HΦの概要

三澤　貴宏

東京大学物性研究所計算物質科学研究センター

http://www.pasums.issp.u-tokyo.ac.jp/hphi/

Outline

SCES = Strongly Correlated Electron Systems

1. Introduction

   Basics of models for SCES

   Why do we want to solve models for SCES ?

   How difficult to solve models for SCES ?

2. Algorithms used in HΦ

   - exact diagonalization (Lanczos, LOBCG)
   - finite-temperature calculations

3. How to use HΦ

Introduction

What is *models* for SCES ?

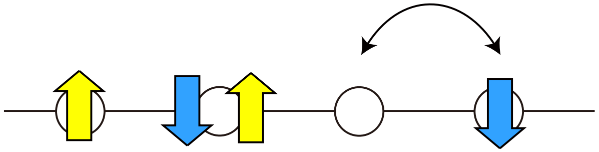
e.g. Hubbard model

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_t + \hat{\mathcal{H}}_U$$

Electrons as waves

$$\hat{H}_t = -t \sum_{\langle i,j\rangle,\sigma} (\hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma} + \hat{c}^\dagger_{j\sigma}\hat{c}_{i\sigma})$$

t: hopping



U: onsite Coulomb

Electrons as particles

$$\hat{H}_U = \sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow},\ \hat{n}_{i\sigma} = \hat{c}^\dagger_{i\sigma}\hat{c}_{i\sigma}$$

Relations between 2nd-quantized operators (these are all !)

$$\{\hat{c}^\dagger_{i\sigma}, \hat{c}_{j\sigma'}\} = \hat{c}^\dagger_{i\sigma}\hat{c}_{j\sigma'} + \hat{c}_{j\sigma'}\hat{c}^\dagger_{i\sigma} = \delta_{i,j}\delta_{\sigma,\sigma'}$$

$$\{\hat{c}^\dagger_{i\sigma}, \hat{c}^\dagger_{j\sigma'}\} = 0 \to \hat{c}^\dagger_{i\sigma}\hat{c}^\dagger_{i\sigma} = 0$$

Pauli's principle

$$\{\hat{c}_{i\sigma}, \hat{c}_{j\sigma'}\} = 0 \to \hat{c}_{i\sigma}\hat{c}_{i\sigma} = 0$$

Why do we want to solve ?

Rules (=models) are known, but results are unknown

Japanese chess (将棋)

We know how each piece moves,

but we don’t know which one wins.

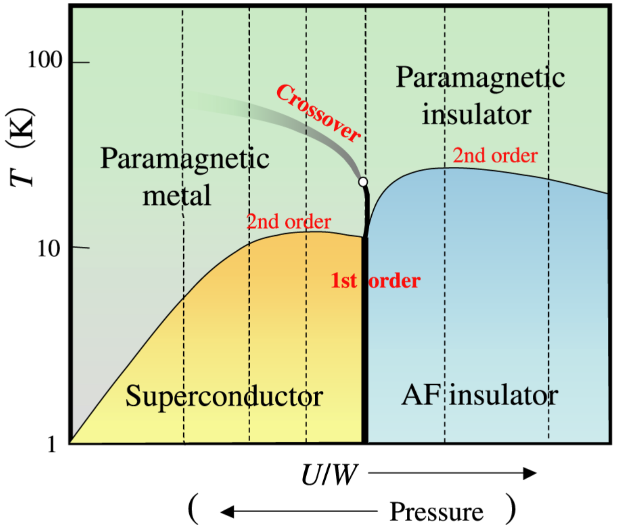
Solis state physics

We know how each electron moves,

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_t + \hat{\mathcal{H}}_U$$

$$\{\hat{c}^{\dagger}_{i\sigma}, \hat{c}_{j\sigma'}\} = \hat{c}^{\dagger}_{i\sigma}\hat{c}_{j\sigma'} + \hat{c}_{j\sigma'}\hat{c}^{\dagger}_{i\sigma} = \delta_{i,j}\delta_{\sigma,\sigma'}$$
$$\{\hat{c}^{\dagger}_{i\sigma}, \hat{c}^{\dagger}_{j\sigma'}\} = 0 \rightarrow \hat{c}^{\dagger}_{i\sigma}\hat{c}^{\dagger}_{i\sigma} = 0$$
$$\{\hat{c}_{i\sigma}, \hat{c}_{j\sigma'}\} = 0 \rightarrow \hat{c}_{i\sigma}\hat{c}_{i\sigma} = 0$$

but we don’t know what happens.



K. Kanoda, JPSJ 75, 051007 (2006)

Wave function = eigenvectors of Hamiltonian

Matrix representation of Hamiltonian (ex. 2 site Hubbard model)

Real-space configuration $|\uparrow,\downarrow\rangle = c^\dagger_{1\uparrow}c^\dagger_{2\uparrow}|0\rangle$

After some *tedious* calculations, $\langle\uparrow,\downarrow|\hat{H}_t|\uparrow\downarrow,0\rangle = \langle\uparrow,\downarrow|(t\sum_\sigma c^\dagger_{1\sigma}c_{2\sigma} + c^\dagger_{2\sigma}c_{1\sigma})|\uparrow\downarrow,0\rangle = -t$

$$\mathcal{H} = \begin{array}{c|cccc} & |\uparrow,\downarrow\rangle & |\downarrow,\uparrow\rangle & |\uparrow\downarrow,0\rangle & |0,\uparrow\downarrow\rangle \\ \hline \langle\uparrow,\downarrow| & 0 & 0 & -t & -t \\ \langle\downarrow,\uparrow| & 0 & 0 & t & t \\ \langle\uparrow\downarrow,0| & -t & t & U & 0 \\ \langle 0,\uparrow\downarrow| & -t & t & 0 & U \end{array}$$

$$|\phi\rangle = a_0|\uparrow,\downarrow\rangle + a_1|\downarrow,\uparrow\rangle + a_2|\uparrow\downarrow,0\rangle + a_3|0,\uparrow\downarrow\rangle$$

Diagonalization → eigenvalues, eigenvectors
→ Problem is completely solved (HΦ)

How difficult to solve models ?

“nightmare” for theorists

- Exact diagonalization ~ $4^{Ns}\times4^{Ns}$
→ Only for small sizes [HΦ]

- Unbiased Quantum Monte Carlo method [DSQSS]
→ Negative sign problem

- Many methods have been developed
DMFT (∞ dimensions → 2,3 dimensions) [DCore]
DMRG (1 dimension → 2 dimensions)
tensor-network (1,2 dimensions) [TeNES]
VMC (wide applicable range, but ..) [mVMC]

No all-round method exists

For small systems, using exact diagonalizations, we can obtain *exact* results.

Introduction to HΦ

Developers of HΦ (2015-)

Y. Yamaji

S. Todo

N. Kawashima

K. Ido

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

Basics of HΦ

What can we do by HΦ?

For Hubbard model, spin-S Heisenberg model, Kondo-lattice model …

- **Full diagonalization [Scalapack & GPGPU]**
- **Ground state calculations by Lanczos method**
- **Finite-temperature calculations by thermal pure quantum (TPQ) states**

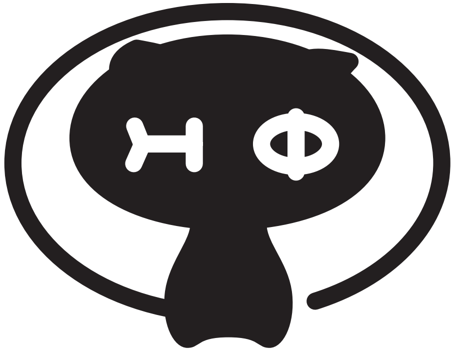
-Low-energy states calculation by LOBCG method

-Dynamical properties [$S(q,\omega)$], real-time evo.

maximum system sizes@ ISSP system B (sekirei)

- spin 1/2: ~ 40 sites (Sz conserved)
- Hubbard model: ~ 20sites (# of particles & Sz conserved)

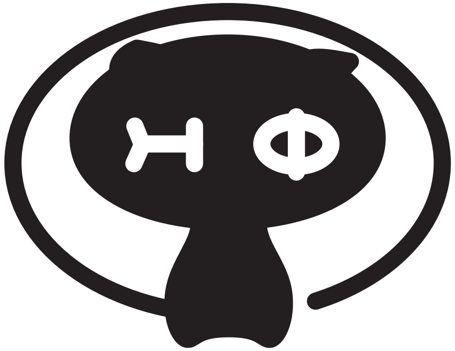
Meaning of name & logo



- Multiplying H to Φ (HΦ)
- The cat means wave function in two ways

cat is a symbol of superposition.. (Schrödinger’s cat)

Meaning of name & logo



- Multiplying H to Φ (HΦ)
- The cat means wave function in two ways

cat is a symbol of superposition.. (Schrödinger’s cat)

$\Phi =$

wake

$+$

sleep

Algorithms used in HΦ !

Full diagonalization

Matrix representation of Hamiltonian

(real space basis) → Full diagonalization for the matrix

$$H_{ij} = \langle i|\hat{H}|j\rangle$$

$|i\rangle$ **real-space basis**

ex. 2-site Hubbard $\mathcal{H} = \begin{pmatrix} 0 & 0 & -t & -t \\ 0 & 0 & t & t \\ -t & t & U & 0 \\ -t & t & 0 & U \end{pmatrix}$

dim. of matrix=

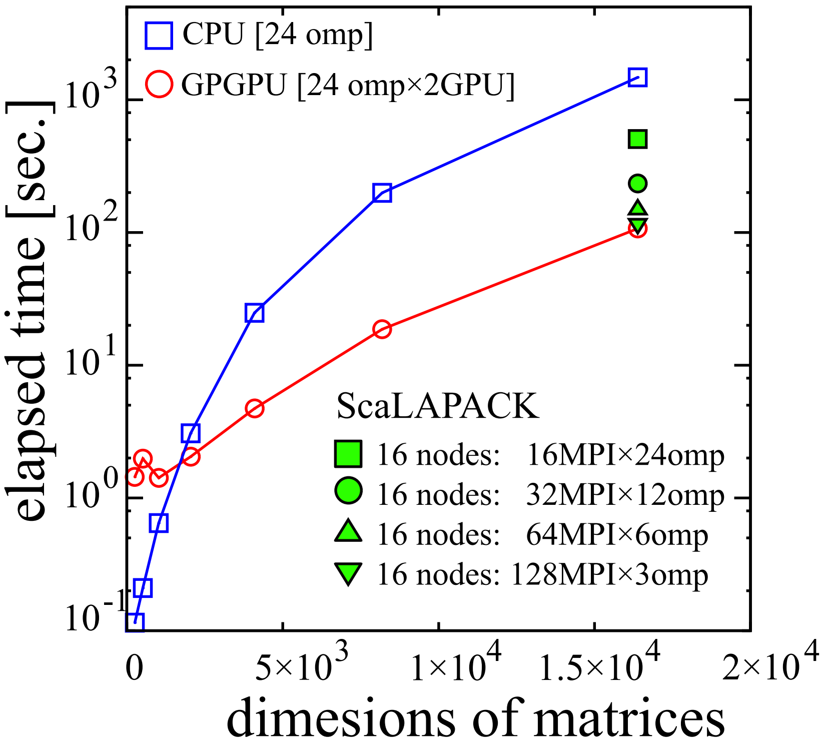
# of real-space bases = exponentially large

ex. spin1/2 system: S_z=0 $\quad {}_{N_s}C_{N_s/2}$

- Ns=16: dim.=12800, required memory ~ 1 GB
- Ns=32: dim.~6×10^8 required memory ~ 3 E B !

Full diagonalization

In HΦ, full diagonalization can be done using LAPACK, SCALAPACK, MAGMA(GPGPU)



Input matrix can specified by MATRIX MARKET format

```
%%%% matrix coordinate complex hermitian
4 4 6
1 1 4.000000 0.000000
2 1 1.000000 0.000000
3 1 -1.000000 0.000000
4 2 1.000000 0.000000
4 3 -1.000000 0.000000
4 4 4.000000 0.000000
```

A visual repository of test data for use in comparative studies of algorithms for numerical linear algebra, featuring nearly 500 sparse matrices from a variety of applications, as well as matrix generation tools and services.

Lanczos method

By multiplying the Hamiltonian to initial vector, we can obtain the ground state (power method)

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

A few (at least two) vectors are necessary→

We can treat larger system size than full diagonalization

ex. spin 1/2 sysmte: **S_z=0**

- Ns=16: dim. =12800,required memory (vectors) ~0.1 MB
- Ns=32: dim. ~6×10^8,required memory (vectors) ~5 GB !
- Ns=36: dim. ~9×10^9,required memory (vectors) ~72 GB !

LOBCG method

Lanczos method + conjugate gradient (CG) method
[Conventional exact diagonalization (ex. TITPACK)]

Lanczos[eigenvalues,2 vectors] → Lanczos[eigenvectors,3 vectors]
→ Inverted iteration [5 vectors]

[merit] Only two vectors are necessary
[demerit] Redundant, *Failure* of orthogonality, degeneracy, etc…

Locally Optimal (Preconditioning) Block Conjugate [LO(P)BCG]

A. V. Knyazev, SIAM J. Sci. Compute. 23, 517 (2001)
B. 山田進 他, 日本計算工学会論文集, 20060027 (2006)

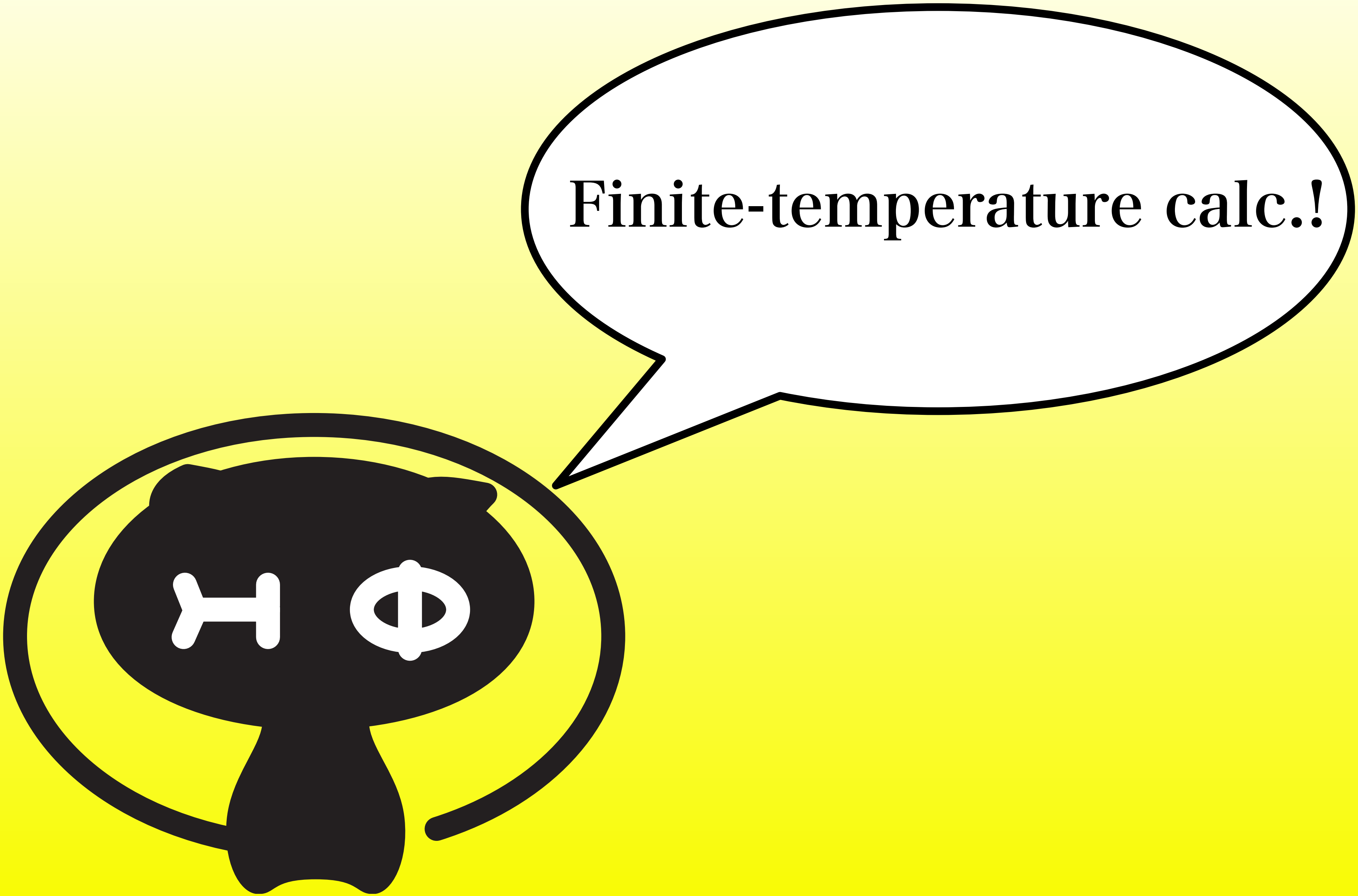
$$\text{Eigenvectors } \{|\Phi_0\rangle, |\Phi_1\rangle, \cdots, |\Phi_{M-1}\rangle\}$$

$$\text{residual vectors } \{|r_i\rangle = H|\Phi_i\rangle - \epsilon_i|\Phi_i\rangle\}$$

$$\text{conjugate gradient vectors } \{|p_0\rangle, |p_1\rangle, \cdots, |p_{M-1}\rangle\}$$

Diagonalize & Optimize with 3M vectors

[merit]
- Obtaining many low-energy states at once
- Degeneracy & orthogonality can be well reproduced
[demerit] Required memory is slightly larger (~3M-6M vectors)


HΦ
Finite-temperature calc.!

Finite-temperature calculations

- Is ensemble average necessary for finite-temperature calculations ? No !

$$\langle E\rangle = \frac{\sum_n E_n e^{-\beta E_n}}{\sum_n e^{-\beta E_n}}$$

- Is it possible to replace ensemble average by the average values of wave function ? (Partially)Yes !

$$\langle E(\beta)\rangle = \frac{\langle\Phi(\beta)|H|\Phi(\beta)\rangle}{\langle\Phi(\beta)|\Phi(\beta)\rangle} \; ? \qquad |\Phi(\beta)\rangle = \sum_i a_i(\beta)|i\rangle$$

Exact: Double Hilbert space (二重ヒルベルト空間,熱場ダイナミクス)

高橋康, 物性研究 20, 97 (1973); 鈴木増雄, 統計力学(岩波書店)

Yasushi TAKAHASHI and Hiroomi UMEZAWA, *Int. J. Mod. Phys. B* 10, 1755 (1996).
Reprinted with permission from Collective Phenomena 1975, Vol. 2 pp. 55–80

Stochastic: Quantum Transfer Monte Carlo (Imada-Takahashi 1986)

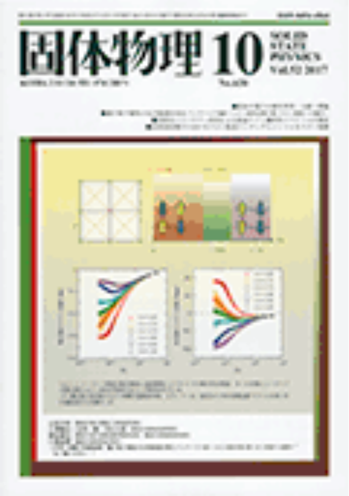
Finite-temperature Lanczos (Jaklic-Prelovsek 1994)

Hams-De Raedt method (2000)

Thermal Pure Quantum [TPQ] state (Sugiura-Shimizu 2012,2013)

Essence of TPQ

cf. 固体物理, vol. 52, No. 10
「量子格子模型の汎用数値対角化パッケージHΦ
—スピン液体近傍の熱・スピン励起への適用—」


固体物理 10

1. Random vector (high-temperature limit)
***equally* includes all eigenvectors**

$$|\Phi_{\rm rand}\rangle = \sum_n a_n|n\rangle, |a_n|^2 \sim 1/N_{\rm H}$$

$$\langle \hat{H}\rangle = \frac{\langle\Phi_{\rm rand}|\hat{H}|\Phi_{\rm rand}\rangle}{\langle\Phi_{\rm rand}|\Phi_{\rm rand}\rangle} = \frac{\sum_n |a_n|^2 E_n}{\sum_n |a_n|^2} \sim \frac{1}{N_{\rm H}}\sum_n E_n$$

2. Commutative quantities can be calculated by *single wave function*

$$|\Phi(\beta)\rangle = e^{-\beta\hat{H}/2}|\Phi_{\rm rand}\rangle$$

$$\frac{\langle\Phi(\beta)|\hat{H}|\Phi(\beta)\rangle}{\langle\Phi(\beta)|\Phi(\beta)\rangle} = \frac{\sum_n |a_n|^2 E_n e^{-\beta E_n}}{\sum_n |a_n|^2 e^{-\beta E_n}} \sim \frac{\sum_n E_n e^{-\beta E_n}}{\sum_n e^{-\beta E_n}} = E(\beta)$$

3. Non-commutative quantities can be also calculated by *single wave function*

Proofs: Hams and De Raedt PRE 2000; Sugiura and Shimizu PRL 2012,2013
Thermal Pure Quantum state (熱的純粋量子状態) by Sugiura and Shimizu

Sugiura-Shimizu method [TPQ method]

Procedure

Simple implementation

S. Sugiura and A. Shimizu, PRL 2012

$$|\psi_0\rangle : \text{random vector}$$

$$|\psi_k\rangle \equiv \frac{(l-\hat{H}/N_{\rm s})|\psi_{k-1}\rangle}{|(l-\hat{H}/N_{\rm s})|\psi_{k-1}\rangle|}$$

***l*:constant larger the maximum eigenvalues**

$$u_k \sim \langle\psi_k|\hat{H}|\psi_k\rangle/N_{\rm s}$$

$$\beta_k \sim \frac{2k/N_{\rm s}}{(l-u_k)}, \quad \langle\hat{A}\rangle_{\beta_k} \sim \langle\psi_k|\hat{A}|\psi_k\rangle$$

All the finite temperature properties can be calculated by using *one* thermal pure quantum [TPQ] state.

Errors of estimation are bounded by *S*
→ Powerful tools for studying frustrated systems !
Error of estimation ~ exp[-*S*(*T)/2] (*T*/2<*T**<*T*)**

Drastic reduction of numerical cost

Heisenberg model, 32 sites, S_z=0

Full diagonalization:

Dimension of Hamiltonian ~ $10^8 \times 10^8$

Memory ~ 3E Byte → Almost impossible.

TPQ method:

Only two vectors are required:

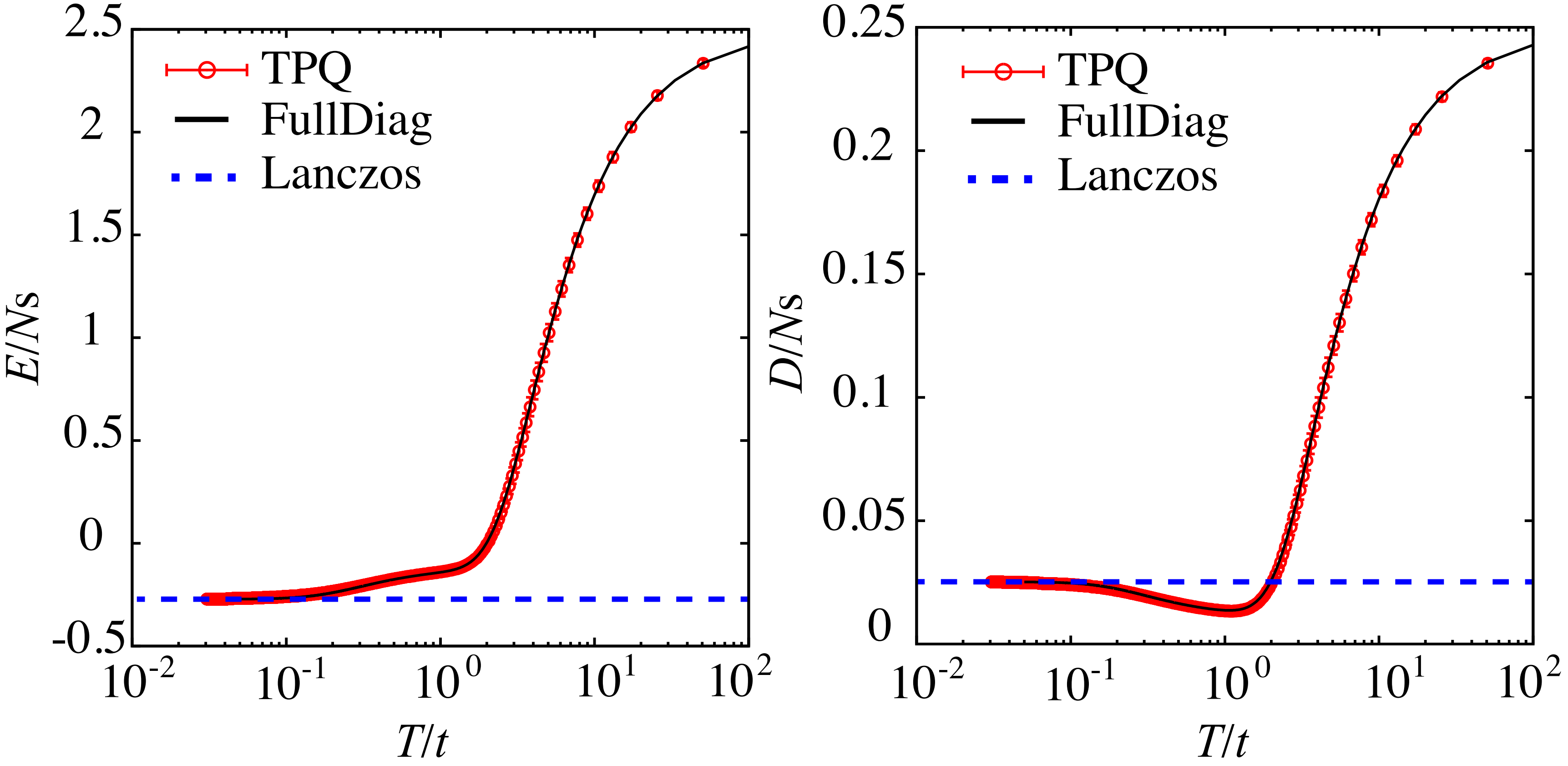
dimension of vector ~ 10^8~~$\times 10^8$~~

Memory ~ 10 G Byte

→ Possible even in lab's cluster machine !

Validity of TPQ method

Comparison of FullDiag, TPQ, Lanczos method
Hubbard model, *L*=8, *U/t*=8, half filling, S_z=0



TPQ method works well !

Let’s start HΦ !
(Standard mode)

How to use HΦ: Standard mode I (Lanczos)

Only StdFace.def is necessary (< 10 lines) !

ex. 4×4 2d Heisenberg model,
GS by Lanczos method

```
L          = 4
model    = “Spin”
method  = “Lanczos”
lattice    = “square lattice”
J            = 1.0
2Sz         = 0
```

⇩

```
HPhi -s StdFace.def
```

./ouput : results are output

Method

Lanczos — ground state

CG - LOBCG

TPQ — finite-temperature

FullDiag — full-diagonalization

Important files

./output/zvo_energy.dat	→ energy
./output/zvo_Lanczos_Step.dat	→ convergence
./output/zvo_cisajs.dat	→ one-body Green func.
./output/zvo_cisajscktalt.dat	→ two-body Green func.

How to use HΦ: Standard mode II

ex. 4by4, 2d Heisenberg model,
GS calculations by Lanczos

./output/zvo_energy.dat

```
$ cat output/zvo_energy.dat
Energy  -11.2284832084288109
Doublon  0.000000000000000
Sz       0.000000000000000
```

GS energy

./output/zvo_Lanczos_Step.dat

```
$ tail output/zvo_Lanczos_Step.dat
stp=28 -11.2284832084 -9.5176841765 -8.7981539671 -8.5328120558
stp=30 -11.2284832084 -9.5176875029 -8.8254961060 -8.7872255591
stp=32 -11.2284832084 -9.5176879460 -8.8776934418 -8.7939798590
stp=34 -11.2284832084 -9.5176879812 -8.8852955092 -8.7943260103
stp=36 -11.2284832084 -9.5176879838 -8.8863380562 -8.7943736678
stp=38 -11.2284832084 -9.5176879839 -8.8864307327 -8.7943782609
stp=40 -11.2284832084 -9.5176879839 -8.8864405361 -8.7943787937
stp=42 -11.2284832084 -9.5176879839 -8.8864422628 -8.7943788984
stp=44 -11.2284832084 -9.5176879839 -8.8864424018 -8.7943789077
stp=46 -11.2284832084 -9.5176879839 -8.8864424075 -8.7943789081
```

convergence process by Lanczos method

How to use HΦ: Standard mode III

./output/zvo_cisajs.dat

$\langle c^{\dagger}_{i\sigma} c_{j\tau} \rangle$

```
$ head output/zvo_cisajs.dat
    0    0    0    0 0.5000000000 0.0000000000

    0    1    0    1 0.5000000000 0.0000000000
```

$\langle c^{\dagger}_{0\downarrow} c_{0\downarrow} \rangle$

$\langle c^{\dagger}_{0\uparrow} c_{0\uparrow} \rangle$

./output/zvo_cisajscktalt.dat

```
$ head output/zvo_cisajscktalt.dat
    0    0    0    0    0    0    0    0 0.5000000000 0.0000000000
    0    0    0    0    0    1    0    1 0.0000000000 0.0000000000
    0    0    0    0    1    0    1    0 0.1330366332 0.0000000000
    0    0    0    0    1    1    1    1 0.3669633668 0.0000000000
```

$\langle c^{\dagger}_{0\downarrow} c_{0\downarrow} c^{\dagger}_{0\downarrow} c_{0\downarrow} \rangle$

$\langle c^{\dagger}_{0\downarrow} c_{0\downarrow} c^{\dagger}_{0\uparrow} c_{0\uparrow} \rangle$

$\langle c^{\dagger}_{0\downarrow} c_{0\downarrow} c^{\dagger}_{1\downarrow} c_{1\downarrow} \rangle$

$\langle c^{\dagger}_{0\downarrow} c_{0\downarrow} c^{\dagger}_{1\uparrow} c_{1\uparrow} \rangle$

ex. onsite・nn-site correlation func.

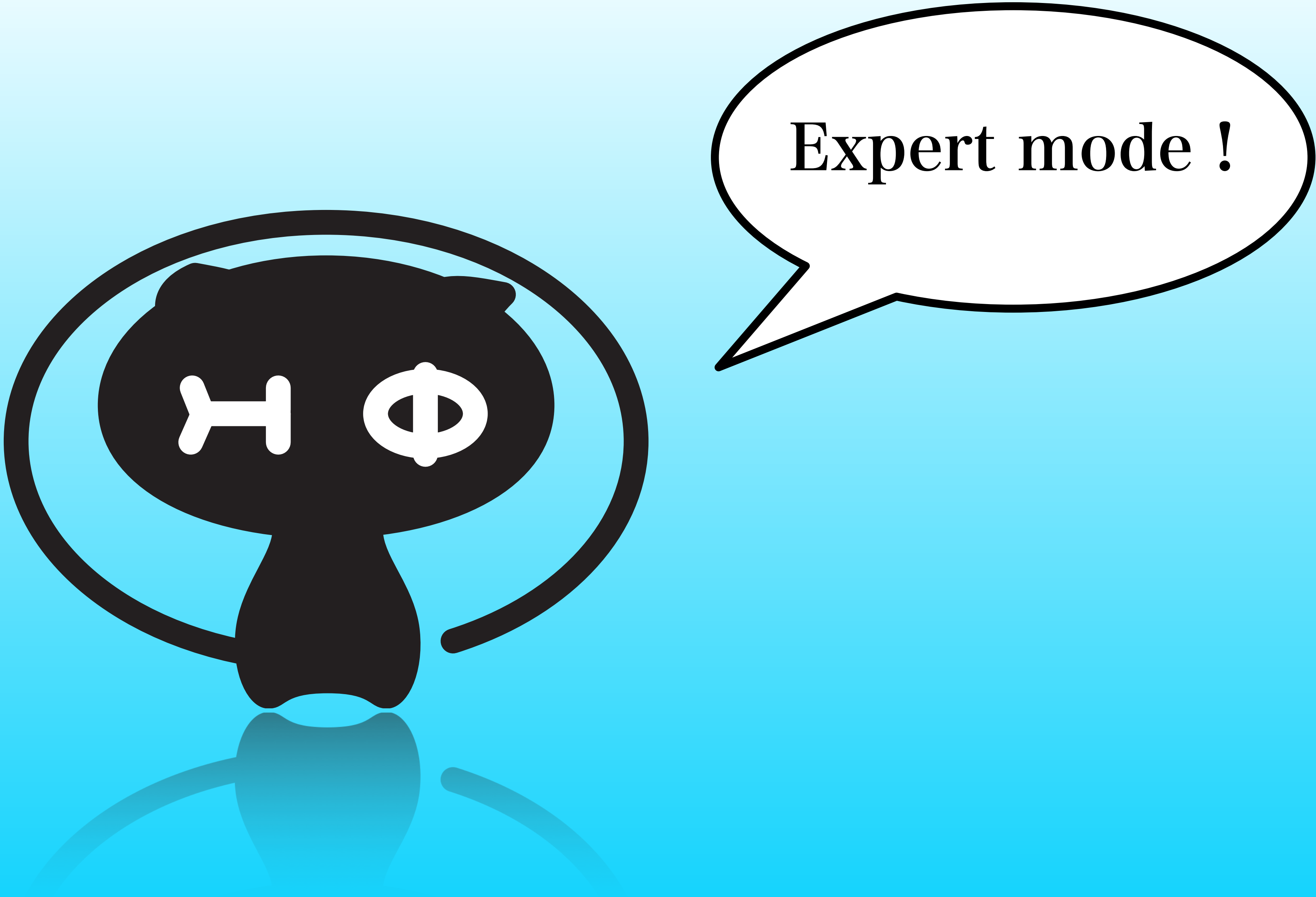
How to use HΦ: Standard mode IV

HPhi/samples/Standard/
StdFace.def for
Hubbard model, Heisenberg model, Kitaev model,
Kondo-lattice model

By changing StdFace.def slightly, you can easily perform the calculations for different models.

Cautions：

- Do not input too large system size
(upper limit@laptop: spin 1/2→24 sites, Hubbard model 12 sites)
- Lanczos method is unstable for too small size
(dim. > 1000)
-TPQ method does no work well for small size
(dim. > 1000)


Expert mode !

How to use HΦ: What is Expert mode ?

```
HPhi -s StdFace.def
```

Standard mode: Necessary input files are automatically generated

Files for Hamiltonian
CoulombIntra.def,Trans.def, locspn.def…

Files for basic parameters (two files)
modpara.def,calcmod.def

Files for correlations functions (two files)
greenone.def, greentwo.def

+ list of input files: namelist.def

Expert mode: preparing the following files by yourself

How to use HΦ: What is Expert mode ?

Expert mode: preparing the following files by yourself

Files for Hamiltonian
CoulombIntra.def,Trans.def, locspn.def…

Files for basic parameters (two files)
modpara.def,calcmod.def

Files for correlations functions (two files)
greenone.def, greentwo.def

execute following command

```
HPhi -e namelist.def
```

Enjoy HΦ
HΦ

Questions ?