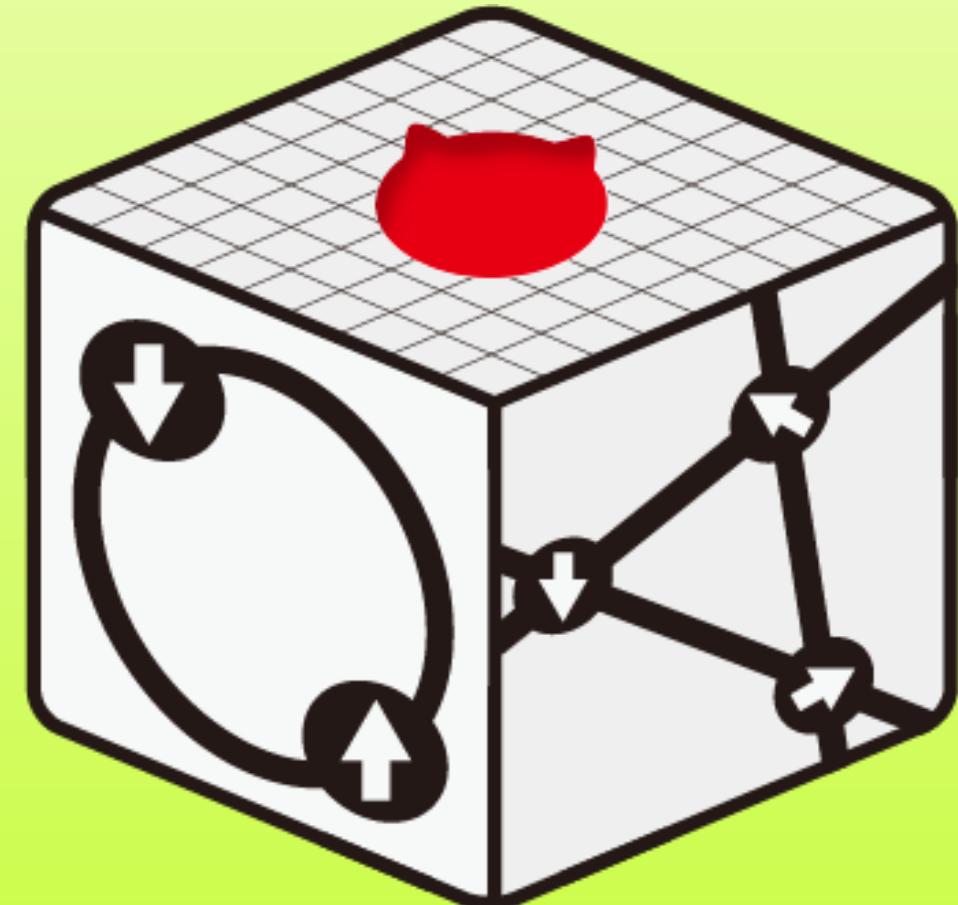


Introduction to mVMC

三澤 貴宏 (Takahiro MISAWA)

Institute for Solid State Physics (ISSP), University of Tokyo



mVMC

<https://www.pasums.issp.u-tokyo.ac.jp/mvmc/>

Outline

1. Introduction

- Strongly correlated electron systems (SCES)
- *Ab initio* (non-empirical) method for SCES

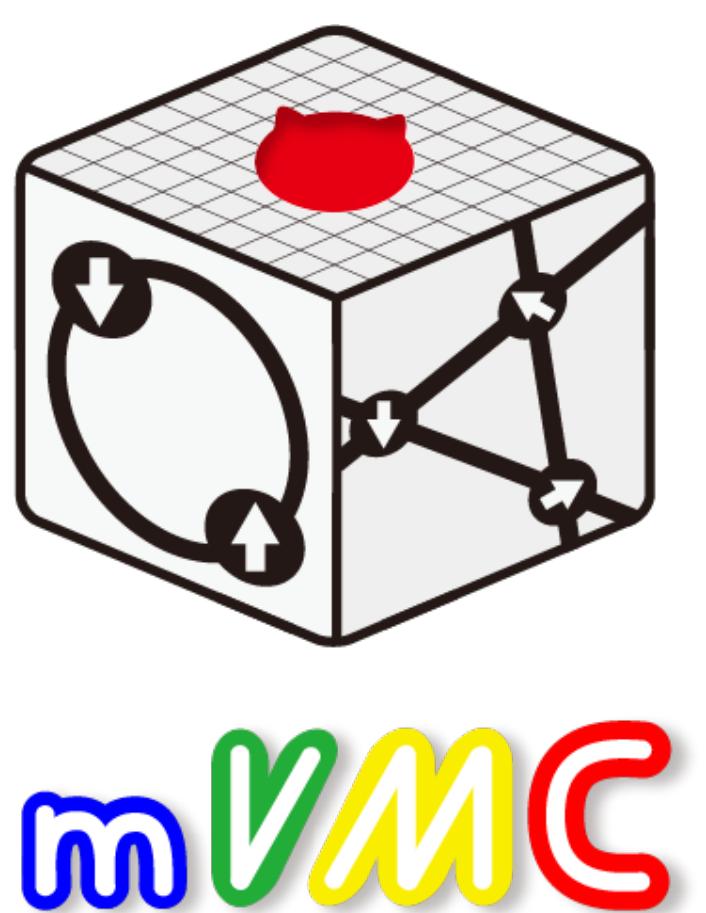
2. Basics of mVMC

- What is **variational Monte Carlo (VMC)** ?
- Conventional VMC vs mVMC
- Optimization method (SR method) based on **time-dependent variational principle**

3. Open-source software of mVMC

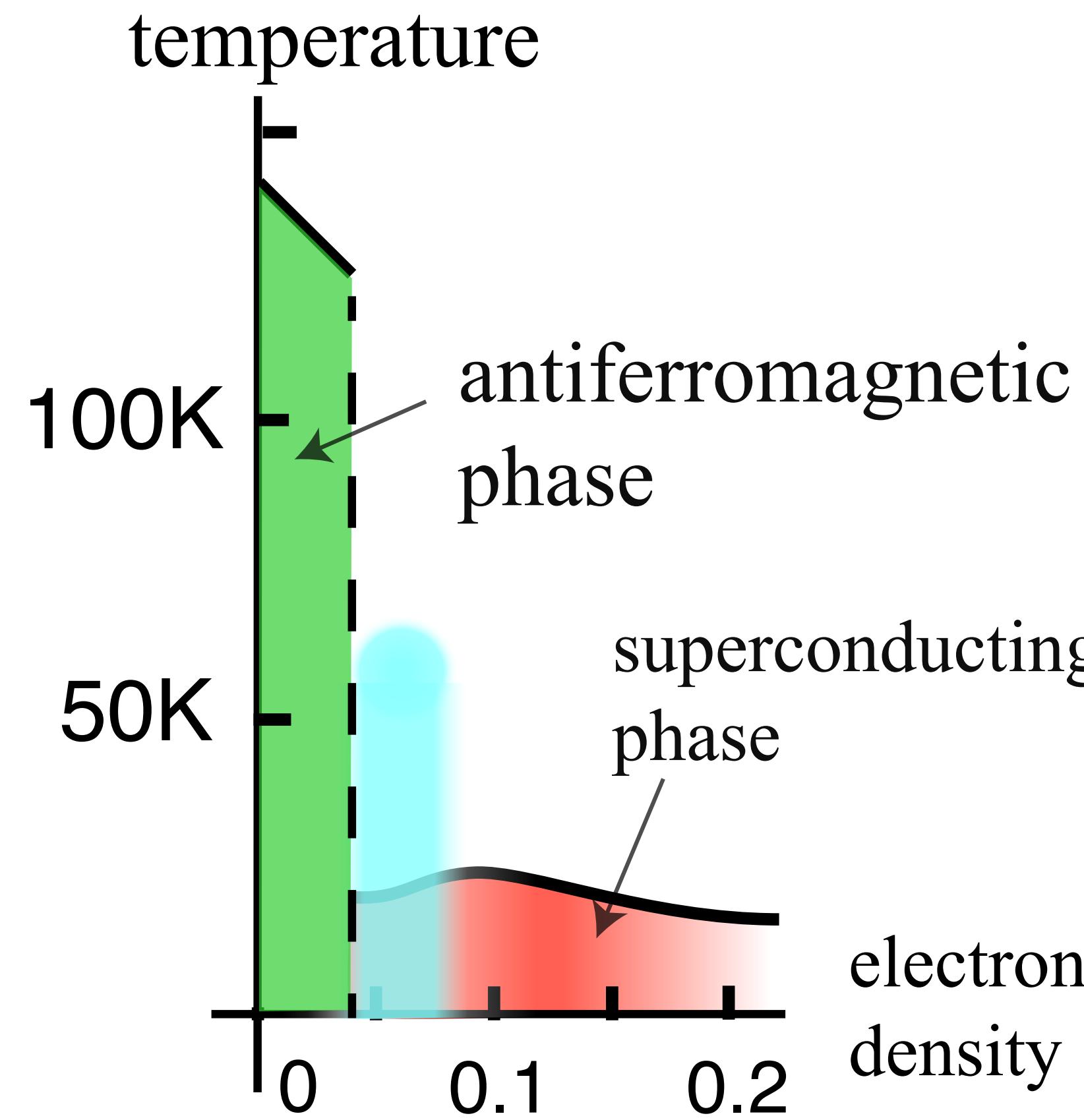
- How to get mVMC
- How to use mVMC [Standard & Expert mode]

4. Tutorial (1D Heisenberg model)



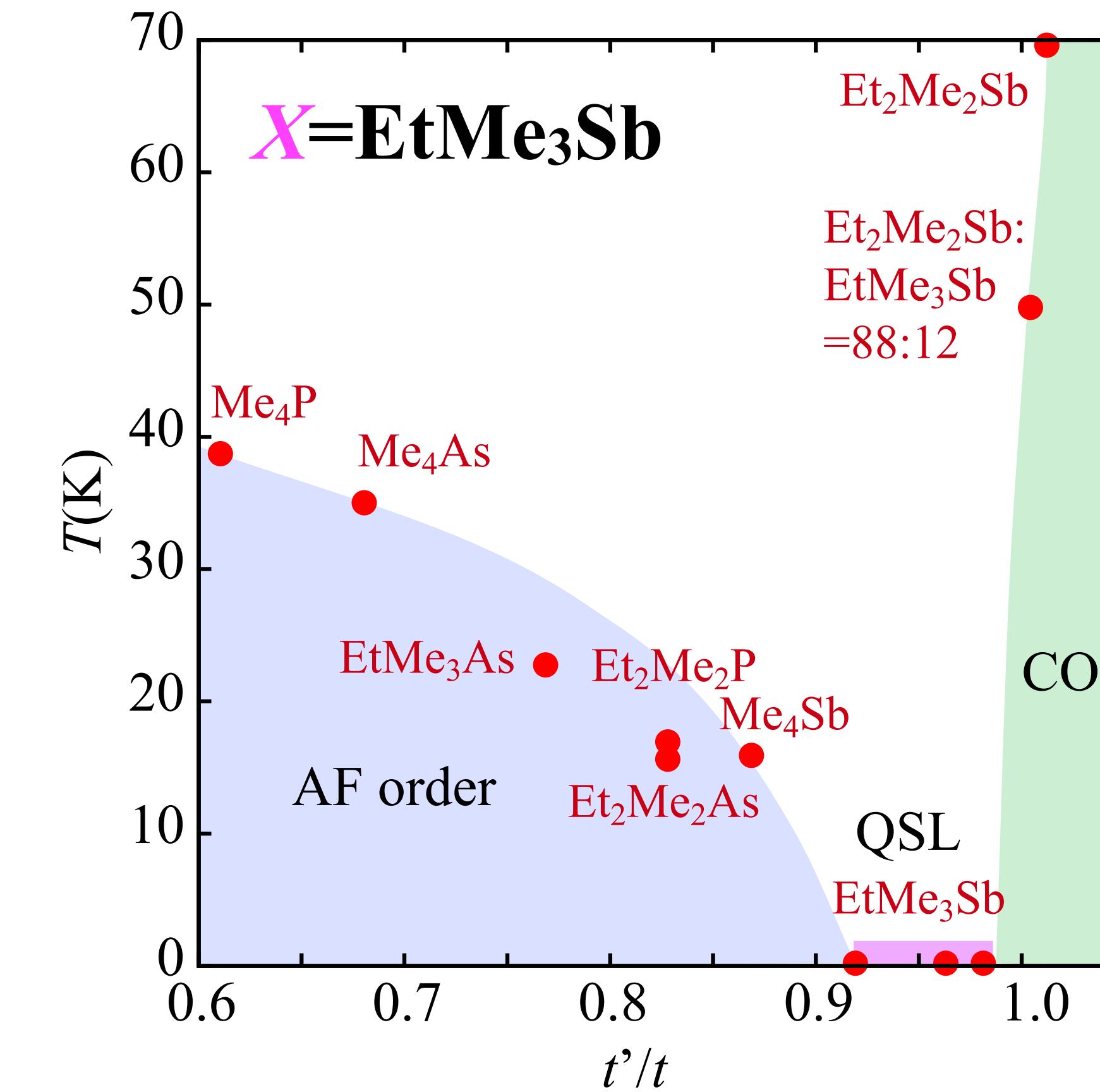
Exotic quantum phases in SCES

High- T_c SC: LaFeAsO



Y. Kamihara *et al*, JACS (2008)

Quantum spin liquid: β' -X[Pd(dmit)₂]₂



Read data from Fig. 17 in “K. Kanoda and R. Kato, Annu. Rev. Condens. Matter Phys. 2, 167-188 (2011)” and replotted

To clarify and predict exotic phenomena in SCES

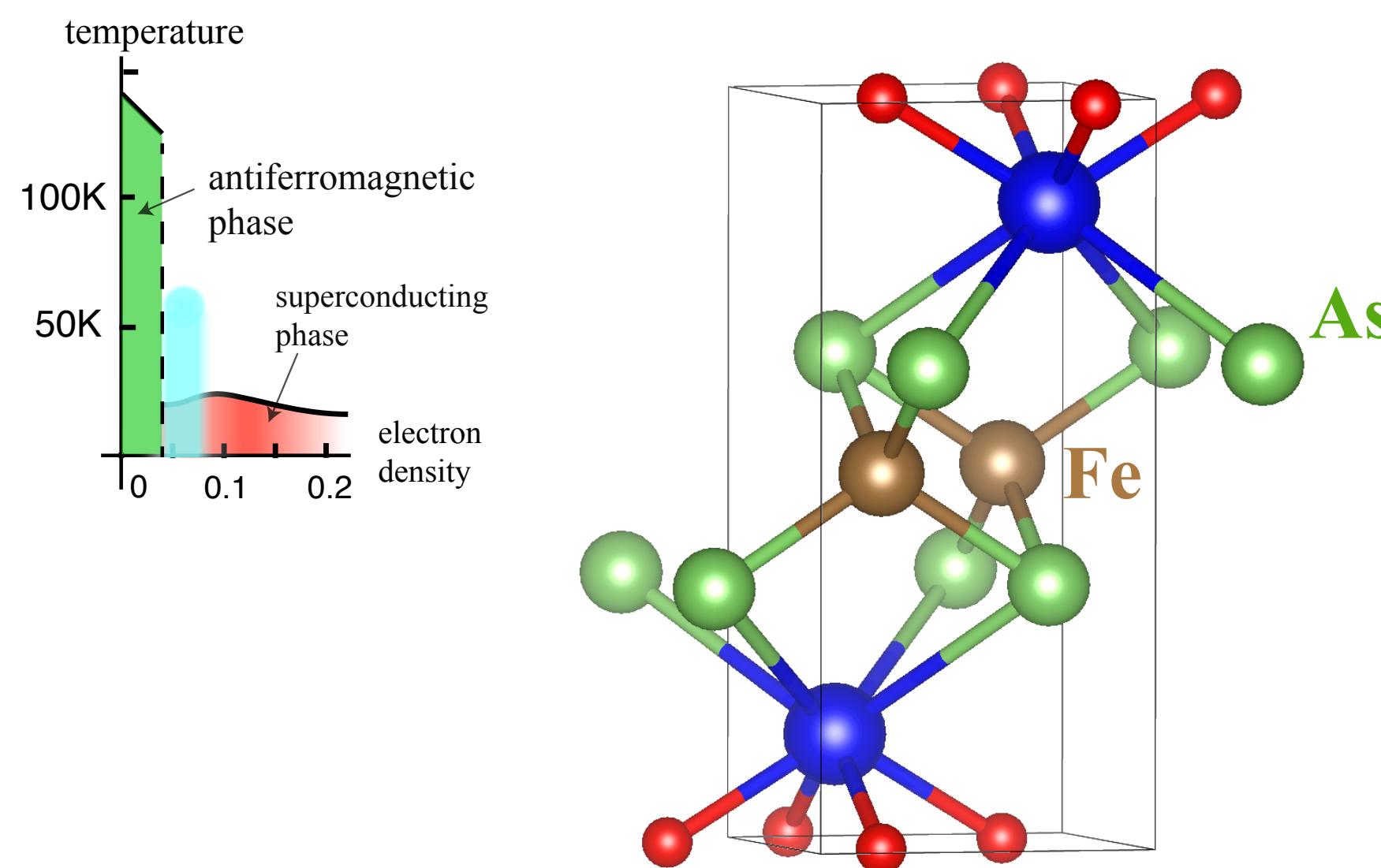
→ Accurate numerical methods for solving low-energy effective models are necessary

Low-energy effective models

High- T_c SC: LaFeAsO

5-orbital Hubbard Hamiltonians
obtained by *ab initio* calculations

$$\begin{aligned} \mathcal{H} = & \sum_{\sigma} \sum_{RR'} \sum_{nm} t_{mRnR'} a_{nR}^{\sigma\dagger} a_{mR'}^{\sigma} \rightarrow \text{Hopping Term} \\ & + \frac{1}{2} \sum_{\sigma\rho} \sum_{RR'} \sum_{nm} \left\{ U_{mRnR'} a_{nR}^{\sigma\dagger} a_{mR'}^{\rho\dagger} a_{mR'}^{\rho} a_{nR}^{\sigma} \right\} \rightarrow \text{Coulomb Term} \\ & + J_{mRnR'} (a_{nR}^{\sigma\dagger} a_{mR'}^{\rho\dagger} a_{nR}^{\rho} a_{mR'}^{\sigma} + a_{nR}^{\sigma\dagger} a_{nR}^{\rho\dagger} a_{mR}^{\rho} a_{mR'}^{\sigma}) \} \rightarrow \text{Exchange Term} \end{aligned}$$

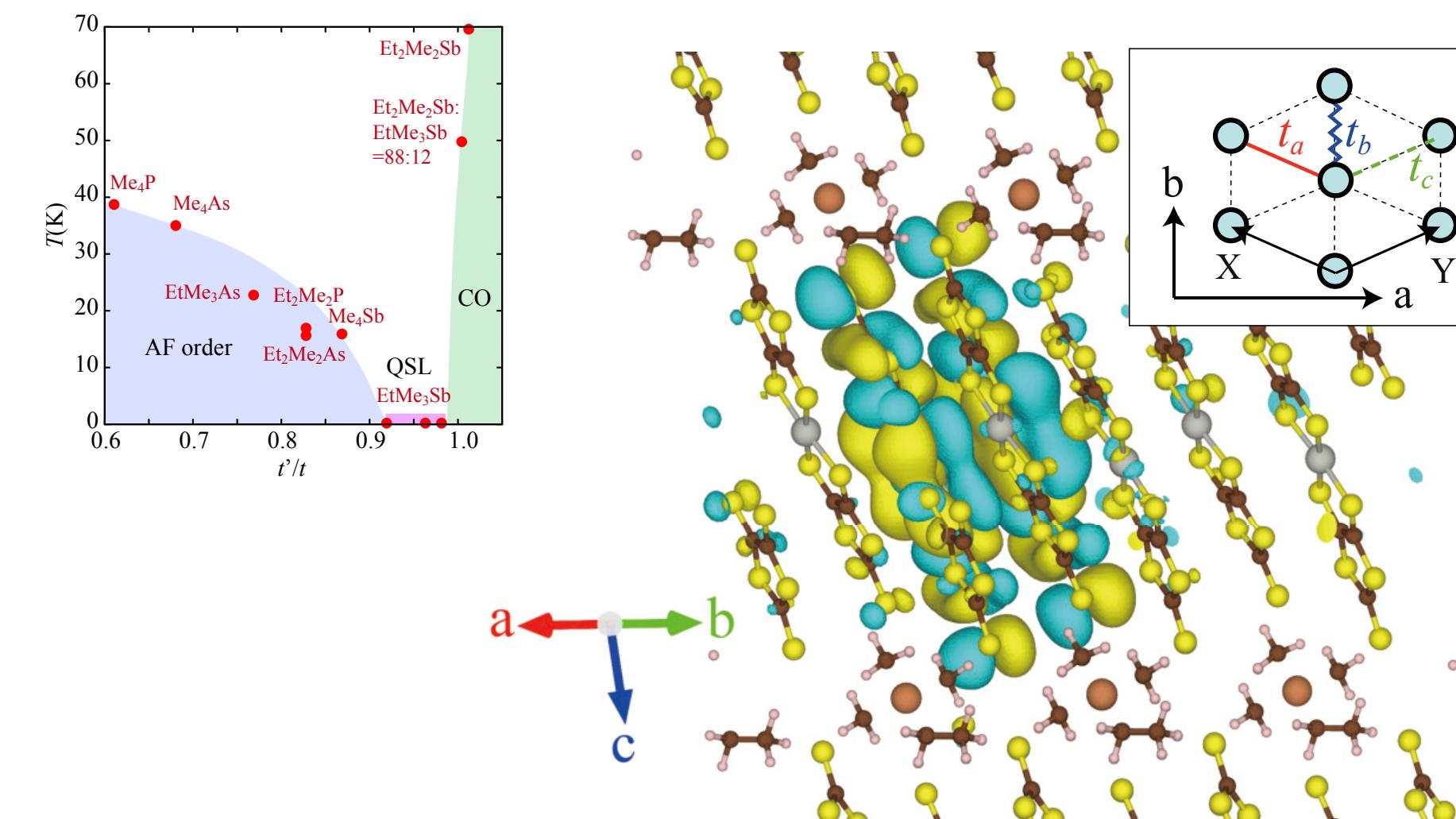


T. Misawa *et al.*, Nat. Com. 5, 5738 (2014)

Quantum spin liquid: β' -X[Pd(dmit)₂]₂

Single-band Hubbard Hamiltonians
obtained by *ab initio* calculations

$$H = \sum_{i,j} t_{ij} (c_{i\sigma}^\dagger c_{j\sigma} + \text{H.c.}) + U \sum_i n_{i\uparrow} n_{i\downarrow} + \sum_{i,j} V_{i,j} n_i n_j$$



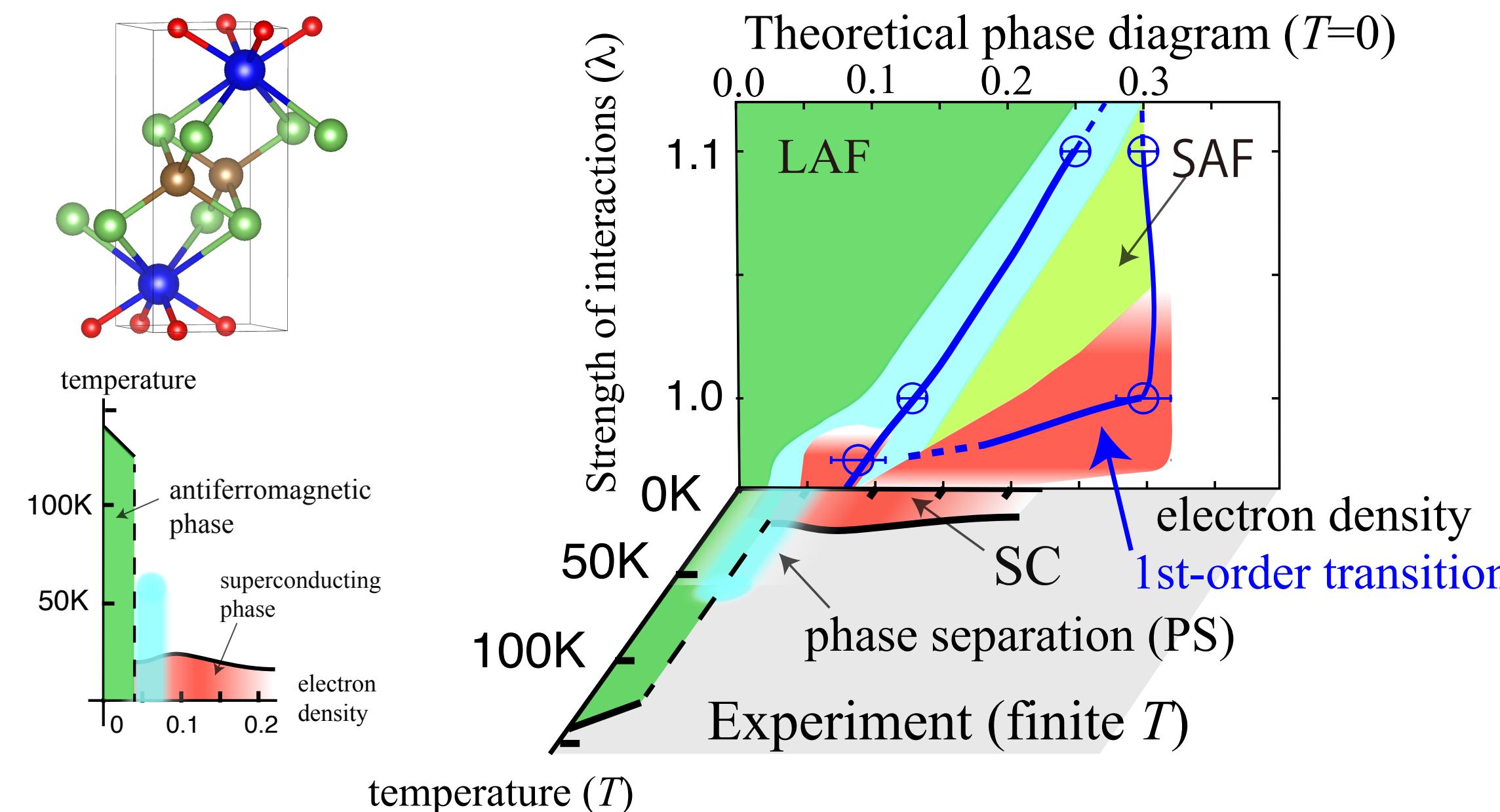
T. Misawa *et al.*, PRR 2, 032072(R) (2020)
K. Yoshimi *et al.*, PRR 3, 043224 (2021)
K. Ido *et al.*, npj Quantum mat. 7, 48 (2022)

Solving low-energy effective models by mVMC

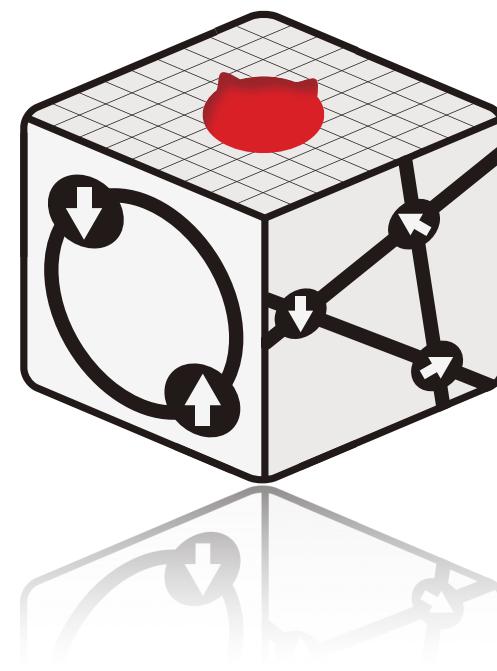
High- T_c SC: LaFeAsO

5-orbital Hubbard Hamiltonians obtained by *ab initio* calculations

$$\begin{aligned} \mathcal{H} = & \sum_{\sigma} \sum_{\mathbf{R}\mathbf{R}'} \sum_{nm} t_{m\mathbf{R}n\mathbf{R}'} a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}'}^{\sigma} \rightarrow \text{Hopping Term} \\ & + \frac{1}{2} \sum_{\sigma\rho} \sum_{\mathbf{R}\mathbf{R}'} \sum_{nm} \left\{ U_{m\mathbf{R}n\mathbf{R}'} a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}'}^{\rho\dagger} a_{m\mathbf{R}'}^{\rho} a_{n\mathbf{R}}^{\sigma} \rightarrow \text{Coulomb Term} \right. \\ & \left. + J_{m\mathbf{R}n\mathbf{R}'} (a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}}^{\rho\dagger} a_{n\mathbf{R}}^{\rho} a_{m\mathbf{R}'}^{\sigma} + a_{n\mathbf{R}}^{\sigma\dagger} a_{n\mathbf{R}}^{\rho\dagger} a_{m\mathbf{R}}^{\rho} a_{m\mathbf{R}'}^{\sigma}) \right\} \rightarrow \text{Exchange Term} \end{aligned}$$



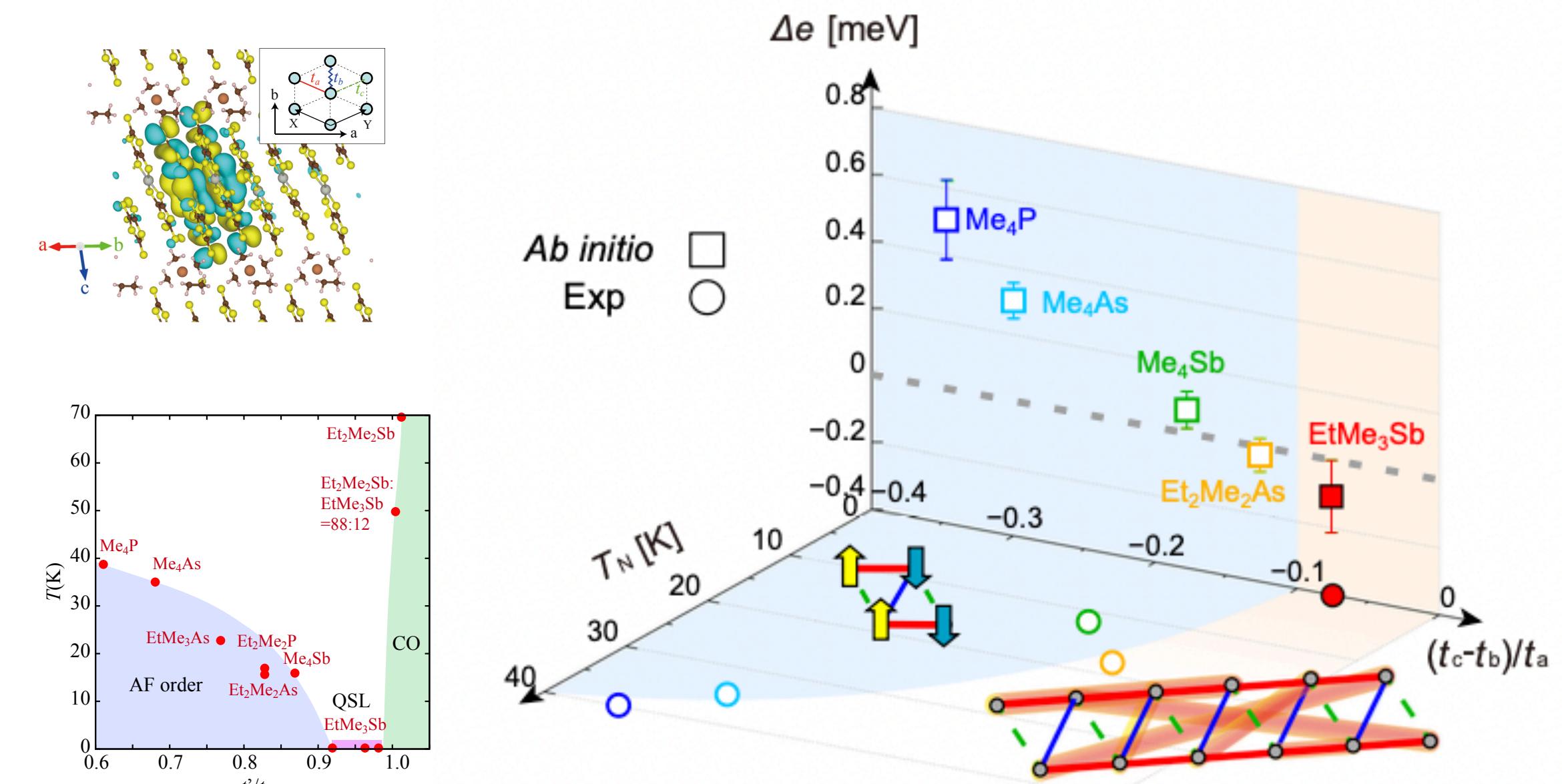
T. Misawa *et al.*, Nat. Com. 5, 5738 (2014)



Quantum spin liquid: β' -X[Pd(dmit)₂]₂

Single-band Hubbard Hamiltonians obtained by *ab initio* calculations

$$H = \sum_{i,j} t_{ij} (c_{i\sigma}^\dagger c_{j\sigma} + \text{H.c.}) + U \sum_i n_{i\uparrow} n_{i\downarrow} + \sum_{i,j} V_{i,j} n_i n_j$$



T. Misawa *et al.*, PRR 2, 032072(R) (2020)
K. Yoshimi *et al.*, PRR 3, 043224 (2021)
K. Ido *et al.*, npj Quantum mat. 7, 48 (2022)

Open-source software packages

HΦ: exact diagonalization

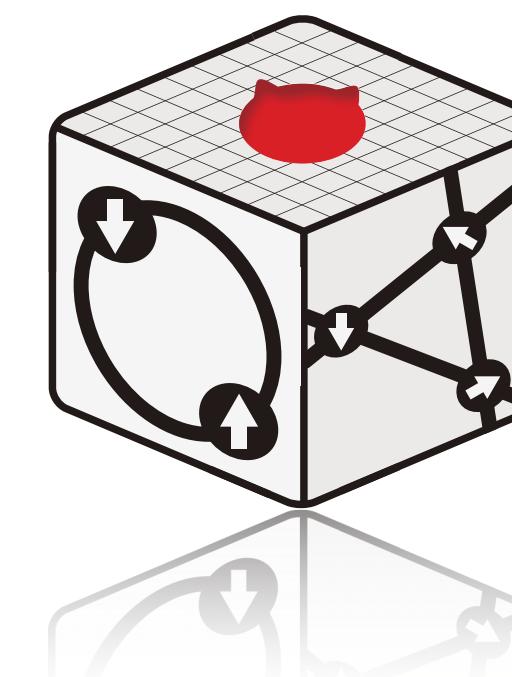


- Exact calc. for small system sizes (~40 sites)
- Cutting-edge theoretical and mathematical method
- Applications to QSL

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

② 第一回HPCIソフトウェア優秀賞(開発部門)受賞!

mVMC: many-variable variational MC



- Applicable for large system sizes (~1000 sites)
- Highly accurate and flexible method
- # of variational parameters > 10^4
- Applications to high-Tc SCs & QSL

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

① 第一回HPCIソフトウェア最優秀賞(開発部門)受賞!

RESPACK



- Derivation of low-energy effective Hamiltonians
- Using [wan2respack](#), user can use wannier functions obtained by Wannier90 as input for RESPACK

RESPACK: <https://sites.google.com/view/kazuma7k6r>
wan2respack: <https://github.com/respack-dev/wan2respack>

Seamless combination with the *ab initio* derivation of the low-energy effective models
→ Systematic & comprehensive calculations for SCES are now possible
(eg. Applications to molecular solids)

A part of development is supported by PASUMS@ ISSP.

Basics of variational Monte Carlo

- *Optimization* of variational parameters
[time-dependent variational principle]
- *Evaluation* of physical quantities [MC sampling]

Variational Monte Carlo (VMC) I

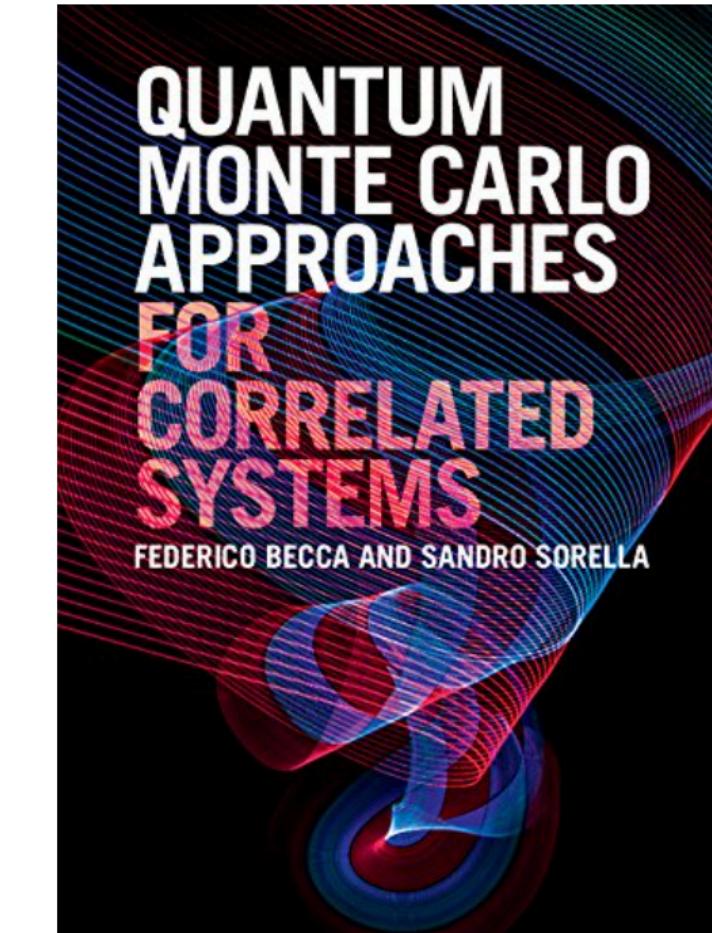
Variational principle

α :variational parameters

review: C. Gros,
Ann. Phys. 189, 53 (1989)

$$\min_{\alpha} E(\alpha) = \min_{\alpha} \frac{\langle \psi(\alpha) | \hat{H} | \psi(\alpha) \rangle}{\langle \psi(\alpha) | \psi(\alpha) \rangle}$$

Physical properties [MC sampling]
 x :real space configuration



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

$$\sim \frac{1}{N_{\text{MC}}} \underset{\text{MC}}{\sum_{\text{sampling}}} \frac{\langle \psi | \hat{A} | x \rangle}{\langle \psi | x \rangle}$$

$$\langle \psi | \hat{A} | x \rangle = \langle \psi | x' \rangle$$

Inner product

positive weight

$$\rho(x) = \frac{|\langle \psi | x \rangle|^2}{\langle \psi | \psi \rangle} > 0$$

F. Becca & S. Sorella

Variational Monte Carlo (VMC) II

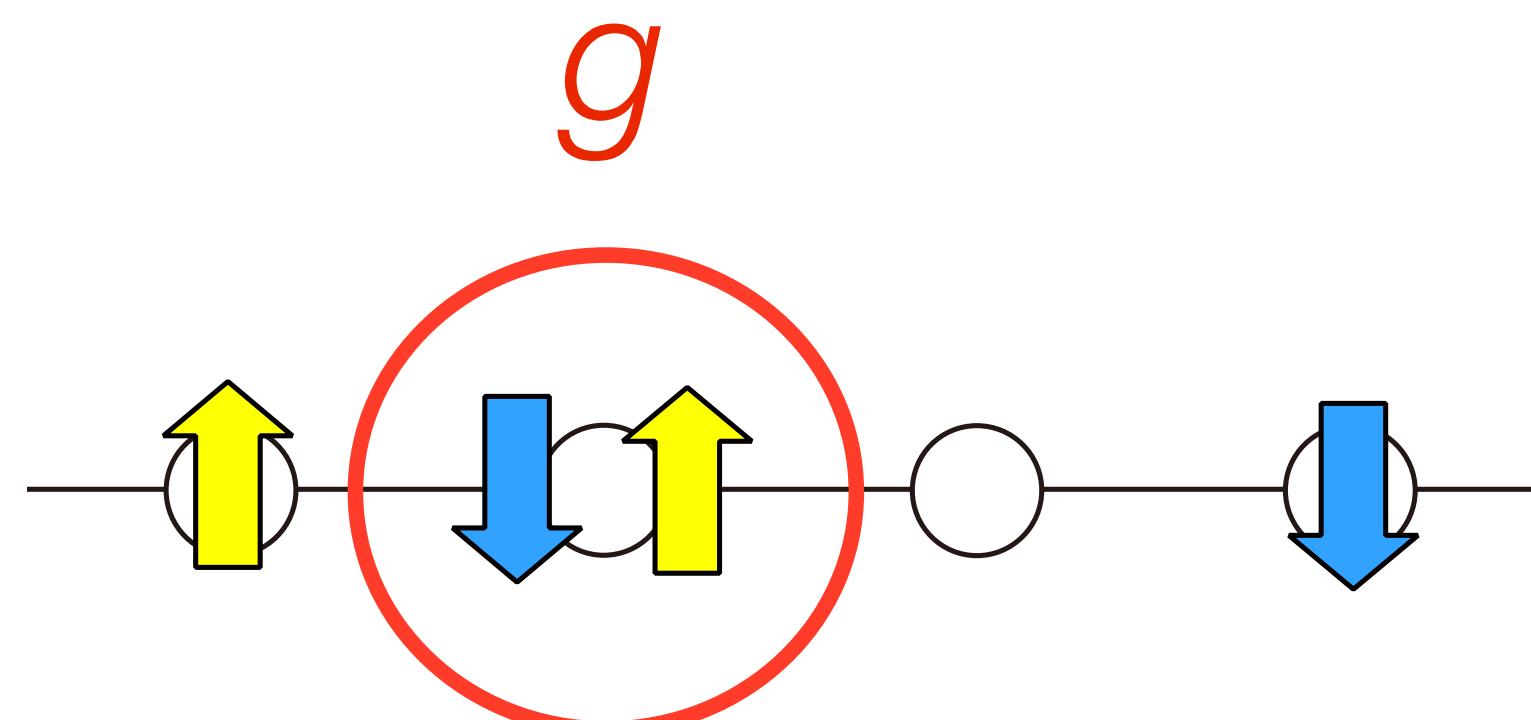
$$|\psi\rangle = \mathcal{P}_{\text{cor}} |\phi_0\rangle \longrightarrow \langle \psi | x \rangle = \langle \phi_0 | x \rangle \mathcal{P}_{\text{cor}}(x)$$

One-body part
correlation factor

determinant or Pfaffian

Ex. Gutzwiller factor

$$\mathcal{P}_G = e^{-g \sum_i n_{i\uparrow} n_{i\downarrow}}$$



$$\mathcal{P}_G |x\rangle = |x\rangle e^{-g D(x)}$$

Real-space diagonal correlation factor =
easy to calculate inner product.

Wave function of mVMC

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

D. Tahara and M. Imada, JPSJ (2008)
T. Misawa *et al.*, CPC (2019)

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

Pair-product part

$$|\phi_{\text{pair}}\rangle = \left[\sum_{i,j,\sigma,\tau}^{N_s} F_{ij}^{\sigma\tau} c_{i\sigma}^\dagger c_{j\tau}^\dagger \right]^{N_e/2} |0\rangle$$

Generalized BCS wave func.
 → correlated metal,
 AF, CO ordered states, SC,
 spin-orbit interactions

Correlation factors

Gutzwiller-Jastrow \mathcal{P}_G \mathcal{P}_J
 doublon-holon $\mathcal{P}_{d-h}^{(2)}$ $\mathcal{P}_{d-h}^{(4)}$

Projections

\mathcal{L}^S : Total spin, $S=0$

\mathcal{L}^K : Total momentum, $K=0$

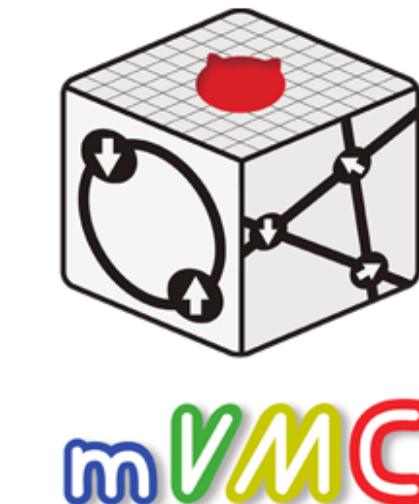
Update (SR method)

Mimimizing $E_{\vec{\alpha}} = \langle H \rangle_{\vec{\alpha}}$

$$\vec{\alpha}_{\text{new}} - \vec{\alpha}_{\text{old}} = -X^{-1} \vec{g}$$

$g_k = \frac{\partial E_{\vec{\alpha}}}{\partial \alpha_k}$ gradient of energy
 (MC Sampling)

S. Sorella PRB (2001) Equivalent to natural gradient method
 S.-I. Amari Neural Comp. (1998)



Optimization of many variational parameters ($>=10000$)
 → High-accuracy wave functions for ground states

Conventional VMC v.s. mVMC

Conventional VMC:

Strong constraint on wave functions [# of parameters~10]

ex. antiferromagnetic phase

$$|\phi_{\text{AF}}\rangle = \prod_{|\mathbf{k}| < k_F, \sigma} a_{\mathbf{k}\sigma}^\dagger |0\rangle$$

$$a_{\mathbf{k}\sigma}^\dagger = u_{\mathbf{k}} c_{\mathbf{k},\sigma}^\dagger + \sigma v_{\mathbf{k}} c_{\mathbf{k}+Q,\sigma}^\dagger$$

$$u_{\mathbf{k}}^2 = \frac{1}{2} \left(1 - \frac{E_{\mathbf{k}}}{\sqrt{E_{\mathbf{k}}^2 + \Delta_{\text{AF}}^2}} \right)$$

Variational parameters = AF order parameter + Gutzwiller factors +etc.

Disadvantages of conventional VMC

- Accuracy is *not enough* due to the strong constraint
- *Overestimating* the stability of ordered phases
- It is difficult to treat *realistic models* (*ab initio* effective models)

$$\begin{aligned} \mathcal{H} = & \sum_{\sigma} \sum_{\mathbf{R}\mathbf{R}'} \sum_{nm} t_{m\mathbf{R}n\mathbf{R}'} a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}'}^{\sigma} \rightarrow \text{Hopping Term} \\ & + \frac{1}{2} \sum_{\sigma\rho} \sum_{\mathbf{R}\mathbf{R}'} \sum_{nm} \left\{ U_{m\mathbf{R}n\mathbf{R}'} a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}'}^{\rho\dagger} a_{m\mathbf{R}'}^{\rho} a_{n\mathbf{R}}^{\sigma} \right\} \rightarrow \text{Coulomb Term} \\ & + J_{m\mathbf{R}n\mathbf{R}'} (a_{n\mathbf{R}}^{\sigma\dagger} a_{m\mathbf{R}}^{\rho\dagger} a_{n\mathbf{R}}^{\rho} a_{m\mathbf{R}'}^{\sigma} + a_{n\mathbf{R}}^{\sigma\dagger} a_{n\mathbf{R}'}^{\rho\dagger} a_{m\mathbf{R}'}^{\rho} a_{m\mathbf{R}'}^{\sigma}) \} \rightarrow \text{Exchange Term} \end{aligned}$$

Example of *ab initio* effective models for iron-based superconductors
= Five-orbital Hubbard model

Conventional VMC v.s. mVMC

Conventional VMC:

Strong constraint on wave functions [# of parameters~10]

ex. antiferromagnetic phase

$$|\phi_{\text{AF}}\rangle = \prod_{|\mathbf{k}| < k_F, \sigma} a_{\mathbf{k}\sigma}^\dagger |0\rangle$$

$$a_{\mathbf{k}\sigma}^\dagger = u_{\mathbf{k}} c_{\mathbf{k},\sigma}^\dagger + \sigma v_{\mathbf{k}} c_{\mathbf{k}+Q,\sigma}^\dagger$$

$$u_{\mathbf{k}}^2 = \frac{1}{2} \left(1 - \frac{E_{\mathbf{k}}}{\sqrt{E_{\mathbf{k}}^2 + \Delta_{\text{AF}}^2}} \right)$$

Variational parameters = AF order parameter + etc.

many-variable VMC (mVMC):

flexibility of one-body part [# of parameters > 10000]

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

$$|\phi_{\text{AP+P}}\rangle = \left(\sum_{i\sigma, j\tau} F_{i\sigma, j\tau} c_{i\sigma}^\dagger c_{j\tau}^\dagger \right)^{N_e/2} |0\rangle$$

f_{ij} , F_{IJ} : variational
parameters

f_{ij} [i, j real-space indices] → correlated paramagnetic state,
symmetry breaking phase (AF etc.), SC states

Optimization method

(General) Gradient method α :variational parameters

$$\Delta \alpha = \alpha_{\text{new}} - \alpha_{\text{old}} = -X^{-1}g \quad \left(g_k = \frac{\partial E_\alpha}{\partial \alpha_k} \right)$$

Optimization method

(General) Gradient method α :variational parameters

$$\Delta \alpha = \alpha_{\text{new}} - \alpha_{\text{old}} = -X^{-1}g \quad \left(g_k = \frac{\partial E_\alpha}{\partial \alpha_k} \right)$$

Steepest decent method [slow due to *redundancy*]

$X = I$ (identity matrix)

Optimization method

(General) Gradient method

α :variational parameters

$$\Delta \alpha = \alpha_{\text{new}} - \alpha_{\text{old}} = -X^{-1}g \quad \left(g_k = \frac{\partial E_\alpha}{\partial \alpha_k} \right)$$

Steepest decent method [slow due to *redundancy*]

$X = I$ (identity matrix)

Newton method [second derivatives are expensive]

$X = h$ (Hessian : $h_{\alpha\beta} = \frac{\partial^2 E}{\partial \alpha \partial \beta}$)

Optimization method

(General) Gradient method

α :variational parameters

$$\Delta \alpha = \alpha_{\text{new}} - \alpha_{\text{old}} = -X^{-1}g \quad \left(g_k = \frac{\partial E_\alpha}{\partial \alpha_k} \right)$$

Steepest decent method [slow due to *redundancy*]

$X = I$ (identity matrix)

Newton method [second derivatives are expensive]

$$X = h \quad (\text{Hessian} : h_{\alpha\beta} = \frac{\partial^2 E}{\partial \alpha \partial \beta})$$

Stochastic reconfiguration (SR) method [fast & stable]

$X = S$ (overlap matrix : $S_{\alpha\beta} = \langle \bar{\psi}_\alpha | \bar{\psi}_\beta \rangle$)

$$|\bar{\psi}_\alpha\rangle = \frac{\partial |\bar{\psi}\rangle}{\partial \alpha}, \quad |\bar{\psi}\rangle = \frac{|\psi\rangle}{\sqrt{\langle\psi|\psi\rangle}}$$

SR method [S. Sorella, PRB 2001]
Natural gradient [S.-I. Amari, Neural Comp. 1998]

Time-dependent variational principle

Imaginary time evolution

$$\frac{\partial|\psi\rangle}{\partial\tau} = -\hat{H}|\psi\rangle$$

$$\left\| \frac{\partial|\psi\rangle}{\partial\tau} + \hat{H}|\psi\rangle \right\| = 0 \rightarrow \min_{\dot{\alpha}} \left\| \sum_k \frac{\partial\alpha_k}{\partial\tau} \frac{\partial|\psi\rangle}{\partial\alpha_k} + \hat{H}|\psi\rangle \right\|$$

(imaginary) time-dependent variational principle

A. D. McLachlan, Mol. Phys. 8, 39 (1964)

SR method = Imaginary-time evolution in restricted Hilbert space

$$\min_{\dot{\alpha}} \left\| \sum_k \frac{\partial\alpha_k}{\partial\tau} \frac{\partial|\bar{\psi}\rangle}{\partial\alpha_k} + (\hat{H} - \langle\hat{H}\rangle)|\bar{\psi}\rangle \right\|$$

$$\rightarrow \Delta\alpha = -\frac{\Delta\tau}{2} S^{-1} g$$

S : overlap matrix

SR method can be used for real-time evolution (Ido *et al.*, PRB 2015)
& finite-temperature calculations (Takai *et al.*, JPSJ 2016)

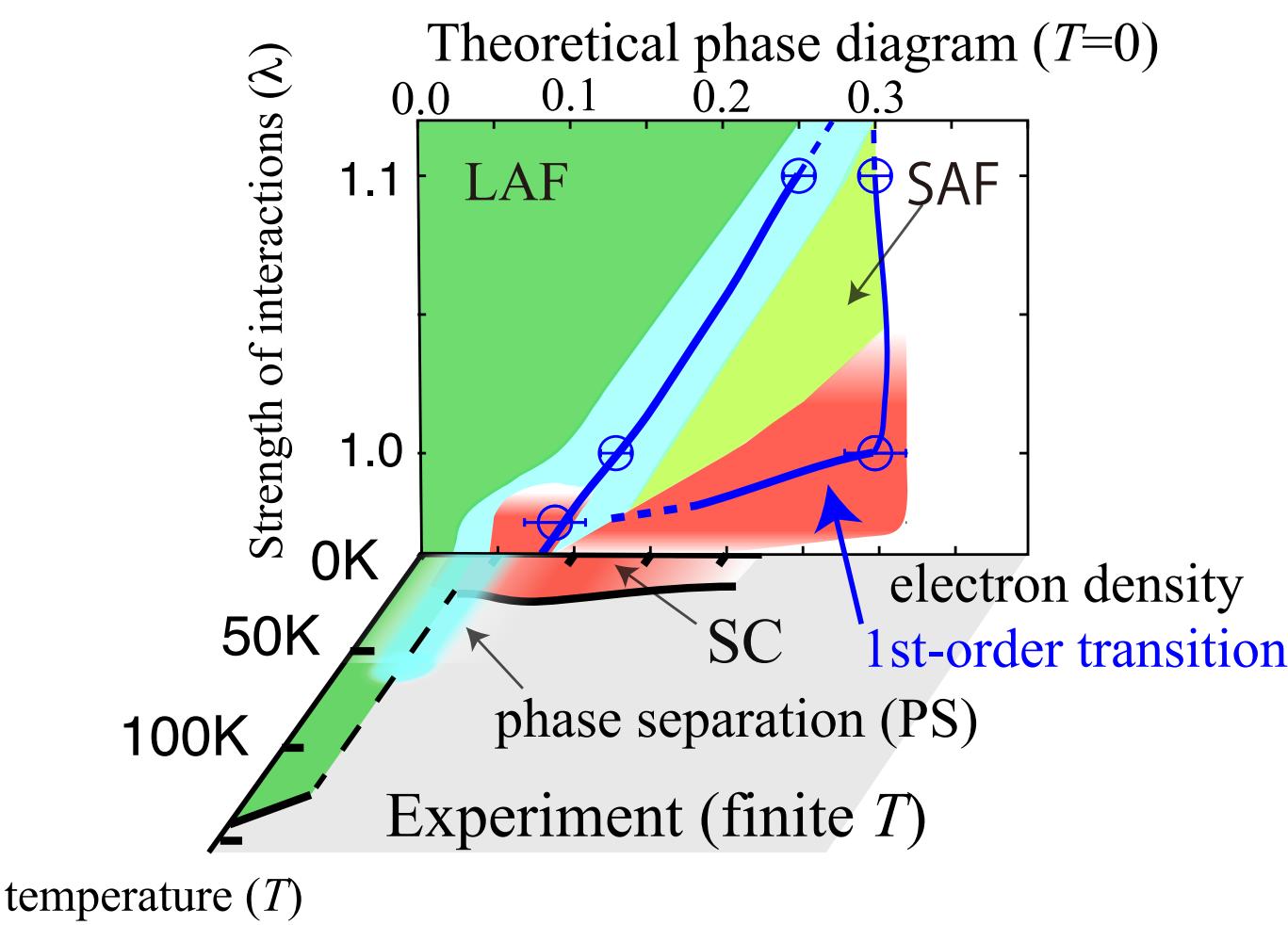
Advantages of mVMC



- No negative-sign problem
positive weight $\rho(x) > 0$
- Wide applicable range [strong correlations, geometrical frustration, multi orbital system, any dimensions ...]
- Natural extensions of mean-field calculations
- Easy to include many-body correlations through correlation factors (Gutzwiller, Jastrow, Doublon-Holon..)
- Systematic improvement is possible (power Lanczos, backflow, multi Pfaffian method ...)
 - Not only for ground-state calculations →
finite-temperature calculations, real-time evolution !

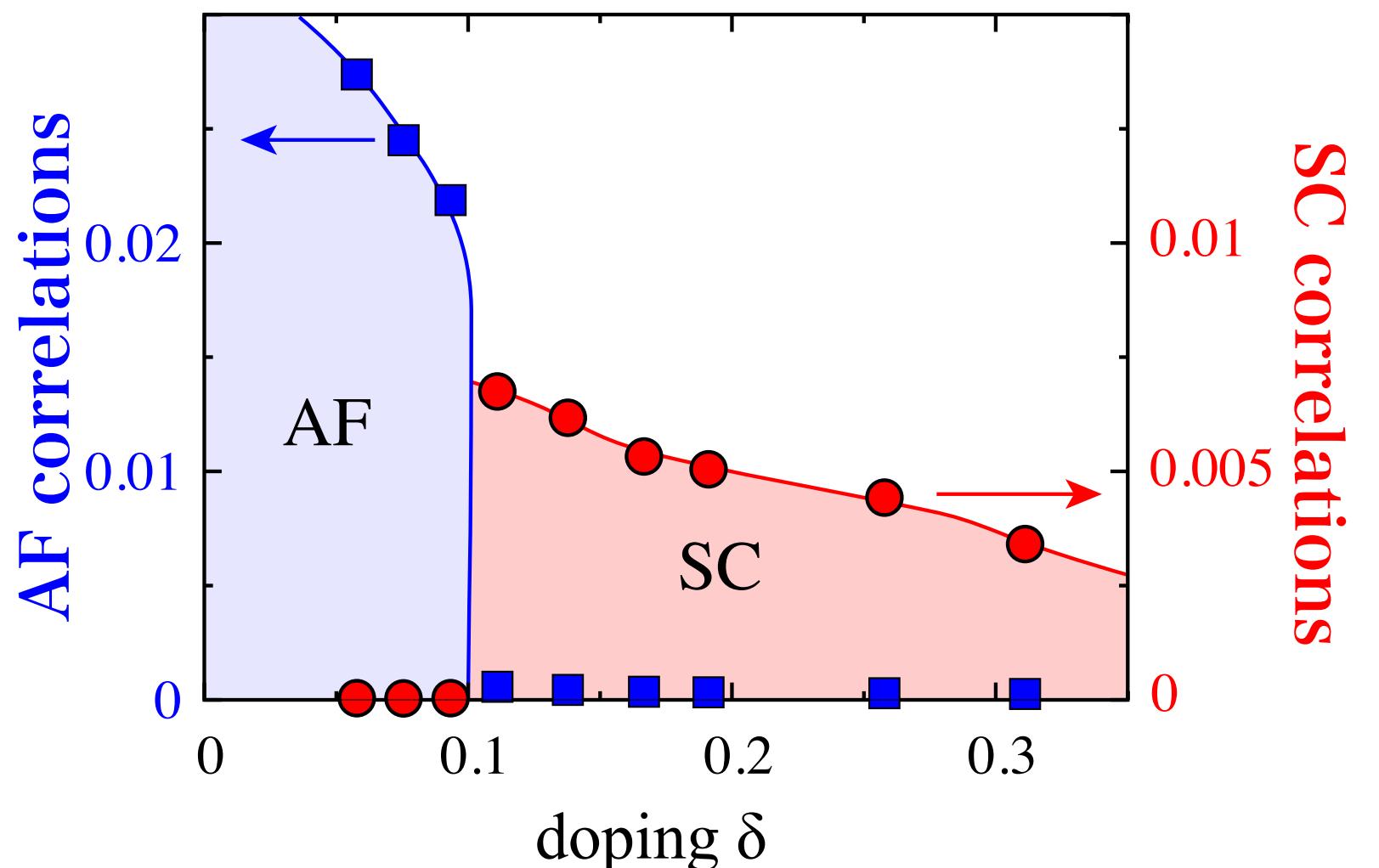
Applications of mVMC

LaFeAsO



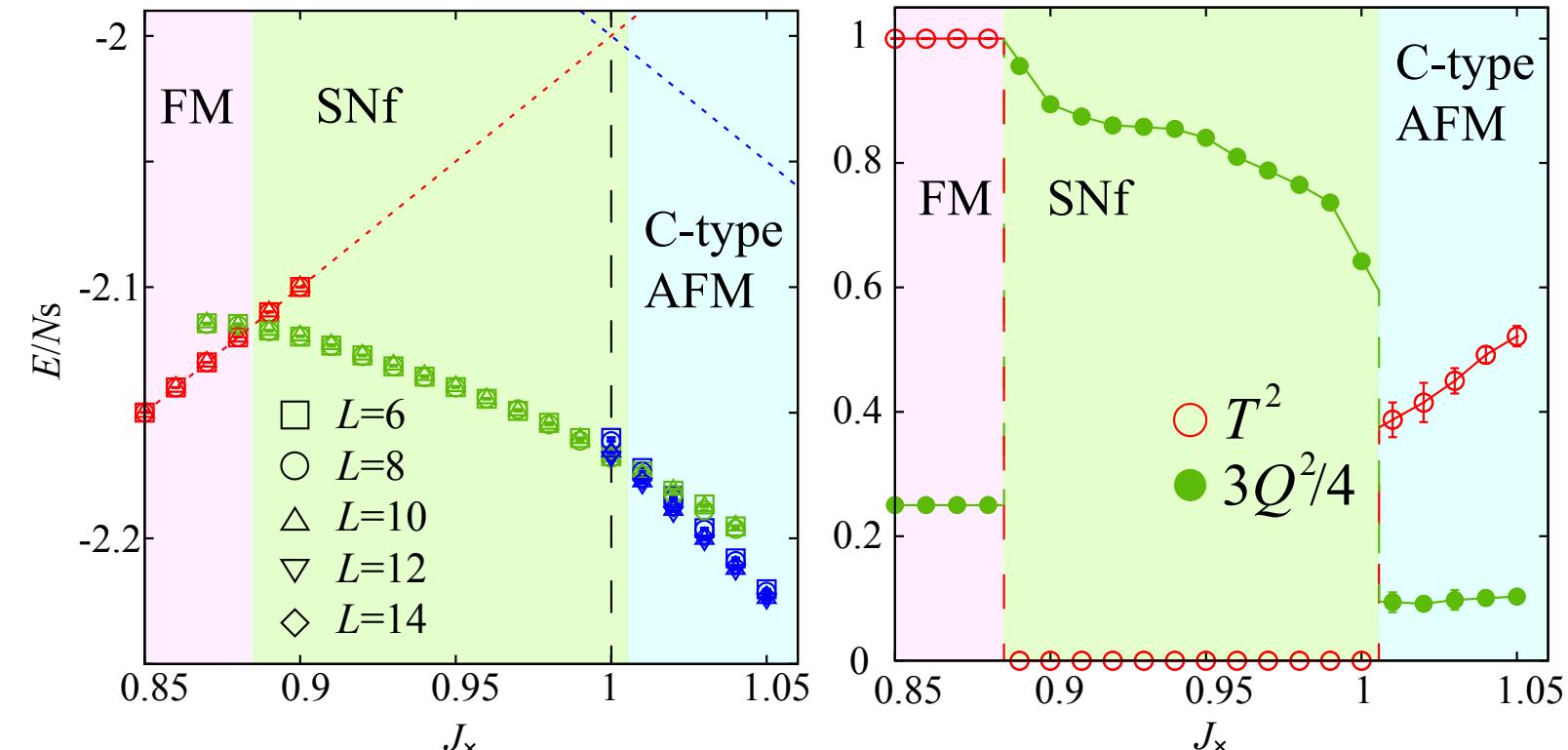
T. Misawa and M. Imada, Nat. Commun (2014).

HgBa₂CuO_{4+δ}



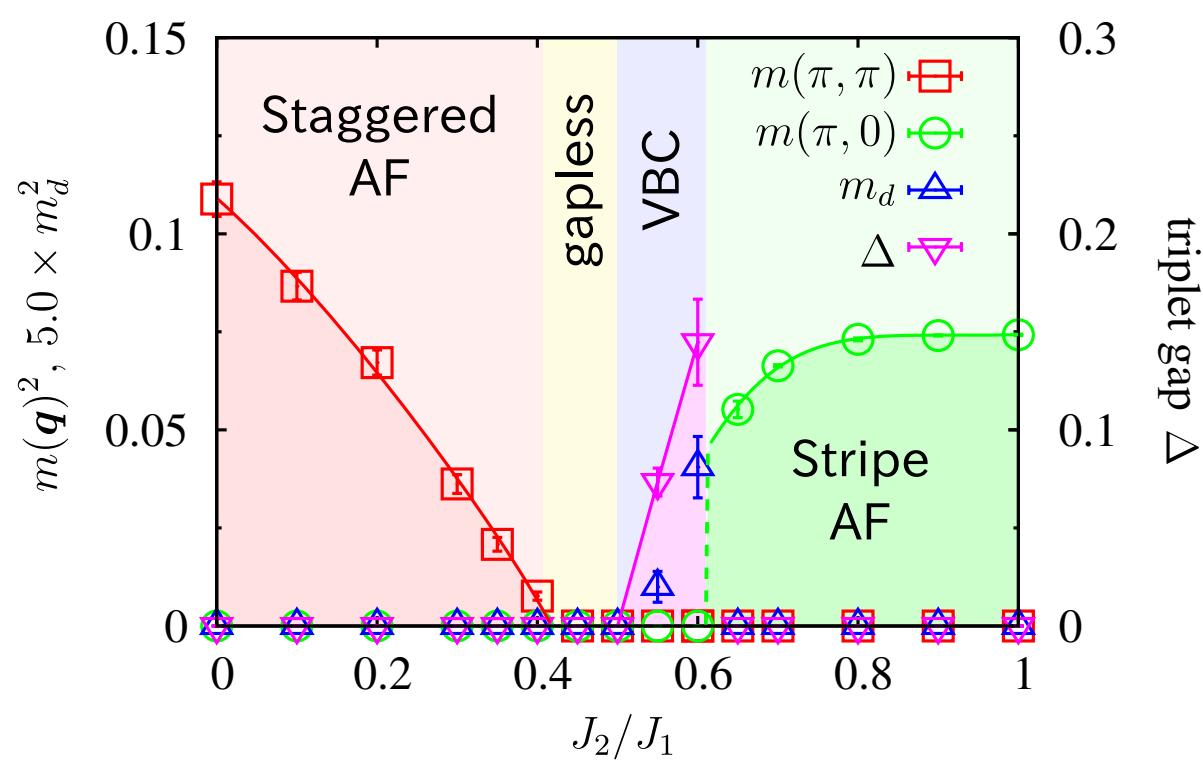
T. Ohgoe *et al.*, PRB (2020).

Spin nematic phase



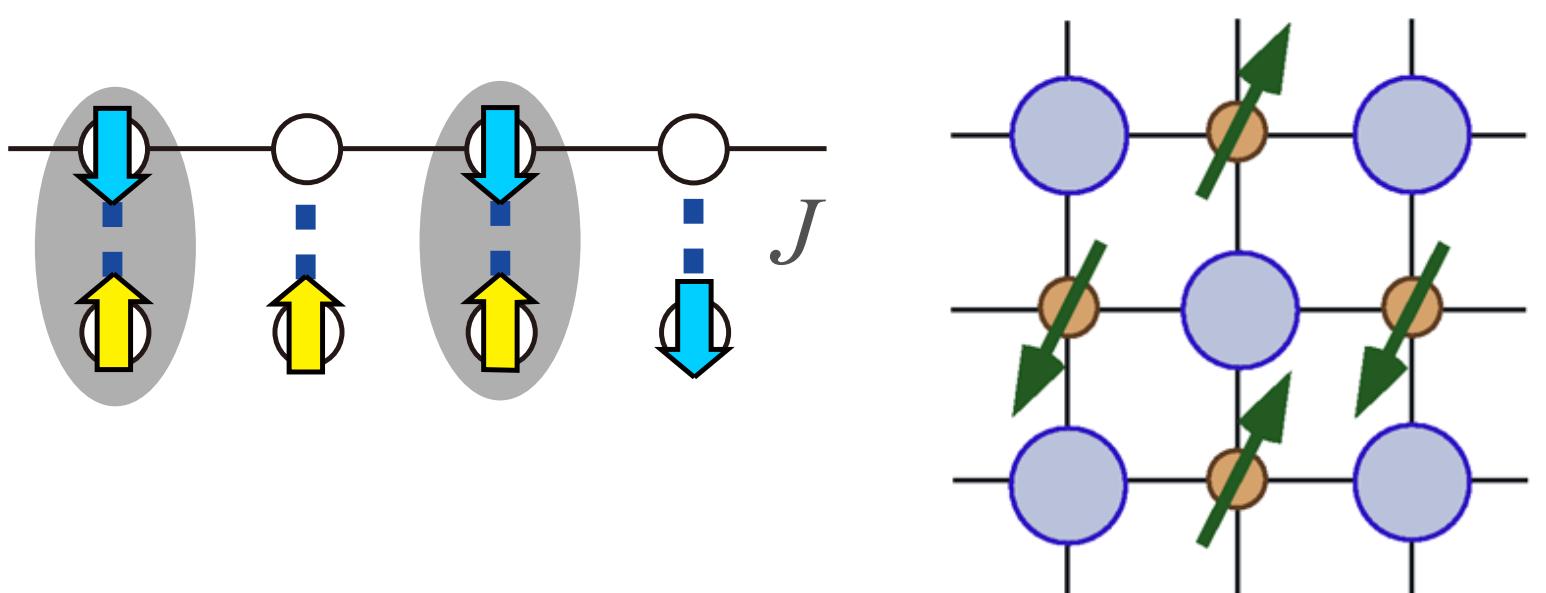
T. Hikihara *et al.*, PRB(2019)

J_1 - J_2 Heisenberg



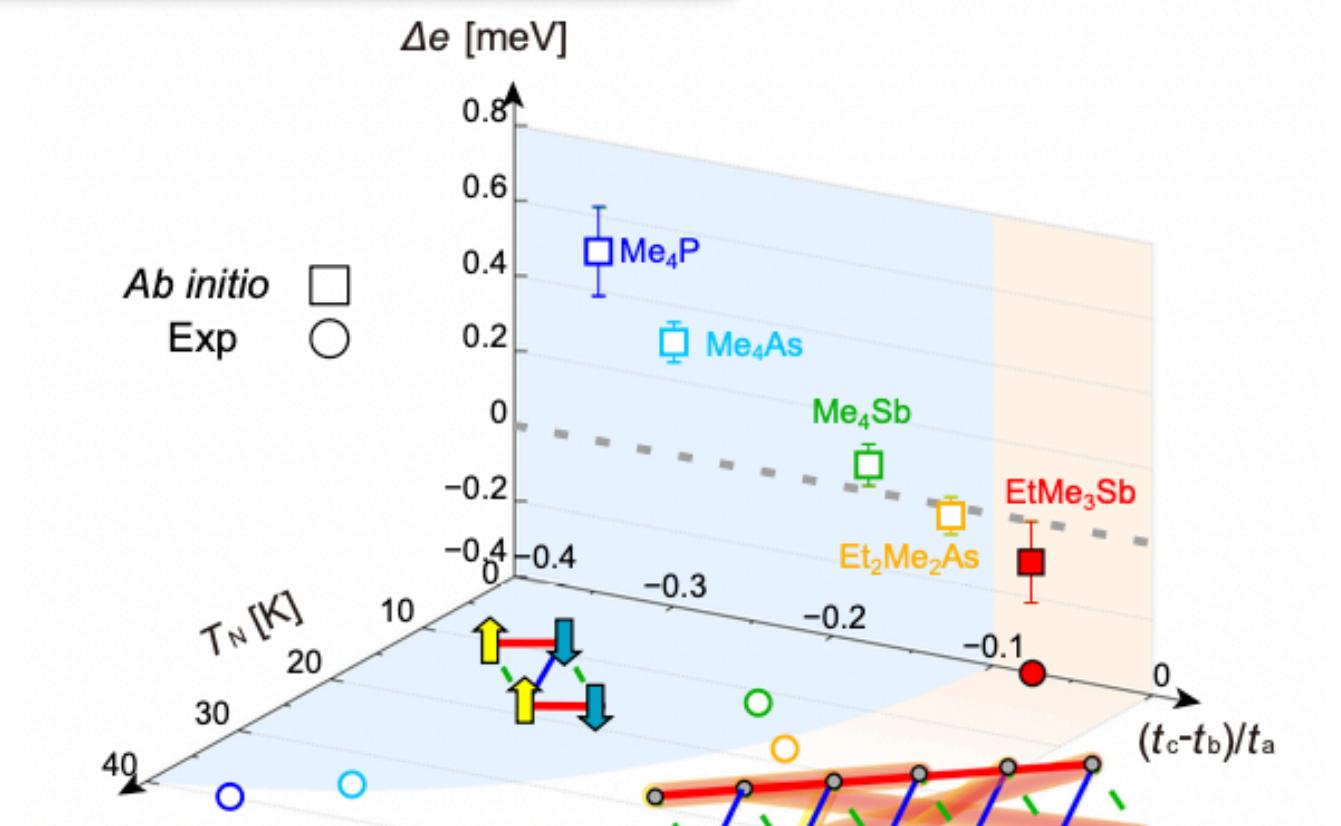
S. Motira and M. Imada JPSJ (2014).
see also Y. Nomura and M. Imada PRX (2021).

CO in Kondo lattice model



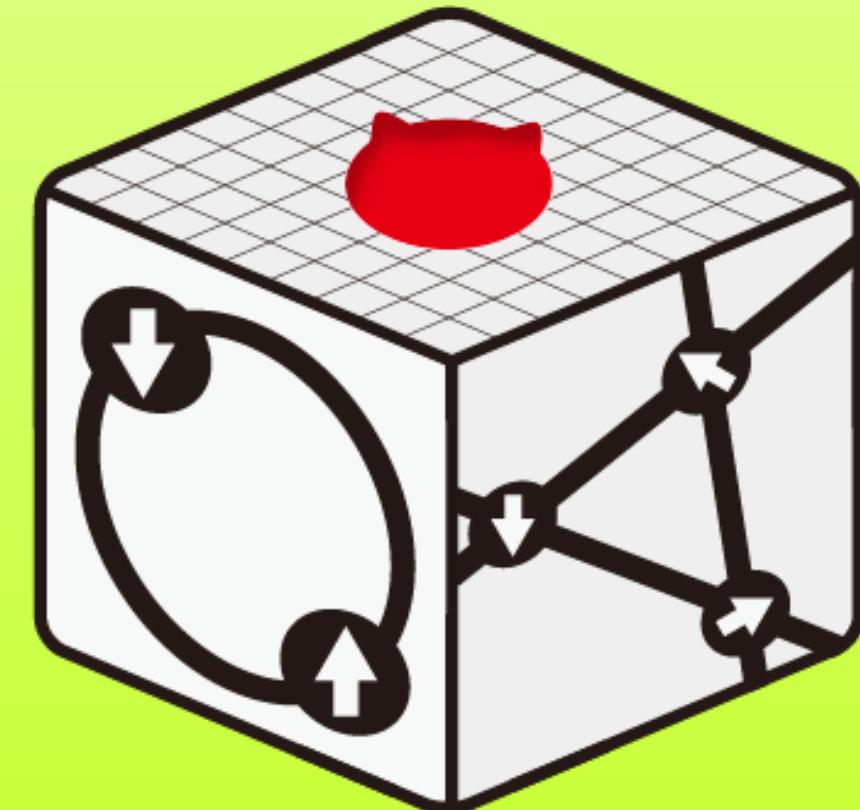
T. Misawa *et al.*, PRL (2013).

QSL in dmit salts



K. Ido *et al.*, npj QM (2022)

Open-source software of mVMC



mVMC

<https://www.pasums.issp.u-tokyo.ac.jp/mvmc/>

Main developers of mVMC (2018-)

M. Kawamura



K. Ido



Y. Motoyama



K. Yoshimi



T. Kato



T. Misawa



S. Morita



T. Ohgoe



RuQing Xu

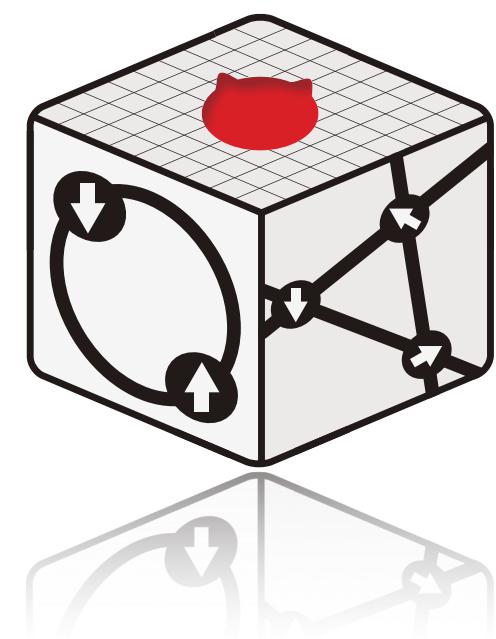


M. Imada



🥇 第一回HPCIソフトウェア最優秀賞(開発部門)受賞!

Development of mVMC is supported by
“*Project for advancement of software
usability in materials science*” by ISSP

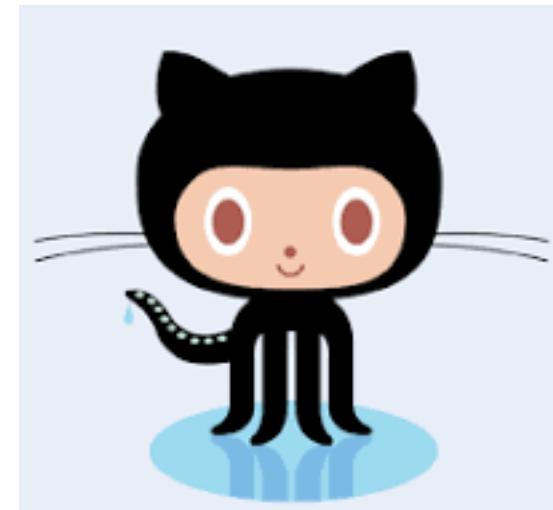


How to get mVMC

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

HP: <https://www.pasums.issp.u-tokyo.ac.jp/mvmc/>

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



mVMC is pre-installed in supercomputer in ISSP (ohtaka, kugui)

/home/issp/materiapps/intel/mvmc/

全国のスパコンにもプレインストール [RIST]

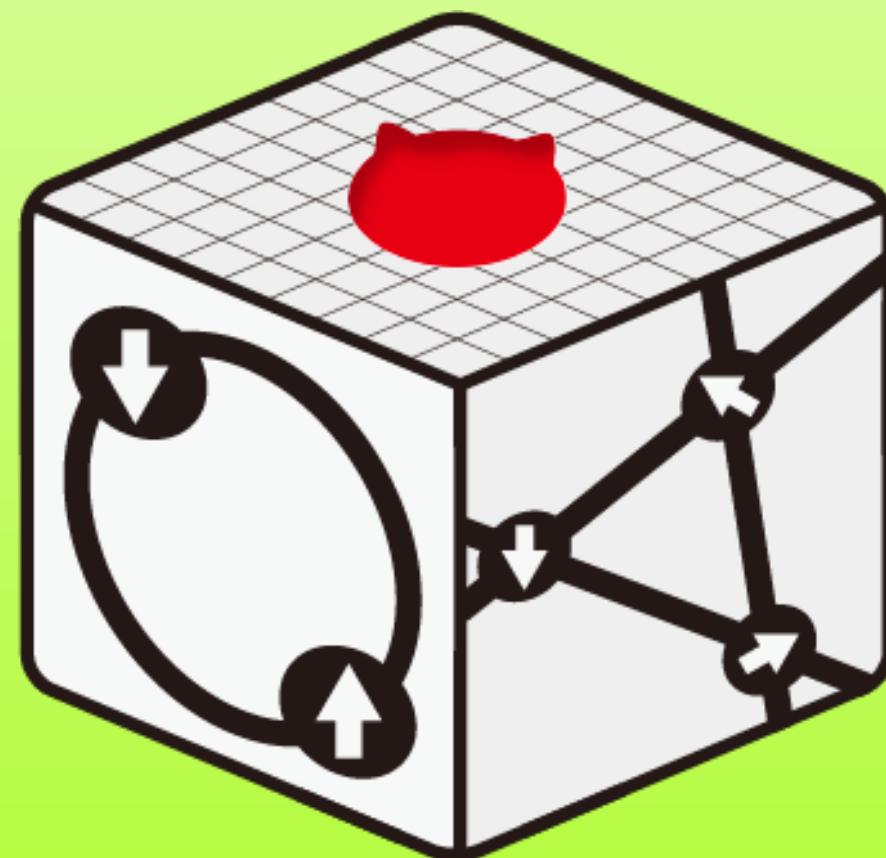
https://www.hpci-office.jp/for_users/appli_software/appli_mvmc

北海道大学 情報基盤センター (Grand Chariot), 東北大学 サイバーサイエンスセンター (AOBA)

東京大 情報基盤センター(Wisteria, Oakbridge-CX), 東工大 学術国際情報センター (TUBAME3.0)

名古屋大 情報基盤センター (不老 [Furou]), 大阪大 サイバーメディアセンター (OCTOPUS)

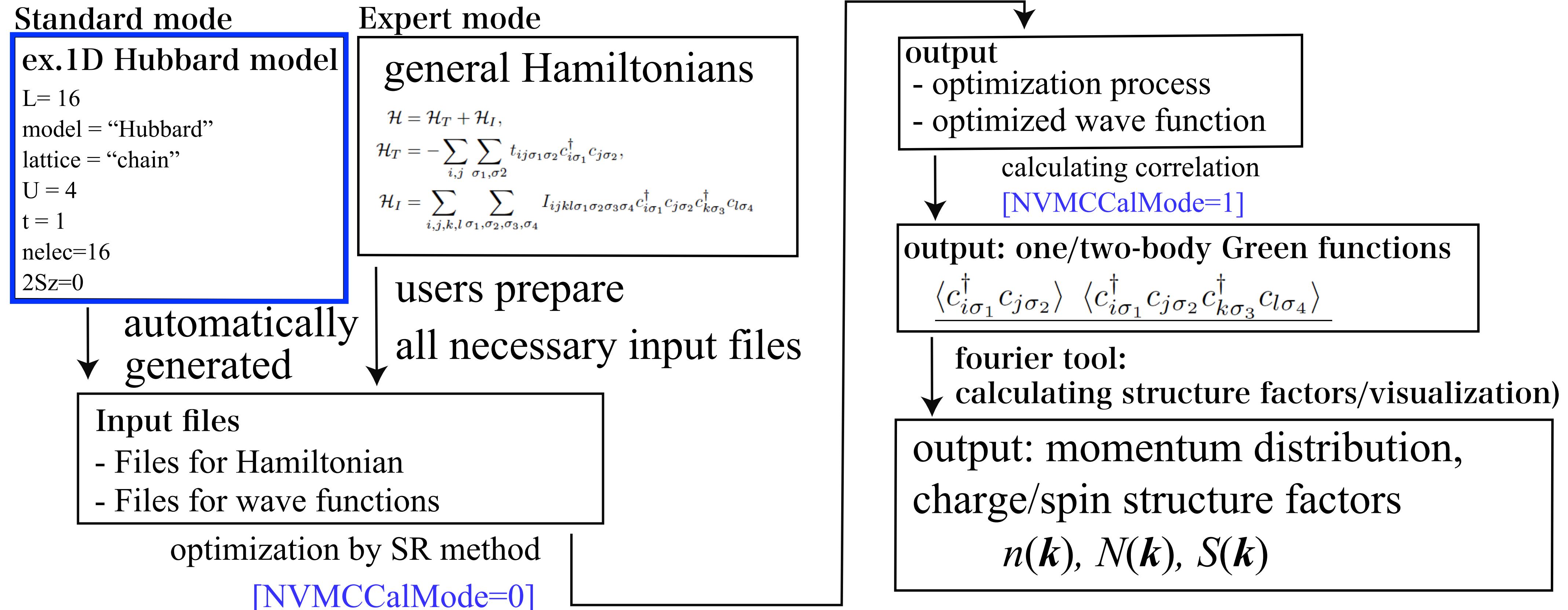
理研 計算科学研究センター(富岳 [Fugaku]), 九州大学 情報基盤研究開発センター (ITO)



mVMC

Let's start mVMC !

Flow of mVMC



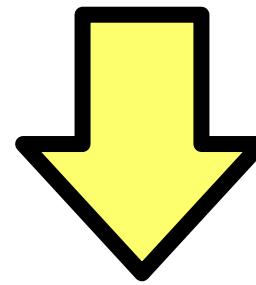
Standard mode

1. Preparing one input file → Optimization will be done automatically
2. Structure factors can be calculated from correlations functions

It is better to use *script languages* (python, perl, ruby) for preparing input files and calculation physical properties

How to use mVMC: What is Standard mode ?

vmcdry stan_opt.in



[Common in HΦ, mVMC]

Standard mode:

Automatically generating input files

Hamiltonians

coulombintra.def, trans.def, zlocspn.def ...

Green functions

greenone.def, greentwo.def

Specifying calculations conditions

modpara.def

[mVMC]

Specifying wave functions

orbitalidx.def, gutzwilleridx.def, jastrowidx.def...

+List of input files: namelist.def

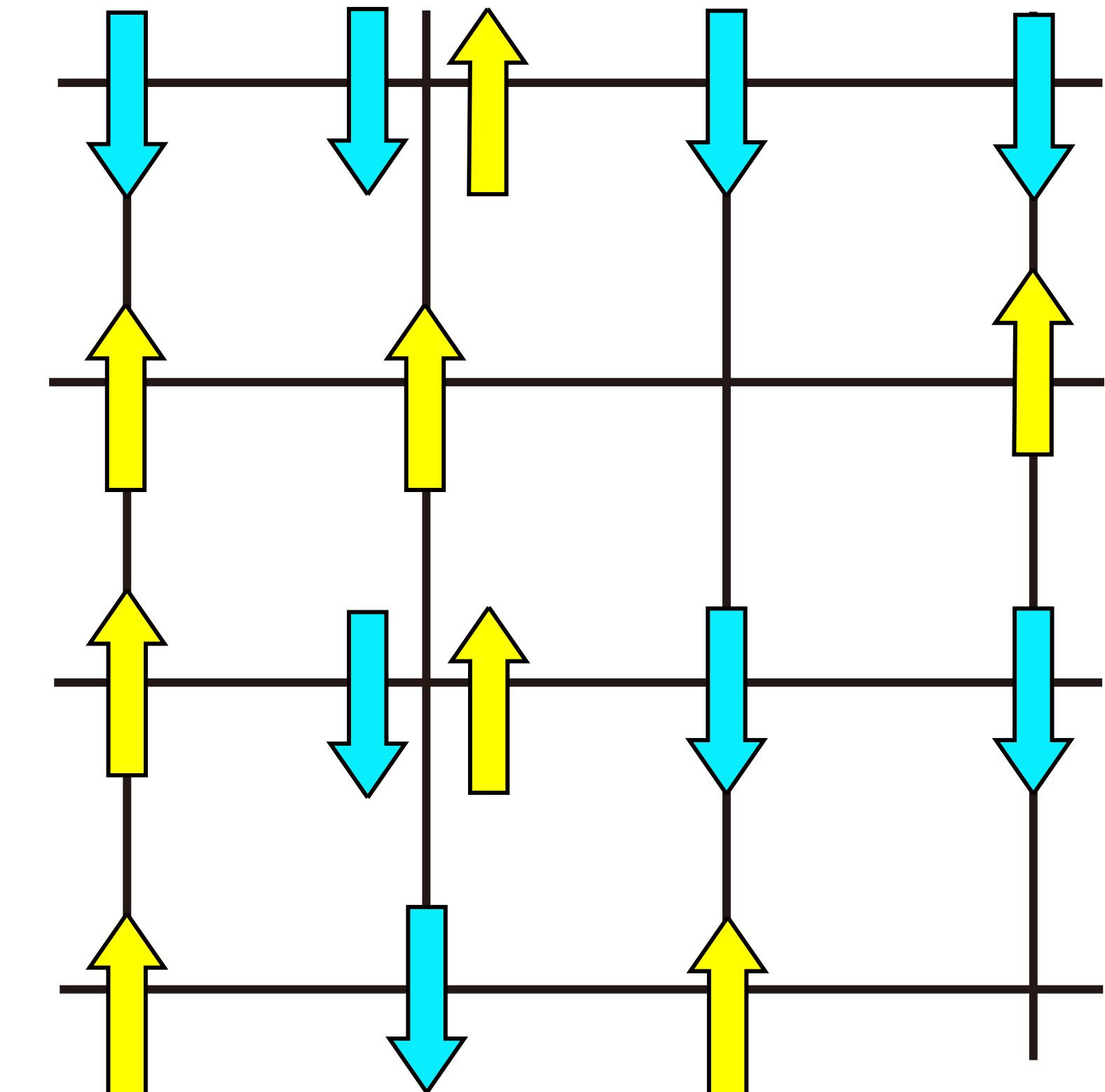
Expert mode: preparing input files manually

How to use mVMC: Standard mode

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

stan_opt.in

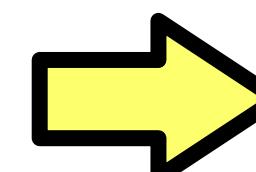
```
W = 4
L = 4
Wsub = 2
Lsub = 2
model = "FermionHubbard"
lattice = "Tetragonal"
t = 1.0
U = 4.0
nelec = 16
```



Simple input files for the 2D Hubbard model

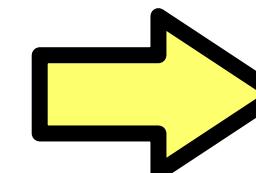
How to use mVMC: Standard mode II

vmcdry stan_opt.in

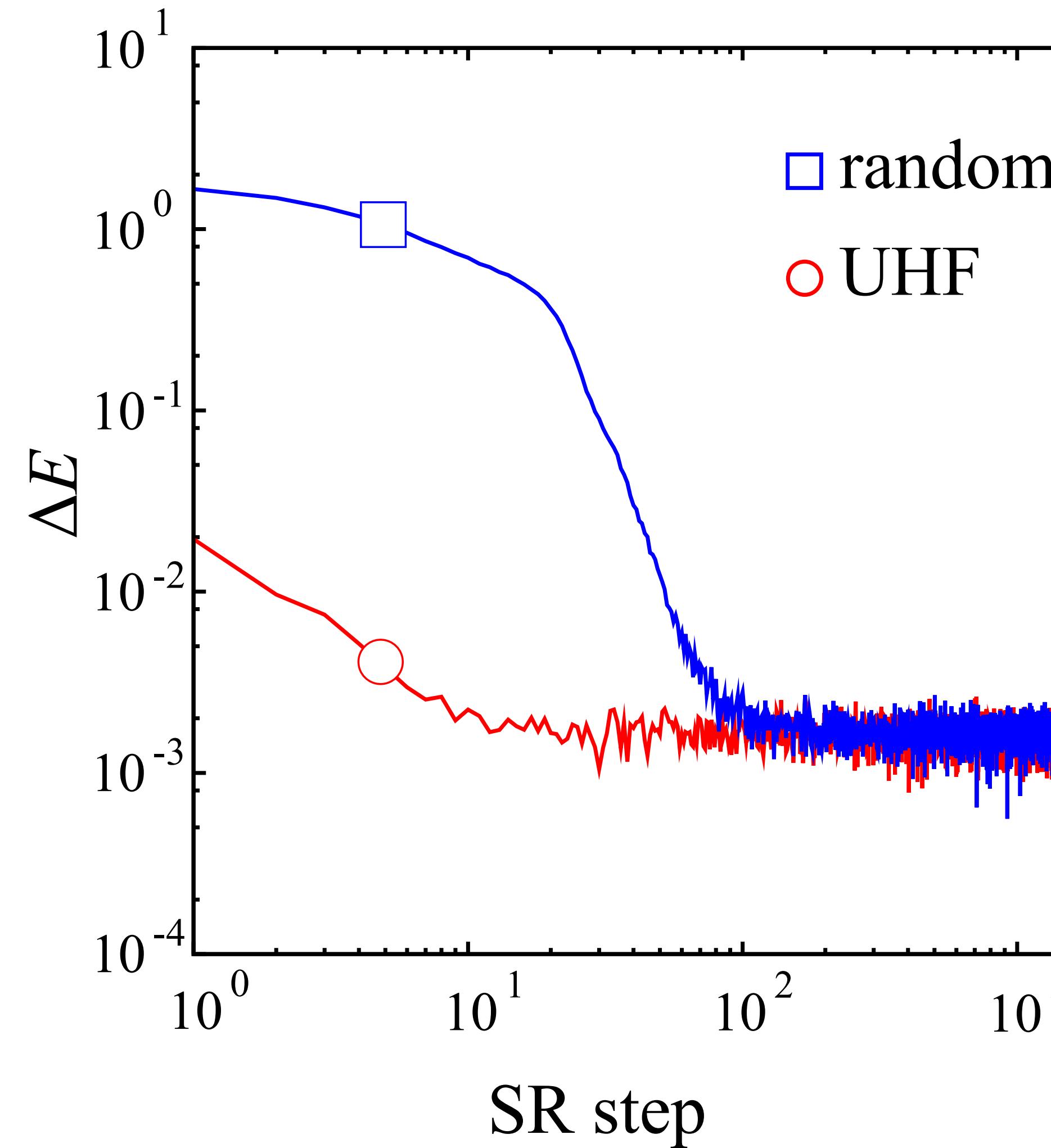


Generating input files

vmc namelist.def



Optimization



2D Hubbard model,
4×4, U/t=4, n=1

on laptop
~ 2-3 minutes

$$\Delta E = (E_{\text{mVMC}} - E_{\text{exact}})/E_{\text{exact}}$$

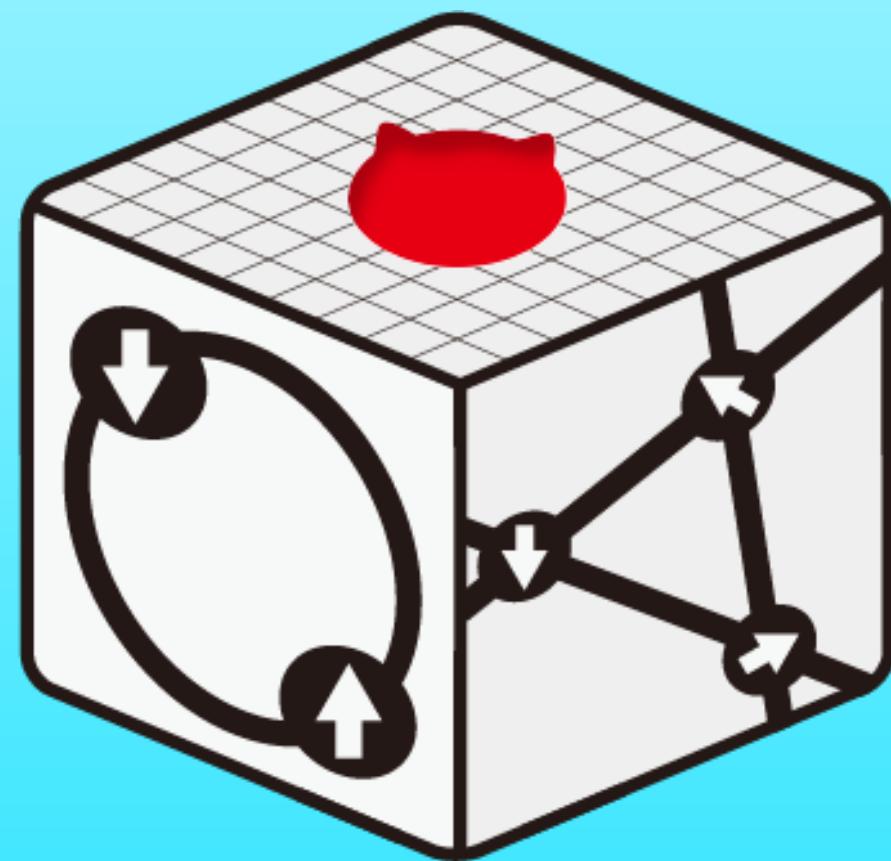
Hubbard model (Benchmark)

Physical Properties	mVMC(2×2)	ED
$4 \times 4(\text{PP}), n = 1$		
Energy per site	-0.8500(1)	-0.8513
$S(\mathbf{q}_{\text{peak}})/N_s$	0.0575(2)	0.0569
\mathbf{q}_{peak}	(π, π)	(π, π)
$\langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle$	-0.2063(14)	-0.2063
$4 \times 4(\text{PP}), n = 0.625$		
Energy per site	-1.2196(1)	-1.22380
$S(\mathbf{q}_{\text{peak}})/N_s$	0.0130(1)	0.01300
\mathbf{q}_{peak}	$(\pi/2, \pi)$	$(\pi/2, \pi)$
$\langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle$	-0.0704(5)	-0.0683



$$S(\mathbf{q}) = \frac{1}{3N_s} \sum_{i,j} \langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle e^{i\mathbf{q} \cdot (\mathbf{r}_i - \mathbf{r}_j)}$$

mVMC well reproduces results of exact diagonalization!
It is possible to calculate larger system sizes (100-1000 sites)



mVMC

Expert mode !

How to use mVMC: What is Expert mode ?

Expert mode: preparing input files by yourself

[Common in HΦ, mVMC]

Hamiltonians

coulombintra.def, trans.def, zlocspn.def ...

Green functions

greenone.def, greentwo.def

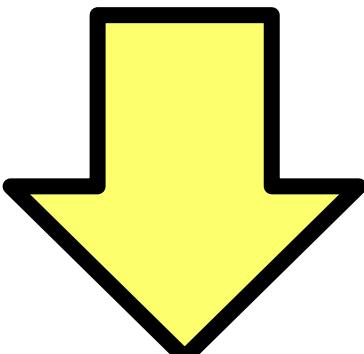
Specifying calculations conditions

modpara.def

[mVMC]

Specifying wave functions

orbitalidx.def, gutzwilleridx.def, jastrowidx.def...



vmc.out namelist.def

How to use mVMC: Expert mode

For standards interactions

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

=====
NCoulombintra 2
=====

=====Exchange=====

0 4.0
1 4.0

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

=====
NExchange 2
=====

=====Exchange=====

0 1 0.5
1 2 0.5

For details, see manuals

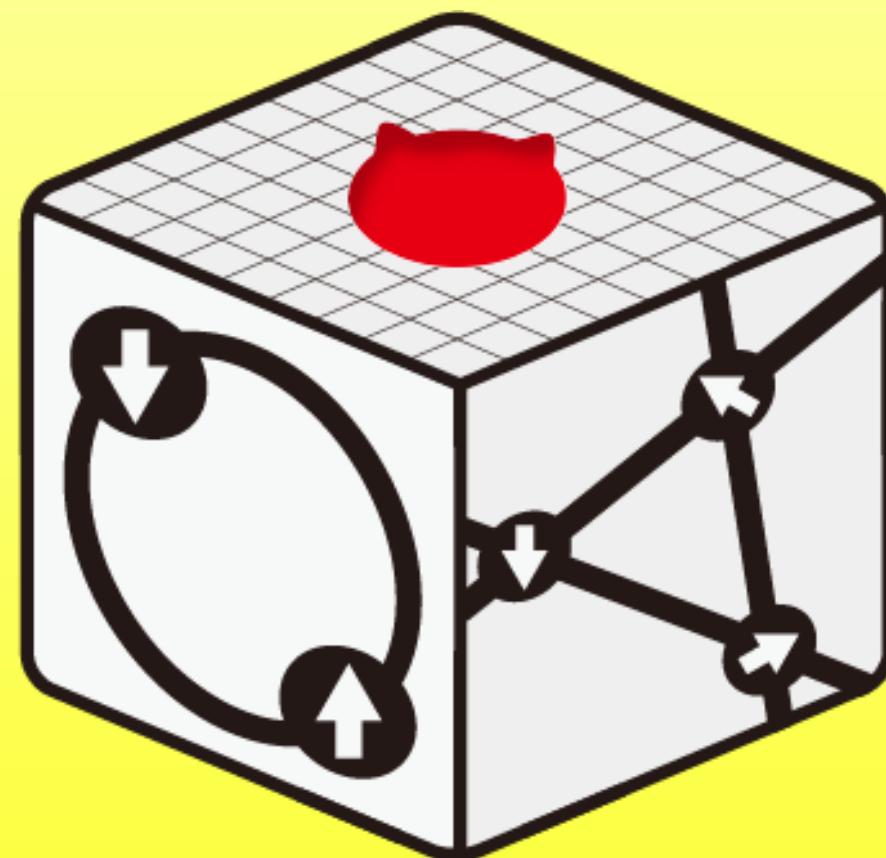
How to use mVMC: Interall.def

For general interactions

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

=====		# of interactions parameters								
NInterAll	96									
=====zInterAll=====									real part	imaginary part
0	0	0	0	1	0	1	0	0.500000	0.000000	
0	0	0	0	1	1	1	1	-0.500000	0.000000	
0	1	0	1	1	0	1	0	-0.500000	0.000000	
0	1	0	1	1	1	1	1	0.500000	0.000000	
0	0	0	1	1	1	1	0	1.000000	0.000000	
0	1	0	0	1	0	1	1	1.000000	0.000000	
...	<i>i</i>	σ_1	<i>j</i>	σ_2	<i>k</i>	σ_3	<i>l</i>	σ_4		

Arbitrary two-body interactions can be treated



mVMC

Tutorial!

0. Calculation flow

Step 1: Optimization [stan_opt.in]

Step 2: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]

Step 3: Calc. of correlations functions such as
spin/charge structure factors
from 1- and 2-body Green functions
[VMCcor.py]



Tutorial will be done in MateriAppsLive!
(様々なソフトがプレインストールされた仮想環境)

<http://cmsi.github.io/MateriAppsLive/>

1. Heisenberg chain

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

https://github.com/issp-center-dev/mVMC-tutorial/tree/master/HandsOn/2022_1128/2022_1128/Samples/1D_Heisenberg

Step 1: Optimization [stan_opt.in]

stan_opt.in

```
L          = 4
Lsub       = 2
lattice    = "chain"
model      = "Spin"
J          = 1.0
2Sz        = 0
NVMCSSample = 200
NSROptItrStep = 600
NMPTrans    = 1
NSPStot     = 0
```

Exercise: Yellow shaded boxes

```
cp -r 1D_Heisenberg L4_1D_Heisenberg
cd ./L4_1D_Heisenberg
```

```
vmcdry stan_opt.in
```

```
vmc ./namelist.def
```

```
gnuplot
plot ./output/zvo_out_001.dat u 1
```

1. Heisenberg chain

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

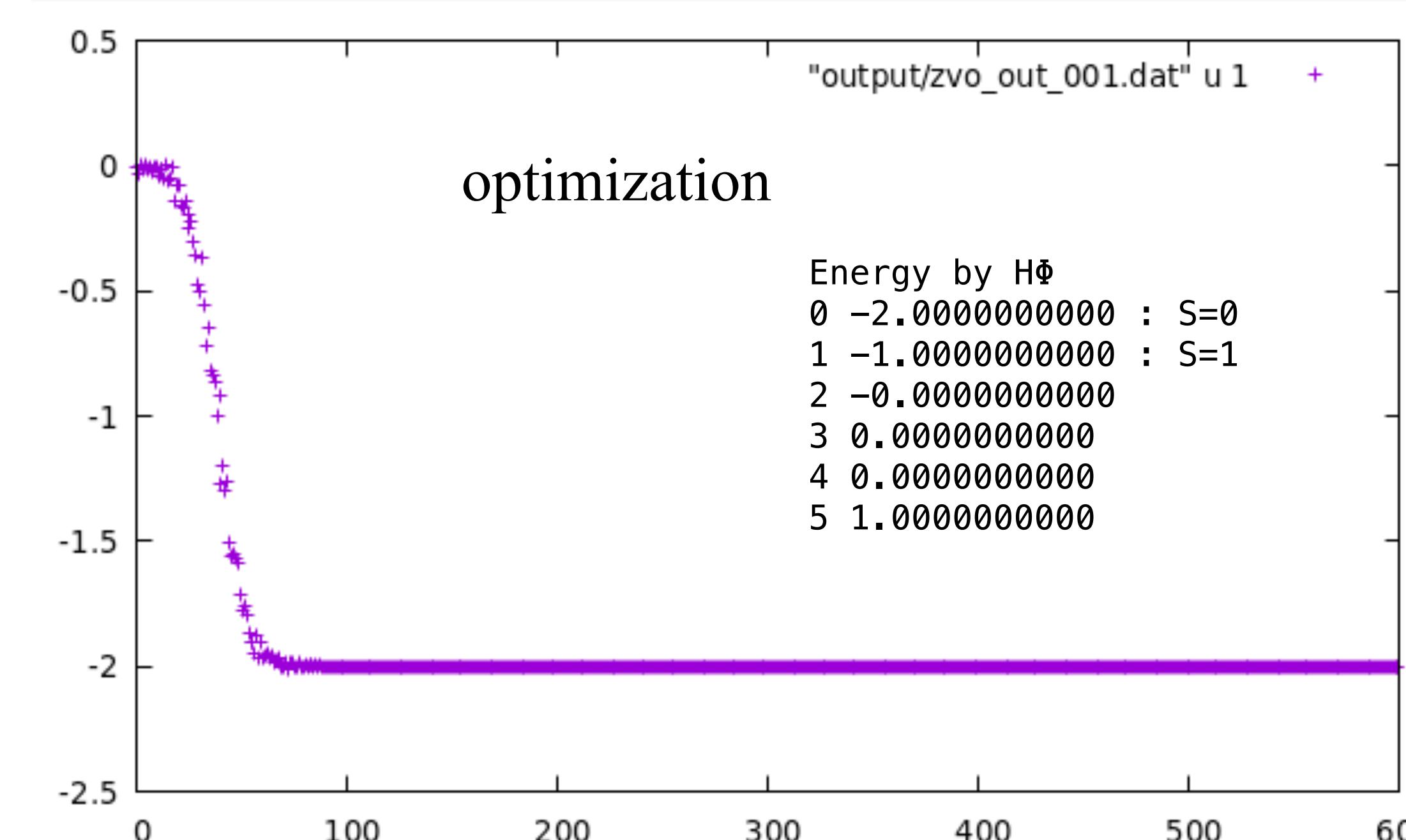
mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

Step 1: Optimization [stan_opt.in]

stan_opt.in

```
L          = 4
Lsub       = 2
lattice    = "chain"
model      = "Spin"
J          = 1.0
2Sz        = 0
NVMCsample = 200
NSROptItrStep = 600
NMPTrans   = 1
NSPStot    = 0
```

plot “./output/zvo_out_001.dat” u 1



1. Heisenberg chain [Exact diag by HΦ]

Sample script for HΦ

HPhi -s stan_hphi.in

```
L          = 4
model      = "Spin"
lattice    = "chain"
method     = "CG"
J          = 1.0
2Sz       = 0
exct      = 4
```

exct: # of excited states

For small, system sizes, you can do
full diagonalization

```
L          = 4
model      = "Spin"
lattice    = "chain"
method     = "fulldiag"
J          = 1.0
2Sz       = 0
```

1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

Step 1: Optimization [stan_opt.in]

output/zvo_out_001.dat

1	2	3	4	5	6
Re[<H>]	Im[<H>]	<H ² >	[<H ² >-<H> ²]/<H> ²	<S ^z >	<(S ^z) ² >

1st row: Energy = Re[<H>]

4th row: Variance = [<H²>-<H>²]/<H>²

By seeing energy, you can check the convergence of optimization.

Note that if the variational wave function becomes an eigenstate, variance=0. However, VMC is *not* exact method, variance becomes finite in most cases. Exceptions: small system sizes, non-interacting systems

1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

stan_aft.in

```
L          = 4
Lsub       = 2
lattice    = "chain"
model      = "Spin"
J          = 1.0
2Sz        = 0
NVMCSSample = 200
NSROptItrStep = 600
NMPTrans   = 1
NSPStot    = 0
NVMCCalMode = 1 # 1=calculation of physical quantities, 0=optimization
NDatadxStart = 0 # the first index
NDataQtySmp = 5 # number of bins
```

```
cp ./output/zqp_opt.dat .
mv output opt
```

```
vmcdry ./stan_aft.in
cp green1 greenone.def
cp green2 greentwo.def
vmc ./namelist.def ./zqp_opt.dat
```

NB: output is overwritten!

zqp_opt.dat = optimized wave functions !

1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

ls -l ./output

In output

zvo_out_000.dat - zvo_out_004.dat	[energy, variance ...]
zvo_cisajs_000.dat - zvo_cisajs_004.dat	[1-body Green functions]
zvo_cisajscktalt_000.dat - zvo_cisajscktalt_004.dat	[2-body Green functions]

From these files, you can calculate average values and statistical errors of physical quantities such as energy, charge/spin structure factors, etc...

1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

mv output aft

**python3 VMCloca.py input.toml
python3 VMCCor.py input.toml**

VMCloca.py → Calculation of energies and local charge/spin density

VMCCor.py → Calculation of spin/charge structure factors, and spin correlations

1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

cat Ene.dat

```
# Ene err_Ene Ene/(All_site) err_Ene/(All_site)
-2.000000 0.000000 -0.500000 0.000000
```

mVMC generates exact ground state for L=4

cat occ.dat

```
# occ err_occ AF err_AF
1.000000 0.000000 0.032000 0.038425
```

$$\text{occ} = \frac{1}{N_s} \sum_i (n_{i\uparrow} + n_{i\downarrow})$$

$$\text{AF} = \frac{1}{N_s} \sum_i (n_{i\uparrow} - n_{i\downarrow}) e^{i\pi r_i}$$

Averaged charge density $\text{occ}=1$. Antiferromagnetic spin moment is zero $\text{AF}=0.03(4)$ with in error bar.

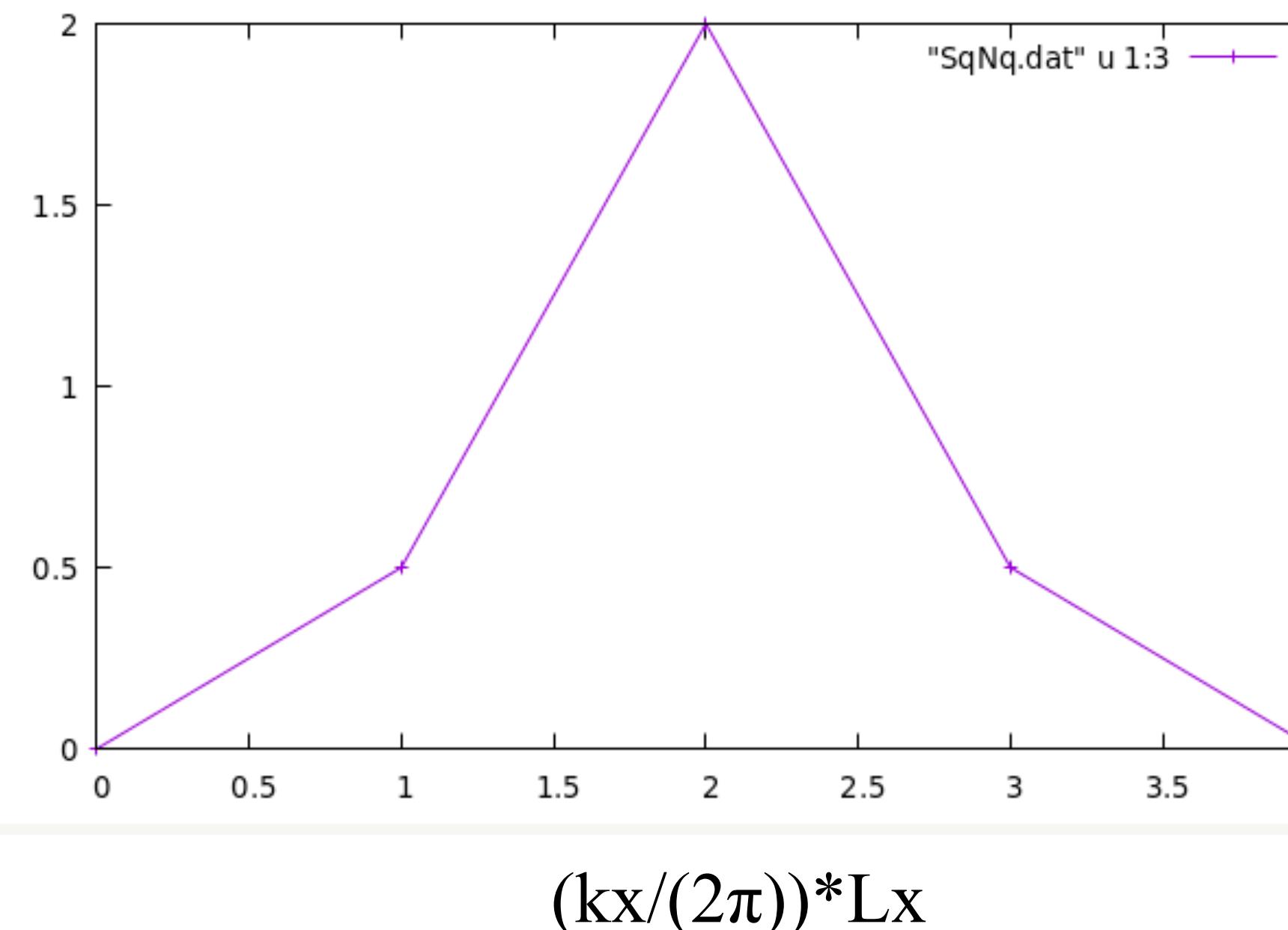
1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

```
gnuplot  
p "SqNq.dat" u 1:3 w lp
```



$$S(q) = \frac{1}{N_s} \sum_{i,j} \vec{S}_i \cdot \vec{S}_j e^{iqr_i}$$

NB: In some cases,
pre-factor of $S(q)$
is $1/(3N_s)$.

If you want to check error bars,

```
gnuplot  
p "SqNq.dat" u 1:3:4 w err
```

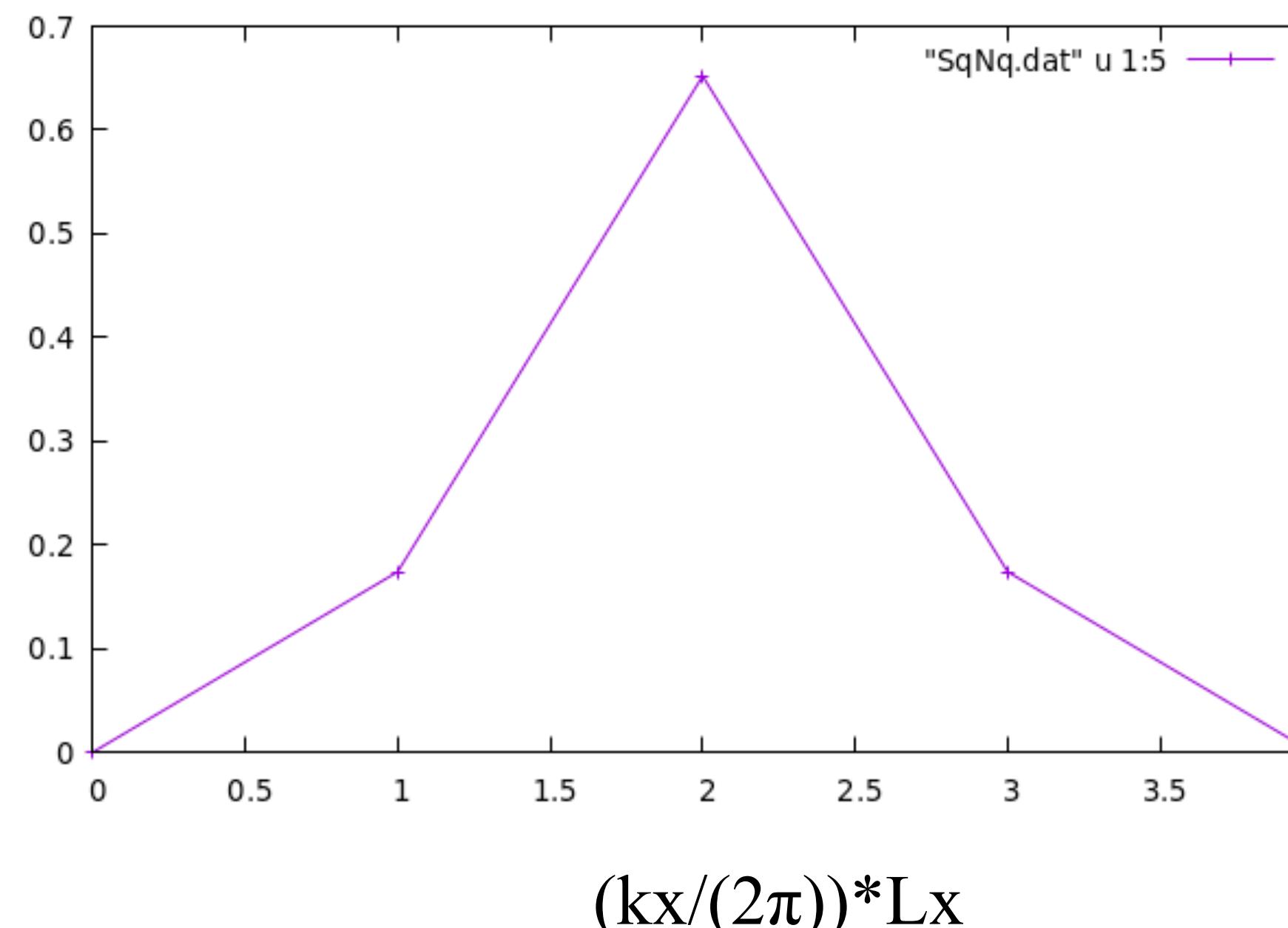
1. Heisenberg chain

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

mVMC-tutorial/HandsOn/2022_1128/Samples/1D_Heisenberg

**Step 1: Calc. of 1- and 2-body Green functions
for optimized wave functions [stan_aft.in]**

```
gnuplot  
p "SqNq.dat" u 1:5 w lp
```



$$S^z(q) = \frac{1}{N_s} \sum_{i,j} S_i^z \cdot S_j^z e^{iqr_i}, S_i^z = (n_{i\uparrow} - n_{i\downarrow})/2$$

NB: $S^z(q) = S(q)/3$ is satisfied if SU(2) symmetry is preserved.

If you want to check error bars,

```
gnuplot  
p "SqNq.dat" u 1:5:6 w err
```

1. Heisenberg chain

Let's change system sizes!

Edit “Lx” in input.toml

```
cp -r 1D_Heisenberg L8_1D_Heisenberg  
cd ./L8_1D_Heisenberg
```

```
python3 MakeInput.py input.toml  
vmcdry stan_opt.in  
vmc namelist.def  
cp output/zqp_opt.dat ./  
mv output opt  
  
vmcdry stan_aft.in  
cp green1 greenone.def  
cp green2 greenone.def  
vmcdry stan_aft.in  
vmc namelist.def ./zqp_opt.dat  
mv output aft
```

```
python3 VMCllocal.py input.toml  
python3 VMCCor.py input.toml
```

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

For L = 8 sites

[lattice]
Lx = 8
Ly = 1
Lz = 1
orb_num = 1
model_type = "Spin"
[mVMC]
sub_x = 2
sub_y = 1
sub_z = 1
[mVMC_aft]
modpara = "modpara.def"
directory = "aft"

1. Heisenberg chain

These energies are obtained by exact diagonalization ($H\Phi$)

$L=6:$

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 !

If you change

NSPStot=0 → NSPStot=1 in stan_opt.in & stan_aft.in

you can generate $S=1$ state.

Please try this! How about $S=2, S=3 \dots ?$

Scripts X.sh & Clean.sh

sh ./X.sh → Performing all calculations

sh ./Clean.sh → Delete all generated files (Initialization)

**After editing input.toml,
by executing “sh X.sh”,
you can do optimization,
calculation of Green functions,
and post process.**

```
#[s] definitions of executions
MPI=" "
VMC="vmc" #MAL
VMCDRY="vmcdry" #MAL
#[e] definitions of executions

python3 MakelInput.py input.toml

#[s] opt
${VMCDRY} ./stan_opt.in
${MPI} ${VMC} namelist.def
cp ./output/zqp_opt.dat .
mv output opt
#[e] opt

#[s] aft
${VMCDRY} ./stan_aft.in
cp green1 greenone.def
cp green2 greentwo.def
${MPI} ${VMC} namelist.def ./zqp_opt.dat
mv output aft
#[e] aft

#[s] post process
python3 VMClocaL.py input.toml
python3 VMCCor.py input.toml
#[e] post process
```

Other examples

Basic exercises [demonstration]

1. Heisenberg & Hubbard chain
2. Heisenberg & Hubbard (square lattice)
3. Using Hartree-Fock solutions as initial states

Advanced exercise

1. Hubbard + V → Charge order
2. Heisenberg+J2 → stripe magnetic order
3. Attractive Hubbard → superconductivity
4. 1D Kondo lattice → Kondo insulator
5. Data used in actual researches [Data repository in ISSP]

git clone <https://github.com/issp-center-dev/mVMC-tutorial.git>

Sample scripts: [mVMC-tutorial/HandsOn/2022_1128/Samples](https://github.com/issp-center-dev/mVMC-tutorial/HandsOn/2022_1128/Samples)

Summary

Basics of mVMC:

- Flexible wave functions (# of parameter $> 10^4$)
- *Time-dependent variational principle*
 - optimization of many variational parameters
 - finite-temperature calculations & real-time evolutions

How to use mVMC:

- Simple & Flexible user interfaces
- *Very easy* to study conventional models
- *Easy* to study general models

