3$  into single orbital 1D Hubbard model, and simulate that model with `HPhi/mVMC`. We use `QuantumESPRESSO` for the DFT calculation. Input files are served in `samples/Wannier/Sr2CuO3` directory.

In actual studies, the input files etc. of each solver should be modified for more high accuracy calculation. Please refer to the manuals of each solver for the details of the input files.

### B.2.1 SCF calculation of charge density

First, we perform the SCF calculation of the charge density. The input file is as follows:

`scf.in`

```

&CONTROL
  calculation = 'scf'
  pseudo_dir = '../pseudo'
  prefix = 'sr2cuo3'
/
&SYSTEM
  ibrav = 0
  nat = 6
  ntyp = 3
  ecutwfc = 30.000000
  ecutrho = 240.000000

```

(continues on next page)

(continued from previous page)

```

occupations = 'tetrahedra_opt'
/
&ELECTRONS
mixing_beta = 0.3
/
CELL_PARAMETERS angstrom
-1.749305 1.955690 6.351200
1.749305 -1.955690 6.351200
1.749305 1.955690 -6.351200
ATOMIC_SPECIES
Sr 87.620000 Sr_ONCV_PBE-1.0.upf
Cu 63.546000 Cu_ONCV_PBE-1.0.upf
O 15.999400 O_ONCV_PBE-1.0.upf
ATOMIC_POSITIONS crystal
Sr 0.851940 0.351940 0.500000
Sr 0.148060 0.648060 0.500000
Cu 0.500000 0.000000 0.500000
O 0.654110 0.154110 0.500000
O 0.345890 0.845890 0.500000
O 0.000000 0.000000 0.000000
K_POINTS automatic
4 4 4 0 0 0

```

The pseudopotential (UPF file) are downloaded from [The SG15 Optimized Norm-Conserving Vanderbilt \(ONCV\) pseudopotentials](http://www.quantum-simulation.org/potentials/sg15_oncv/sg15_oncv_upf_2015-10-07.tar.gz). Put the pseudopotential files into `../pseudo` directory.

[http://www.quantum-simulation.org/potentials/sg15\\_oncv/sg15\\_oncv\\_upf\\_2015-10-07.tar.gz](http://www.quantum-simulation.org/potentials/sg15_oncv/sg15_oncv_upf_2015-10-07.tar.gz)

We use the program `pw.x` in QuantumESPRESSO as follows.

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

## B.2.2 (Optional) Band structure

`band.in`

```

&CONTROL
calculation = 'bands'
pseudo_dir = '../pseudo'
prefix = 'sr2cuo3'
/
&SYSTEM
ibrav = 0
nat = 6
ntyp = 3
ecutwfc = 30.000000
ecutrho = 240.000000
nbnd = 35
/
&ELECTRONS
/
CELL_PARAMETERS angstrom

```

(continues on next page)

(continued from previous page)

```

-1.749305 1.955690 6.351200
1.749305 -1.955690 6.351200
1.749305 1.955690 -6.351200
ATOMIC_SPECIES
Sr 87.620000 Sr_ONCV_PBE-1.0.upf
Cu 63.546000 Cu_ONCV_PBE-1.0.upf
O 15.999400 O_ONCV_PBE-1.0.upf
ATOMIC_POSITIONS crystal
Sr 0.851940 0.351940 0.500000
Sr 0.148060 0.648060 0.500000
Cu 0.500000 0.000000 0.500000
O 0.654110 0.154110 0.500000
O 0.345890 0.845890 0.500000
O 0.000000 0.000000 0.000000
K_POINTS crystal
50
0.50000000000 0.50000000000 -0.50000000000 1.0
0.4994075000 0.5005925000 -0.4428337500 1.0
0.4988150000 0.5011850000 -0.3856675000 1.0
0.4982225000 0.5017775000 -0.3285012500 1.0
0.4976300000 0.5023700000 -0.2713350000 1.0
0.4970375000 0.5029625000 -0.2141687500 1.0
0.4964450000 0.5035550000 -0.1570025000 1.0
0.4958525000 0.5041475000 -0.0998362500 1.0
0.4952600000 0.5047400000 -0.0426700000 1.0
0.5337666667 0.4662333333 -0.0811750000 1.0
0.5722733333 0.4277266667 -0.1196800000 1.0
0.6107800000 0.3892200000 -0.1581850000 1.0
0.6492866667 0.3507133333 -0.1966900000 1.0
0.6877933333 0.3122066667 -0.2351950000 1.0
0.7263000000 0.2737000000 -0.2737000000 1.0
0.6810400000 0.3189600000 -0.3189600000 1.0
0.6357800000 0.3642200000 -0.3642200000 1.0
0.5905200000 0.4094800000 -0.4094800000 1.0
0.5452600000 0.4547400000 -0.4547400000 1.0
0.5000000000 0.5000000000 -0.5000000000 1.0
0.3333333333 0.3333333333 -0.3333333333 1.0
0.1666666667 0.1666666667 -0.1666666667 1.0
0.0000000000 0.0000000000 0.0000000000 1.0
0.0000000000 0.0000000000 0.0625000000 1.0
0.0000000000 0.0000000000 0.1250000000 1.0
0.0000000000 0.0000000000 0.1875000000 1.0
0.0000000000 0.0000000000 0.2500000000 1.0
0.0000000000 0.0000000000 0.3125000000 1.0
0.0000000000 0.0000000000 0.3750000000 1.0
0.0000000000 0.0000000000 0.4375000000 1.0
0.0000000000 0.0000000000 0.5000000000 1.0
0.0426700000 -0.0426700000 0.5047400000 1.0
0.0811750000 -0.0811750000 0.4662333333 1.0
0.1196800000 -0.1196800000 0.4277266667 1.0
0.1581850000 -0.1581850000 0.3892200000 1.0
0.1966900000 -0.1966900000 0.3507133333 1.0

```

(continues on next page)

(continued from previous page)

0.2351950000	-0.2351950000	0.3122066667	1.0
0.2737000000	-0.2737000000	0.2737000000	1.0
0.2280833333	-0.2280833333	0.2280833333	1.0
0.1824666667	-0.1824666667	0.1824666667	1.0
0.1368500000	-0.1368500000	0.1368500000	1.0
0.0912333333	-0.0912333333	0.0912333333	1.0
0.0456166667	-0.0456166667	0.0456166667	1.0
0.0000000000	0.0000000000	0.0000000000	1.0
0.0833333333	0.0000000000	0.0000000000	1.0
0.1666666667	0.0000000000	0.0000000000	1.0
0.2500000000	0.0000000000	0.0000000000	1.0
0.3333333333	0.0000000000	0.0000000000	1.0
0.4166666667	0.0000000000	0.0000000000	1.0
0.5000000000	0.0000000000	-0.0000000000	1.0

We use `pw.x`.

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

`bands.in`

```
&BANDS
  prefix = 'sr2cuo3'
  lsym = .false.
/
```

We use `bands.x` QuantumESPRESSO.

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

We can plot the band structure by reading output `bands.out.gnu` from GnuPlot etc.

## B.2.3 Kohn-Sham orbitals for Wannier

`nscf.in`

```
&CONTROL
  calculation = 'nscf'
  pseudo_dir = './pseudo'
  wf_collect = .true.
  prefix = 'sr2cuo3'
/
&SYSTEM
  ibrav = 0
  nat = 6
  ntyp = 3
  ecutwfc = 30.000000
  ecutrho = 240.000000
  occupations = 'tetrahedra_opt'
  nbnd = 35
/
&ELECTRONS
```

(continues on next page)

(continued from previous page)

```

/
CELL_PARAMETERS angstrom
-1.749305 1.955690 6.351200
1.749305 -1.955690 6.351200
1.749305 1.955690 -6.351200
ATOMIC_SPECIES
Sr 87.620000 Sr_ONCV_PBE-1.0.upf
Cu 63.546000 Cu_ONCV_PBE-1.0.upf
O 15.999400 O_ONCV_PBE-1.0.upf
ATOMIC_POSITIONS crystal
Sr 0.851940 0.351940 0.500000
Sr 0.148060 0.648060 0.500000
Cu 0.500000 0.000000 0.500000
O 0.654110 0.154110 0.500000
O 0.345890 0.845890 0.500000
O 0.000000 0.000000 0.000000
K_POINTS automatic
4 4 4 0 0 0

```

We use `pw.x` as

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

Then, we use the utility `qe2respack.py` which is included in the RESPACK package. The command-line argument is the name of `[prefix].save` directory.

```
$ qe2respack.py sr2cuo3.save
```

## B.2.4 Wannier function, dielectric function, effective interaction

`respack.in`

```

&PARAM_CHIQW
Num_freq_grid = 1
!Ecut_for_eps =
flg_cRPA = 1
/
&PARAM_WANNIER
N_wannier = 1
Lower_energy_window = 8
Upper_energy_window = 13
N_initial_guess = 1
/
dx2 0.2 0.500000 0.000000 0.50000 0.0 0.0 1.0 0.0 1.0 0.0 1.0 0.0 0.0
&PARAM_INTERPOLATION
N_sym_points = 10
!dense = 20, 24, 28
/
0.50000 0.50000 -0.50000
0.49526 0.50474 -0.04267
0.72630 0.27370 -0.27370

```

(continues on next page)

(continued from previous page)

```

0.500000 0.500000 -0.500000
0.000000 0.000000 0.000000
0.000000 0.000000 0.500000
0.04267 -0.04267 0.50474
0.27370 -0.27370 0.27370
0.000000 0.000000 0.000000
0.500000 0.000000 0.000000
&PARAM_VISUALIZATION
flg_vis_wannier = 1,
ix_vis_min = -1,
ix_vis_max = 2,
iy_vis_min = -1,
iy_vis_max = 1,
iz_vis_min = -1,
iz_vis_max = 2
/
&PARAM_CALC_INT
calc_ifreq = 1
ix_intJ_min = -1
ix_intJ_max = 1
iy_intJ_min = -1
iy_intJ_max = 1
iz_intJ_min = -1
iz_intJ_max = 1
/

```

We use `calc_wannier`, `calc_chiqw`, `calc_j3d`, `calc_w3d` in RESPACK.

```

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

```

After finishing calculations, the files are outputted in `dir-model` folder. The format of these files is Wannier90 format and the data such as the hopping integrals are written. (If you use the old version of RESPACK (20190226), the folder name is `dir-mvmc`.)

## B.2.5 Quantum lattice mode for HPhi/mVMC

Using standard mode of HPhi/mVMC, the calculation will be done by reading the files in `dir-model` folder. First, the files in `dir-model` directory should be moved to the current directry. Then, the calculation will be started by using standard mode. For example, in HPhi, the calculation will be dobe by typing the following command:

`stan.in`

```

model = "Hubbard"
lattice = "wannier90"
a0w = 8
a0l = 0
a0h = 8
a1w = 0
a1l = 1

```

(continues on next page)

(continued from previous page)

```
a1h = 0
a2w = 1
a2l = 0
a2h = 0
method = "TPQ"
nelec = 8
exct = 1
cutoff_t = 0.5
cutoff_u = 1.0
cutoff_j = 0.02
```

```
$ cp ./dir-model/* .
$ HPhi -s stan.in
```

## B.3 Input parameters for Standard mode

We show the following example of the input file.

stan.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
```

The input parameters for the Standard mode to perform calculation of the downfolded model are as follows:

- lattice
  - lattice = "wannier90"
- Parameters related to the lattice
  - W, L, Height
    - Type :** int
    - Description :** The alignment of original unit cells is specified.
  - a0W, a0L, a0H, a1W, a1L, a1H, a2W, a2L, a2H



**Type :** int

**Description :** Three vectors ( $\vec{a}_0, \vec{a}_1, \vec{a}_2$ ) that specify the lattice . These vectors should be written in the Fractional coordinates of the original translational vectors.

- Wsub, Lsub, Hsub

**Type :** int (positive). In the default setting, Wsub=W, Lsub=L, Hsub=Height . Namely, there is no sublattice.

**Description :** They are available only in mVMC. By using these parameters, we can force the pair-orbital symmetrical to the translation of the sublattice. If the sublattice is incommensurate with the original lattice, vmcdry.out stops.

- a0Wsub, a0Lsub, a0Hsub, a1Wsub, a1Lsub, a1Hsub, a2Wsub, a2Lsub, a2Hsub

**Type :** int (positive). In the default setting, a0Wsub=a0W, a0Lsub=a0L, a0Hsub=a0H, a1Wsub=a1W, a1Lsub=a1L, a1Hsub=a1H, a2Wsub=a2W, a2Lsub=a2L, a2Hsub=a2H. Namely, there is no sublattice.

**Description :** The manner to set these aparameters is same as that for a0W, a0L, a0H, a1W, a1L, a1H, a2W, a2L, a2H. If the sublattice is incommensurate with the original lattice, vmcdry.out stops.

- Parameters related to interactions

- lambda\_u

**Type :** float (greater than or equal to 0)

**Default :** 1.0

**Description :** A parameter to tune the strength of Coulomb interactions by multiplying :math: lambda\_u by them.

- lambda\_j

**Type :** float (greater than or equal to 0)

**Default :** 1.0

**Description :** A parameter to tune the strength of exchange Coulomb interactions by multiplying :math: lambda\_j by them.

- lambda

**Type :** float (greater than or equal to 0)

**Default :** 1.0

**Description :** A parameter to tune the strength of Coulomb and exchange interactions by multiplying :math: lambda by them. When  $\lambda_U, \lambda_J$  are specified, these settings are used.

- cutoff\_t, cutoff\_u, cutoff\_j

**Type :** float

**Default :** 1.0e-8

**Description :** The cutoff parameters for the hopping, Coulomb, exchange integrals. We ignore these integrals smaller than cutoff values.

- cutoff\_tW, cutoff\_tL, cutoff\_tH
- cutoff\_UW, cutoff\_UL, cutoff\_UH
- cutoff\_JW, cutoff\_JL, cutoff\_JH

**Type :**

**Default :** `cutoff_tW = int((W-1)/2)`, `cutoff_tL=int((L-1)/2)`, `cutoff_tH=int((Height-1)/2)` (when `W`, `L` and `Height` are not defined, the values are set to 0) and others are set to 0.

**Description :** The cutoff parameters for the hopping, Coulomb, exchange integrals. We ignore these integrals that have lattice vector  $\mathbf{R}$  larger than these values.

– `cutoff_length_t`, `cutoff_length_U`, `cutoff_length_J`

**Type :** float

**Default :** `cutoff_length_t = -1.0` (include all terms), others are set to 0.3.

**Description**

The cutoff parameters for the hopping, Coulomb, exchange integrals. We ignore these integrals whose distances are longer than these values. The distances are computed from the position of the Wannier center and unit lattice vectors.

- Parameters for one body correction

To avoid double countings in analyzing the lattice model, one body correction is done by subtracting the following terms from one body terms:

$$\begin{aligned}
 t_{mm}^{\text{DC}}(\mathbf{0}) &\equiv \alpha U_{mm}(\mathbf{0}) D_{mm}(\mathbf{0}) + \sum_{(\mathbf{R},n) \neq (\mathbf{0},m)} U_{mn}(\mathbf{R}) D_{nn}(\mathbf{0}) \\
 &\quad - (1 - \alpha) \sum_{(\mathbf{R},n) \neq (\mathbf{0},0)} J_{mn}(\mathbf{R}) D_{nn}(\mathbf{R}), \\
 t_{mn}^{\text{DC}}(\mathbf{R}_{ij}) &\equiv \frac{1}{2} J_{mn}(\mathbf{R}_{ij}) (D_{nm}(\mathbf{R}_{ji}) + 2\text{Re}[D_{nm}(\mathbf{R}_{ji})]) \\
 &\quad - \frac{1}{2} U_{mn}(\mathbf{R}_{ij}) D_{nm}(\mathbf{R}_{ji}), \quad (\mathbf{R}_{ij}, m) \neq (\mathbf{0}, n), \\
 D_{mn}(\mathbf{R}_{ij}) &\equiv \sum_{\sigma} \left\langle c_{im\sigma}^{\dagger} c_{jn\sigma} \right\rangle_{\text{KS}},
 \end{aligned} \tag{2.3}$$

where, the first and second terms correspond to the Hartree and Fock corrections, respectively.  $\alpha$  is a tuning parameter for one body correction from the on-site Coulomb interactions.

– `doublecounting`

**Type :** char

**Default :** none

**Description :**

none: One body correction is not considered. `Hartree_U`: Hartree correction only considered the contribution from Coulomb interactions  $U_{Rii}$ . `Hartree`: Hartree correction. `full`: One body correction including Fock correction. Charge densities  $D_{Rij}$  are obtained by `[CDataFileHead]_dr.dat` which is automatically outputted by RESPACK. It is noted that the charge densities are assumed not to depend on spin components.

– `alpha`

**Type :** float

**Default :** 0.5

**Description :**

A tuning parameter for one body correction from the on-site Coulomb interactions ( $0 \leq \alpha \leq 1$ ).

## B.4 File format

Standard mode of HPhi/mVMC reads the following files. We can obtain these files by executing RESPACK. The files are outputted in the `dir-model` directory.

### B.4.1 Geometry

The file name is `[CDataFileHead]_geom.dat`. By editing this file, we can modify the number of orbitals treated in HPhi/mVMC.

```
-1.917800 1.917800 6.280100
1.917800 -1.917800 6.280100
1.917800 1.917800 -6.280100
3
0.0000000 -0.0000000 -0.0000000
-0.0000000 -0.0000000 -0.0000000
0.0000000 0.0000000 0.0000000
```

- Lines 1 - 3

Unit lattice vectors in the Cartesian coordinate (arbitrary unit).

- Line 4

The number of orbitals par unit cell treated by mVMC/HPhi. When we reduce the number by editing this file, the model including the same number of orbitals from the top.

- Line 5 - end

Wannier centers in the fractional coordinate. They are used by the Fourier utility. The order in which wannier functions are defined corresponds to the index of wannier functions in the following four files.

### B.4.2 Hopping, Coulomb, exchange integrals, charge densities

The file names are `[CDataFileHead]_hr.dat`, `[CDataFileHead]_ur.dat`, `[CDataFileHead]_jr.dat` and `[CDataFileHead]_dr.dat`, respectively. They are formatted as the hopping-integral file of Wannier90 is used. See details in the 8.19 `seedname_hr.dat` in the `user_guide` for wannier90.

```
wannier90 format for mvmcdry
8
343
1 1 1 1 1 1 1 1 1 1 1 1
...
-3 -3 -3 1 1 0.0004104251 -0.0000000000
-3 -3 -3 1 2 0.0001515941 -0.0000000006
-3 -3 -3 1 3 -0.0001515941 0.0000000002
```

- Line 1

File Header

- Line 2

Total number of wannier functions.

- Line 3  
Total number of super cells `nrpts` .
- Line 4- Line 5 + `int(nrpts/15)`  
The degeneracy at each super cell (basically, the number is set as 1).
- Line 6 + `int(nrpts/15)` -  
1-3rd columns: The supercell lattice vectors.  
4-th column: The index of wannier functions at the original cell.  
5-th column: The index of wannier functions at the supercell.  
6(7)-th column: The real (imaginary) value.

## B.5 Contact

If you have any comments, questions, bug reports etc. about this utility, please contact to the main developer (Mitsuaki Kawamura) by sending the e-mail (the address is shown below).

`mkawamura_at_issp.u-tokyo.ac.jp`

Please change `_at_` into `@`, when you will send the e-mail.