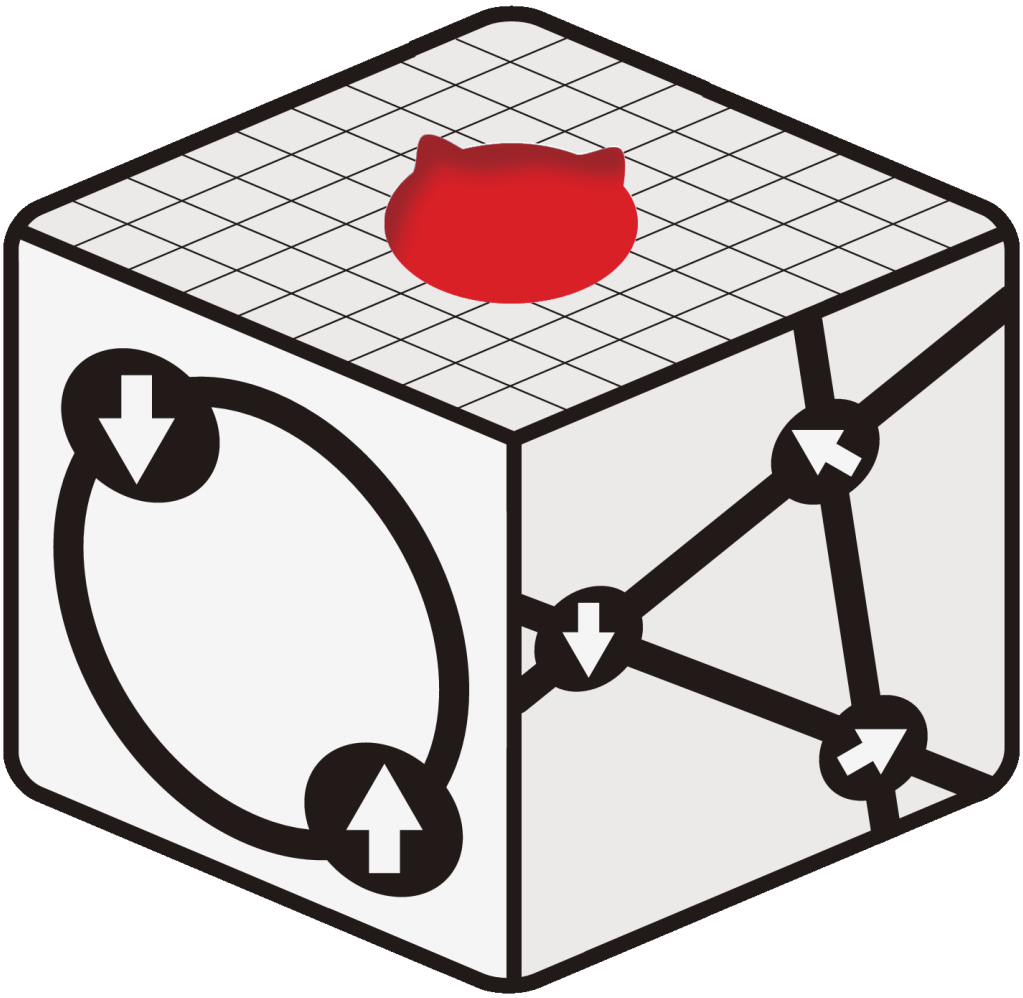

mVMC Documentation

Release 1.1.0

mVMC team

Nov 15, 2019

CONTENTS

1 What is mVMC? 1
- 1.1 Overview of mVMC . . . 1
- 1.2 License . . . 2
- 1.3 Copyright . . . 2
- 1.4 Contributors . . . 2
- 1.5 Operating environment . . . 3

2 How to use mVMC? 5
- 2.1 Prerequisite . . . 5
- 2.2 Installation . . . 5
- 2.3 Directory structure . . . 7
- 2.4 Basic usage . . . 9
- 2.5 Printing version ID . . . 10

3 Tutorial 11
- 3.1 List of sample files . . . 11
- 3.2 Heisenberg model . . . 11
- 3.3 Input files for Expert mode . . . 15
- 3.4 Fourier transformation of correlation functions . . . 16

4 Input files for Standard mode 17
- 4.1 Parameters about the kind of a calculation . . . 17
- 4.2 Parameters for the lattice . . . 18
- 4.3 Sublattice . . . 22
- 4.4 Parameters for the Hamiltonian . . . 22
- 4.5 Parameters for the numerical condition . . . 26

5 Input files for Expert mode 31
- 5.1 List file for Input files (namelist.def) . . . 33
- 5.2 ModPara file (modpara.def) . . . 34
- 5.3 LocSpin file (locspn.def) . . . 39
- 5.4 Trans file (trans.def) . . . 40
- 5.5 InterAll file . . . 42
- 5.6 CoulombIntra file (coulombintra.def) . . . 43
- 5.7 CoulombInter file (coulombinter.def) . . . 44
- 5.8 Hund file (hund.def) . . . 46
- 5.9 PairHop file . . . 47
- 5.10 Exchange file (exchange.def) . . . 48
- 5.11 Gutzwiller file (gutzwiller.def) . . . 49
- 5.12 Jastrow file (jastrow.def) . . . 51

5.13 DH2 file . . . 52
5.14 DH4 file . . . 54
5.15 Orbital/OrbitalAntiParallel file (orbitalidx.def) . . . 56
5.16 OrbitalParallel file . . . 58
5.17 OrbitalGeneral file . . . 60
5.18 TransSym file (qptransidx.def) . . . 62
5.19 Files to set initial values of variational parameters . . . 63
5.20 OneBodyG file (greenone.def) . . . 64
5.21 TwoBodyG file (greentwo.def) . . . 66

6 Output files **69**
6.1 Output file for variational parameters (***_opt.dat) . . . 69
6.2 Output files for variational parameters at each steps (xxx_var_yyy.dat) . . . 69
6.3 Output file for gutzwiller factors (***_gutzwiller_opt.dat) . . . 70
6.4 Output file for jastrow factors (***_jastrow_opt.dat) . . . 70
6.5 Output file for doublonHolon 2-site factors (***_doublonHolon2site_opt.dat) . . . 70
6.6 Output file for doublonHolon 4-site factors (***_doublonHolon4site_opt.dat) . . . 70
6.7 Output file for pair orbitals (***_orbital_opt.dat) . . . 70
6.8 xxx_out_yyy.dat . . . 70
6.9 xxx_CalcTimer.dat . . . 71
6.10 xxx_time_zzz.dat . . . 71
6.11 xxx_cisajs_yyy.dat . . . 71
6.12 xxx_cisajscktalt_yyy.dat . . . 72
6.13 xxx_ls_out_yyy.dat . . . 73
6.14 xxx_ls_cisajs_yyy.dat . . . 73
6.15 xxx_ls_cisajscktalt_yyy.dat . . . 74

7 Algorithm **75**
7.1 Variational Monte Calro Method . . . 75
7.2 Bogoliubov representation . . . 75
7.3 Properties of the Pfaffian-Slater determinant . . . 76
7.4 Power Lanczos method . . . 78

8 Program for the unrestricted Hartree-Fock approximation **81**
8.1 Overview . . . 81
8.2 Usage . . . 83

9 HPhi/mVMC Fourier-Transformation utility **85**
9.1 Overview . . . 85
9.2 Tutorial . . . 86
9.3 File format . . . 87
9.4 Behavior of `greenr2k` utility . . . 90
9.5 Contact . . . 92

10 Downfolding with Wannier functions **93**
10.1 Overview . . . 93
10.2 Tutorial . . . 94
10.3 Input parameters for Standard mode . . . 100
10.4 File format . . . 102
10.5 Contact . . . 103

11 Acknowledgement **105**

CHAPTER

ONE

WHAT IS MVMC?

High-accuracy analyses of theoretical models for quantum many-body systems are expected to play important role for clarifying the nature of novel quantum phases such as high-temperature superconductivities and quantum spin liquids. Furthermore, recent theoretical progress enables us to obtain low-energy effective models for real materials with non-empirical way [ImadaMiyake]. To clarify the electronic structures of real materials and control the physical properties, it is important to perform the high-accuracy analyses for the low-energy effective models. One of the most reliable theoretical tools for treating the strongly correlated electron systems is the exact diagonalization method. However, applicable range of system size is strongly limited in the exact diagonalization method. Variational Monte Carlo method [Gros] is one of promising way to perform the high-accuracy calculations for the larger system sizes beyond exact diagonalization method. Although the strong limitation of the variational wave function is the origin of the poor accuracy of the variational Monte Carlo method, recent development of the theoretical method and computers relaxes the limitation of the variational wave functions and enable us to perform the highly-accurate calculations [Tahara2008 , Misawa2014 , Morita2015].

mVMC (many-variable Variational Monte Carlo method) is a software for performing the highly-accurate variational Monte Carlo calculations with the simple and flexible user interface. mVMC also supports the large-scale parallelization. For the conventional models in strongly correlated electron systems such as the Hubbard model, the Heisenberg model, and the Kondo-lattice model, users can perform the calculation by preparing the one input files whose length is shorter than ten lines. By using the same input file, users can perform the exact diagonalization through HPhi [HPhi]. Thus, it is easy to examine the accuracy of the variational calculation for small system sizes and to perform the calculations for large system sizes that can not be treated by the exact diagonalization. A broad spectrum of users including experimental scientists is cordially welcome.

1.1 Overview of mVMC

By using mVMC, the following calculation can be done:

- The variational wave function which gives the minimum value of the expected value of energy in the range of the degree of freedoms of variational parameters is numerically generated. The calculation limited to the partial space divided by quantum numbers is also possible.
- The expected values of the physical quantities such as correlation functions can be calculated by using the generated variational wave functions.

The calculation flow in mVMC is shown as follows:

1. Read input files (*.def)
2. Optimize variational parameters $\vec{\alpha}$ to minimize $\langle \mathcal{H} \rangle$
3. Calculate one body and two body Green functions
4. Output variational parameters and expected values

In calculation, the simple parallelization for the generation of the real space arrangement $|x\rangle$ and the collecting samples and the calculation result of expected energies is implemented. Following the procedure for each cluster computers, the parallelized calculation using MPI is automatically done by indicating the parallel number. However, mVMC cannot execute under the environment where the MPI job is forbidden such as the front-end of system B at ISSP. In mVMC, we use PFAPACK [PFAPACK] to compute the Pfaffian matrix.

1.2 License

The distribution of the program package and the source codes for mVMC follows GNU General Public License version 3 (GPL v3).

We hope that you cite the following URL, https://github.com/issp-center-dev/mVMC, when you publish the results using mVMC (after writing the paper for mVMC, we will replace the URL to the paper's reference).

1.3 Copyright

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

This software is developed under the support of "*Project for advancement of software usability in materials science* " by The Institute for Solid State Physics, The University of Tokyo.

1.4 Contributors

This software is developed by following contributors.

- ver.1.1.0 (released at 2019/11/15)
- ver.1.0.3 (released at 2018/7/23)
- ver.1.0.2 (released at 2017/8/25)
- ver.1.0.1 (released at 2017/6/8)
- ver.1.0.0 (released at 2017/5/23)
- ver.0.2.0 (released at 2017/3/16)
- ver.0.1.1 (released at 2016/12/16)
- ver.0.1.0 (released at 2016/10/26)
  - Developers
    - Takahiro Misawa (The Institute for Solid State Physics, The University of Tokyo)
    - Satoshi Morita (The Institute for Solid State Physics, The University of Tokyo)
    - Takahiro Ogoe (Department of Applied Physics, The University of Tokyo)
    - Kota Ido (Department of Applied Physics, The University of Tokyo)
    - Masatoshi Imada (Department of Applied Physics, The University of Tokyo)
    - Yuichi Motoyama (The Institute for Solid State Physics, The University of Tokyo)
    - Mitsuaki Kawamura (The Institute for Solid State Physics, The University of Tokyo)
    - Kazusyohi Yoshimi (The Institute for Solid State Physics, The University of Tokyo)

  - Project coordinator
    * Takeo Kato (The Institute for Solid State Physics, The University of Tokyo)

1.5 Operating environment

mVMC is tested in the following platform:

- The supercomputer system-B “sekirei”S in ISSP
- K computer
- OpenMPI + Intel Compiler + MKL
- MPICH + Intel Compiler + MKL
- MPICH + GNU Compiler + MKL

CHAPTER

TWO

HOW TO USE MVMC?

2.1 Prerequisite

mVMC requires the following packages:

- C compiler (intel, Fujitsu, GNU, etc.)
- MPI library
- Option: ScaLAPACK library (intel MKL, Fujitsu, ATLAS, etc.)

Note: Settings of intel compiler

When you use the intel compiler, you can use easily scripts attached to the compiler. In the case of the bash in 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 manuals of your compiler/library for more information.

2.2 Installation

You can download mVMC in the following place. https://github.com/issp-center-dev/mVMC/releases

You can obtain the mVMC directory by typing

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

There are two kind of procedures to install mVMC.

2.2.1 Using `mVMCconfig.sh`

Please run `mVMCconfig.sh` script in the mVMC directory as follow (for ISSP system-B “sekirei”):

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

Then environmental configuration file `make.sys` is generated in `src/` directory. The command-line argument of `mVMCconfig.sh` is as follows:

- `sekirei` : ISSP system-B "sekirei"
- `kei` : K computer and ISSP system-C "maki"
- `intel-openmpi` : Intel compiler + OpenMPI
- `intel-mpich` : Intel compiler + MPICH2
- `intel-intelmpi` : Intel compiler + IntelMPI
- `gcc-openmpi` : GCC + OpenMPI
- `gcc-mpich-mkl` : GCC + MPICH + MKL

`make.sys` is as follows (for ISSP-system-B "sekirei"):

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

We explain macros of this file as:

- `CC` : C compiler (`mpicc`, `mpifccpx`)
- `F90` : fortran compiler (`ifort`, `frtpx`)
- `Libs` : Linker option
- `CFLAGS` : C compile option
- `FFLAGS` : fortran compile option

Then you are ready to compile mVMC. Please type

```
$ make mvmc
```

and obtain `vmc.out` and `vmcdry.out` in `src/` directory; you should add this directory to the `$PATH`.

You can make a PATH to mVMC as follows: `$ export PATH=${PATH}:`*mVMC_top_directory*`/src/` If you keep this PATH, you should write above in `~/.bashrc` (for `bash` as a login shell)

2.2.2 Using `cmake`

We can compile mVMC as

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

Here, we set a path to mVMC as `$PathTomvmc` and to a build directory as '' $HOME/build/mvmc''. After compilation, `src` directory is constructed below a `$HOME/build/mvmc` directory and we obtain an executable `vmc.out` in `src/` directory.

In the above example, we compile mVMC by using a gcc compiler. We can select a compiler by using the following options:

- `sekirei` : ISSP system-B "sekirei"
- `fujitsu` : Fujitsu compiler
- `intel` : Intel compiler + Linux PC
- `gcc` : GCC compiler + Linux PC.

An example of compiling mVMC by using the Intel compiler is shown as follows:

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

After compilation, `src/` directory is created below the `build/` directory and an execute `vmc.out` in the `src/` directory. We can select ScaLAPACK instead of LAPACK for vmc calculation by adding the following as the cmake option

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

Here, xxx is the libraries to use ScaLAPACK. Please note that we must delete the `build/` directory and repeat the above operations when we change the compiler.

Note: Before using cmake for sekirei, you must type

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

When we type the following in sekirei,

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

`-DSCALAPACK_LIBRARIES` is automatically set as

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

When the path to libraries for ScaLAPACK is different in your circumstance, please set `-DSCALAPACK_LIBRARIES` as the correct path.

2.3 Directory structure

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

```
├──COPYING
├──mVMCconfig.sh
├──doc/
│    ├──bib/
│    │    ├──elsart-num_mod.bst
│    │    └──userguide.bib
│    ├──figs/
│    │    ├──*.pdf
```

```
│   │   └──*.xbb
│   ├──fourier/
│   │   ├──en/
│   │   ├──figs/
│   │   └──ja/
│   ├──jp/
│   │   └──*.tex
│   └──en/
│       └──*.tex
├──sample/
│   └──Standard/
│       ├──Hubbard/
│       │   ├──square/
│       │   │   ├──StdFace.def
│       │   │   └──reference/
│       │   │       └──**.dat
│       │   └──triangular/
│       │       └── . . .
│       ├──Kondo/
│       │   └──chain/
│       │       └── . . .
│       └──Spin/
│           ├──HeisenbergChain/
│           │   └── . . .
│           ├──HeisenbergSquare/
│           │   └── . . .
│           └──Kagome/
│               └── . . .
├──src/
│   ├──mVMC/
│   │   ├── **.c
│   │   └──include/
│   │       └──**.h
│   ├──ComplexUHF/
│   │   ├── **.c
│   │   └──include/
│   │       └──**.h
│   ├──StdFace/
│   │   ├──**.c
│   │   └──**.h
│   ├──pfapack/
│   │   ├──makefile_pfapack
│   │   └──**.f
│   └──sfmt/
│       ├──makefkie_sfmt
│       ├──**.c
│       └──**.h
│
```

```
└─tool/
      ├─**.f90
      └─makefile_tool
```

2.4 Basic usage

mVMC works as whether the following two modes:

- Expert mode

  mVMC supports the arbitrary fermion-/spin-lattice system; we can specify the hopping, etc. at each site independently. Although this makes us able to specify flexibly the target this requires many input-files, and the setup of the calculation is complicated.

- Standard mode

  For some typical models (such as the Heisenberg model on the square lattice), we can start calculation with a few parameters (for example, the size of the simulation cell, the common coupling parameter). In this case, the input-files for Expert mode are automatically generated. Although the number of available systems is smaller than that number of Expert mode, the setup of the calculation is easier than in Expert mode.

We can calculate by using these modes as follows:

1. Prepare a minimal input file

   You can choose a model (the Heisenberg model, the Hubbard model, etc.) and a lattice (the square lattice, the triangular lattice, etc.) from ones provided; you can specify some parameters (such as the first/second nearest neighbor hopping integrals, the on-site Coulomb integral, etc.) for them. The input file format is described in *Input files for Standard mode*.

2. Run

   Run a executable `vmc.out` in terminal by specifying the name of input file written in previous step (option `-s` is required).

   ```
   $ mpiexec -np number-of-processes Path/vmc.out -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 manuals for your system.

3. Watch calculation logs

   Log files are outputted in the `output/` directory which is automatically made in the directory for a calculation scenario. The details of output files are shown in *Output files*.

4. Results

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

5. Prepare and run Expert mode

   In the above case, the calculation starts as soon as input files for Expert mode are generated. If we only generate files without starting the calculation, we can use a executable `vmcdry.out` as follows (MPI is not used in this step):

   ```
   $ Path/vmcdry.out Input-file-name
   ```

Then, we can edit generated files by hand and run a executable `vmc.out` with `namelist.def` as an argument (option `-e` is required) as follows:

```
$ mpiexec -np number-of-processes Path/vmc.out -e namelist.def
```

Note: The number of threads for OpenMP

If you specify the number of OpenMP threads for mVMC, you should set it as follows (in case of 16 threads) before the running:

```
$ export OMP_NUM_THREADS=16
```

2.5 Printing version ID

By using `-v` option as follows, you can check which version of mVMC you are using.

```
$ PATH/vmcdry.out -v
$ PATH/vmc.out -v
```

CHAPTER

THREE

TUTORIAL

3.1 List of sample files

There are following tutorials in `samples/Standard/`.

- The Hubbard model on the two dimensional square lattice
  (`samples/Standard/Hubbard/square/`)
- The Hubbard model on the two dimensional triangular lattice
  (`samples/Standard/Hubbard/triangular/`)
- The one dimensional Kondo chain
  (`samples/Standard/Kondo/chain/`)
- The one dimensional antiferromagnetic Heisenberg chain
  (`samples/Standard/Spin/HeisenbergChain/HeisenbergChain/`)
- The antiferromagnetic Heisenberg model on the two dimensional square lattice
  (`samples/Standard/Spin/HeisenbergSquare/`)
- The antiferromagnetic Heisenberg model on the two dimensional Kagome lattice
  (`samples/Standard/Spin/Kagome/`)

We can perform these tutorials in the same way. In the following, the tutorial of the one dimensional antiferromagnetic Heisenberg chain is shown.

3.2 Heisenberg model

This tutorial should be performed in

```
sample/Standard/Spin/HeisenbergChain/.
```

This directory contains the following things:

The input file: StdFace.def

reference outputs: reference/

In this case, we treat the one dimensional antiferromagnetic Heisenberg chain which has a nearest neighbor spin coupling.

$$\hat{H} = J \sum_{i=1}^{L} \hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_{i+1}$$

The input file is as follows:

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

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

3.2.1 Running

We execute the following command.

```
$ mpiexec -np number-of-processes Path/vmcdry.out -s StdFace.def
```

The MPI command depends on your system (such as `mpiexec`, `mpirun`, `mpijob`, `poe`, etc.). Then, the calculation starts and the following standard message is outputted in the terminal.

```
######  Standard Intarface Mode STARTS  ######

  Open Standard-Mode Inputfile StdFace.def

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

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

  @ Lattice Size & Shape

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

  @ Hamiltonian

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

(continues on next page)

(continued from previous page)

```
              J0z = 1.00000

  @ Numerical conditions

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

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

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

######  Input files are generated.  ######
-----------
Start: Read *def files.
  Read File namelist.def .
  Read File 'modpara.def' for ModPara.
  Read File 'locspn.def' for LocSpin.
  Read File 'trans.def' for Trans.
  Read File 'coulombintra.def' for CoulombIntra.
  Read File 'coulombinter.def' for CoulombInter.
```

(continues on next page)

(continued from previous page)

```
  Read File 'hund.def' for Hund.
  Read File 'exchange.def' for Exchange.
  Read File 'gutzwilleridx.def' for Gutzwiller.
  Read File 'jastrowidx.def' for Jastrow.
  Read File 'orbitalidx.def' for Orbital.
  Read File 'qptransidx.def' for TransSym.
  Read File 'greenone.def' for OneBodyG.
  Read File 'greentwo.def' for TwoBodyG.
End  : Read *def files.
Start: Read parameters from *def files.
End  : Read parameters from *def files.
Start: Set memories.
End  : Set memories.
Start: Initialize parameters.
End  : Initialize parameters.
Start: Initialize variables for quantum projection.
End  : Initialize variables for quantum projection.
Start: Optimize VMC parameters.
End  : Optimize VMC parameters.
-----------
```

In the beginning of this run, files describing the detail of considered Hamiltonian

- `locspin.def`
- `trans.def`
- `coulombinter.def`
- `coulombintra.def`
- `exchange.def`
- `hund.def`
- `namelist.def`
- `modpara.def`

and files for setting variational parameters

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

and files specifying elements of correlation functions that will be calculated

- `greenone.def`
- `greentwo.def`

are generated. The details of these files are shown in *Input files for Expert mode*.

During the calculation, the following files are outputted in `output` directory:

```
zvo_SRinfo.dat
zvo_out_001.dat
zvo_time_001.dat
```

(continues on next page)

(continued from previous page)

```
zvo_var_001.dat
zvo_CalcTimer.dat
```

In `zvo_out_001.dat`, the following quantities are outputted at each bins

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

By seeing these informations, the conversion of the calculation can be judged. By using gnuplot, we can check the evolution of $\langle H \rangle$ as follows:

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

The details of these outputted files are shown in *Output files*.

3.2.2 Output results

After finishing calculation normally, the files for the energy, the deviation, the optimized variational parameters and the time of execution for each calculation steps are outputted in `output/` directory. In the following, the outputted files are shown

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

The details of these outputted files are shown in *Output files*.

3.2.3 Calculation of Green functions

After changing the value of `NVMCCalMode` from 0 to 1 in `modpara.def` file, we execute the following command. When we add `"zqp_opt.dat"` after `"namelist.dat"` as a command-line argument as follows, the calculation of Green functions is done by using the optimized variational parameters.

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

After the calculation finishes, the following files are outputted in `output/` directory.

```
zvo_cisajs_001.dat
zvo_cisajscktalt_001.dat
```

The details of these outputted files are shown in *Output files*.

3.3 Input files for Expert mode

In mVMC, the calculation is done by reading input files categorized by the following six parts.

(1) List: Specify the kinds and names of input files.

(2) Basic parameters: Specify the basic parameters.

(3) Set Hamiltonian: Specify the Hamiltonian.

(4) Set condition of variational parameters : Specify the variational parameters to be optimized.

(5) Initial variational parameters: Specify the initial values of the variational parameters.

(6) Output: Specify the components of one-body and two-body Green's functions to be outputted.

The calculation for complex models can be done by directly making above input files. The details for each files are shown in *Input files for Expert mode*.

3.4 Fourier transformation of correlation functions

This package has a utility which performs the Fourier transformation of the correlation function and plots that function. For more details, please see *HPhi/mVMC Fourier-Transformation utility*.

CHAPTER

FOUR

INPUT FILES FOR STANDARD MODE

An example of input file for the standard mode is shown below:

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

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

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 in a "//"(comment outs) are skipped.
- Upper- and lowercase are not distinguished. Double quotes and blanks are ignored.
- There are three kinds of parameters.
  1. Parameters that must be specified (if not, `vmcdry.out` will stop with error messages),
  2. Parameters that is not necessary be specified (if not, default values are used),
  3. Parameters that must not be specified (if specified, `vmcdry.out` will stop with error messages).

  An example of 3 is transfer t for the Heisenberg spin system. If you choose "model=spin", you should not specify "t".

We explain each keywords as follows:

4.1 Parameters about the kind of a calculation

- `model`

Type : String (Choose from `"Fermion Hubbard"`, `"Spin"`, `"Kondo Lattice"` , `"Fermion HubbardGC"`, `"SpinGC"`, `"Kondo LatticeGC"`)[1]

Description : The target model is specified with this parameter; `"Fermion Hubbard"` denotes the canonical ensemble of the Fermion in the Hubbard model

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

`"Spin"` denotes canonical ensemble in the Spin model($\{\sigma_1, \sigma_2\} = x, y, z$)

$$\begin{aligned} H = &-h \sum_{i} S_{iz} - \Gamma \sum_{i} S_{ix} + D \sum_{i} S_{iz} S_{iz} \\ &+ \sum_{ij,\sigma_1} J_{ij\sigma_1} S_{i\sigma_1} S_{j\sigma_1} + \sum_{ij,\sigma_1 \neq \sigma_2} J_{ij\sigma_1\sigma_2} S_{i\sigma_1} S_{j\sigma_2}, \end{aligned}$$

`"Kondo Lattice"` denotes canonical ensemble in the Kondo lattice model

$$H = -\mu \sum_{i\sigma} c^{\dagger}_{i\sigma} c_{i\sigma} - t \sum_{\langle ij \rangle \sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + \frac{J}{2} \sum_{i} \left\{ S^{+}_{i} c^{\dagger}_{i\downarrow} c_{i\uparrow} + S^{-}_{i} c^{\dagger}_{i\uparrow} c_{i\downarrow} + S_{iz} (n_{i\uparrow} - n_{i\downarrow}) \right\}. \tag{4.2}$$

`"Fermion HubbardGC"`, `"SpinGC"`, and `"Kondo LatticeGC"` indicate the S_z-unconserved Fermion in the Hubbard model [Eqn. (4.1)], the S_z-unconserved Spin model [Eqn. (4.2)], and the S_z-unconserved Kondo lattice model [Eqn. (4.2)], respectively. Note: Although these flags has a word "GC"(=grandcanonical), the number of electrons are conserved in these system.

- `lattice`

  Type : String (Choose from `"Chain Lattice"`, `"Square Lattice"`, `"Triangular Lattice"`, `"Honeycomb Lattice"`, `"Kagome"`, `"Ladder"`)

  Description : The lattice shape is specified with this parameter; above words denote the one dimensional chain lattice (Fig. 1 (a)), the two dimensional square lattice (Fig. 1 (b)), the two dimensional triangular lattice (Fig. 1 (c)), the two dimensional anisotropic honeycomb lattice (Fig. 2), the Kagome Lattice(Fig. 3), and the ladder lattice (Fig. 4), respectively.

4.2 Parameters for the lattice

4.2.1 Chain lattice

Fig. 1 (a)

- `L`

  Type : Integer

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

4.2.2 Ladder lattice

Fig. 4

- `L`

  Type : Integer

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

[1] GC=Grand Canonical

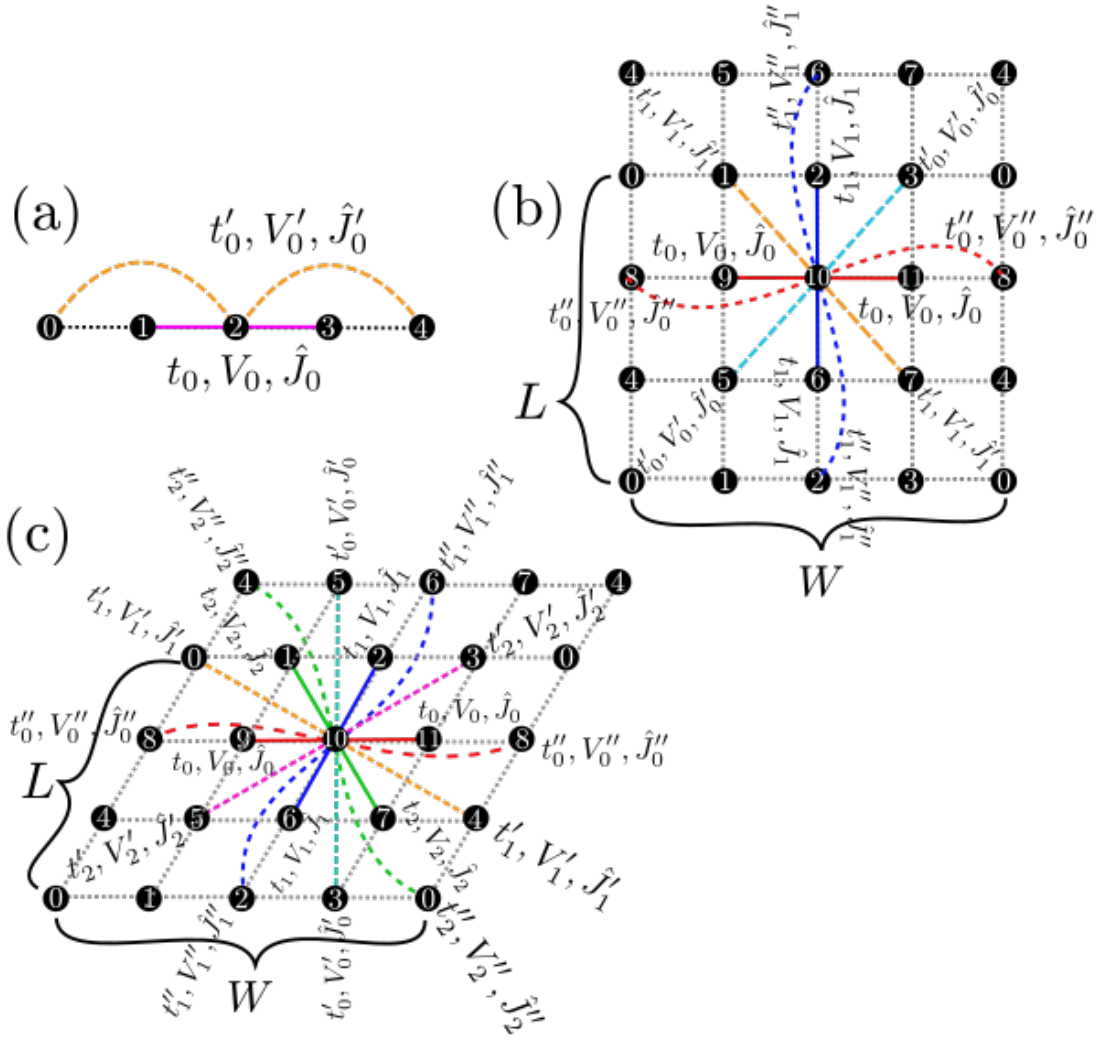

Fig. 1: Schematic illustration of (a) one dimensional chain lattice, (b) two dimensional square lattice, and (c) two dimensional triangular lattice. They have t, V, and J as a nearest neighbor hopping, an offsite Coulomb integral, and a spin-coupling constant, respectively (magenta solid lines); They also have t', V', and J' as a next nearest neighbor hopping, offsite Coulomb integral, and spin-coupling constant, respectively (green dashed line).

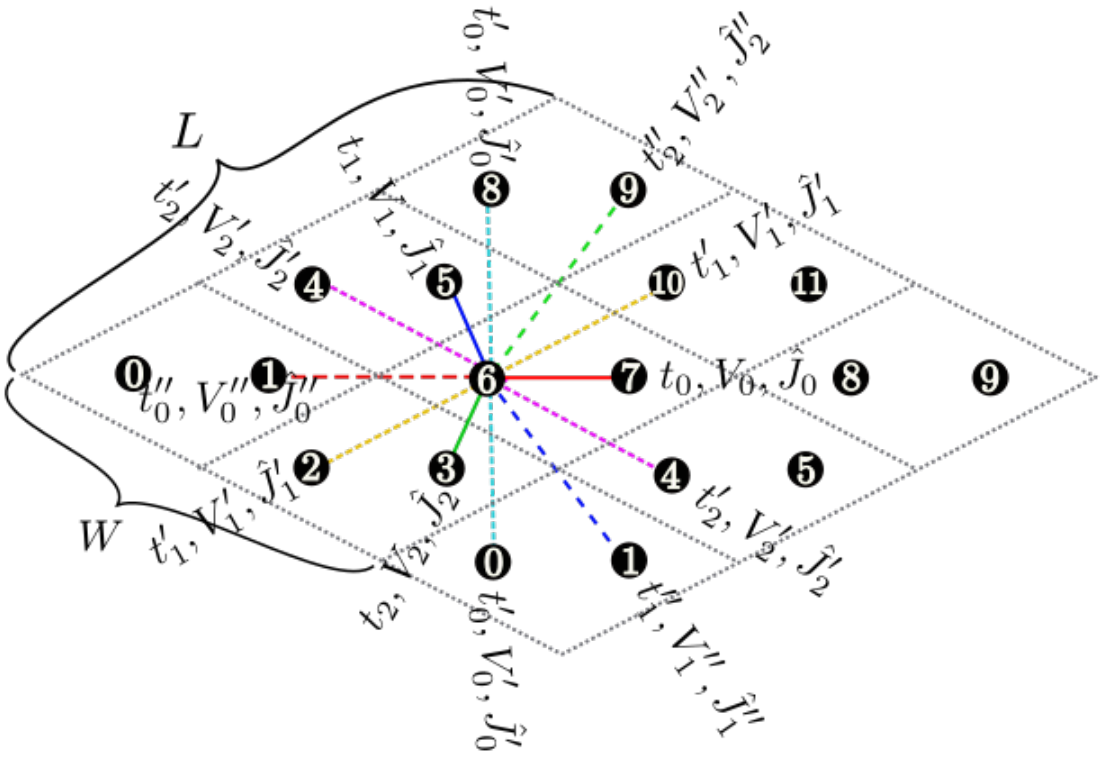

Fig. 2: Schematic illustration of the anisotropic honeycomb lattice. The first/second/third nearest neighbor hopping integral, spin coupling, and offsite Coulomb integral depend on the bond direction.

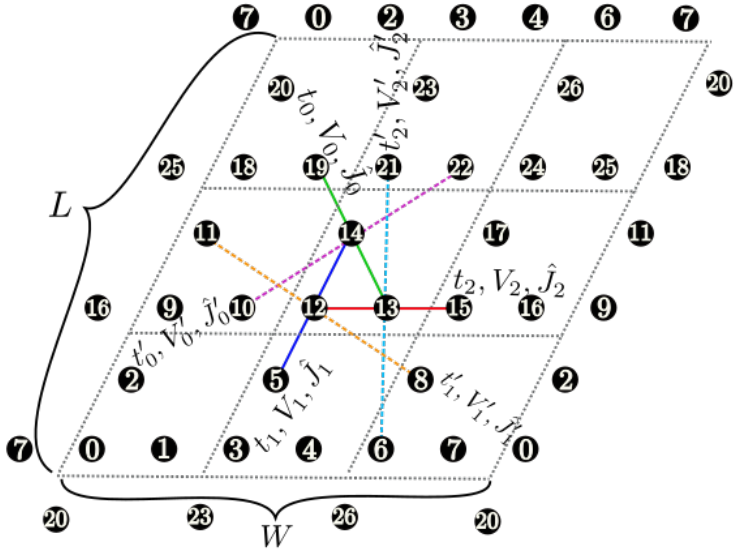

Fig. 3: Schematic illustration of the Kagome lattice.

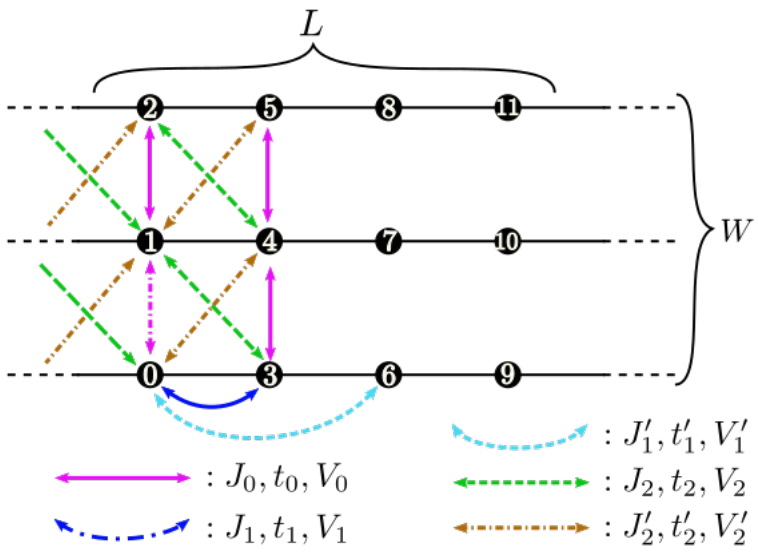

Fig. 4: Schematic illustration of the ladder lattice.

- `W`

  Type : Integer

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

4.2.3 Square lattice , Triangular lattice, Honeycomb lattice, Kagome lattice

Square lattice [Fig. 1 (b)], Triangular lattice[Fig. 1 (c)], Honeycomb lattice(Fig. 2), Kagome lattice(Fig. 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 original unit cells (dashed black lines in Figs. 1 - 3) is specified with these parameter.

- `a0W`, `a0L`, `a1W`, `a1L`

  Type : Integer

  Description : We can specify two vectors ($\vec{a}_0, \vec{a}_1$) that surrounds the numerical cell (Fig. 5). These vectors should be specified in the Fractional coordinate.

If we use both of these method, `vmcdry.out` stops.

We can check the shape of the numerical cell by using a file `lattice.gp` (only for square, trianguler, honeycomb, and kagome lattice) which is written in the Standard mode. This file can be read by `gnuplot` as follows:

```
$ gnuplot lattice.gp
```

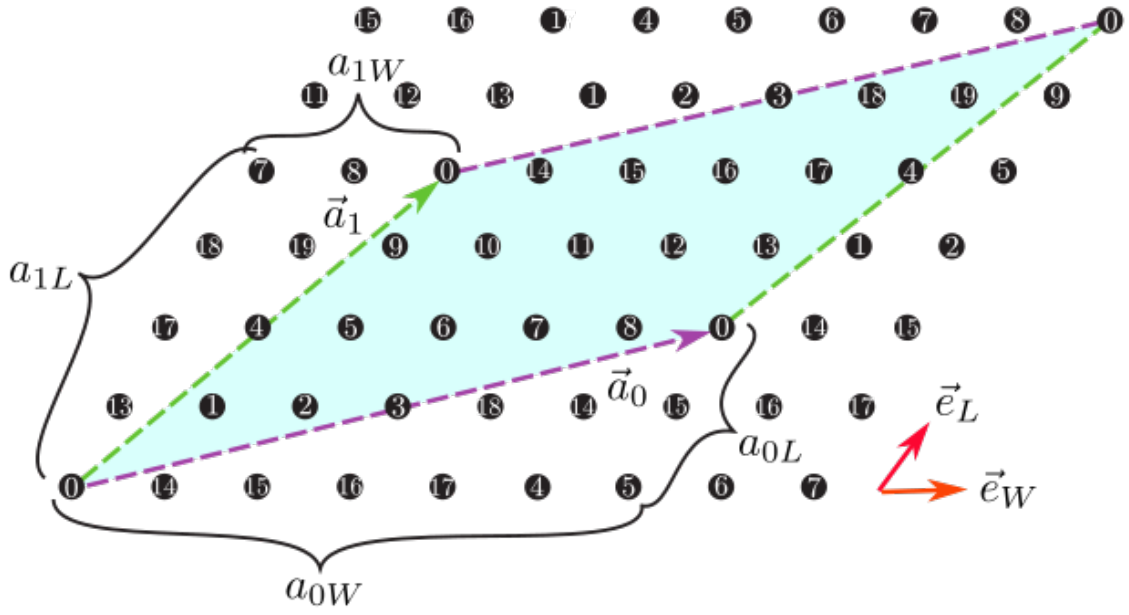

Fig. 5: The shape of the numerical cell when $\vec{a}_0 = (6, 2), \vec{a}_1 = (2, 4)$ in the triangular lattice. The region surrounded by $\vec{a}_0$ (Magenta dashed arrow) and $\vec{a}_1$ (Green dashed arrow) becomes the cell to be calculated (20 sites).

4.3 Sublattice

By using the following parameters, we can force the pair-orbital symmetrical to the translation of the sublattice.

- `a0Wsub`, `a0Lsub`, `a1Wsub`, `a1Lsub`, `Wsub`, `Lsub`

  Type : Positive integer. In the default setting, `a0Wsub=a0W`, `a0Lsub=a0L`, `a1Wsub=a1W`, `a1Lsub=a1L`, `Wsub=W`, and `Lsub=L`. Namely, there is no sublattice.

  Description : We can specify these parameter as we specify `a0W`, `a0L`, `a1W`, `a1L`, `W`, `L`. If the sublattice is incommensurate with the original lattice, `vmcdry.out` stops.

4.4 Parameters for the Hamiltonian

A default value is set as 0 unless a specific value is not defined in a description. Table [table_interactions] 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* ".

4.4.1 Local terms

- `mu`

  Type : Real

  Description : (Hubbard and Kondo lattice model) The chemical potential μ (including the site potential) is specified with this parameter.

- `U`

  Type : Real

Description : (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 : (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, `vmcdry.out` will stop.

- `h`, `Gamma`, `D`

  Type : Real

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

The non-local terms described below should be specified in different ways depending on the lattice structure: For `lattice=Ladder`, the non-local terms are specified in the different way from `lattice=Chain Lattice`, `Square Lattice`, `Triangular Lattice`, `Honeycomb Lattice`, `Kagome`. Below, the available parameters for each lattice are shown in Table [table_interactions].

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

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

4.4.2 Non-local terms for Ladder lattice

Fig. 4

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

  Type : Complex

  Description : (Hubbard and Kondo lattice model) Hopping integrals in the ladder lattice (See Fig. 4) is 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. 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) are specified with these parameter. 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, vmcdry.out will stop. The above rules are also valid for the simplified parameters, J1, J1', J2, and J2'.

4.4.3 Non-local terms for other than Ladder

Figs. 1 , 2 , 3

- t, t0, t1, t2

  Type : Complex

  Description : (Hubbard and Kondo lattice model) The nearest neighbor hoppings for each direction (see **??** - **??**) are specified with these parameters. If there is no bond dependence of the hoppings, the simplified parameter t is available to specify t0, t1, and t2 as t0 = t1 = t2 = t. If both t and the set of the hoppings (t0, t1, t2) are specified, the program will stop.

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

  Type : Complex

  Description : (Hubbard and Kondo lattice model) The second nearest neighbor hoppings for each direction (see **??** - **??**) are specified with these parameter. If there is no bond dependence of the hoppings, the simplified parameter t' is available to specify t0', t1', and t2' as t0' = t1' = t2' = t'. If both t' and the set of the hoppings (t0', t1', t2') are specified, the program will stop.

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

  Type : Complex

  Description : (Hubbard and Kondo lattice model) The third nearest neighbor hoppings for each direction (see **??** - **??**) are specified with these parameter. If there is no bond dependence of the hoppings, the simplified parameter t'' is available to specify t0'', t1'', and t2'' as t0'' = t1'' = t2'' = t''. If both t'' and the set of the hoppings (t0'', t1'', t2'') are specified, the program will stop.

- V, V0, V1, V2

  Type : Real

  Description : (Hubbard and Kondo lattice model) The nearest neighbor offsite Coulomb integrals V for each direction (see **??** - **??**) are specified with these parameters. If there is no bond dependence of the offsite Coulomb integrals, the simplified parameter V is available to specify V0, V1, and V2 as V0 = V1 = V2 = V. If both V and the set of the Coulomb integrals (V0, V1, V2) are specified, the program will stop.

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

  Type : Real

  Description : (Hubbard and Kondo lattice model) The second nearest neighbor-offsite Coulomb integrals V' for each direction (see **??** - **??**) are specified with this parameter. If there is no bond dependence of the offsite

Coulomb integrals, the simplified parameter `V'` is available to specify `V0'`, `V1'`, and `V2'` as `V0' = V1' = V2' = V'`. If both `V'` and the set of the Coulomb integrals (`V0'`, `V1'`, `V2'`) are specified, the program will stop.

- `V''`, `V0''`, `V1''`, `V2''`

  Type : Real

  Description : (Hubbard and Kondo lattice model) The third nearest neighbor-offsite Coulomb integrals V' for each direction (see **??** - **??**) are specified with this parameter. If there is no bond dependence of the offsite Coulomb integrals, the simplified parameter `V''` is available to specify `V0''`, `V1''`, and `V2''` as `V0'' = V1'' = V2'' = V''`. If both `V''` and the set of the Coulomb integrals (`V0''`, `V1''`, `V2''`) are specified, the program will stop.

- `J0x`, `J0y`, `J0z`, `J0xy`, `J0yx`, `J0xz`, `J0zx`, `J0yz`, `J0zy`
- `J1x`, `J1y`, `J1z`, `J1xy`, `J1yx`, `J1xz`, `J1zx`, `J1yz`, `J1zy`
- `J2x`, `J2y`, `J2z`, `J2xy`, `J2yx`, `J2xz`, `J2zx`, `J2yz`, `J2zy`

  Type : Real

  Description : (Spin model) Nearest-neighbor exchange couplings for each direction are specified with thees parameters. If the simplified parameter `J0` is specified, instead of `J0x, J0y, J0z`, the exchange couplings, `J0x, J0y, J0z`, are set as `J0x = J0y = J0z = J0`. If both `J0` and the set of the exchange couplings (`J0x, J0y, J0z`) are specified, `vmcdry.out` will stop. The above rules are valid for `J1` and `J2`.

  If there is no bond dependence of the nearest-neighbor exchange couplings, the simplified parameters, `Jx`, `Jy`, `Jz`, `Jxy`, `Jyx`, `Jxz`, `Jzx`, `Jyz`, `Jzy`, are available to specify the exchange couplings for every bond as `J0x = J1x = J2x = Jx`. If any simplified parameter (`Jx` $\sim$ `Jzy`) is specified in addition to its counter parts (`J0x` $\sim$ `J2zy`), `vmcdry.out` will stop. Below, examples of parameter sets for nearest-neighbor exchange couplings are shown.

  - If there are no bond-dependent, 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 are anisotropic couplings, please specify the non-zero couplings in the diagonal parameters, `Jx, Jy, Jz`.
  - If there are no bond-dependent exchange couplings but are anisotropic and offdiagonal exchange couplings, please specify the non-zero couplings in the nine parameters, `Jx, Jy, Jz, Jxy, Jyz, Jxz, Jyx, Jzy, Jzx`.
  - If there are no anisotropic and offdiagonal exchange couplings, but are bond-dependent couplings, please specify the non-zero couplings in the three parameters, `J0, J1, J2`.
  - If there are no anisotropic exchange couplings, but are bond-dependent and offdiagonal couplings, please specify the non-zero couplings in the nine parameters, `J0x, J0y, J0z, J1x, J1y, J1z, J2x, J2y, J2z`.
  - If there are bond-dependent, anisotropic and offdiagonal exchange couplings, please specify the non-zero couplings in the twenty-seven parameters from `J0x` to `J2zy`.

- `J'x`, `J'y`, `J'z`, `J'xy`, `J'yx`, `J'xz`, `J'zx`, `J'yz`, `J'zy`
- `J0'x`, `J0'y`, `J0'z`, `J0'xy`, `J0'yx`, `J0'xz`, `J0'zx`, `J0'yz`, `J0'zy`
- `J1'x`, `J1'y`, `J1'z`, `J1'xy`, `J1'yx`, `J1'xz`, `J1'zx`, `J1'yz`, `J1'zy`
- `J2'x`, `J2'y`, `J2'z`, `J2'xy`, `J2'yx`, `J2'xz`, `J2'zx`, `J2'yz`, `J2'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, the program will stop.

- `J''x`, `J''y`, `J''z`, `J''xy`, `J''yx`, `J''xz`, `J''zx`, `J''yz`, `J''zy`
- `J0''x`, `J0''y`, `J0''z`, `J0''xy`, `J0''yx`, `J0''xz`, `J0''zx`, `J0''yz`, `J0''zy`
- `J1''x`, `J1''y`, `J1''z`, `J1''xy`, `J1''yx`, `J1''xz`, `J1''zx`, `J1''yz`, `J1''zy`
- `J2''x`, `J2''y`, `J2''z`, `J2''xy`, `J2''yx`, `J2''xz`, `J2''zx`, `J2''yz`, `J2''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, the program 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 $\vec{a}_0$ direction and the $\vec{a}_1$ direction can be specified independently. For the one-dimensional system, only `phase0` can be used. For example, a fopping from i-th site to j-th site through the cell boundary with the positive direction becomes as

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

4.5 Parameters for the numerical condition

- `nelec`

Type : int-type (must be specified)

Description : The number of itinerant electrons. It is the sum of the ↑ and ↓ electrons.

- `NVMCCalMode`

Type : int-type (default value: 0)

Description : [0] Optimization of variational parameters, [1] Calculation of one body and two body Green's functions.

- `NDataIdxStart`

Type : int-type (default value: 1)

Description : An integer for numbering of output files. For `NVMCCalMode` = 0 , `NDataIdxStart` is added at the end of the output files. For `NVMCCalMode` = 1, the files are outputted with the number from `NDataIdxStart` to `NDataIdxStart` + `NDataQtySmp`-1.

- `NDataQtySmp`

Type : int-type (default value: 1)

Description : The set number for outputted files (only used for `NVMCCalMode` = 1).

- `NSPGaussLeg`

  Type : int-type (Positive integer, default value is 8 for `2Sz=0`)

  Description : The mesh number for the Gauss-legendre quadrature about β integration (S_y rotation) for the spin quantum-number projection in actual numerical calculation.

- `NSPStot`

  Type : int-type (greater equal 0, default value is 0 for `2Sz=0`)

  Description : The total spin quantum-number.

- `2Sz`

  Type : int-type (greater equal 0, default value is 0)

  Description : The spin quantum-number S_z.

- `NMPTrans`

  Type : int-type (Positive integer. Default `1`)

  Description : The number of the momentum and lattice translational quantum-number projection. In the case of not to apply the projection, this value must be set as 1.

- `NSROptItrStep`

  Type : int-type (Positive integer, default value: 1000)

  Description : The whole step number to optimize variational parameters by SR method. Only used for `NVMCCalMode` =0.

- `NSROptItrSmp`

  Type : int-type (Positive integer, default value: `NSROptItrStep`/10)

  Description : In the `NSROptItrStep` step, the average values of the each variational parameters at the `NSROptItrStep` step are adopted as the optimized values. Only used for `NVMCCalMode` =0.

- `DSROptRedCut`

  Type : double-type (default value: 0.001)

  Description : The stabilized factor for the SR method by truncation of redundant directions corresponding to $\varepsilon_{\mathrm{wf}}$ in the ref. [Tahara2008].

- `DSROptStaDel`

  Type : double-type (default value: 0.02)

  Description : The stabilized factor for the SR method by modifying diagonal elements in the overwrap matrix corresponding to ε in the ref. [Tahara2008].

- `DSROptStepDt`

  Type : double-type (default value: 0.02)

  Description : The time step using in the SR method.

- `NVMCWarmUp`

  Type : int-type (Positive integer, default value: 10)

  Description : Idling number for the Malkov chain Montecarlo Methods.

- `NVMCInterval`

  Type : int-type (Positive integer, default value: 1)

Description : The interval step between samples. The local update will be performed `Nsite` × `NVMCInterval` times.

- `NVMCSample`

  Type : int-type (Positive integer, default value: 1000)

  Description : The sample numbers to calculate the expected values.

- `NExUpdatePath`

  Type : int-type (Positive integer)

  Description : The option for local update about exchange terms. 0: not update, 1: update. The default value is set as 1 when the local spin exists, otherwise 0.

- `RndSeed`

  Type : int-type (default value: 123456789)

  Description : The initial seed of generating random number. For MPI parallelization, the initial seeds are given by `RndSeed` +my rank+1 at each ranks.

- `NSplitSize`

  Type : int-type (Positive integer, default value: 1)

  Description : The number of processes of MPI parallelization.

- `NStore`

  Type : int-type (0 or 1, default value: 1)

  Description : The option of applying matrix-matrix product to calculate expected values $\langle O_k O_l \rangle$ (0: off, 1: on). This speeds up calculation but increases the amount of memory usage from $O(N_{\mathrm{p}}^2)$ to $O(N_{\mathrm{p}}^2) + O(N_{\mathrm{p}} N_{\mathrm{MCS}})$, where N_{p} is the number of the variational parameters and N_{MCS} is the number of Monte Carlo sampling.

- `NSRCG`

  Type : int-type (0 or 1, default value: 0)

  Description : The option of solving $Sx = g$ in the SR method without constructing S matrix [NeuscammanUmrigarChan]. (0: off, 1: on). This reduces the amount of memory usage from $O(N_{\mathrm{p}}^2) + O(N_{\mathrm{p}} N_{\mathrm{MCS}})$ to $O(N_{\mathrm{p}}) + O(N_{\mathrm{p}} N_{\mathrm{MCS}})$ when $N_{\mathrm{p}} > N_{\mathrm{MCS}}$.

- `ComplexType`

  Type : int-type (`0` or `1`. Default value is `0` for the S_z-conserved system and `1` for the S_z-unconserved system.)

  Description : If it is `0`, only the real part of the variational parameters are optimized. And the real and the imaginary part of them are optimized if this parameter is `1`.

- `OutputMode`

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

  Description : Indices of correlation functions are specified with this keyword. `"none"` indicates correlation functions will not 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, $\langle c^\dagger_{i\sigma} c_{j\sigma'} \rangle$ is computed at all i, j, σ, σ', and $\langle c^\dagger_{i_1\sigma} c_{i_2\sigma} c^\dagger_{i_3\sigma'} c_{i_4\sigma'} \rangle$ is computed at all $i_1, i_2, i_3, i_4, \sigma, \sigma'$.

  In spin system, indices are specified as those on the Bogoliubov representation (See [sec_bogoliubov_rep]).

- `CDataFileHead`

  Type : string-type (default : `"zvo"`)

Description : A header for output files. For example, the output filename for one body Green's function becomes " **xxx_cisajs_yyy.dat**" (xxx are characters set by `CDataFileHead` and yyy are numbers given by numbers from `NDataIdxStart` to `NDataIdxStart` + `NDataQtySmp`).

- `CParaFileHead`

  Type : string-type (default : `"zqp"`)

  Description : A header for output files of the optimized variational parameters. For example, the optimized variational parameters are outputted as **zzz_opt_yyy.dat** (zzz are characters set by `CParaFileHead` and yyy are numbers given by numbers from `NDataIdxStart` to `NDataIdxStart` + `NDataQtySmp` -1).

CHAPTER

FIVE

INPUT FILES FOR EXPERT MODE

In this section, detailed input files (*.def) are explained. Input files are categorized by the following six parts. The files that are listed in parentheses correspond to the file made by vmcdry.out.

(1) List:

No keyword (namelist.def): This file is a list of input file names with keywords. Each keywords is fixed, but file names are free to be determined.

(2) Basic parameters:

ModPara (modpara.def): Set the parameters for basic parameters such as site number, electron number, Lanczos step *etc*. **LocSpin (locspn.def)**: Set the location of local spin.

(3) Hamiltonian:

Hamiltonian for mVMC is denoted by

$$
\begin{aligned}
\mathcal{H} &= \mathcal{H}_T + \mathcal{H}_U + \mathcal{H}_V + \mathcal{H}_H + \mathcal{H}_E + \mathcal{H}_P + \mathcal{H}_I, \\
\mathcal{H}_T &= -\sum_{i,j}\sum_{\sigma_1,\sigma_2} t_{ij\sigma_1\sigma_2} c^\dagger_{i\sigma_1} c_{j\sigma_2}, \\
\mathcal{H}_U &= \sum_i U_i n_{i\uparrow} n_{i\downarrow}, \\
\mathcal{H}_V &= \sum_{i,j} V_{ij} n_i n_j, \\
\mathcal{H}_H &= -\sum_{i,j} J^{\mathrm{Hund}}_{ij} (n_{i\uparrow} n_{j\uparrow} + n_{i\downarrow} n_{j\downarrow}), \\
\mathcal{H}_E &= \sum_{i,j} J^{\mathrm{Ex}}_{ij} (c^\dagger_{i\uparrow} c_{j\uparrow} c^\dagger_{j\downarrow} c_{i\downarrow} + c^\dagger_{i\downarrow} c_{j\downarrow} c^\dagger_{j\uparrow} c_{i\uparrow}), \\
\mathcal{H}_P &= \sum_{i,j} J^{\mathrm{Pair}}_{ij} c^\dagger_{i\uparrow} c_{j\uparrow} c^\dagger_{i\downarrow} c_{j\downarrow}, \\
\mathcal{H}_I &= \sum_{i,j,k,l}\sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} c^\dagger_{i\sigma_1} c_{j\sigma_2} c^\dagger_{k\sigma_3} c_{l\sigma_4},
\end{aligned}
$$

as the format of interactions for electron system. Here, we define the charge density operator with spin σ at site i as $n_{i\sigma} = c^\dagger_{i\sigma} c_{i\sigma}$ and the total charge density operator at site i as $n_i = n_{i\uparrow} + n_{i\downarrow}$. Each parameters are specified by the following files, respectively;

Trans (trans.def): $t_{ij\sigma_1\sigma_2}$ in $\mathcal{H}_T$,

CoulombIntra (coulombintra.def): U_i in $\mathcal{H}_U$,

CoulombInter (coulombinter.def): V_{ij} in $\mathcal{H}_V$,

Hund (hund.def): J^{Hund}_{ij} in $\mathcal{H}_H$,

Exchange (exchange.def): J_{ij}^{Ex} in $\mathcal{H}_E$,

PairHop: J_{ij}^{Pair} in $\mathcal{H}_P$,

InterAll: $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$ in $\mathcal{H}_I$.

(4) Variational parameters to be optimized:

The variational parameters to be optimized are specified by using this categorized files. In mVMC, the variational wave function is given as

$$
\begin{aligned}
|\psi\rangle &= \mathcal{P}_G\mathcal{P}_J\mathcal{P}_{d-h}^{(2)}\mathcal{P}_{d-h}^{(4)}\mathcal{L}^S\mathcal{L}^K\mathcal{L}^P|\phi_{\mathrm{pair}}\rangle,\\
\mathcal{P}_G &= \exp\left[\sum_i g_i n_{i\uparrow}n_{i\downarrow}\right],\\
\mathcal{P}_J &= \exp\left[\frac{1}{2}\sum_{i\neq j} v_{ij}(n_i-1)(n_j-1)\right],\\
\mathcal{P}_{d-h}^{(2)} &= \exp\left[\sum_t\sum_{n=0}^{2}(\alpha_{2nt}^d\sum_i \xi_{i2nt}^d + \alpha_{2nt}^h\sum_i\xi_{i2nt}^h)\right],\\
\mathcal{P}_{d-h}^{(4)} &= \exp\left[\sum_t\sum_{n=0}^{4}(\alpha_{4nt}^d\sum_i \xi_{i4nt}^d + \alpha_{4nt}^h\sum_i\xi_{i4nt}^h)\right],\\
\mathcal{L}_S &= \frac{2S+1}{8\pi^2}\int d\Omega P_s(\cos\beta)\hat{R}(\Omega),\\
\mathcal{L}_K &= \frac{1}{N_s}\sum_{\boldsymbol{R}} e^{i\boldsymbol{K}\cdot\boldsymbol{R}}\hat{T}_{\boldsymbol{R}},\\
\mathcal{L}_P &= \sum_\alpha p_\alpha \hat{G}_\alpha,
\end{aligned}
$$

where $\Omega=(\alpha,\beta,\gamma)$ is the Euler angle, $\hat{R}(\Omega)$ is the rotational operator, $P_S(x)$ is the S-th polynomial, $\boldsymbol{K}$ is the momentum operator of the whole system and $\hat{T}_{\boldsymbol{R}}$ is the translational operators corresponding to the translational vector $\boldsymbol{R}$, $\hat{G}_\alpha$ is the point group operator, and p_α is the parity operator, respectively. The details of $\mathcal{P}_{d-h}^{(2)}$ and $\mathcal{P}_{d-h}^{(4)}$ are shown in [Tahara2008]. The one body part of the wavefunction is represented as the pair function of the real space:

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

where N is the number of electrons and N_s is the number of sites. The setting for optimizing variational parameters or not is given by the following files (the parameters for $\mathcal{L}_S$ are specified in the **ModPara** file).

Gutzwiller (gutzwilleridx.def): Set the target parameters g_i in $\mathcal{P}_G$ to be optimized.

Jastrow (jastrowidx.def): Set the target parameters v_{ij} in $\mathcal{P}_J$ to be optimized.

DH2: Set the target 2-site doublon-holon correlation factor $\alpha_{2nt}^{d(h)}$ in $\mathcal{P}_{d-h}^{(2)}$ to be optimized.

DH4: Set the target 4-site doublon-holon correlation factor $\alpha_{4nt}^{d(h)}$ in $\mathcal{P}_{d-h}^{(4)}$ to be optimized.

Orbital/OrbitalAntiParallel (orbitalidx.def): Set the pair orbital with anti-parallel spins $f_{i\uparrow j\downarrow}$ in $|\phi_{\mathrm{pair}}\rangle$ to be optimized.

OrbitalParallel: Set the pair orbital with anti-parallel spins $f_{i\sigma j\sigma}$ in $|\phi_{\mathrm{pair}}\rangle$ to be optimized.

OrbitalGeneral: Set the pair orbital with anti-parallel spins $f_{i\sigma j\sigma'}$ in $|\phi_{\mathrm{pair}}\rangle$ to be optimized.

TransSym (qptransidx.def): Set the the momentum projection operators $\mathcal{L}_K$ and the lattice translational projection operators $\mathcal{L}_P$.

(5) Initial variational parameters:

Set the initial values of the variational parameters. When the keyword is not setting, the corresponding parameters are given by random values as default values.

InGutzwiller: Set the initial values of g_i in $\mathcal{P}_G$.

InJastrow: Set the initial values of v_{ij} in $\mathcal{P}_J$.

InDH2: Set the initial values of $\alpha_{2nt}^{d(h)}$ in $\mathcal{P}_{d-h}^{(2)}$.

InDH4: Set the initial values of $\alpha_{4nt}^{d(h)}$ in $\mathcal{P}_{d-h}^{(4)}$.

InOrbital /InOrbitalAntiParallel: Set the initial values of $f_{i\uparrow j\downarrow}$ in $|\phi_{\mathrm{pair}}\rangle$.

InOrbitalParallel: Set the initial values of $f_{i\sigma j\sigma}$ in $|\phi_{\mathrm{pair}}\rangle$.

InOrbitalGeneral: Set the initial values of $f_{i\sigma j\sigma'}$ in $|\phi_{\mathrm{pair}}\rangle$.

(6) Output:

OneBodyG (greenone.def): Set the components of one-body green functions to output.

TwoBodyG (greentwo.def): Set the components of two-body green functions to output.

5.1 List file for Input files (namelist.def)

This file determines input filenames correlated with keywords. An example of the file format is shown as follows.

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

5.1.1 File format

[string01] [string02]

5.1.2 Parameters

- [string01]

  Type : string-type

  Description : Select a word from keywords.

- [string02]

  Type : string-type

  Description : An input filename which is correlated with keywords.

5.1.3 User rules

- After setting keywords at [string 01], half-width state is needed for writing a filename. You can set the filename freely.
- Keywords for input files are shown in Table [Table:Defs].
- Essential keywords are “CalcMod”, “ModPara” , “LocSpin”, “Orbital” and “TransSym”.
- Keywords can be set in random order.
- If keywords or filenames are incorrect, the program is terminated.
- When the head of line is #, the line is skipped.

Keywords	Details for corresponding files
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.
Gutzwiller	Gutzwiller factors.
Jastrow	Charge Jastrow factors.
DH2	2-site doublon-holon correlation factors.
DH4	4-site doublon-holon correlation factors.
Orbital *	Pair orbital factors with anti-parallel spins $f_{i\uparrow j\downarrow}$.
OrbitalAntiParallel	Pair orbital factors with anti-parallel spins $f_{i\uparrow j\downarrow}$.
OrbitalParallel	Pair orbital factors with parallel spins $f_{i\sigma j\sigma}$.
OrbitalGeneral	Pair orbital factors $f_{i\sigma_1 j\sigma_2}$.
TransSym *	Translational and lattice symmetry operation.
InGutzwiller	Initial values of Gutzwiller factors.
InJastrow	Initial values of charge Jastrow factors.
InDH2	Initial values of 2-site doublon-holon correlation factors.
InDH4	Initial values of 4-site doublon-holon correlation factors.
InOrbital	Initial values of pair orbital factors $f_{i\uparrow j\downarrow}$.
InOrbitalAntiParallel	Initial values of pair orbital factors $f_{i\uparrow j\downarrow}$.
InOrbitalParallel	Initial values of pair orbital factors $f_{i\sigma j\sigma}$.
InOrbitalGeneral	Initial values of pair orbital factors $f_{i\sigma_1 j\sigma_2}$.
OneBodyG	Output components for Green functions $\langle c^\dagger_{i\sigma} c_{j\sigma} \rangle$
TwoBodyG	Output components for Correlation functions $\langle c^\dagger_{i\sigma} c_{j\sigma} c^\dagger_{k\tau} c_{l\tau} \rangle$

5.2 ModPara file (modpara.def)

This file determines parameters for calculation. An example of the file format is shown as follows.

```
--------------------
Model_Parameters   0
--------------------
VMC_Cal_Parameters
```

(continues on next page)

(continued from previous page)

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

5.2.1 File format

- Lines 1 - 5: Header
- Line 6: [string01] [string02]
- Line 7: [string03] [string04]
- Line 8: Header
- Lines 9 - : [string05] [int01] (or [double01])

5.2.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : Set a keyword for header of output files.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : Set a header of output files. For example, the output file of one-body green's functions are named as **xxx_cisajs.dat**, where **xxx** is [string02].

- [string03]

  Type : string-type (blank parameter not allowed)

Description : Set a keyword for header of output files for variational parameters.

- [string04]

  Type : string-type (blank parameter not allowed)

  Description : Set a header of output files for variational parameters. For example, the output file of optimized variational parameters are named as **xxx_opt.dat**, where **xxx** is [string04].

- [string05]

  Type : string-type

  Description : Select a word from keywords.

- [int01] ([double01])

  Type : int (double)-type (blank parameter not allowed)

  Description : A parameter which is correlated with a keyword.

5.2.3 User rules

- From Line 9: After setting keywords at [string 01], a half-width blank is needed for setting a parameter.
- From Line 9: When the first character of the line is "-", the line is not read and skipped.

5.2.4 Keywords and parameters

- `NVMCCalMode`

  Type : int-type (default value: 0)

  Description : [0] Optimization of variational parameters, [1] Calculation of one body and two body Green's functions.

- `NLanczosMode`

  Type : int-type (default value: 0)

  Description : [0] Not using single Lanczos step, [1] Calculating energy by using Single Lanczos Step, [2] Calculating one body and two body Green's functions by using Single Lanczos Step (Condition: The options 1 and 2 can be selected when `NVMCCalMode` = 1).

- `NDataIdxStart`

  Type : int-type (default value: 0)

  Description : An integer for numbering of output files. For `NVMCCalMode` = 0 , `NDataIdxStart` is added at the end of the output files. For `NVMCCalMode` = 1, the files are outputted with the number from `NDataIdxStart` to `NDataIdxStart` + `NDataQtySmp`-1.

- `NDataQtySmp`

  Type : int-type (default value: 1)

  Description : The set number for outputted files (only used for `NVMCCalMode` = 1).

- `Nsite`

  Type : int-type (Positive integer)

  Description : The number of sites.

- `Nelectron`

  Type : int-type (Positive integer)

  Description : The number of electron pairs (the electron number is given by 2 `Nelectron`).

- `Ncond`

  Type : int-type (greater than 0)

  Description : The number of conduction electrons.

- `2Sz`

  Type : int-type

  Description : The value of $2S_z$. Since the electrons form pair, $2S_z$ must be even number.

- `NSPGaussLeg`

  Type : int-type (Positive integer, default value: 8)

  Description : The mesh number for the Gauss-legendre quadrature about β integration (S_y rotation) for the spin quantum-number projection in actual numerical calculation.

- `NSPStot`

  Type : int-type (greater than 0, default value: 0)

  Description : The spin quantum-number.

- `NMPTrans`

  Type : int-type (default value: 1)

  Description : The absolute value gives the number of the momentum and lattice translational quantum-number projection. When the value is negative, the mode of anti-periodic condition turns on. The quantum-number projection is used from the top to `NMPTrans` with the specified weight indicated in `TransSym` file. In the case of not applying the projection, this value must be equal to 1.

- `NSROptItrStep`

  Type : int-type (Positive integer, default value: 1000)

  Description : The whole step number to optimize variational parameters by SR method. Only used for `NVMCCalMode` =0.

- `NSROptItrSmp`

  Type : int-type (Positive integer, default value: `NSROptItrStep`/10)

  Description : In the `NSROptItrStep` step, the average values of the each variational parameters at the `NSROptItrStep` step are adopted as the optimized values. Only used for `NVMCCalMode` =0.

- `DSROptRedCut`

  Type : double-type (default value: 0.001)

  Description : The stabilized factor for the SR method by truncation of redundant directions corresponding to $\varepsilon_{\mathrm{wf}}$ in the ref. [Tahara2008].

- `DSROptStaDel`

  Type : double-type (default value: 0.02)

  Description : The stabilized factor for the SR method by modifying diagonal elements in the overwrap matrix corresponding to ε in the ref. [Tahara2008].

- `DSROptStepDt`

  Type : double-type

  Description : The time step using in the SR method.

- `NSROptCGMaxIter`

  Type : int-type (default value: 0)

  Description : The maximum number of CG steps for the SR method. If this is zero or negative, CG steps will be run as many as the size of S matrix at maximum. Only used for `NSRCG`!=0.

- `DSROptCGTol`

  Type : double-type (default value: 1.0e-10)

  Description : The convergence condition of a CG step in the SR method. CG method runs until the root mean square of the residues becomes below this value. Only used for `NSRCG`!=0.

- `NVMCWarmUp`

  Type : int-type (Positive integer, default value: 10)

  Description : Idling number for the Malkov chain Montecarlo Methods.

- `NVMCInterval`

  Type : int-type (Positive integer, default value: 1)

  Description : The interval step between samples. The local update will be performed `Nsite` × `NVMCInterval` times.

- `NVMCSample`

  Type : int-type (Positive integer, default value: 1000)

  Description : The sample numbers to calculate the expected values.

- `NExUpdatePath`

  Type : int-type (Positive integer)

  Description : The option for local update about exchange terms. 0: not update, 1: update for electron system. For Spin system, the value must be 2.

- `RndSeed`

  Type : int-type

  Description : The initial seed of generating random number. For MPI parallelization, the initial seeds are given by `RndSeed` +my rank+1 at each ranks.

- `NSplitSize`

  Type : int-type (Positive integer, default value: 1)

  Description : The number of processes of MPI parallelization.

- `NStore`

  Type : int-type (0 or 1, default value: 1)

  Description : The option of applying matrix-matrix product to calculate expected values $\langle O_k O_l \rangle$ (0: off, 1: on). This speeds up calculation but increases the amount of memory usage from $O(N_{\mathrm{p}}^2)$ to $O(N_{\mathrm{p}}^2) + O(N_{\mathrm{p}} N_{\mathrm{MCS}})$, where N_{p} is the number of the variational parameters and N_{MCS} is the number of Monte Carlo sampling.

- NSRCG

  Type : int-type (0 or 1, default value: 0)

  Description : The option of solving $Sx = g$ in the SR method without constructing S matrix [NeuscammanUmrigarChan]. (0: off, 1: on). This reduces the amount of memory usage from $O(N_{\mathrm{p}}^2) + O(N_{\mathrm{p}}N_{\mathrm{MCS}})$ to $O(N_{\mathrm{p}}) + O(N_{\mathrm{p}}N_{\mathrm{MCS}})$ when $N_{\mathrm{p}} > N_{\mathrm{MCS}}$.

5.3 LocSpin file (locspn.def)

This file determines sites with localized spins. An example of the file format is shown 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
```

5.3.1 File format

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

5.3.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of localized spins. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of localized spins.

- [int02]

  Type : int-type (blank parameter not allowed)

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

- [int03]

  Type : int-type (blank parameter not allowed)

  Description : An integer for selecting an electron state whether localized spin or itinerant electron states (0: Itinerant electron state, 1: localized spin state with $S = 1/2$).

5.3.3 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` $\leq$ [int02].

5.4 Trans file (trans.def)

The Hamiltonian for general one-body interactions

$$\mathcal{H}_T = -\sum_{ij\sigma_1\sigma_2} t_{ij\sigma_1\sigma_2} c^{\dagger}_{i\sigma_1} c_{j\sigma_2},$$

is added to the whole Hamiltonian by setting the parameters $t_{ij\sigma_1\sigma 2}$. An example of the file format is shown 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
...
```

5.4.1 File format

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

- Lines 6-: [int02] [int03] [int04] [int05] [double01] [double02]

5.4.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

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

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of transfer integrals.

- [int02], [int04]

  Type : int-type (blank parameter not allowed)

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

- [int03], [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a spin index,

  0: up-spin,

  1: down-spin.

- [double01]

  Type : double-type (blank parameter not allowed)

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

- [double02]

  Type : double-type (blank parameter not allowed)

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

5.4.3 Use rules

- Headers cannot be omitted.
- Blank line is not allowed.
- 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 out of range from the defined values.
- Since Hamiltonian must be Hermitian, the following relation must be satisfied, $t_{ij\sigma_1\sigma_2} = t^{\dagger}_{ji\sigma_2\sigma_1}$.

5.5 InterAll file

The Hamiltonian for general two-body interactions

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

is added to the whole Hamiltonian by setting the parameters $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4}$. An example of file format is shown 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
...
```

5.5.1 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]

5.5.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

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

- [int01]

  Type : int-type (blank parameter not allowed)

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

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

  Type : int-type (blank parameter not allowed)

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

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

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a spin index,

  0: up-spin,

  1: down-spin.

- [double01]

  Type : double-type (blank parameter not allowed)

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

- [double02]

  Type : double-type (blank parameter not allowed)

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

5.5.3 Use rules

- Headers cannot be omitted.
- Since Hamiltonian must be Hermitian, the following relation must be satisfied, $I_{ijkl\sigma_1\sigma_2\sigma_3\sigma_4} = I^{\dagger}_{lkji\sigma_4\sigma_3\sigma_2\sigma_1}$.
- 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 out of range from the defined values.

5.6 CoulombIntra file (coulombintra.def)

The Hamiltonian for the coulombintra interactions

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

is added to the whole Hamiltonian by setting U_i. An example of the file format is shown as follows.

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

5.6.1 File format

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

5.6.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of on-site interactions. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of on-site interactions.

- [int02]

  Type : int-type (blank parameter not allowed)

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

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : A value for U_i.

5.6.3 Use rules

- Headers cannot be omitted.
- 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 out of range from the defined values.

5.7 CoulombInter file (coulombinter.def)

The Hamiltonian for the coulombinter interactions

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

is added to the whole Hamiltonian by setting V_{ij}. An example of the file format is shown as follows.

```
======================
NCoulombInter 6
======================
========CoulombInter ======
======================
```

(continues on next page)

(continued from previous page)

```
0     1  1.0000
1     2  1.0000
2     3  1.0000
3     4  1.0000
4     5  1.0000
5     0  1.0000
```

5.7.1 File format

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

5.7.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of off-site interactions. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of off-site interactions.

- [int02], [int03]

  Type : int-type (blank parameter not allowed)

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

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : A value for V_{ij}.

5.7.3 Use rules

- Headers cannot be omitted.
- 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] are out of range from the defined values.

5.8 Hund file (hund.def)

The Hamiltonian for Hund couplings

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

is added to the whole Hamiltonian by setting the parameters J_{ij}^{Hund}. An example of the file format is shown 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
```

5.8.1 File format

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

5.8.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of Hund couplings. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of Hund couplings.

- [int02], [int03]

  Type : int-type (blank parameter not allowed)

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

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : A value for J_{ij}^{Hund}.

5.8.3 Use rules

- Headers cannot be omitted.
- 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] are out of range from the defined values.

5.9 PairHop file

The Hamiltonian for PairHop couplings

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

is added to the whole Hamiltonian by setting the parameters J_{ij}^{Pair}. An example of the file format is shown 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
```

5.9.1 File format

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

5.9.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of PairHop couplings. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of PairHop couplings.

- [int02], [int03]

  Type : int-type (blank parameter not allowed)

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

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : A value for J_{ij}^{Pair}.

5.9.3 Use rules

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

5.10 Exchange file (exchange.def)

The Hamiltonian for exchange couplings

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

is added to the whole Hamiltonian by setting J_{ij}^{Ex}. An example of the file format is shown 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
```

5.10.1 File format

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

5.10.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of Exchange couplings. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of Exchange couplings.

- [int02], [int03]

  Type : int-type (blank parameter not allowed)

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

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : A value for J_{ij}^{Ex}.

5.10.3 Use rules

- Headers cannot be omitted.
- 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] are out of range from the defined values.

5.11 Gutzwiller file (gutzwiller.def)

This file sets the calculation conditions of Gutzwiller factors

$$\mathcal{P}_G = \exp\left[\sum_i g_i n_{i\uparrow} n_{i\downarrow}\right].$$

A site number i and the variational parameters g_i are specified. An example of the file format is shown as follows.

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

5.11.1 File format

In the following, we define the whole number of sites as N_s and variational parameters as N_g, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ N_s): [int03] [int04]
- Lines (6+ N_s) - (5+ N_s + N_g): [int05] [int06]

5.11.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters g_i. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters g_i.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters g_i. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters g_i (0: double, 1: complex).

- [int03]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03] $<$ `Nsite`).

- [int04]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters g_i ($0 \leq$ [int04] $<$ [int01]).

- [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters ($0 \leq$ [int05] $<$ [int01]).

- [int06]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int05] to be optimized or not (0: not optimize, 1: optimize).

5.11.3 User rules

- Headers cannot be omitted.
- A program is terminated, when components of variational parameters are double counted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int06] are out of range from the defined values.

5.12 Jastrow file (jastrow.def)

This file sets the calculation conditions of Jastrow factors

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

Site numbers i j, and the variational parameters v_{ij} are specified. An example of the file format is shown as follows.

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

5.12.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_j, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ $N_s \times (N_s - 1)$): [int03] [int04] [int05]
- Lines (6+ $N_s \times (N_s - 1)$) - (5+ $N_s \times (N_s - 1)$ + N_j): [int06] [int07]

5.12.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

Description : A keyword for total number of variational parameters v_{ij}. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters v_{ij}.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters v_{ij}. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters v_{ij} (0: double, 1: complex).

- [int03], [int04]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03], [int04] $<$ `Nsite`).

- [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters v_{ij} ($0 \leq$ [int05] $<$ [int01]).

- [int06]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters ($0 \leq$ [int06] $<$ [int01]).

- [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int06] to be optimized or not (0: not optimize, 1: optimize).

5.12.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int07] are out of range from the defined values.

5.13 DH2 file

This file sets the calculation conditions of 2-site doublon-holon correlation factors

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

A site number i, the two sites around i site and the variational parameters $\alpha_{2nt}^{d(h)}$ which have t kinds at each sites are specified. The details of the parameters $\alpha_{2nt}^{d(h)}$ and the operator $\xi_{i2nt}^{d(h)}$ are shown in ref. [Tahara2008]. An example of the file format is shown as follows.

```
====================================
NDoublonHolon2siteIdx 2
ComplexType 0
====================================
====================================
   0     5    15     0
   0    13     7     1
 (continue...)
  15     8     2     1
   0     1
(continue...)
  11     1
```

5.13.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_{dh2}, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ $N_s \times N_{\mathrm{dh2}}$): [int03] [int04] [int05] [int06]
- Lines (6+ $N_s \times N_{\mathrm{dh2}}$) - (5+ $(N_s + 6) \times N_{\mathrm{dh2}}$): [int07] [int08]

5.13.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters (0: double, 1: complex).

- [int03], [int04], [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03], [int04], [int05] $<$ `Nsite`).

- [int06]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters ($0 \leq$ [int06] $<$ [int01]).

- [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters. The value is $(2n+s)\times$ [int01]:math:+*t*, where n, s and t are given by the following relation:

  - n: The number of doublon (holon) around the center site (0, 1, 2),
  - s: When the center is doublon (holon), s=0 (1),
  - t: The kind of variational parameters (0, $\cdots$ [int1]-1).

- [int08]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int07] to be optimized or not (0: not optimize, 1: optimize).

5.13.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int08] are out of range from the defined values.

5.14 DH4 file

This file sets the calculation conditions of 4-site doublon-holon correlation factors

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

A site number i, the four sites around i site and the variational parameters $\alpha_{4nt}^{d(h)}$ which have t kinds at each sites are specified. The details of the parameters $\alpha_{4nt}^{d(h)}$ and the operator $\xi_{i4nt}^{d(h)}$ are shown in ref. [Tahara2008]. An example of the file format is shown as follows.

```
==================================
NDoublonHolon4siteIdx 1
ComplexType 0
==================================
==================================
   0     1     3     4    12     0
   1     2     0     5    13     0
```

(continues on next page)

(continued from previous page)

```
 (continue...)
  15    12   14    3   11    0
   0     1
(continue...)
   9     1
```

5.14.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_{dh4}, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ $N_s \times N_{\mathrm{dh4}}$): [int03] [int04] [int05] [int06] [int07] [int08]
- Lines (6+ $N_s \times N_{\mathrm{dh4}}$) - (5+ $(N_s + 10) \times N_{\mathrm{dh4}}$): [int09] [int10]

5.14.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters (0: double, 1: complex).

- [int03], [int04], [int05], [int06], [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03], $\cdots$, [int07] $<$ `Nsite`).

- [int08]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters ($0 \leq$ [int08] $<$ [int01]).

- [int09]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters. The value is $(2n+s)\times$ [int01]:math:+t, where n, s and t are given by the following relation:

  - n: The number of doublon (holon) around the center site (0, 1, 2, 3, 4),
  - s: When the center is doublon (holon), s=0 (1),
  - t: The kind of variational parameters (0, $\cdots$ [int1]-1).

- [int10]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int09] to be optimized or not (0: not optimize, 1: optimize).

5.14.3 User rules

- Headers cannot be omitted.
- A program is terminated, when components of variational parameters are double counted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int10] are out of range from the defined values.

5.15 Orbital/OrbitalAntiParallel file (orbitalidx.def)

This file sets the calculation conditions of pair orbitals

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

Site numbers i, j and the variational parameters f_{ij} are indicated. An example of the file format is shown as follows.

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

5.15.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_{o}, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ N_s^2): [int03] [int04] [int05] [int06]
- Lines (6+ N_s^2)- (5+ $N_s^2 + N_{\mathrm{o}}$): [int06] [int07]

5.15.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters (0: double, 1: complex).

- [int03], [int04]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index (0 $\leq$ [int03], [int04] $<$ `Nsite`).

- [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters (0 $\leq$ [int05] $<$ [int01]).

- [int06]

  Type : int-type

  Description : When the mode of the anti-periodic condition turns on (the mode turns on when the value of `NMPTrans` in `ModPara` file is negative), the sign of f_{ij} is specified by setting [int06] $= \pm 1$. This term can be omitted when the mode of the anti-periodic condition is off.

- [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters (0 $\leq$ [int06] $<$ [int01]).

- [int08]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int06] to be optimized or not (0: not optimize, 1: optimize).

5.15.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int09] are out of range from the defined values.

5.16 OrbitalParallel file

This file sets the calculation conditions of pair orbitals

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

Site numbers i, j, the spin index σ and the variational parameters $f_{i\sigma j\sigma}$ are indicated. The indexes of $f_{i\sigma j\sigma}$ must satisfy the condition $i < j$sigma = 0‘ or 1 and process will terminate when the condition is broken. An example of the file format is shown as follows.

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

5.16.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_{o}, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header

- Lines 6 - (5+ $N_s * (N_s - 1)/2$): [int03] [int04] [int05] [int06]
- Lines (6+ $N_s * (N_s - 1)/2$)- (5+ $N_s * (N_s - 1)/2 + N_o$): [int06] [int07]

5.16.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters (0: double, 1: complex).

- [int03], [int04]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03], [int04] $<$ `Nsite`).

- [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters ($0 \leq$ [int05] $<$ [int01]).

- [int06]

  Type : int-type

  Description : When the mode of the anti-periodic condition turns on (the mode turns on when the value of `NMPTrans` in `ModPara` file is negative), the sign of f_{ij} is specified by setting [int06] $= \pm 1$. This term can be omitted when the mode of the anti-periodic condition is off.

- [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters ($0 \leq$ [int06] $<$ [int01]).

- [int08]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int06] to be optimized or not (0: not optimize, 1: optimize).

5.16.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int09] are out of range from the defined values.

5.17 OrbitalGeneral file

This file sets the calculation conditions of pair orbitals

$$|\phi_{\mathrm{pair}}\rangle = \left[\sum_{i,j=1}^{N_s}\sum_{\sigma_1,\sigma_2} f_{i\sigma_1 j\sigma_2} c^{\dagger}_{i\sigma_1} c^{\dagger}_{j\sigma_2}\right]^{N/2} |0\rangle.$$

Site numbers i, j, spin indexes σ_1, σ_2 and the variational parameters $f_{i\sigma_1 j\sigma_2}$ are indicated. The indexes of $f_{i\sigma_1 j\sigma_2}$ must satisfy the condition $i + \sigma_1 N_s < j + \sigma_2 N_s$, where $\sigma_i = 0$ or 1 and process will terminate when the condition is broken. An example of the file format is shown as follows.

12.5cm

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

5.17.1 File format

In the following, we define the total number of sites as N_s and variational parameters as N_{o}, respectively. A total number of variational parameters N_p is given by N_s^2, $2N_s^2 - N_s$ in Sz conserved and S_z unconserved system, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Line 3: [string02] [int02]
- Lines 4 - 5: Header
- Lines 6 - (5+ N_p): [int03] [int04] [int05] [int06] [int07] [int08]
- Lines (6+ N_p) - (5+ $N_p + N_{\mathrm{o}}$): [int09] [int10]

5.17.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [string02]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for indicating the double or complex type of variational parameters. You can freely give a name of the keyword.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer indicates the double or complex type of variational parameters (0: double, 1: complex).

- [int03], [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int03], [int04] $<$ `Nsite`).

- [int04], [int06]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a spin index (0 :↑ spin, 1 :↓ spin).

- [int07]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters ($0 \leq$ [int07] $<$ [int01]).

- [int08]

  Type : int-type

  Description : When the mode of the anti-periodic condition turns on (the mode turns on when the value of `NMPTrans` in `ModPara` file is negative), the sign of $f_{i\sigma_1 j\sigma_2}$ is specified by setting [int08] $= \pm 1$. This term can be omitted when the mode of the anti-periodic condition is off.

- [int09]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of variational parameters ($0 \leq$ [int09] $<$ [int01]).

- [int10]

  Type : int-type (blank parameter not allowed)

  Description : An integer to select the target of variational parameters indicated at [int09] to be optimized or not (0: not optimize, 1: optimize).

5.17.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.
- A program is terminated, when [int02] - [int10] are out of range from the defined values.

5.18 TransSym file (qptransidx.def)

This file sets the weight and corresponding site numbers of momentum projection $\mathcal{L}_K = \frac{1}{N_s}\sum_{\boldsymbol{R}} e^{i\boldsymbol{K}\cdot\boldsymbol{R}}\hat{T}_{\boldsymbol{R}}$ and lattice translational projection $\mathcal{L}_P = \sum_\alpha p_\alpha \hat{G}_\alpha$. The patterns of projection are indicated by $(\alpha, \boldsymbol{R})$. We note that the weight must be equal to 1.0 when the projection is not done. An example of the file format is shown as follows.

12.5cm

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

5.18.1 File format

In the following, we define the total number of sites as N_s and projection patterns as N_{TS}, respectively.

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3 - 5: Header
- Lines 6 - (5+ N_{TS}): [int02] [double01]
- Lines (6+ N_{TS}) - (5+ $(N_s+1)\times N_{\mathrm{TS}}$): [int03] [int04] [int05] [int06]

5.18.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of projection patterns. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

Description : An integer giving total number of projection patterns.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving the projection pattern $(\alpha, \boldsymbol{R})$ ($0 \leq$ [int02] $<$ [int01]).

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : The weight $p_\alpha \cos(\boldsymbol{K} \cdot \boldsymbol{R})$ of the projection pattern $(\alpha, \boldsymbol{R})$.

- [int03]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving kinds of the projection pattern $(\alpha, \boldsymbol{R})$ ($0 \leq$ [int03] $<$ [int01]).

- [int04], [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a site index ($0 \leq$ [int04], [int05] $<$ `Nsite`). The site number [int05] is given by applying the translation and point group transformation indicated by [int03] to the site [int04].

- [int06]

  Type : int-type

  Description : When the mode of the anti-periodic condition turns on (the mode turns on when the value of `NMPTrans` in `ModPara` file is negative), the sign of the translational operator is specified by setting [int06] $= \pm 1$. This term can be omitted when the mode of the anti-periodic condition is off.

5.18.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of projection patterns defined in this file.
- A program is terminated, when [int02] - [int06] are out of range from the defined values.

5.19 Files to set initial values of variational parameters

This file sets the initial values of variational parameters. The kinds of variational parameters are specified by setting the following keywords in `List` file (namelist.def): `InGutzwiller`, `InJastrow`, `InDH2`, `InDH4`, `InOrbital`, `InOrbitalAntiParallel`, `InOrbitalParallel`, `InOrbitalGeneral`. The file format is common and an example of the `InJastrow` file is shown as follows.

```
=======================
NJastrowIdx  28
=======================
== i_j_JastrowIdx  ===
=======================
0 -8.909963465082626488e-02  0.00000000000000000e+00
1  5.521681211878626955e-02  0.00000000000000000e+00
(continue...)
27 -9.017586139930480749e-02  0.00000000000000000e+00
```

5.19.1 File format

In the following, we define the total number of variational parameters as N_v.

- Line 1: Header
- Line 2: [string01] [int01]
- Lines 3 - 5: Header
- Lines 6 - (5+ N_v): [int03] [double01] [double02]

5.19.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

  Description : A keyword for total number of variational parameters. You can freely give a name of the keyword.

- [int01]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving total number of variational parameters.

- [int02]

  Type : int-type (blank parameter not allowed)

  Description : An integer setting kinds of variational parameters ($0 \leq$ [int02] $<$ [int01]).

- [double01]

  Type : double-type (blank parameter not allowed)

  Description : The real part of the variational parameter indicated by [int01].

- [double02]

  Type : double-type

  Description : The imaginary part of the variational parameter indicated by [int01].

5.19.3 User rules

- Headers cannot be omitted.
- A program is terminated, when [int01] is different from the total number of variational parameters defined in this file.

5.20 OneBodyG file (greenone.def)

This file determines the target components to calculate and output one-body Green's function $\langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2} \rangle$. An example of file format is shown 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
```

5.20.1 File format

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

5.20.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

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

- [int01]

  Type : int-type (blank parameter not allowed)

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

- [int02], [int04]

  Type : int-type (blank parameter not allowed)

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

- [int03], [int05]

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a spin index,

  0: up-spin,

  1: down-spin.

5.20.3 Use rules

- Headers cannot be omitted.
- 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 out of range from the defined values.

5.21 TwoBodyG file (greentwo.def)

This file determines the target components to calculate and output two-body Green's function $\langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2} c^{\dagger}_{k\sigma_3} c_{l\sigma_4} \rangle$. For Spin, the condition $i = j$ and $k = l$ must be satisfied. An example of file format is shown 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
    ...
```

5.21.1 File format

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

5.21.2 Parameters

- [string01]

  Type : string-type (blank parameter not allowed)

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

- [int01]

  Type : int-type (blank parameter not allowed)

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

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

  Type : int-type (blank parameter not allowed)

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

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

  Type : int-type (blank parameter not allowed)

  Description : An integer giving a spin index,
  0: up-spin,
  1: down-spin.

5.21.3 Use rules

- Headers cannot be omitted.
- 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 out of range from the defined values.

CHAPTER

SIX

OUTPUT FILES

The list of output files are shown in Table [Table:Output], where *** and xxx are the header indicated by `CParaFileHead`,‘‘CDataFileHead‘‘ in `ModPara` file, respectively. yyy is a number given by `NDataIdxStart` ⋯ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file and zzz is a number given by `NDataIdxStart` in `ModPara`.

Name	Details for corresponding files
***_opt.dat	All optimized parameters.
***_gutzwiller_opt.dat	Optimized gutzwiller factors.
***_jastrow_opt.dat	Optimized jastrow factors.
***_doublonHolon2site_opt.dat	Optimized 2-site doublon-holon correlation factors.
***_doublonHolon4site_opt.dat	Optimized 4-site doublon-holon correlation factors.
***_orbital_opt.dat	Optimized pair orbital factors.
xxx_out_yyy.dat	Energy and deviation.
xxx_var_yyy.dat	Progress information for optimizing variational parameters.
xxx_CalcTimer.dat	Computation time for each processes.
xxx_time_zzz.dat	Progress information for MonteCalro samplings.
xxx_cisajs_yyy.dat	One body Green’s functions.
xxx_cisajscktalt_yyy.dat	Correlation functions.

6.1 Output file for variational parameters (***_opt.dat)

The average and deviation values of variational parameters and the energy optimized by the SR method are outputted in the following order:

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

The type of average values is a complex number, while that of the deviation is a real number. Since the initial values of all variational parameters are specified at the beginning of the calculation, the calculation of physical quantities is done by using this file file. Here, *** is the header indicated by `CParaFileHead` in `ModPara` file.

6.2 Output files for variational parameters at each steps (xxx_var_yyy.dat)

The average and deviation values of variational parameters and the energy optimized by the SR method are postscripted by the order same as $$$_opt.dat file (the deviation is always outputted as 0). Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and yyy is a number given by `NDataIdxStart` ⋯ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file.

6.3 Output file for gutzwiller factors (***_gutzwiller_opt.dat)

The optimized Gutzwiller factors by SR method are outputted. The file format is same as the `InGutzwiller` file defined in Sec. *Files to set initial values of variational parameters*.

6.4 Output file for jastrow factors (***_jastrow_opt.dat)

The optimized Jastrow factors by SR method are outputted. The file format is same as the `InJastrow` file defined in Sec. *Files to set initial values of variational parameters*.

6.5 Output file for doublonHolon 2-site factors (***_doublonHolon2site_opt.dat)

The optimized 2-site doublon-holon correlation factors by SR method are outputted. The file format is same as the `InDH2` file defined in Sec. *Files to set initial values of variational parameters*.

6.6 Output file for doublonHolon 4-site factors (***_doublonHolon4site_opt.dat)

The optimized 4-site doublon-holon correlation factors by SR method are outputted. The file format is same as the `InDH4` file defined in Sec. *Files to set initial values of variational parameters*.

6.7 Output file for pair orbitals (***_orbital_opt.dat)

The optimized pair orbitals by SR method are outputted. The file format is same as the `InOrbital` file defined in Sec. *Files to set initial values of variational parameters*.

6.8 xxx_out_yyy.dat

The calculation information at each bins are outputted in the order:

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

The type of $\langle H\rangle$ is a complex number, and that of the others is a real number. Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and yyy is a number given by `NDataIdxStart` $\cdots$ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file. An example of outputted file is shown as follows.

```
1.15198376570421299e+01  8.12462241836090948e-01  \
1.61908295543888726e+02  2.01990520393908495e-01
1.28848261381742315e+01  5.00690373326284743e-01
1.97200032527695782e+02  1.82450519369579289e-01
1.30889720601188042e+01  5.70124488695657016e-01  \
```

(continues on next page)

(continued from previous page)

```
2.072610167083121837e+02  2.029162857569105916e-01
...
```

6.9 xxx_CalcTimer.dat

After finishing calculation, the processing time is outputted in the order of the name, the number assigned by the process and the seconds at each processes. An example of outputted file is shown as follows.

```
All                          [0]     15.90724
Initialization               [1]      0.04357
  read options              [10]      0.00012
  ReadDefFile               [11]      0.00082
  SetMemory                 [12]      0.00002
  InitParameter             [13]      0.03026
VMCParaOpt                   [2]     15.86367
  VMCMakeSample              [3]     12.85650
...
```

6.10 xxx_time_zzz.dat

The calculation information at each bins are outputted in the order of the sampling number, the acceptance ratio for hopping and exchange term (acc_hopp, acc_ex), trial numbers to update for hopping and exchange term (n_hopp, n_ex) and the time stamp. Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and zzz is a number given by `NDataIdxStart` in `ModPara`. An example of outputted file is shown as follows.

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

6.11 xxx_cisajs_yyy.dat

This file is the outputted files for one-body Green's function $\langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2} \rangle$. The target components are set in the input file with the keyword "OneBodyG". Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and yyy is a number given by `NDataIdxStart` $\cdots$ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file.

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
```

(continues on next page)

(continued from previous page)

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

6.11.1 File format

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

6.11.2 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 σ_1 and σ_2, respectively.

- [double01], [double02]

  Type : Double

  Description : The value of $\langle c^\dagger_{i\sigma_1} c_{j\sigma_2} \rangle$.
  [double01] and [double02] show the real and imaginary part of $\langle c^\dagger_{i\sigma_1} c_{j\sigma_2} \rangle$, respectively.

6.12 xxx_cisajscktalt_yyy.dat

This file is the outputted files for the two-body Green's function $\langle c^\dagger_{i\sigma_1} c_{j\sigma_2} c^\dagger_{k\sigma_3} c_{l\sigma_4} \rangle$. The target components are set in the input file with the keyword "TwoBodyG". Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and yyy is a number given by `NDataIdxStart` $\cdots$ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file. 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
```

(continues on next page)

(continued from previous page)

```
0    0    0    0    3    0    3    0 0.2512810778 0.0000000000
0    0    0    0    3    1    3    1 0.1939965962 0.0000000000
...
```

6.12.1 File format

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

6.12.2 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 σ_1, σ_2, σ_3, and σ_4, respectively.

- [double01], [double02]

  Type : Double

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

6.13 xxx_ls_out_yyy.dat

This file is the outputted files for $\langle H \rangle$, $\langle H^2 \rangle$, and the optimized parameter α obtained by Power Lanczos method. This file is outputted when `NVMCCalMode` = 1, `NLanczosmode` = 1 or 2 are set in `ModPara` file. Here, xxx is the header indicated by `CDataFileHead` in `ModPara` file and yyy is a number given by `NDataIdxStart` $\cdots$ `NDataIdxStart` + `NDataQtySmp`, where both `NDataIdxStart` and `NDataQtySmp` are defined in `ModPara` file.

6.14 xxx_ls_cisajs_yyy.dat

This file is the outputted files for one-body Green's function $\langle c^\dagger_{i\sigma_1} c_{j\sigma_2} \rangle$ obtained by Power Lanczos method. The file format is same as the xxx_cisajs_yyy.dat file. This file is outputted when `NVMCCalMode` = 1, `NLanczosmode` = 2 are set in `ModPara` file.

6.15 xxx_ls_cisajscktalt_yyy.dat

This file is the outputted files for the two-body Green's function $\langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2} c^{\dagger}_{k\sigma_3} c_{l\sigma_4} \rangle$ obtained by Power Lanczos method. The file format is same as the xxx_cisajscktalt_yyy.dat file. This file is outputted when `NVMCCalMode` = 1, `NLanczosmode` = 2 are set in `ModPara` file.

CHAPTER

SEVEN

ALGORITHM

7.1 Variational Monte Calro Method

The variational Monte Carlo (VMC) method is a method for calculating approximate wave functions of a ground state and low-lying excited states by optimizing variational parameters included in a trial wave function. In calculating expectation values of physical quantities for the trial wave functions, the Markov chain Monte Carlo method is applied for efficient important sampling.

In the mVMC package, we choose a spatial configuration for electrons as a complete set of bases in sampling:

$$|x\rangle = \prod_{n=1}^{N_e/2} c^\dagger_{r_{n\uparrow}} \prod_{n=1}^{N_e/2} c^\dagger_{r_{n\downarrow}} |0\rangle,$$

where $r_{n\sigma}$ is a position of n-th electron with $\sigma(=\uparrow \text{ or } \downarrow)$ spin, and $c^\dagger_{r_{n\sigma}}$ is a creation operator of electrons. By using this basis set, the expectation value of an operator A is expressed as

$$\langle A \rangle = \frac{\langle \psi | A | \psi \rangle}{\langle \psi | \psi \rangle} = \sum_x \frac{\langle \psi | A | x \rangle \langle x | \psi \rangle}{\langle \psi | \psi \rangle}.$$

If we define a weight of the Markov chain Monte Carlo method as

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

we can rewrite $\langle A \rangle$ in the following form:

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

By using this form, the Markov chain Monte Carlo method is performed for sampling with respect to x. The local Green's function $G_{ij\sigma\sigma'}(x)$, which is defined as

$$G_{ij\sigma\sigma'}(x) = \frac{\langle \psi | c^\dagger_{i\sigma} c_{j\sigma'} | \psi \rangle}{\langle \psi | x \rangle},$$

is also evaluated by the same sampling method by taking $A = c^\dagger_{i\sigma} c_{j\sigma'}$. We adopt the Mersenne twister method as a random number generator for sampling [Mutsuo2008].

7.2 Bogoliubov representation

In the VMC calculation for spin systems, we use the Bogoliubov representation. In the input files defining the one-body term (`transfer`) and the two-body term (`InterAll`), and the output files for correlation functions, the

indices must be assigned by the Bogoliubov representation, in which the spin operators are generally expressed by creation/annihilation operators of fermions as

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

Since the present package support only $S = 1/2$ spin systems, the Bogoliubov representation obtained by substituting $S = 1/2$ into the above equations is used.

7.3 Properties of the Pfaffian-Slater determinant

In this section, we explain some properties of the Pfaffian-Slater determinant. We derive the general relation between a Pfaffian-Slater determinant and a single Slater determinant in *Antiparallel Pfaffian* and *General Pfaffian* . We also discuss meaning of the singular value decomposition of coefficients f_{ij} in *SVD*.

7.3.1 Relation between f_{ij} and $\Phi_{in\sigma}$ (the case of the anti-parallel pairing)

In the many-variable variational Monte Carlo (mVMC) method, the one-body part of the trial wave function is expressed by the Pfaffian Slater determinant defined as

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

where N_s is number of sites, N_e is number of total particles, and f_{ij} are variational parameters. For simplicity, we assume that f_{ij} are a real number. The single Slater determinant is defined as

$$
\begin{aligned}
|\phi_{\mathrm{SL}}\rangle &= \Big(\prod_{n=1}^{N_e/2} \psi_{n\uparrow}^{\dagger} \Big) \Big(\prod_{m=1}^{N_e/2} \psi_{m\downarrow}^{\dagger} \Big) |0\rangle, \\
\psi_{n\sigma}^{\dagger} &= \sum_{i=1}^{N_s} \Phi_{in\sigma} c_{i\sigma}^{\dagger},
\end{aligned}
$$

Here, $\Phi_{in\sigma}$ is an orthonormal basis, i.e., satisfies

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

where δ_{nm} is the Kronecker's delta. From this orthogonality, we can prove the relation

$$
\begin{aligned}
{}_{+} &= \delta_{nm}, \\
G_{ij\sigma} = \langle c_{i\sigma}^{\dagger} c_{j\sigma} \rangle &= \frac{\langle \phi_{\mathrm{SL}} | c_{i\sigma}^{\dagger} c_{j\sigma} | \phi_{\mathrm{SL}} \rangle}{\langle \phi_{\mathrm{SL}} | \phi_{\mathrm{SL}} \rangle} \\
&= \sum_{n} \Phi_{in\sigma} \Phi_{jn\sigma}.
\end{aligned}
$$

Next, let us prove the relation between f_{ij} and $\Phi_{in\sigma}$ by modifying $|\phi_{\rm SL}\rangle$. By the commutation relation for $\psi^\dagger_{n\sigma}$, $|\phi_{\rm SL}\rangle$ is rewritten as

$$|\phi_{\rm SL}\rangle \propto \prod_{n=1}^{N_e/2} \left(\psi^\dagger_{n\uparrow}\psi^\dagger_{\mu(n)\downarrow}\right)|0\rangle,$$

where $\mu(n)$ represents permutation of a sequence of natural numbers, $n = 1, 2, \cdots, N_e/2$. For simplicity, let us take identity permutation ($\mu(n) = n$). By defining $K^\dagger_n = \psi^\dagger_{n\uparrow}\psi^\dagger_{n\downarrow}$, and by using the relation $K^\dagger_n K^\dagger_m = K^\dagger_m K^\dagger_n$, we can derive the relation

$$\begin{aligned}|\phi_{\rm SL}\rangle &\propto \prod_{n=1}^{N_e/2} \left(\psi^\dagger_{n\uparrow}\psi^\dagger_{n\downarrow}\right)|0\rangle = \prod_{n=1}^{N_e/2} K^\dagger_n|0\rangle \\ &\propto \Big(\sum_{n=1}^{\frac{N_e}{2}} K^\dagger_n\Big)^{\frac{N_e}{2}}|0\rangle = \Big(\sum_{i,j=1}^{N_s}\Big[\sum_{n=1}^{\frac{N_e}{2}}\Phi_{in\uparrow}\Phi_{jn\downarrow}\Big]c^\dagger_{i\uparrow}c^\dagger_{j\downarrow}\Big)|0\rangle.\end{aligned}$$

This result indicates that f_{ij} is expressed by the coefficients of the single Slater determinant as

$$f_{ij} = \sum_{n=1}^{\frac{N_e}{2}}\Phi_{in\uparrow}\Phi_{jn\downarrow}.$$

We note that this is one of a number of possible expressions of f_{ij} derived from one single Slater determinant. Since f_{ij} depends not only on the choice of the pairing degrees of freedom (i.e., the choice of $\mu(n)$) but also on the choice of the gauge degrees of freedom (i.e., the sign of $\Phi_{in\sigma}$), the parameter f_{ij} has huge redundancy.

7.3.2 Relation between F_{IJ} and Φ_{In} (the case of the general pairing)

We extend the relation between the Pfaffian-Slater wave function and the single Slater wave function into the general pairing case including the spin-parallel pairing. We define the Pfaffian-Slater wave function and the single Slater wave function as

$$\begin{aligned}|\phi_{\rm Pf}\rangle &= \Big(\sum_{I,J=1}^{2N_s} F_{IJ}c^\dagger_I c^\dagger_J\Big)^{N_e/2}|0\rangle, \\ |\phi_{\rm SL}\rangle &= \Big(\prod_{n=1}^{N_e}\psi^\dagger_n\Big)|0\rangle, \quad \psi^\dagger_n = \sum_{I=1}^{2N_s}\Phi_{In}c^\dagger_I,\end{aligned}$$

respectively, where I, J denote the site index including the spin degrees of freedom. By the similar argument as the anti-parallel pairing case, we can derive the following relation:

$$F_{IJ} = \sum_{n=1}^{\frac{N_e}{2}}\Big(\Phi_{I,2n-1}\Phi_{J,2n} - \Phi_{J,2n-1}\Phi_{I,2n}\Big).$$

Because this relation hold for the case of anti-parallel pairing, we employ this relation in mVMC ver 1.0 and later.

7.3.3 Singular value decomposition of f_{ij}

We define matrices F, $\Phi_\uparrow$, $\Phi_\downarrow$, and Σ as

$$\begin{aligned}&(F)_{ij} = f_{ij}, \quad (\Phi_\uparrow)_{in} = \Phi_{in\uparrow}, \quad (\Phi_\downarrow)_{in} = \Phi_{in\downarrow}, \\ &\Sigma = \mathrm{diag}[\underbrace{1, \cdots, 1}_{N_e/2}, 0, 0, 0].\end{aligned}$$

When f_{ij} (i.e., the matrix F) is related with a single Slater determinant of the wave function, we can show that the singular value decomposition of F becomes

$$F = \Phi_{\uparrow} \Sigma \Phi_{\downarrow}^{t}.$$

This result indicates that when the number of nonzero singular values is $N_e/2$, and when all the nonzero singular values of F are one in the singular value decomposition of F, the Pfaffian-Slater wave function parametrized by f_{ij} coincides with a single Slater determinant (i.e. a solution of the mean-field approximation). In other words, the numbers of the nonzero singular values and their difference from one offer a quantitative criterion how the Pfaffian-Slater determinant deviates from the single Slate determinant.

7.4 Power Lanczos method

In this section, we show how to determine α in the power-Lanczos method. We also explain the calculation of physical quantities after the single-step Lanczos method.

7.4.1 Determination of α

First, we briefly explain the sampling procedure of the variational Monte Carlo (VMC) method. Physical properties $\hat{A}$ are calculated as follows:

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

There are two ways to calculate the product of the operators $\hat{A}\hat{B}$.

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

As we explain later, in general, the latter way is numerical stable one. For example, we consider the expectation value of the variance, which is defined as $\sigma^2 = \langle (\hat{H} - \langle \hat{H} \rangle)^2 \rangle$. There are two ways to calculate the variance.

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

From its definition, the latter way gives the positive definitive variance even for the finite sampling while the former way does not guarantee the positive definitiveness of the variance. Here, we consider the expectation values of energy and variance for the (single-step) power Lanczos wave function $|\phi\rangle = (1 + \alpha \hat{H})|\psi\rangle$. The energy is calculated as

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

where we define h_1, $h_{2(11)}$, $h_{2(20)}$, and $h_{3(12)}$ as

$$
\begin{aligned}
h_1 &= \sum_x \rho(x) F^\dagger(x, \hat{H}), \\
h_{2(11)} &= \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{H}), \\
h_{2(20)} &= \sum_x \rho(x) F^\dagger(x, \hat{H}^2), \\
h_{3(12)} &= \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{H}^2).
\end{aligned}
$$

From the condition $\frac{\partial E_{LS}(\alpha)}{\partial \alpha} = 0$, i.e., by solving the quadratic equations, we can determine the α. The variance is calculate in the similar way.

7.4.2 Calculation of physical quantities

By using the optimized parameter α, we can calculate the expected value of the operator $\hat{A}$ as

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

where we define A_0, $A_{1(10)}$, $A_{1(01)}$, and $A_{2(11)}$ as

$$
\begin{aligned}
A_0 &= \sum_x \rho(x) F(x, \hat{A}), \\
A_{1(10)} &= \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{A}), \\
A_{1(01)} &= \sum_x \rho(x) F(x, \hat{A}\hat{H}), \\
A_{2(11)} &= \sum_x \rho(x) F^\dagger(x, \hat{H}) F(x, \hat{A}\hat{H}).
\end{aligned}
$$

CHAPTER

EIGHT

PROGRAM FOR THE UNRESTRICTED HARTREE-FOCK APPROXIMATION

In mVMC package, there is a program to calculate the initial values of the pair orbital parameters f_{ij} by using the unrestricted Hartree-Fock (UHF) approximation (relation between Pfaffian Slater determinant and single Slater determinant is explained in sec. *Properties of the Pfaffian-Slater determinant*). It is noted that the target system of this program is the itinerant electron system.

8.1 Overview

In UHF approximation, two-body interaction terms are approximated as one-body interaction terms by taking into account of the fluctuation, $\delta A \equiv A - \langle A \rangle$, up to the first order. As an example, we consider the inter-site coulomb interactions

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

where we define $i \equiv (i, \sigma)$, $j \equiv (j, \sigma')$ for simplicity. Then, the interaction terms can be approximated as

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

Also for other types of interaction, the problem can be attributed to a one-body problem by using a similar approximation. Actual calculation is performed iteratively until that self-consistent solution for the mean values of the above observables are obtained.

8.1.1 Source code

A set of source codes are included in the directory `src/ComplexUHF/src`.

8.1.2 How to compile

To compile source codes, move to the directory just below the main directory of mVMC, and execute

```
$ make mvmc
```

in a similar way as the compile of mVMC. After compiling, an executable file `UHF` is generated in `src/ComplexUHF/src`.

8.1.3 Input files

A file for assigning input files (namelist.def)

The following files are needed to use the program of UHF. The format of `namelist.def` is the same as defined in *List file for Input files (namelist.def)*.

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

Although the format of these files are the same as those for mVMC basically, the following items are different:

- Parameters assigned in `ModPara` file.
- Addition of `Initial` file.

We explain details of the format of these files as follows.

Parameters assigned in ModPara file

The parameters needed in the program of UHF are as follows:

- `Nsite`
- `Ne`
- `Mix`
- `EPS`
- `IterationMax`

The parameters, `Nsite` and `Ne`, are common as mVMC. The other three parameters are specific to UHF:

- `Mix`

  Linear mixing is assigned by double-type. When mix=1, a new Green's function is fully updated without using a old one.

- `EPS`

  A condition for convergence is assigned by int-type. When a residual error between a new Green's function and a previous one is less than $10^{-\texttt{eps}}$, the iteration of calculation is stopped.

- `IterationMax`

  A maximum number of the loop is assigned by int-type.

If there are the other parameters for mVMC in this file , warning is output to the standard output (the calculation is not stopped).

Initial file

Initial values of Green's function $G_{ij\sigma_1\sigma_2} \equiv \langle c^\dagger_{i\sigma_1} c_{j\sigma_2} \rangle$ are given. The format is the same as `Trans` file, and instead of $t_{ij\sigma_1\sigma_2}$, values of $G_{ij\sigma_1\sigma_2}$ are described. Green's function is set as zero if values are not given.

8.2 Usage

Calculation of UHF is performed by the same way as mVMC, i.e., by executing the command

```
$ UHF namelist.def
```

The routine of the calculation is as follows.

1. Reading files
2. Construction of a Hamiltonian
3. Self-consistent calculation of Green's function
4. Output of f_{ij} and other files

Examples of output after calculation are as follows.

- zvo_result.dat:

  The energy and the particle number are output.

```
energy -15.2265348135
num    36.0000000000
```

- zvo_check.dat:

  The step number of the iteration, the mean of the absolute value of the residual error in Green's function, the energy in convergence process, and the particle number are output in order.

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

- zvo_UHF_cisajs.dat:

  Convergent one-body Green's function $G_{ij\sigma_1\sigma_2} \equiv \langle c^\dagger_{i\sigma_1} c_{j\sigma_2} \rangle$ is output. For all the components, $i, \sigma_1, j, \sigma_2, \mathrm{Re}\left[G_{ij\sigma_1\sigma_2}\right], \mathrm{Im}\left[G_{ij\sigma_1\sigma_2}\right]$ are output in order.

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

- zvo_eigen.dat:

  Convergent eigenvalues of the Hamiltonian are output in ascending order.

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

- zvo_gap.dat:

  For the total electron number N_{tot}, the energy difference $\Delta E = E(N_{\mathrm{tot}}+1) - E(N_{\mathrm{tot}})$ is output.

```
5.2208232631
```

- zvo_orbital_opt.dat:

  f_{ij} generated from the Slater determinant. The file with the same format as `InOrbital`,`InOrbitalAntiParallel`, `InOrbitalParallel`,`InOrbitalAntiGeneral` file is output. By referring `Orbital`, `OrbitalAntiParallel`, `OrbitalParallel`,`OrbitalAntiGeneral` file, f_{ij} is calculated (for the same type of parameters, the averaged value is calculated).

CHAPTER

NINE

HPHI/MVMC FOURIER-TRANSFORMATION UTILITY

9.1 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 $\mathcal{H}\Phi$.

9.1.1 Prerequisite

The prerequisite of this utility is the same as that of mVMC or $\mathcal{H}\Phi$.

9.1.2 Supported quantities

This utility supports the Fourier transformation of the following quantities:

One-body correlations

$$\langle \hat{c}^{\dagger}_{\mathbf{k}\alpha\uparrow}\hat{c}_{\mathbf{k}\beta\uparrow}\rangle \equiv \sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle \hat{c}^{\dagger}_{\mathbf{0}\alpha\uparrow}\hat{c}_{\mathbf{R}\beta\uparrow}\rangle \tag{9.1}$$

$$\langle \hat{c}^{\dagger}_{\mathbf{k}\alpha\downarrow}\hat{c}_{\mathbf{k}\beta\downarrow}\rangle \equiv \sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle \hat{c}^{\dagger}_{\mathbf{0}\alpha\downarrow}\hat{c}_{\mathbf{R}\beta\downarrow}\rangle \tag{9.2}$$

Density-density correlation

$$\langle \hat{\rho}_{\mathbf{k}\alpha}\hat{\rho}_{\mathbf{k}\beta}\rangle \equiv \frac{1}{N_{\mathbf{R}}}\sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle(\hat{\rho}_{\mathbf{0}\alpha}-\langle\hat{\rho}_{\mathbf{0}\alpha}\rangle)(\hat{\rho}_{\mathbf{R}\beta}-\langle\hat{\rho}_{\mathbf{R}\beta}\rangle)\rangle \tag{9.3}$$

Spin-Spin correlations

$$\langle \hat{S}^{z}_{\mathbf{k}\alpha}\hat{S}^{z}_{\mathbf{k}\beta}\rangle \equiv \frac{1}{N_{\mathbf{R}}}\sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle \hat{S}^{z}_{\mathbf{0}\alpha}\hat{S}^{z}_{\mathbf{R}\beta}\rangle \tag{9.4}$$

$$\langle \hat{S}^{+}_{\mathbf{k}\alpha}\hat{S}^{-}_{\mathbf{k}\beta}\rangle \equiv \frac{1}{N_{\mathbf{R}}}\sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle \hat{S}^{+}_{\mathbf{0}\alpha}\hat{S}^{-}_{\mathbf{R}\beta}\rangle \tag{9.5}$$

$$\langle \hat{\mathbf{S}}_{\mathbf{k}\alpha}\cdot\hat{\mathbf{S}}_{\mathbf{k}\beta}\rangle \equiv \frac{1}{N_{\mathbf{R}}}\sum_{\mathbf{R}}^{N_{\mathbf{R}}} e^{-i\mathbf{k}\cdot\mathbf{R}}\langle \hat{\mathbf{S}}_{\mathbf{0}\alpha}\cdot\hat{\mathbf{S}}_{\mathbf{R}\beta}\rangle \tag{9.6}$$

9.2 Tutorial

In this tutorial, we explain through a sample calculation of the 8-site Hubbard model on the square lattice.

9.2.1 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*)

9.2.2 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_cisajscktalt_001.dat (*Results of correlation function in the site representation*)
- output/zvo_cisajscktalt.dat (*Results of correlation function in the site representation*)
- geometry.dat (*Geometry*)
- output/zvo_corr.dat (*Correlation functions on the k path*)

9.2.3 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*)

9.3 File format

9.3.1 Geometry

The file name in the *Tutorial* is `geometry.dat`. When we use Standard mode of mVMC/$\mathcal{H}\Phi$, the information of the cell and geometry is generated automatically.

```
1.000000000000000e+00     0.000000000000000e+00     0.000000000000000e+00 (1)
0.000000000000000e+00     1.000000000000000e+00     0.000000000000000e+00 (1)
0.000000000000000e+00     0.000000000000000e+00     1.000000000000000e+00 (1)
0.000000000000000e+00     0.000000000000000e+00     0.000000000000000e+00 (2)
2 2 0         (3)
-2 2 0        (3)
```

(continues on next page)

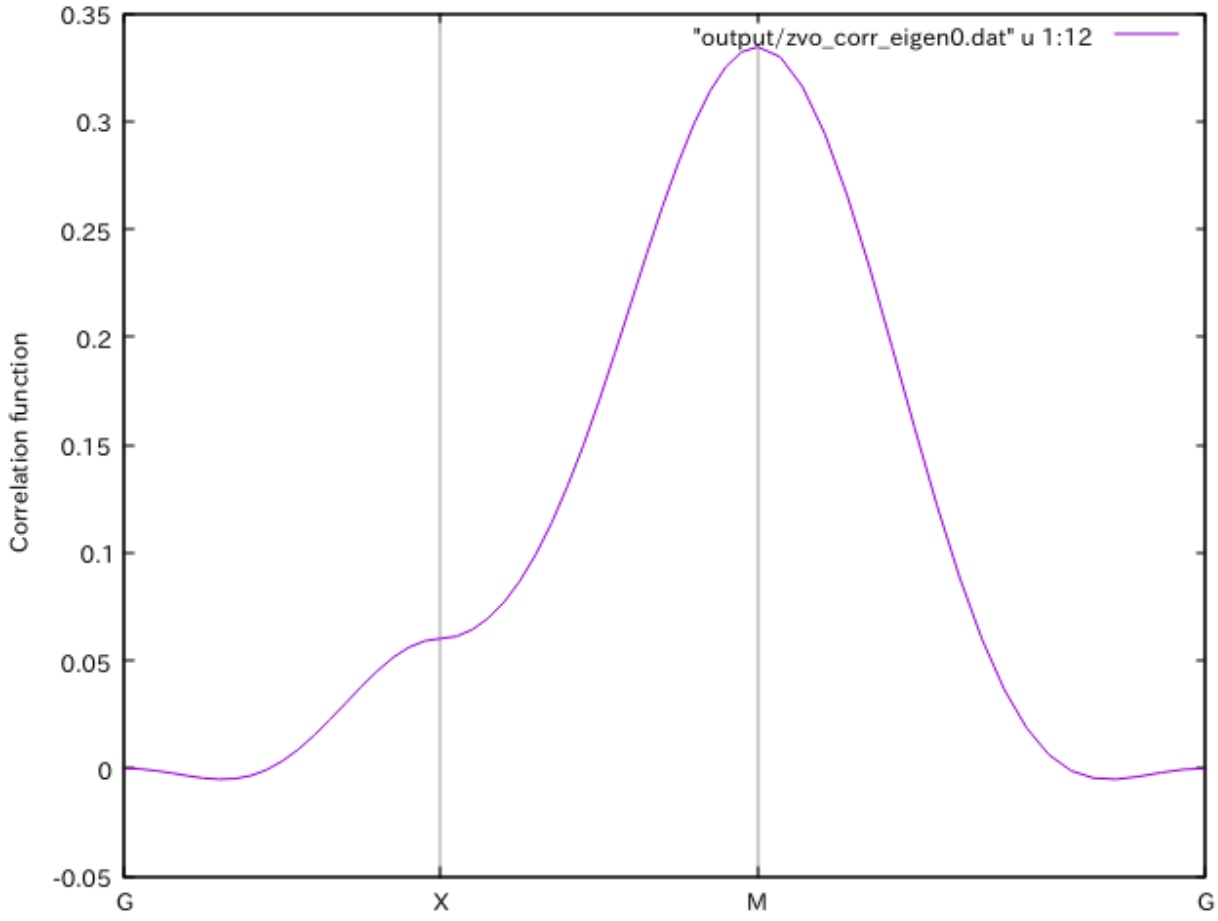

Fig. 1: The spin-spin correlation $\langle \mathbf{S}_\mathbf{k} \cdot \mathbf{S}_\mathbf{k} \rangle$ (Column 12).

(continued from previous page)

```
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.

9.3.2 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/$\mathcal{H}\Phi$. When we use the standard mode, this file is generated automatically. The general description is written in the manuals for mVMC/$\mathcal{H}\Phi$. The file names in the *Tutorial* are `greenone.def` (one body) and `greentwo.def` (two body).

For calculating correlation functions in *Supported quantities*, indices must be specified as follows:

- $\langle \hat{c}^\dagger_{\mathbf{k}\alpha\uparrow} \hat{c}_{\mathbf{k}\beta\uparrow} \rangle$

  $\langle \hat{c}^\dagger_{\mathbf{0}\alpha\uparrow} \hat{c}_{\mathbf{R}\beta\uparrow} \rangle$ with $\mathbf{R}$ ranging on the all unit cell, and (α, β) ranging on the all orbitals in the unit cell.

- $\langle \hat{c}^\dagger_{\mathbf{k}\alpha\downarrow} \hat{c}_{\mathbf{k}\beta\downarrow} \rangle$

  $\langle \hat{c}^\dagger_{\mathbf{0}\alpha\downarrow} \hat{c}_{\mathbf{R}\beta\downarrow} \rangle$ with $\mathbf{R}$ ranging on the all unit cell, and (α, β) ranging on the all orbitals in the unit cell.

- $\langle \hat{\rho}_{\mathbf{k}\alpha} \hat{\rho}_{\mathbf{k}\beta} \rangle$ and $\langle \hat{S}^z_{\mathbf{k}\alpha} \hat{S}^z_{\mathbf{k}\beta} \rangle$

  $\langle \hat{c}^\dagger_{\mathbf{0}\alpha\sigma} \hat{c}_{\mathbf{0}\alpha\sigma} \hat{c}^\dagger_{\mathbf{R}\beta\sigma'} \hat{c}_{\mathbf{R}\beta\sigma'} \rangle$ with $\mathbf{R}$ ranging on the all unit cell, (α, β) ranging on the all orbitals in the unit cell, and (σ, σ') 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 $\mathcal{H}\Phi$, $\langle \hat{c}^\dagger_{\mathbf{0}\alpha\sigma} \hat{c}_{\mathbf{0}\alpha-\sigma} \hat{c}^\dagger_{\mathbf{R}\beta-\sigma} \hat{c}_{\mathbf{R}\beta\sigma} \rangle$ with $\mathbf{R}$ ranging on the all unit cell, (α, β) ranging on the all orbitals in the unit cell, and σ ranging from $\uparrow$ to $\downarrow$. For mVMC, $\langle \hat{c}^\dagger_{\mathbf{0}\alpha\sigma} \hat{c}_{\mathbf{R}\beta\sigma} \hat{c}^\dagger_{\mathbf{R}\beta-\sigma} \hat{c}_{\mathbf{0}\alpha-\sigma} \rangle$ with $\mathbf{R}$ ranging on the all unit cell,

(α, β) ranging on the all orbitals in the unit cell, and σ 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_cisajscktalt_001.dat` (mVMC), or `output/zvo_cisajs.dat` and `output/zvo_cisajscktalt.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.

9.3.3 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.

9.3.4 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
```

9.3.5 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`.

9.4 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).

9.4.1 HPhi-Lanczos

In this case, HPhi writes correlation functions to the files zvo_cisajs.dat (one body) and zvo_cisajscktalt.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.

9.4.2 HPhi-TPQ

HPhi writes correlation functions to files zvo_cisajs_run*step*.dat (one body), zvo_cisajscktalt_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.

9.4.3 HPhi-Full diagonalization and LOBCG

HPhi writes correlation functions to files zvo_cisajs_eigen*.dat (one body) and zvo_cisajscktalt_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/.

9.4.4 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_cisajscktalt_???.dat (two body) in output/ directory. fourier utility reads all of these files, performs Fourier transformation, computes the average

$$\langle A \rangle = \frac{1}{N_{\mathrm{Try}}} \sum_{i=1}^{N_{\mathrm{Try}}} A_i \tag{9.7}$$

and the standard error

$$\delta A = \frac{1}{N_{\mathrm{Try}} - 1} \sqrt{\frac{1}{N_{\mathrm{Try}}} \sum_{i=1}^{N_{\mathrm{Try}}} (A_i - \langle A \rangle)^2} \tag{9.8}$$

of the real- and imaginary-part of each correlation function, and writes them to a file zvo_corr_eigen*.dat in output/ directory.

9.5 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.

CHAPTER

TEN

DOWNFOLDING WITH WANNIER FUNCTIONS

10.1 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:

$$\begin{aligned} U_{mn}(\mathbf{R},\omega) = & \int_V d\mathbf{r} \int_V d\mathbf{r}' w^*_{m\mathbf{0}}(\mathbf{r}) w_{m\mathbf{0}}(\mathbf{r}) W(\mathbf{r},\mathbf{r}',\omega) w^*_{n\mathbf{R}}(\mathbf{r}') w_{n\mathbf{R}}(\mathbf{r}'), \\ J_{mn}(\mathbf{R},\omega) = & \int_V d\mathbf{r} \int_V d\mathbf{r}' w^*_{m\mathbf{0}}(\mathbf{r}) w_{n\mathbf{R}}(\mathbf{r}) W(\mathbf{r},\mathbf{r}',\omega) w^*_{n\mathbf{R}}(\mathbf{r}') w_{m\mathbf{0}}(\mathbf{r}'). \end{aligned}$$

Here, V is the volume of the crystal, $w_{n\mathbf{R}}(\mathbf{r})$ is the n-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:

$$\begin{aligned} \mathcal{H}_{\rm int} = \frac{1}{2} \sum_{\sigma\rho} \sum_{ij} \sum_{nm} \Big[& U_{mn}(\mathbf{R}_{ij}) c^\dagger_{im,\sigma} c^\dagger_{jn,\rho} c_{jn,\rho} c_{im,\sigma} \\ & + J_{mn}(\mathbf{R}_{ij}) (c^\dagger_{im,\sigma} c^\dagger_{jn,\rho} c_{im,\rho} c_{jn,\sigma} + c^\dagger_{im,\sigma} c^\dagger_{im,\rho} c_{jn,\rho} c_{jn,\sigma}) \Big], \end{aligned}$$

where $\mathbf{R}_{ij} \equiv \mathbf{R}_i - \mathbf{R}_j$. $\mathbf{R}_i$ is the position vector of the i-th cell. Since mVMC and $\mathcal{H}\Phi$ cannot directly treat the following type of interactions $c^\dagger_{i,\sigma} c^\dagger_{j,\rho} c_{k,\rho'} c_{l,\sigma'}$, the Hamiltonian must be rewritten as follows:

$$\begin{aligned} \mathcal{H}_{\rm int} = & \sum_{i,m} U_{mm}(\mathbf{0}) n_{im,\uparrow} n_{im,\downarrow} + \sum_{(i,m)<(j,n)} U_{mn}(\mathbf{R}_{ij}) n_{im} n_{jn} \\ & - \sum_{(i,m)<(j,n)} J_{mn}(\mathbf{R}_{ij}) (n_{im,\uparrow} n_{jn,\uparrow} + n_{im,\downarrow} n_{jn,\downarrow}) \\ & + \sum_{(i,m)<(j,n)} J_{mn}(\mathbf{R}_{ij}) (c^\dagger_{im,\uparrow} c^\dagger_{jn,\downarrow} c_{im,\downarrow} c_{jn,\uparrow} + \text{h.c.}) \\ & + \sum_{(i,m)<(j,n)} J_{mn}(\mathbf{R}_{ij}) (c^\dagger_{im,\uparrow} c^\dagger_{im,\downarrow} c_{jn,\downarrow} c_{jn,\uparrow} + \text{h.c.}). \end{aligned}$$

The lattice model is defined by the following Hamiltonian:

$$\mathcal{H} = \sum_{m,n,i,j,\sigma} \left[t_{mn}(\mathbf{R}_{ij}) - t^{\rm DC}_{mn}(\mathbf{R}_{ij}) \right] c^\dagger_{im\sigma} c_{jn\sigma} + \mathcal{H}_{int},$$

where $t^{\mathrm{DC}}_{mn}(\mathbf{R}_{ij})$ is the one-body correction term given as:

$$
\begin{aligned}
t^{\mathrm{DC}}_{mm}(\mathbf{0}) &\equiv \alpha U_{mm}(\mathbf{0}) D_{mm}(\mathbf{0}) + \sum_{(\mathbf{R},n)\neq(\mathbf{0},m)} U_{mn}(\mathbf{R}) D_{nn}(\mathbf{0}) \\
&\quad - (1-\alpha) \sum_{(\mathbf{R},n)\neq(\mathbf{0},0)} J_{mn}(\mathbf{R}) D_{nn}(\mathbf{R}), \\
t^{\mathrm{DC}}_{mn}(\mathbf{R}_{ij}) &\equiv \frac{1}{2} J_{mn}(\mathbf{R}_{ij}) \left(D_{nm}(\mathbf{R}_{ji}) + 2\mathrm{Re}[D_{nm}(\mathbf{R}_{ji})] \right) \\
&\quad - \frac{1}{2} U_{mn}(\mathbf{R}_{ij}) D_{nm}(\mathbf{R}_{ji}), \quad (\mathbf{R}_{ij}, m) \neq (\mathbf{0}, n), \\
D_{mn}(\mathbf{R}_{ij}) &\equiv \sum_{\sigma} \left\langle c^{\dagger}_{im\sigma} c_{jn\sigma} \right\rangle_{\mathrm{KS}},
\end{aligned}
$$

Here, $t^{\mathrm{DC}}_{mm}(\mathbf{0})$ is the term to correct the chemical potntial, $t^{\mathrm{DC}}_{mn}(\mathbf{R}_{ij})$ is term to correct transfer integrals . α is the tuning parameter for one-body correction from the on-site Coulomb interactions. 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 controled by multiplying tuning parameters λ_U, λ_J. For details, see `Input parameters for Standard mode`.

10.1.1 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 $\mathcal{H}\Phi$. Therefore, these programs must be available in our machine.

10.2 Tutorial

In this tutorial, we downfold Sr_2CuO_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.

10.2.1 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. Put the pseudopotential files into `../pseudo` directory.

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
```

10.2.2 (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
 -1.749305 1.955690 6.351200
 1.749305 -1.955690 6.351200
 1.749305 1.955690 -6.351200
```

(continues on next page)

(continued from previous page)

```
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
        0.5000000000    0.5000000000   -0.5000000000    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
        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
```

(continues on next page)

(continued from previous page)

```
        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.

10.2.3 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
/
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
```

(continues on next page)

(continued from previous page)

```
 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
```

10.2.4 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
0.50000 0.50000 -0.50000
0.00000 0.00000 0.00000
0.00000 0.00000 0.50000
0.04267 -0.04267 0.50474
0.27370 -0.27370 0.27370
0.00000 0.00000 0.00000
0.50000 0.00000 0.00000
&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
```

(continues on next page)

(continued from previous page)

```
/
&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` .)

10.2.5 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
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
```

10.3 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
2Sz = 0
nelec = 4
cutoff_t = 0.1
cutoff_u = 1.0
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 transrational 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 λ_U , λ_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:

$$
\begin{aligned}
t^{\mathrm{DC}}_{mm}(\mathbf{0}) &\equiv \alpha U_{mm}(\mathbf{0}) D_{mm}(\mathbf{0}) + \sum_{(\mathbf{R},n)\neq(\mathbf{0},m)} U_{mn}(\mathbf{R}) D_{nn}(\mathbf{0}) \\
&\quad - (1-\alpha) \sum_{(\mathbf{R},n)\neq(\mathbf{0},0)} J_{mn}(\mathbf{R}) D_{nn}(\mathbf{R}), \\
t^{\mathrm{DC}}_{mn}(\mathbf{R}_{ij}) &\equiv \frac{1}{2} J_{mn}(\mathbf{R}_{ij}) \left(D_{nm}(\mathbf{R}_{ji}) + 2\mathrm{Re}[D_{nm}(\mathbf{R}_{ji})] \right) \\
&\quad - \frac{1}{2} U_{mn}(\mathbf{R}_{ij}) D_{nm}(\mathbf{R}_{ji}), \quad (\mathbf{R}_{ij}, m) \neq (\mathbf{0}, n), \\
D_{mn}(\mathbf{R}_{ij}) &\equiv \sum_{\sigma} \left\langle c^{\dagger}_{im\sigma} c_{jn\sigma} \right\rangle_{\mathrm{KS}},
\end{aligned}
$$

where, the first and second terms correspond to the Hartree and Fock corrections, respectively. α 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 ($0 \leq \alpha \leq 1$).

10.4 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.

10.4.1 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
0.000000 -0.000000 -0.000000
-0.000000 -0.000000 -0.000000
0.000000 0.000000 0.000000
```

- 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.

10.4.2 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    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.

10.5 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.

CHAPTER

ELEVEN

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Mr. Daisuke Tahara for providing us his code of variational Monte Carlo method. A part of mVMC is based on his original code. We also acknowledge Dr. Hiroshi Shinaoka, Dr. Youhei Yamaji, Dr. Moyuru Kurita, and Dr. Ryui Kaneko for their cooperation on developing mVMC. We would also like to thank the support from "Project for advancement of software usability in materials science" by The Institute for Solid State Physics, The University of Tokyo, for development of mVMC ver.0.1, ver. 0.2, and ver. 1.0.