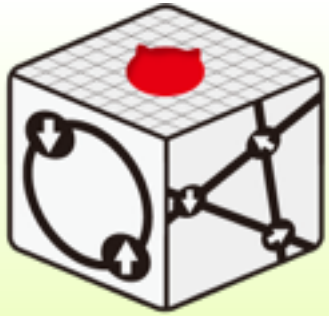
Exercises for mVMC



Takahiro Misawa
ISSP, Univ. of Tokyo

mVMC

Basic exercises

1. Heisenberg, Hubbard chain
2. Heisenberg, Hubbard (square latticd)
3. Tool I: how to use fourier tool
4. Tool II: How to prepare initial wave functions from unrestricted Hartree-Fock (UHF) calculation

You can find sample scripts in mVMC-tutorial/HandsOn/2017_0830/Samples

Advanced exercise

(Let's generate several quantum phases by mVMC)

1. Hubbard + V → Charge order
2. Heisenberg+J2 → stripe magnetic order
3. Attractive Hubbard → superconductivity
4. 1D Kondo lattice → Kondo insulator
5. Kitaev model → Kitaev spin liquid

1. Heisenberg chain

$$H = \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j$$

mVMC-tutorial/HandsOn/2017_0830/Samples/1D_Heisenberg

step 1. For L=4, optimization using mVMC

Ex. of StdFace.def

```
L                = 4
Lsub             = 2
model            = "Spin"
lattice          = "chain"
J                = 1.0
2Sz              = 0
NVMCSample       = 200
NSROptItrStep    = 500
NSROptItrSmp     = 50
NMPTrans         = 1
NSPStot          = 0
```

Ex.
1. ./vmc.out -s StdFace.def
2. using gnuplot
plot ./output/zvo_out_001.dat u 1

```
Energy by HΦ
0 -2.0000000000 : S=0
1 -1.0000000000 : S=1
2 -0.0000000000
3 0.0000000000
4 0.0000000000
5 1.0000000000
```

1. Heisenberg chain

$$H = \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j$$

step 2. L=4, correlation factors by mVMC

StdFace_aft.def

```
L                = 4
Lsub             = 2
model            = "Spin"
lattice          = "chain"
J                = 1.0
2Sz              = 0
NVMCSample       = 200
NVMCCalMode      = 1
NDataIdxStart    = 1
NDataQtySmp      = 5
NMPTrans         = 1
NSPStot          = 0
```

Ex.

1. cp ./output/zqp_opt.dat .
2. ./vmc.out -s StdFace_aft.def ./zqp_opt.dat

1. Heisenberg chain

$$H = \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j$$

step 3. Calculate average values and standard errors of physical propeties

```
output/zvo_out_001.dat
output/zvo_out_002.dat
output/zvo_out_003.dat
output/zvo_out_004.dat
output/zvo_out_005.dat

→Energy in independent bin
is shown at the first line

You can calculate average
values and standard errors
using following command
ln -s output aft
perl -w Aft_energy.pl
```

```
Similarly, in

output/zvo_cisajs_00n.dat,
output/zvo_cisajscktalt_00n.dat,

one-body and two-body correlation
functions are output.

Using following commands, you can
calculate spin structure factors
ln -s output aft
perl -w Aft_Sq.pl

Note: The first line inResult_Sq.dat
represents Lx*kx/(2π)
```

1. Heisenberg chain

$$H = \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j$$

mVMC-tutorial/HandsOn/2017_0830/Samples/1D_Heisenberg

step 4. By executing X.sh, all the calculations will be done

```
sh ./X.sh
[note: vmc.out and vmcdry.out  should exist in the
same directory ]
```

By changing input.txt, you can change system sizes and total spin(NSPStot).

```
Lx 4
Ly 1
orb_num 1
NSPStot 0
```

- **Plot Result_Sq.dat**
- **Changing system sizes and examine system-size dependence**
- **By specifying NSPStot=1, is it possible to obtain S=1 state ?**

[Check whether S(q=0)=S*(S+1)/(3*L^2) is satisfied]

Explanation of script X.sh

sh ./X.sh → Performing all calculations
sh ./Clean.sh → Delete all def files (Initialization)
[Note that calculation results will be deleted !!!]

perl -w MakeMod.pl : Generating
StdFace.def (1st optimization, results are stored in opt1),
StdaFace_2.def (2nd optimization, results are stored in opt2),
StdFace_aft.def (Calculating physical properties, results are stored in aft)

perl -w CisAjs.pl : input file for one-body Green functions
perl -w CisAjsCktAltDC.pl: input file for two-body Green functions

perl -w Aft_Sq.pl : Calculating structure factors from aft/zvo_*.dat
perl -w Aft_energy.pl : Calculating energies /zvo_out_*dat
perl -w Aft_SiSj.pl : Calculating spin correlations from aft/zvo_*dat

1. Heisenberg chain (references)

These energies are obtained by exact diagonalization (HΦ)

L=6:

```
 0 -2.8027756377
 1 -2.1180339887
 2 -1.5000000000
 3 -1.2807764064
 4 -1.2807764064
 5 -1.0000000000
 6 -1.0000000000
 7 -0.5000000000
```

L=8:

```
 0 -3.6510934089
 1 -3.1284190638
 2 -2.6996281483
 3 -2.4587385089
 4 -2.4587385089
 5 -2.1451483739
 6 -2.1451483739
 7 -1.8546376797
```

L=10:

```
 0 -4.5154463545
 1 -4.0922073467
 2 -3.7705974354
 3 -3.5432793743
 4 -3.5432793743
 5 -3.2461649167
 6 -3.2461649167
 7 -2.9759318691
```

Check accuracy of mVMC method !

1. Hubbard chain

$$H = -t \sum_{\langle i,j \rangle} (c^{\dagger}_{i\sigma} c_{j\sigma} + \text{h.c.}) + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$

Step.1 Perform mVMC cal. at L=4, U=4, t=1, half filling

Ex. StdFace.def

```
L               = 4
Lsub            = 2
model           = "FermionHubbard"
lattice         = "chain"
t               = 1.0
U               = 4.0
nelec           = 4
2Sz             = 0
NVMCSample      = 200
NSROptItrStep   = 500
NSROptItrSmp    = 50
NMPTrans        = 1
```

```
Energy by HΦ
 0 -2.1027484835
 1 -1.8064238518
 2 -1.0681403934
 3 -0.8284271247
 4 -0.8284271247
 5 0.0000000000
 6 0.5814492811
 7 2.0000000000
```

1. Hubbard chain

$$H = -t \sum_{\langle i,j \rangle} (c^{\dagger}_{i\sigma} c_{j\sigma} + \text{h.c.}) + U \sum_{i} n_{i\uparrow} n_{i\downarrow}$$

Step. 2 Calculate physical quantities

Step. 3 Calculate average value and standard error of physical quantities

Step. 4 Use all-in-one shell script (X.sh)

Comapre with the results by HΦ [exact diagonalization]

Sample script for HΦ **./HPhi -s StdFace.def**

```
L               = 4
model           = "Hubbard"
lattice         = "chain"
method          = "fulldiag"
U               = 4.0
t               = 1.0
2Sz             = 0
nelec           = 4
```

```
L               = 8
model           = "Hubbard"
lattice         = "chain"
method          = "CG"
U               = 4.0
t               = 1.0
2Sz             = 0
nelec           = 8
exct            = 8
```

1. Hubbard chain (references)

These energies are obtained by exact diagonalization (HΦ)

L=6:

0	−3.668706178872957 1
1	−2.898381474036730 4
2	−2.516376873116143 1
3	−2.422911263847928 9
4	−2.422911263847929 3
5	−2.092753829496921 0
6	−2.092753829496921 0
7	−1.769024823288434 5

L=8:

0	−4.603526299892002
1	−4.299992758433059 9
2	−4.010153957644034 2
3	−3.705764239483940 5
4	−3.705764239483940 5
5	−3.496356310215205 1
6	−3.496356310215204 2
7	−3.244557098464969 4

Check accuracy of mVMC method !

Advanced: Perform the total momentum projections for L=6, 8 by specifying NMPTrans = 2

2. Heisenberg & Hubbard on the square lattice

- **Procedure is basically the same as that of Heisenberg and Hubbard chain. Please note that the elapse time becomes longer !**

For laptop PC, 4×4 =16 is suitable system size. For larger system sizes, we recommend using PC clusters.

- **By using fourier tools, it is possible to plot structure factors.**

2. Heisenberg & Hubbard on the square lattice

Reference data for Heisenberg model

$(N_s = 4 \times 4)$	E/N_s	S_{nn}	S_{nnn}	$S(\boldsymbol{q}_{peak})$
ED	-0.70178020	-0.35089010	0.21376	0.09217
mVMC(2×2)	-0.701769(6)	-0.35088(3)	0.2136(2)	0.09212(6)
mVMC(2×2)+Lanczos	-0.701783(3)	-	-	-
mVMC(4×4)	-0.70178015(8)	0.35089007(4)	0.2139(4)	0.0922(1)
$(N_s = 6 \times 6)$	E/N_s	S_{nn}	S_{nnn}	$S(\boldsymbol{q}_{peak})$
ED	-0.6788721499	-0.33943607	0.207402499	0.069945
mVMC(2×2)	-0.67843(2)	-0.33921(1)	0.20738(1)	0.07019(4)
mVMC(2×2)+Lanczos	-	-	-	-
mVMC(6×6)	-0.678865(5)	-0.339433(3)	0.2072(2)	0.0698(1)
mVMC(6×6)+Lanczos	-0.678881(5)	-	-	-

Table 4: Comparisons with exact diagonalization for 4×4 and 6×6 Heisenberg model with $J = 1$. We note $\boldsymbol{q}_{peak} = (\pi, \pi)$. The relative errors η become 0.000001% for $L = 4$ and 0.001% for $L = 6$, respectively.

2. Heisenberg & Hubbard on the square lattice

Reference data for Hubbard model

	E/N_{s}	D	S_{nn}	$S(\boldsymbol{q}_{\mathrm{peak}})$
ED	-0.85136	0.11512	-0.2063	0.05699
mVMC(2×2)	-0.84982(4)	0.11529(5)	-0.2062(1)	0.05773(4)
mVMC(2×2)+Lanczos	-0.85105(3)	-	-	-
mVMC(4×4)	-0.85068(4)	0.1153(4)	-0.2062(5)	0.0573(2)
mVMC(4×4)+Lanczos	-0.85121(3)	-	-	-

Table 3: Comparisons with exact diagonalization for 4×4 Hubbard model with $U = 4$ and $t = 1$ at half filling. Exact diagonalization (ED) is done by using $\mathcal{H}\Phi$ [38, 39]. mVMC(2×2) means f_{ij} has 2×2 sublattice structures, $\boldsymbol{q}_{\mathrm{peak}} = (\pi, \pi)$, and the parentheses denote the error bars in the last digit. Lanczos means that the first-step Lanczos calculations on top of the mVMC calculations.

3. fourier tool

Calculating and visualizing the strcutreu factors from the calculated one-body/two-body correlations functions

For details, please below document

https://github.com/issp-center-dev/mVMC/releases/download/v1.0.2/mVMC-1.0.2.tar.gz

See, doc/fourier/en/_build/html/index.html

Difference between fourier tool & perl scripts

perl Script(Aft_Sq.pl)

$$S(q) = \frac{1}{3N_s^2} \sum_{\langle i,j \rangle} \langle \vec{S}_i \vec{S}_j \rangle e^{\mathrm{iq(r_i - r_j)}}$$

$$N(q) = \frac{1}{N_s^2} \sum_{\langle i,j \rangle} \langle (\vec{N}_i - \langle \vec{N}_i \rangle) \cdot (\vec{N}_j - \langle \vec{N}_j \rangle) \rangle e^{\mathrm{iq(r_i - r_j)}}$$

fourier tool

$$S(q) = \frac{1}{N_s} \sum_{\langle i,j \rangle} \langle \vec{S}_i \vec{S}_j \rangle e^{\mathrm{iq(r_i - r_j)}}$$

$$N(q) = \frac{1}{N_s} \sum_{\langle i,j \rangle} \langle \vec{N}_i \vec{N}_j) \rangle e^{\mathrm{iq(r_i - r_j)}}$$

4. Initial states from UHF calculations

Generating initial states from unrestricted Hartree-Fock calculations.
Execution file for UHF calculations (UHF) exists in
`/src/ComplexUHF/UHF`

./UHF namelist.def
→UHF calculations

zqp_APOrbital_opt.dat (initial fin)
is generated

Tips: Proper initial Green functions (Weiss fields) are necessary

IniGreen.pl →Generating Green functions for (π, π) magnetic order
Add Initial zinitial.def in namelist.def

4. Initial states from UHF calculations

```
sh ./IniUHF.sh
→Generating initial fij from UHF calculations
(Note: UHF should exist in the same directory)
```

IniUHF.sh

```
perl -w MakeMod.pl
./vmcdry.out StdFace.def
#[s]UHF
mkdir tmpUHF
cp IniGreen.pl ./tmpUHF
cp *def        ./tmpUHF
cp input.txt   ./tmpUHF
cp ./UHF       ./tmpUHF
cd ./tmpUHF
  perl -w IniGreen.pl
  echo "        Initial zinitial.def" >> namelist.def
  ./UHF namelist.def
cd -
#[e]UHF
cp tmpUHF/zqp_APOrbital_opt.dat .
echo "         InOrbital  zqp_APOrbital_opt.dat" >> namelist.def
cp namelist.def xnamelist.def
```

Note: Rename namelist.def in X.sh into xnamelist.def

Advanced exercises [Several quantum phases]

1. Hubbard + V →Charge order

2. Heisenberg+J2 → Stripe magnetic order

3. Attractive Hubbard → 1s (isotropic) SC

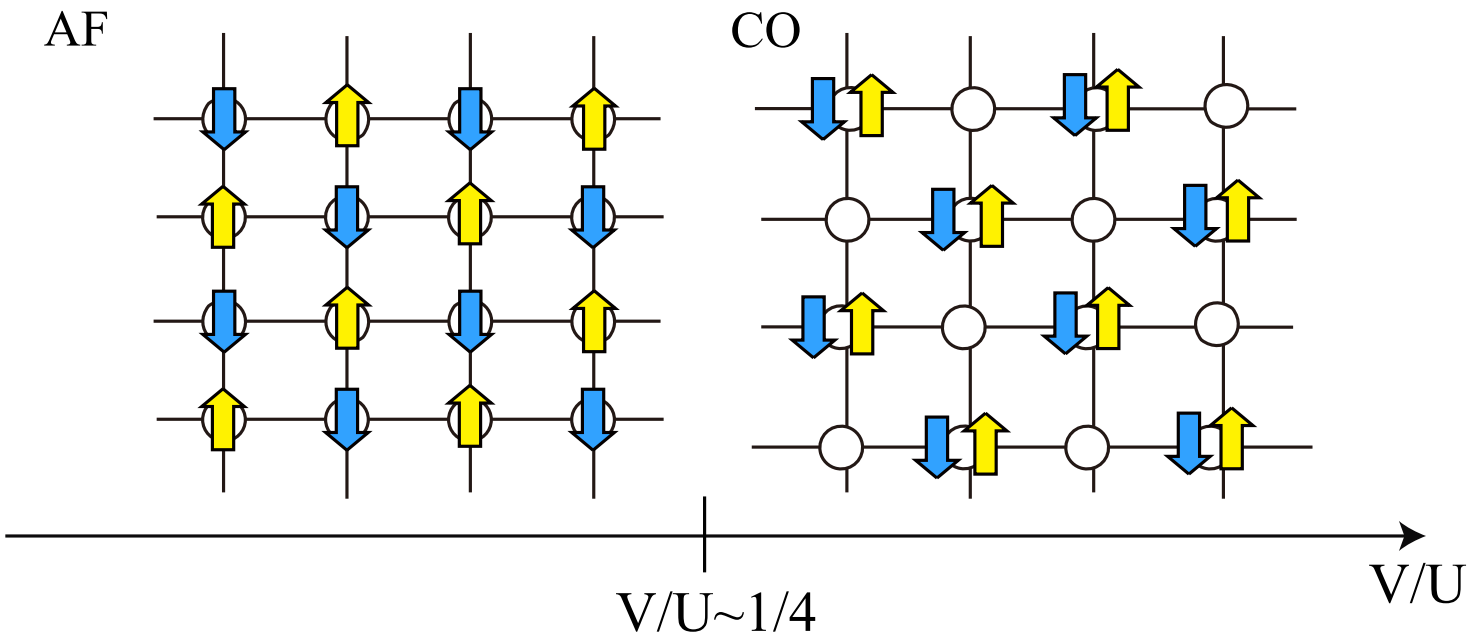
4. 1D Kondo lattice→Kondo ins.

5. Kitaev model → Kitaev spin liquid

Note: Numerical cost for 4, 5 may be large

1. Introducing off-site Coulomb interaction V

$$H_V = V \sum_{\langle i,j \rangle} n_i n_j$$



Check whether the ground state is charge-ordered phase by plotting Result_Nq.dat or using fourier tool !

1. Introducing off-site Coulomb interaction V

$$H_V = V \sum_{\langle i,j \rangle} n_i n_j$$

Ex. StdFace.def

```
W               = 4
L               = 4
Wsub            = 2
Lsub            = 2
model           = "FermionHubbard"
lattice         = "Tetragonal"
t               = 1.0
U               = 4.0
V               = 2.0
nelec           = 16
2Sz             = 0
NVMCSample      = 50
NSROptItrStep = 500
NSROptItrSmp  = 50
NMPTrans        = 4
```

How to rewrite input files

1. perl -w MakeMod.pl
→ generating StFace.def

2. Add "V=2.0" in StdFace.def, StdFace_2.def, StdFace_aft.def

3. Comment out "perl -w MakeMod.pl" in X.sh

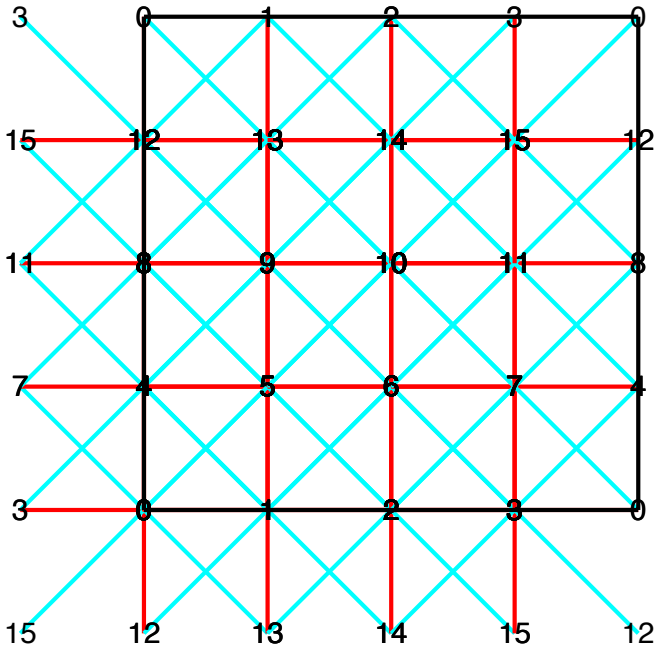
#perl -w MakeMod.pl

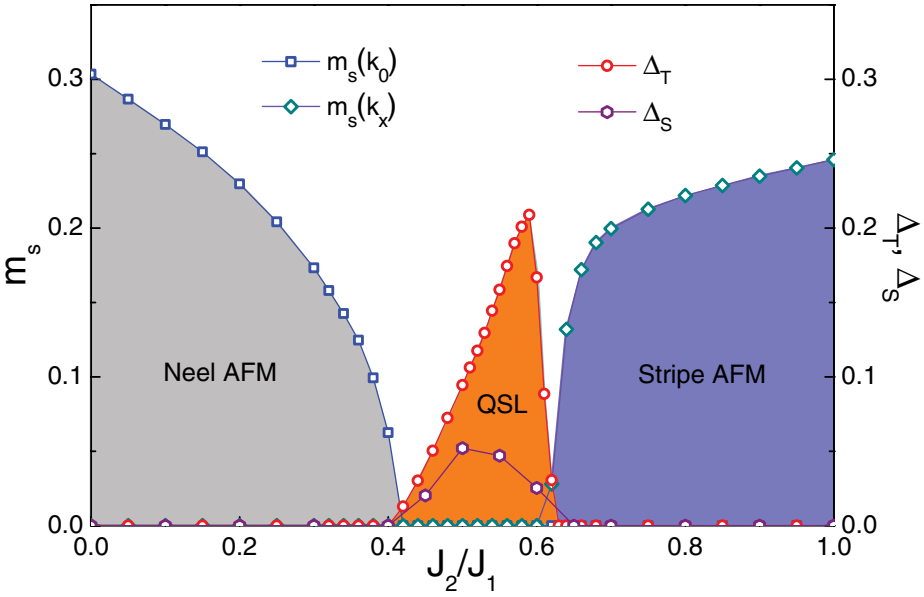
4. sh ./X.sh

2. J1-J2 Heisenberg model

$$H = J_1 \sum_{\langle i,j \rangle} S_i S_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} S_i S_j$$

nearest J1,next-nearest J2





PRB 86, 024424(2012)

plot by lattice.gp

QSL may appear around J2/J1~0.5

2. J1-J2 Heisenberg model

- Cnfirmt that stripe magnetic order occurs at J2=1 from Result_Sq.dat or using fourier tool
- What happens around J2=0.5 ?

StdFace.def の例

```
W               = 4
L               = 4
Wsub            = 2
Lsub            = 2
model           = "Spin"
lattice         = "Tetragonal"
J               = 1.0
J'              = 1.0
2Sz             = 0
NVMCSample      = 50
NSROptItrStep   = 500
NSROptItrSmp    = 100
NMPTrans        = 1
NSPGaussLeg     = 1
```

How to modify script

1. perl -w MakeMod.pl
generating StFace.def

2. Using StdFace.def,
add J'=1.0 in
StdFace_2.def, StdFace_aft.def

3. In X.sh,
comment out
perl -w MakeMod.pl
i.e.,
#perl -w MakeMod.pl

4. sh ./X.sh

Or,directly modify MakeMod.pl

3.attractive Hubbard model

- just make U negative, say, U=-1
- Calculate 1s SC correlations

(it is better to perform calculation for doped case)

How to modify script

1. perl -w MakeMod.pl
-> generating StFace.def

2. Form StdFace.def,
make
StdFace_2.def, StdFace_aft.def
by adding U=-4.0

3. In X.sh,
comment out
perl -w MakeMod.pl

#perl -w MakeMod.pl

4. sh ./X.sh

$$P_{1s}(\vec{r}) = \frac{1}{2N_s}\sum_{i=1}^{N_s}\langle\Delta_{1s}^{\dagger}(\vec{r_i})\Delta_{1s}(\vec{r_i}+\vec{r}) + \Delta_{1s}(\vec{r_i})\Delta_{1s}^{\dagger}(\vec{r_i}+\vec{r})\rangle$$

$$\Delta_{1s}(\vec{r_i}) = \frac{1}{\sqrt{2}}(c_{i\uparrow}c_{i\downarrow} - c_{i\downarrow}c_{i\uparrow}).$$

- Scripts for calculating 1s SC correlating exists at Samples/SC_Correlation

How to use
1. sh X.sh
2. perl -w ForSCCor.pl →SC_cisajscktaltdc.def
3. namelist.defの

TwoBodyG greentwo.def → TwoBodyG SC_cisajscktaltdc.def

4. vmc.out namelist.def
5. ln -s output SC
6. perl -w Aft_SC1.pl
7. plot "Ave_Max_SC_Sum_1s_0.dat" w e

4. 1D Kondo lattice model

StdFace.def の例

```
L                = 8
Lsub             = 2
model            = "Kondo"
lattice          = "chain"
t                = 1.0
J                = 1
nelec            = 8
2Sz              = 0
NVMCSample       = 200
NSROptItrStep    = 2000
NSROptItrSmp     = 200
NMPTrans         = 2
NSPGaussLeg      = 8
```

Calculating spin gap from the energy difference between S=0 and S=1 state !

$(S=0)$	E/N_s	S_{loc}	S^f_{nn}	$S(\pi)$
ED	-1.394104	-0.3151569745	-0.3386218911	0.05685112698
mVMC(2)	-1.39352(1)	-0.3142(6)	-0.3365(2)	0.0575(1)
mVMC(2)+Lanczos	-1.39399(2)	-	-	-
mVMC(8)	-1.39400(1)	-0.3158(3)	-0.3382(3)	0.0567(1)
mVMC(8)+Lanczos	-1.394099(5)	-	-	-
$(S=1)$	E/N_s	S_{loc}	S^f_{nn}	$S(\pi)$
ED	-1.382061	-0.27480917	-0.224015671	0.057478
mVMC(2)	-1.38128(3)	-0.2743(3)	-0.2248(4)	0.05811(6)
mVMC(2)+Lanczos	-1.38187(1)	-	-	-
mVMC(8)	-1.38171(3)	-0.2750(4)	-0.2249(7)	0.0577(1)
mVMC(8)+Lanczos	-1.382011(2)	-	-	-

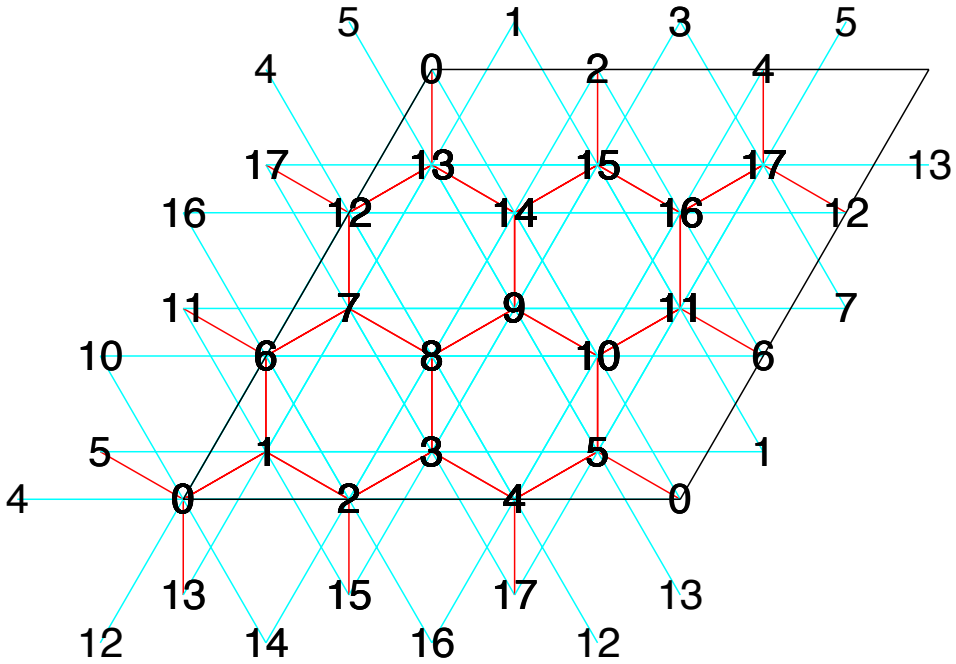
Table 5: Comparisons with exact diagonalization for one-dimensional Kondo-lattice model for $J = 1$ and $t = 1$. Upper (Lower) panel shows the results for spin singlet (triplet) sector. In the triplet sector $(S = 1)$, we take total momentum as $K = \pi$, which gives the lowest energy in $S = 1$, while we take take total momentum as $K = 0$ for $S = 1$.

See, H. Tsunetsugu *et al*., PRB 46, 3175 (1992),
H. Tsunetsugu *et al*., RMP 69, 809 (1997)

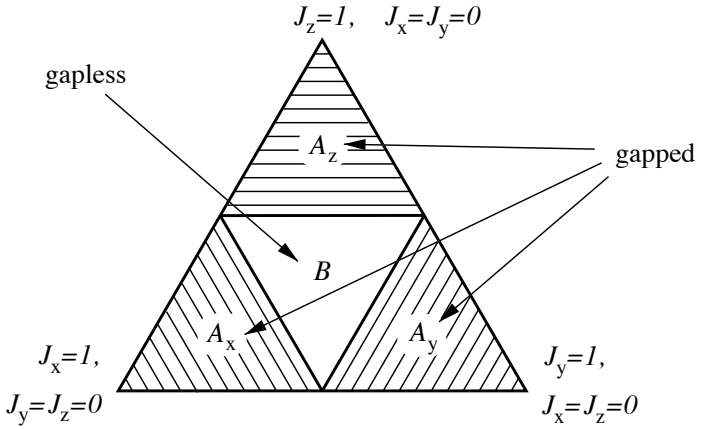
5: Kitaev model

$$H = -J_x \sum_{x-\text{bond}} S_i^x S_j^x - J_y \sum_{y-\text{bond}} S_i^y S_j^y - J_z \sum_{z-\text{bond}} S_i^z S_j^z$$

Bond-dependent interactions (Jx,Jy,Jz)



phase diagram



Annals of Physics 321, 2-111 (2016)

Plot lattice.gp

Exactly solvable →Quantum Spin Liquid

5: Kitaev model

$$H = -J_x \sum_{x-\text{bond}} S_i^x S_j^x - J_y \sum_{y-\text{bond}} S_i^y S_j^y - J_z \sum_{z-\text{bond}} S_i^z S_j^z$$

StdFace.def

```
W               = 2
L               = 3
model           = "SpinGC"
lattice         = "Honeycomb"
NVMCSample      = 200
NSROptItrStep = 5000
NSROptItrSmp  = 100
NMPTrans        = 6
J0x             = -1.0
J0y             =  0.0
J0z             =  0.0
J1x             =  0.0
J1y             = -1.0
J1z             =  0.0
J2x             =  0.0
J2y             =  0.0
J2z             = -1.0
```

Optimization is slow from the random initial states
→How to accurate optimization
[open problem]