HΦの概要
-プログラムとアルゴリズム-
Overview of HΦ: Program & Algorithm

山地 洋平
東京大学大学院工学系物理工学専攻
Youhei Yamaji
Department of Applied Physics, The University of Tokyo

1. Introduction to HΦ
2. Algorithm: Lanczos & LOBCG
3. Algorithm: TPQ

Appendix:
Formulation of many-body quantum systems in a nutshell

Computational Science Alliance
The University of Tokyo

HΦ

For direct comparison between experiments and theory and promoting development of other numerical solvers

Numerical diagonalization package for lattice hamiltonian

-For wide range of quantum lattice hamiltonians

Ab initio effective hamiltonians

-Lanczos method [1] and LOB(P)CG [2]:

Ground state and low-lying excited states

Excitation spectra of ground state

-Thermal pure quantum (TPQ) state [2]: Finite temperatures

-Real-time evolution

-Parallelization with MPI and OpenMP (→並列化性能の紹介)

[1] E. Dagotto, Rev. Mod. Phys. 66, 763 (1994) .
[2] A. V. Knyazev, SIAM J. Sci. Cumput. 23, 517 (2001).
[3] S. Sugiura, A. Shimizu, Phys. Rev. Lett. 108, 240401 (2012).

Open source program package (latest release: ver.3.1.2)

License: GNU GPL version3

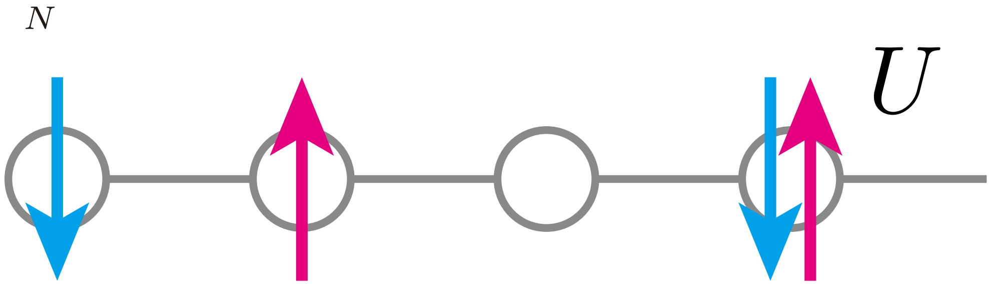
Project for advancement of software usability in materials science" by ISSP

Target Hamiltonian

- Standard Hamiltonian 1

Itinerant electrons: Hubbard-type model

$$H = -\mu \sum_{i=1}^{N} \sum_{\sigma=\uparrow,\downarrow} c_{i\sigma}^{\dagger} c_{i\sigma} - \sum_{i\neq j} \sum_{\sigma=\uparrow,\downarrow} t_{ij} c_{i\sigma}^{\dagger} c_{j\sigma} + U \sum_{i=1}^{N} n_{i\uparrow} n_{i\downarrow} + \sum_{i\neq j} V_{ij} n_i n_j$$



`Fermion Hubbard:` Particle # & total S_z conserved

`HubbardNConserved:` Particle # conserved & total S_z not

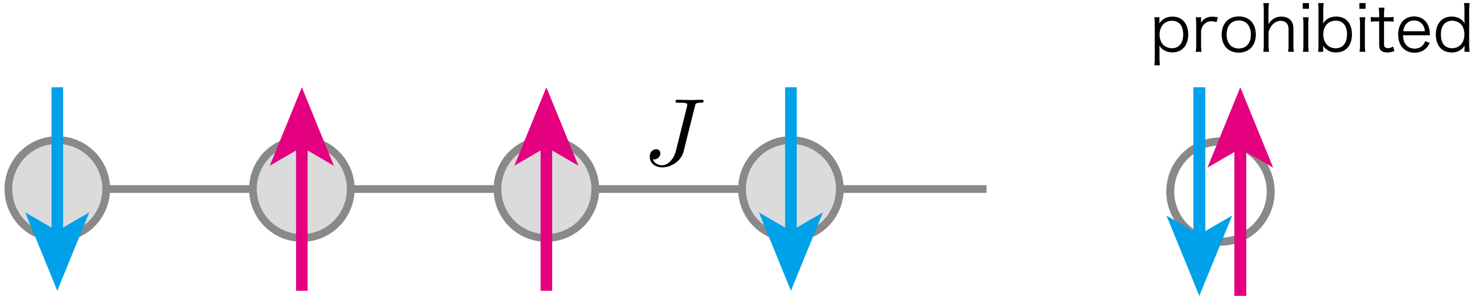
`Fermion HubbardGC:` Particle # & total S_z not conserved

Target Hamiltonian

- Standard Hamiltonian 2

Localized spin: Heisenberg-type model

$$H = -h\sum_{i=1}^{N} S_i^z + \Gamma\sum_i S_i^x + D\sum_i S_i^z S_i^z + \sum_{i,j}\sum_{\alpha,\beta=x,y,z} J_{ij}^{\alpha\beta} S_i^{\alpha} S_j^{\beta}$$



Spin: total S_z conserved

SpinGC: total S_z not conserved

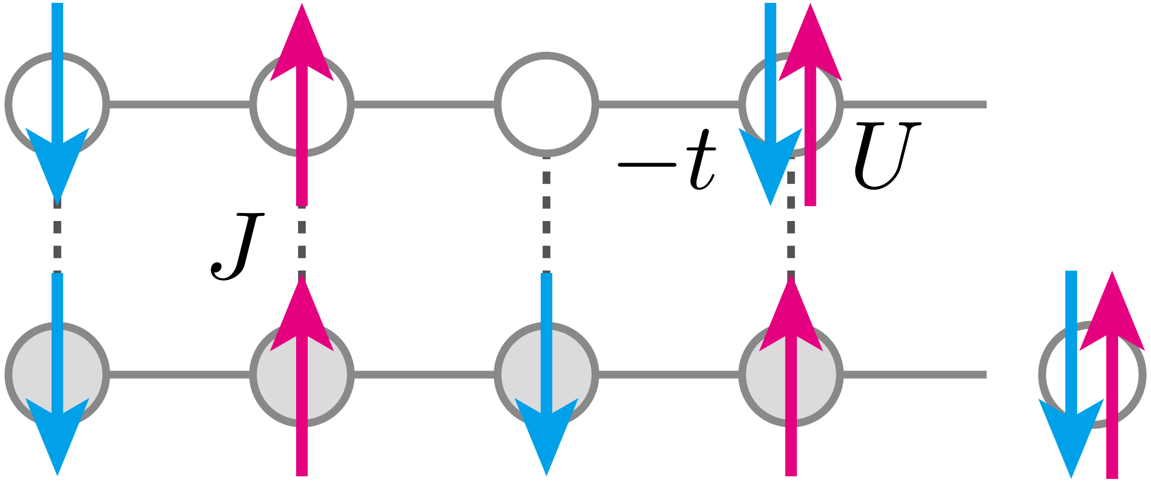
S > 1/2 can be simulated
if your memory is enough large

Target Hamiltonian

- Standard Hamiltonian 3

Mixture: Kondo-lattice-type model

$$H=-\mu\sum_{i=1}^{N}\sum_{\sigma=\uparrow,\downarrow}c_{i\sigma}^{\dagger}c_{i\sigma}-t\sum_{\langle i,j\rangle}\sum_{\sigma=\uparrow,\downarrow}c_{i\sigma}^{\dagger}c_{j\sigma}+\frac{J}{2}\sum_{i=1}^{N}\left\{S_i^{+}c_{i\downarrow}^{\dagger}c_{i\uparrow}+S_i^{-}c_{i\uparrow}^{\dagger}c_{i\downarrow}+S_i^{z}(n_{i\uparrow}-n_{i\downarrow})\right\}$$



prohibited

`Kondo Lattice:` Particle # & total S_z conserved

`Kondo LatticeGC:` Particle # & total S_z not conserved

Standard input: Simplified input for typical lattice models

Hubbard

$$H=-\mu\sum_{i=1}^{N}\sum_{\sigma=\uparrow,\downarrow}c_{i\sigma}^{\dagger}c_{i\sigma}-\sum_{i\neq j}\sum_{\sigma=\uparrow,\downarrow}t_{ij}c_{i\sigma}^{\dagger}c_{j\sigma}+U\sum_{i=1}^{N}n_{i\uparrow}n_{i\downarrow}+\sum_{i\neq j}V_{ij}n_{i}n_{j}$$

Quantum spins

$$H=-h\sum_{i=1}^{N}S_{i}^{z}+\Gamma\sum_{i}S_{i}^{x}+D\sum_{i}S_{i}^{z}S_{i}^{z}+\sum_{i,j}\sum_{\alpha,\beta=x,y,z}J_{ij}^{\alpha\beta}S_{i}^{\alpha}S_{j}^{\beta}$$

Kondo lattice

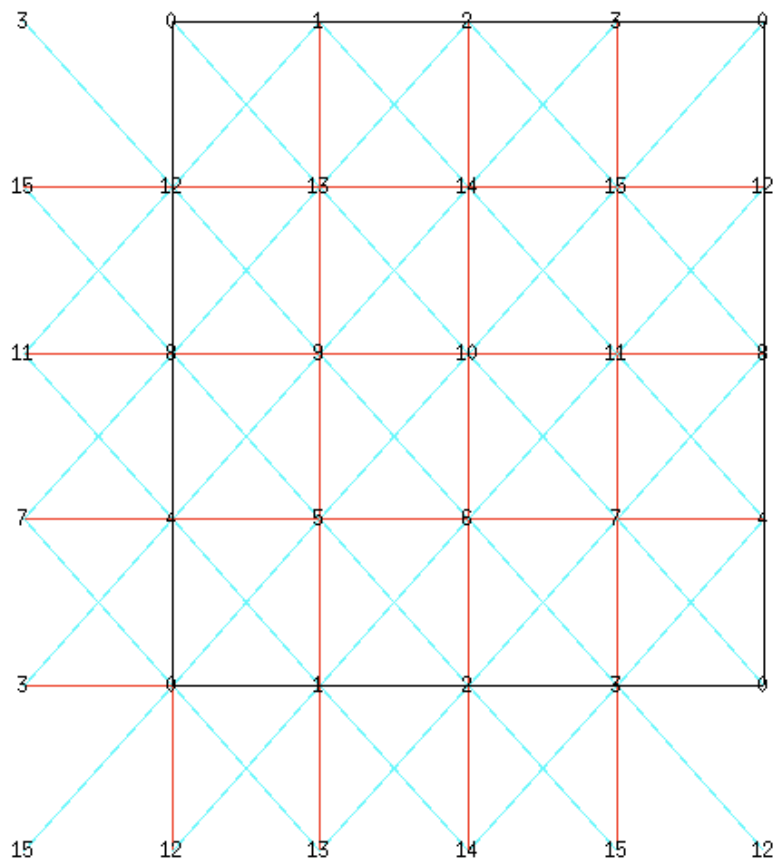
$$H=-\mu\sum_{i=1}^{N}\sum_{\sigma=\uparrow,\downarrow}c_{i\sigma}^{\dagger}c_{i\sigma}-t\sum_{\langle i,j\rangle}\sum_{\sigma=\uparrow,\downarrow}c_{i\sigma}^{\dagger}c_{j\sigma}+\frac{J}{2}\sum_{i=1}^{N}\left\{S_{i}^{+}c_{i\downarrow}^{\dagger}c_{i\uparrow}+S_{i}^{-}c_{i\uparrow}^{\dagger}c_{i\downarrow}+S_{i}^{z}(n_{i\uparrow}-n_{i\downarrow})\right\}$$

Expert input: Flexible input for any one- and two-body hamiltonian

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

Primitive Standard Input File

```
W = 4
L = 4
model = "Hubbard"
//method = "Lanczos”
method = "TPQ"
//method = "FullDiag"
lattice = "Square"
t = 1.0
t' = 0.5
U = 8.0
nelec = 16
2Sz = 0
```



Output

Ground-state/finite-temperature/time-evolution of

-Energy

-Square of energy

-One-body equal time Green’s function

-Two-body equal time Green’s/correlation function

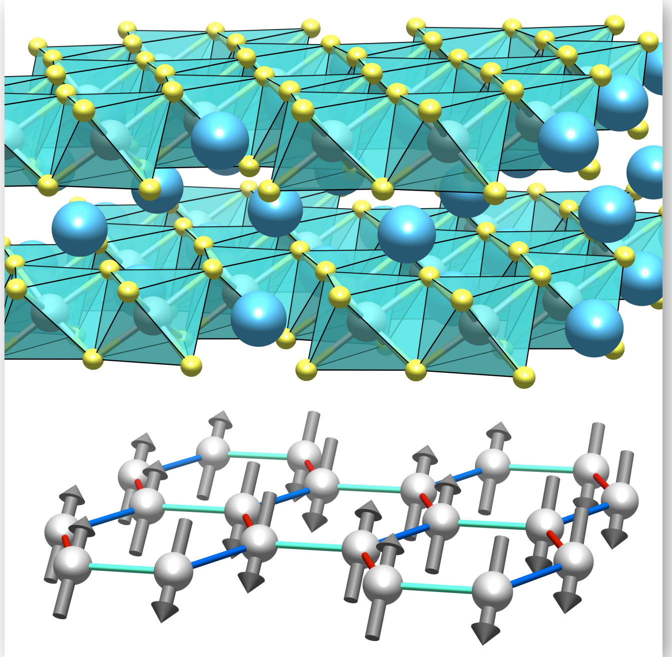
$$\langle H\rangle,\ \langle H^2\rangle,\ \langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2}\rangle,\ \langle c^{\dagger}_{i\sigma_1} c_{j\sigma_2} c^{\dagger}_{k\sigma_3} c_{\ell\sigma_4}\rangle$$

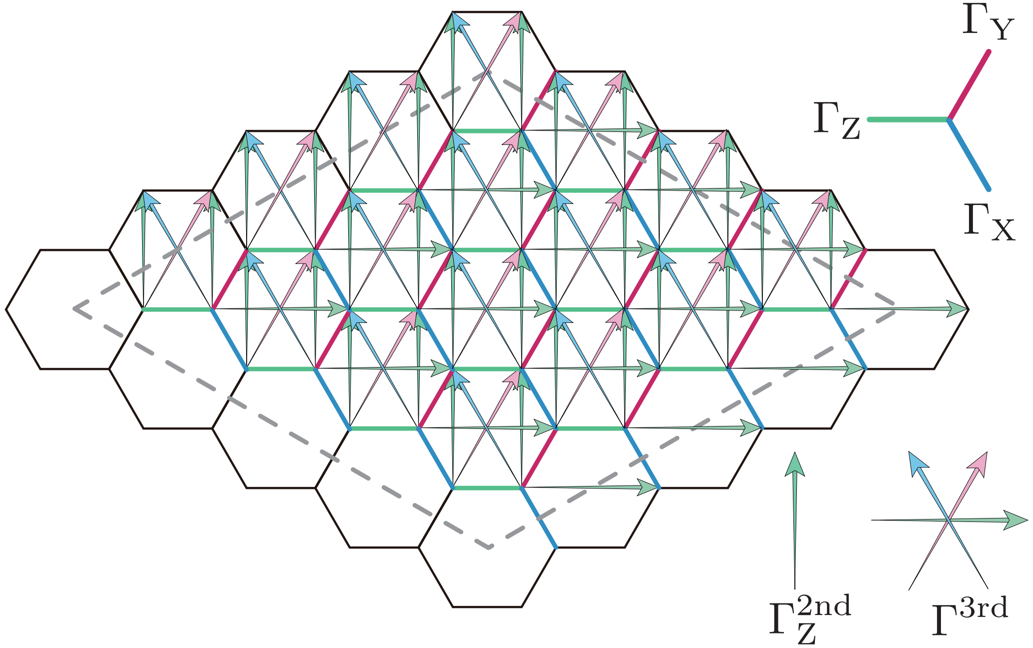
-Dynamical Green’s function is also available

An Example of Expert Input: *Ab Initio* Spin Hamiltonian

Y. Yamaji, Y. Nomura, M. Kurita, R. Arita, & M. Imada, Phys. Rev. Lett. 113, 107201 (2014).

An example: Frustrated magnet Na_2IrO_3





An Example of Expert Input: *Ab Initio* Spin Hamiltonian

Y. Yamaji, Y. Nomura, M. Kurita, R. Arita, & M. Imada, Phys. Rev. Lett. 113, 107201 (2014).

$$\hat{H} = \sum_{\Gamma = X,Y,Z,Z_{\mathrm{2nd}},3} \sum_{\langle \ell, m \rangle \in \Gamma} \vec{\hat{S}}_\ell^T \mathcal{J}_\Gamma \vec{\hat{S}}_m$$

$$\vec{\hat{S}}_\ell^T = (\hat{S}_\ell^x, \hat{S}_\ell^y, \hat{S}_\ell^z)$$

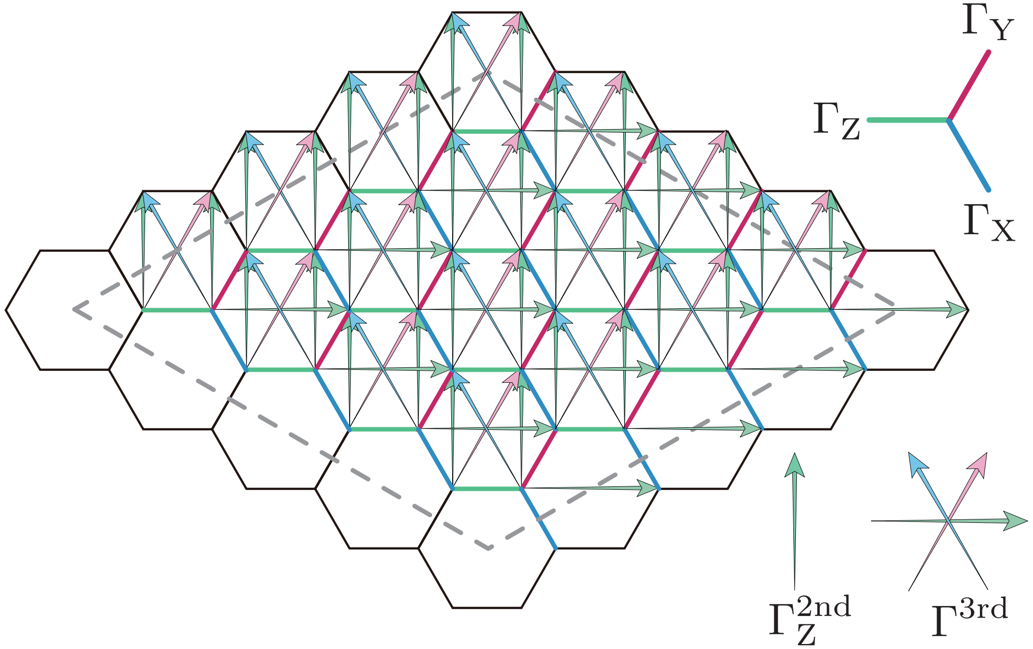
$$\mathcal{J}_X = \begin{bmatrix} \mathbf{-23.9} & -3.1 & -8.4 \\ -3.1 & 3.2 & 1.8 \\ -8.4 & 1.8 & 2.0 \end{bmatrix} (\mathrm{meV})$$

$$\mathcal{J}_Y = \begin{bmatrix} 3.2 & -3.1 & 1.8 \\ -3.1 & \mathbf{-23.9} & -8.4 \\ 1.8 & -8.4 & 2.0 \end{bmatrix} (\mathrm{meV})$$

$$\mathcal{J}_Z = \begin{bmatrix} 4.4 & -0.4 & 1.1 \\ -0.4 & 4.4 & 1.1 \\ 1.1 & 1.1 & \mathbf{-30.7} \end{bmatrix} (\mathrm{meV})$$

$$\mathcal{J}_{Z_{\mathrm{2nd}}} = \begin{bmatrix} -0.8 & 1.0 & -1.4 \\ 1.0 & -0.8 & -1.4 \\ -1.4 & -1.4 & -1.2 \end{bmatrix} (\mathrm{meV})$$

$$\mathcal{J}_3 = \begin{bmatrix} 1.7 & 0.0 & 0.0 \\ 0.0 & 1.7 & 0.0 \\ 0.0 & 0.0 & 1.7 \end{bmatrix} (\mathrm{meV})$$



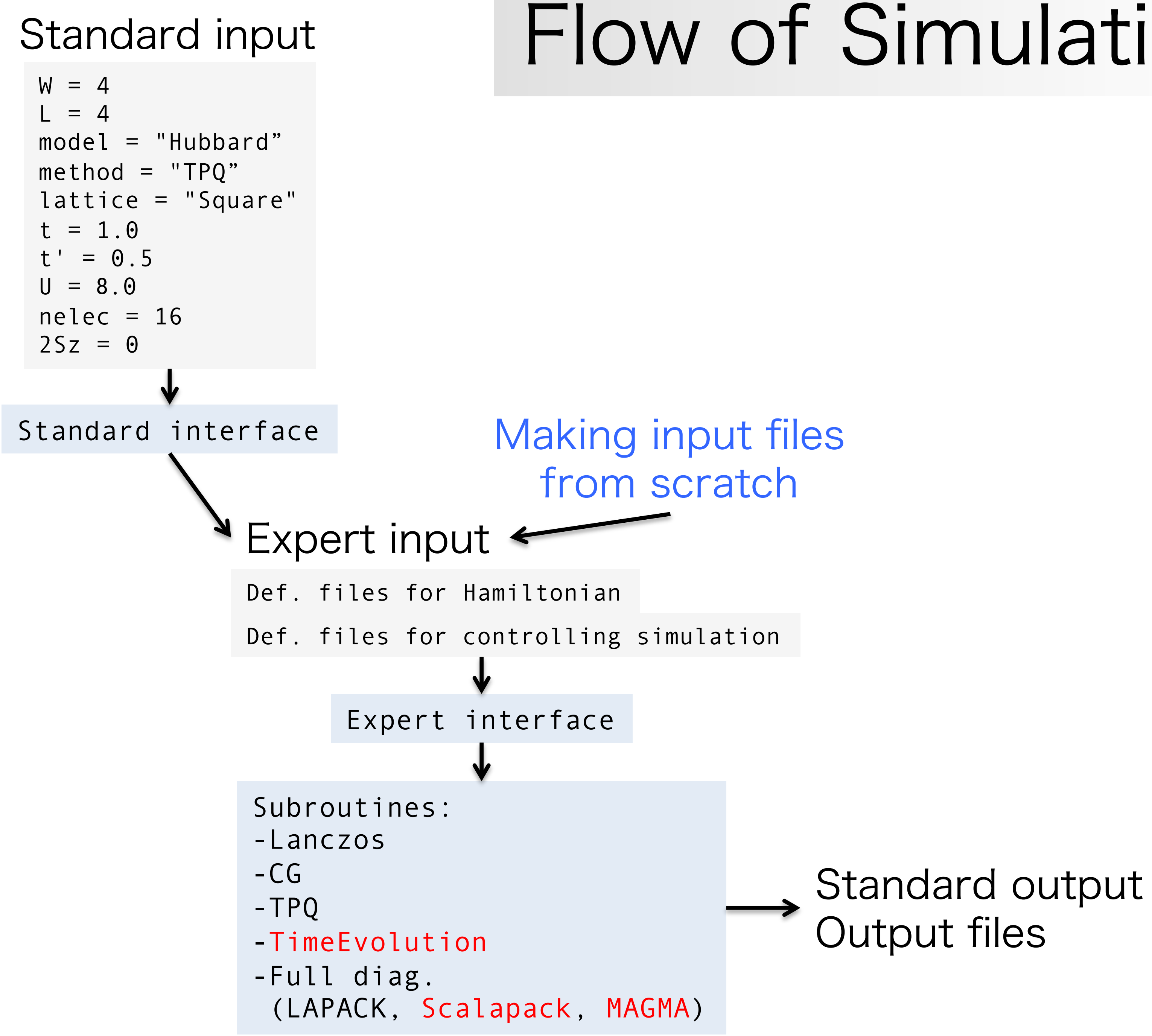
cf.) RESPACK

Overview of Software HΦ

-Language: C
-Compiler: C & Fortran compiler
-Library: BLAS, LAPACK, Kω (distributed with HΦ)
(optional: MPI, Scalapack, MAGMA)
-Parallelization: OpenMP & MPI

For installation, cmake is required

Flow of Simulation



Standard input

```
W = 4
L = 4
model = "Hubbard”
method = "TPQ”
lattice = "Square"
t = 1.0
t' = 0.5
U = 8.0
nelec = 16
2Sz = 0
```

`Standard interface`

Making input files from scratch

Expert input

`Def. files for Hamiltonian`

`Def. files for controlling simulation`

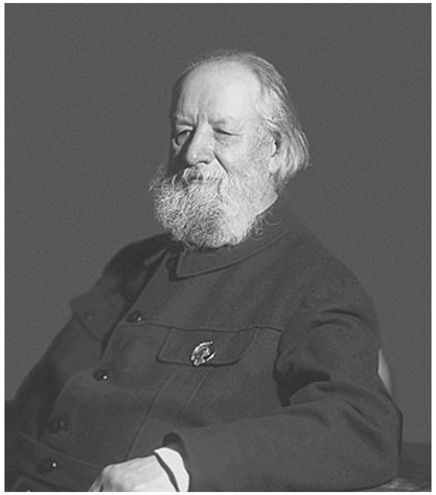
`Expert interface`

```
Subroutines:
-Lanczos
-CG
-TPQ
-TimeEvolution
-Full diag.
 (LAPACK, Scalapack, MAGMA)
```

Standard output
Output files

Algorithm Implemented in HΦ: Lanczos & LOBCG

Krylov Subspace Method for Sparse and Huge Matrices



Alexey Krylov

Aleksey Nikolaevich Krylov
1863-1945
Russian naval engineer and applied mathematician

Krylov subspace $A \in \mathbb{C}^{L \times L}$

$$\mathcal{K}_n(A, \vec{b}) = \mathrm{span}\{\vec{b}, A\vec{b}, \ldots, A^{n-1}\vec{b}\}$$

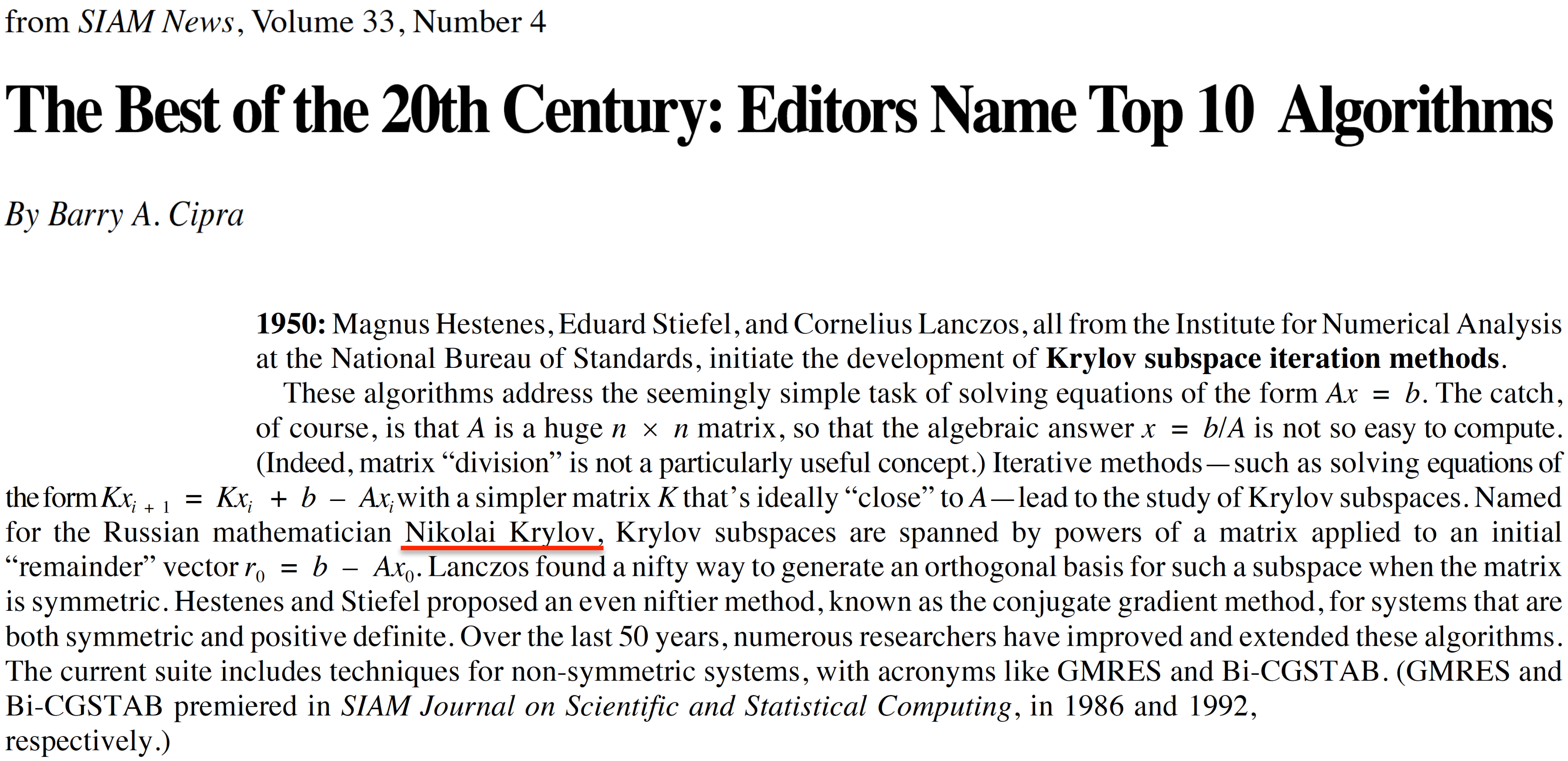
Numerical cost to construct K_n: $\mathcal{O}(\mathrm{nnz}(A) \times n)$

Numerical cost to orthogonalize K_n: $\mathcal{O}(L \times n^2)$

Cornelius Lanczos 1950
Walter Edwin Arnoldi 1951

*nnz: Number of non-zero entries/elements

Krylov Subspace Method



from *SIAM News*, Volume 33, Number 4

The Best of the 20th Century: Editors Name Top 10 Algorithms

By Barry A. Cipra

1950: Magnus Hestenes, Eduard Stiefel, and Cornelius Lanczos, all from the Institute for Numerical Analysis at the National Bureau of Standards, initiate the development of **Krylov subspace iteration methods**.

These algorithms address the seemingly simple task of solving equations of the form $Ax = b$. The catch, of course, is that A is a huge $n \times n$ matrix, so that the algebraic answer $x = b/A$ is not so easy to compute. (Indeed, matrix "division" is not a particularly useful concept.) Iterative methods—such as solving equations of the form $Kx_{i+1} = Kx_i + b - Ax_i$ with a simpler matrix K that's ideally "close" to A—lead to the study of Krylov subspaces. Named for the Russian mathematician Nikolai Krylov, Krylov subspaces are spanned by powers of a matrix applied to an initial "remainder" vector $r_0 = b - Ax_0$. Lanczos found a nifty way to generate an orthogonal basis for such a subspace when the matrix is symmetric. Hestenes and Stiefel proposed an even niftier method, known as the conjugate gradient method, for systems that are both symmetric and positive definite. Over the last 50 years, numerous researchers have improved and extended these algorithms. The current suite includes techniques for non-symmetric systems, with acronyms like GMRES and Bi-CGSTAB. (GMRES and Bi-CGSTAB premiered in *SIAM Journal on Scientific and Statistical Computing*, in 1986 and 1992, respectively.)

Lanczos Method

```
**Initial** : $\beta_1 = 0,\ |v_0\rangle = 0$

**for** $j = 1, 2, \ldots, m$ **do**
    $|w_j\rangle = \hat{H}|v_j\rangle - \beta_j|v_{j-1}\rangle$
    $\alpha_j = \langle w_j|v_j\rangle$
    $|w_j\rangle \leftarrow |w_j\rangle - \alpha_j|v_j\rangle$
    $\beta_{j+1} = \sqrt{\langle w_j|w_j\rangle}$
    $|v_{j+1}\rangle = |w_j\rangle/\beta_{j+1}$
```

Lanczos Method

$$\alpha_j = \langle v_j|\hat{H}|v_j\rangle$$

$$\beta_j = \langle v_{j-1}|\hat{H}|v_j\rangle = \langle v_j|\hat{H}|v_{j-1}\rangle$$

Orthogonalization

$$|v_j\rangle = \frac{\hat{H}|v_{j-1}\rangle - \sum_{\ell=1}^{j-1}|v_\ell\rangle\langle v_\ell|\hat{H}|v_{j-1}\rangle}{\langle v_j|\hat{H}|v_{j-1}\rangle}$$

$$\langle v_\ell|\hat{H}|v_{j-1}\rangle = \begin{cases} 0 & (\ell \leq j-3) \\ \beta_{j-1} & (\ell = j-2) \\ \alpha_{j-1} & (\ell = j-1) \end{cases}$$

Lanczos Method

$$\alpha_j = \langle v_j | \hat{H} | v_j \rangle$$

$$\langle v_j | v_k \rangle = \delta_{j,k}$$

$$\beta_j = \langle v_{j-1} | \hat{H} | v_j \rangle = \langle v_j | \hat{H} | v_{j-1} \rangle$$

Hamiltonian projected onto *m* D Krylov subsace

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \beta_{m-1} & \\ & & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & & \beta_m & \alpha_m \end{pmatrix}$$

Eigenvalues of projected Hamiltonian
→ Approximate eigenvalues of original Hamiltonian

Lanczos Method: # of Vectors Required

```
$\textbf{Initial}: \beta_1 = 0,\ |v_0\rangle = 0$

$\textbf{for}\ j = 1, 2, \ldots, m\ \textbf{do}$
    $|w_j\rangle \leftarrow \hat{H}|v_j\rangle - \beta_j|v_{j-1}\rangle$
    $\alpha_j = \langle w_j|v_j\rangle$
    $|w_j\rangle \leftarrow |w_j\rangle - \alpha_j|v_j\rangle$
    $\beta_{j+1} = \sqrt{\langle w_j|w_j\rangle}$
    $|v_{j+1}\rangle = |w_j\rangle/\beta_{j+1}$
```

$$|v_{j-1}\rangle \rightarrow |w_j\rangle,\ |v_j\rangle$$

$$|w_j\rangle,\ |v_j\rangle$$

$$|w_j\rangle,\ |v_j\rangle$$

$$|w_j\rangle,\ |v_j\rangle$$

$$|w_j\rangle \rightarrow |v_{j+1}\rangle,\ |v_j\rangle$$

Convergence of Lanczos Method

Yousef Saad,
Numerical Methods for Large Eigenvalue Problems (2nd ed)
The Society for Industrial and Applied Mathematics 2011

Assumption: $\lambda_1 > \lambda_2 > \cdots > \lambda_n$

Convergence theorem for the largest eigenvalue

$$0 \leq \lambda_1 - \lambda_1^{(m)} \leq (\lambda_1 - \lambda_n) \left[\frac{\tan\theta(|v_1\rangle, |1\rangle)}{C_{m-1}(1+2\gamma_1)} \right]^2$$

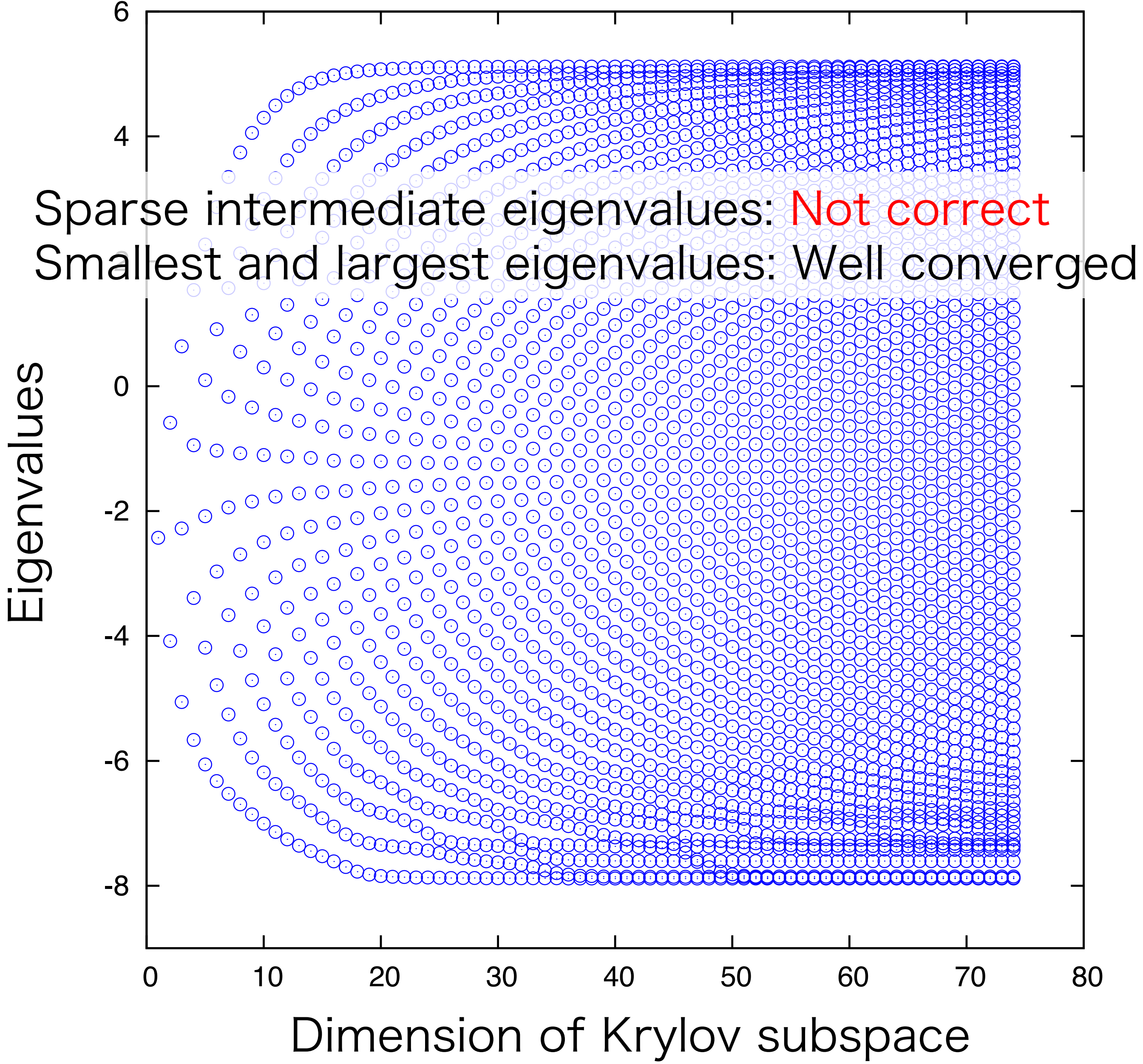
$$\sim 4(\lambda_1 - \lambda_n) \left[\tan\theta(|v_1\rangle, |1\rangle)\right]^2 e^{-4\sqrt{\gamma_1}m}$$

$$\gamma_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}$$

$$C_k(t) = \frac{1}{2}\left[\left(t+\sqrt{t^2-1}\right)^k + \left(t+\sqrt{t^2-1}\right)^{-k}\right]$$

24 site cluster of Kitaev-Γ model (frustrated S =1/2 spins)

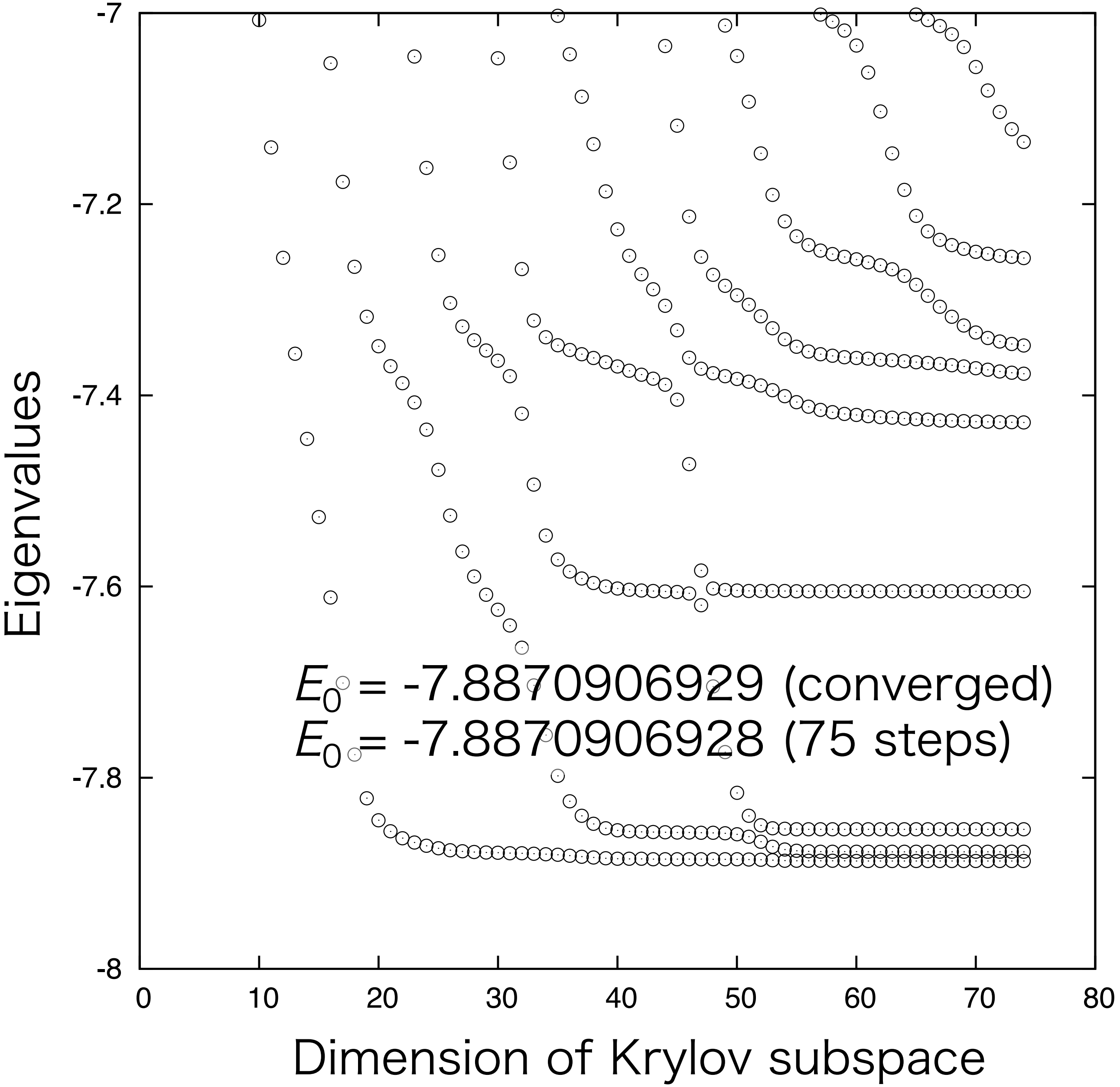
Dimension of Fock space: 2^{24}=16777216



Sparse intermediate eigenvalues: Not correct

Smallest and largest eigenvalues: Well converged

How Lanczos Method Works



E_0 = -7.8870906929 (converged)

E_0 = -7.8870906928 (75 steps)

LOB(P)CG

Algorithm of LOBCG: *m* lowest eigenstates

Initial condition: m orthogonal initial vectors $\boldsymbol{x}_0^{(i)}$
$\boldsymbol{p}_0^{(i)} = \boldsymbol{0}$ $(i = 1, \ldots, m)$

```
for $(k = 0; k < k_{\max}; k{+}{+})$
    $\mu_k^{(i)} = \frac{(\boldsymbol{x}_k^{(i)} A \boldsymbol{x}_k^{(i)})}{(\boldsymbol{x}_k^{(i)}, \boldsymbol{x}_k^{(i)})}$        ←approximation of i th smallest eigenvalue
    $\boldsymbol{w}_k^{(i)} = A\boldsymbol{x}_k^{(i)} - \mu_k^{(i)} \boldsymbol{x}_k^{(i)}$
    $S_A = \{\boldsymbol{w}_k^{(1)}, \ldots, \boldsymbol{w}_k^{(m)}, \boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(m)}, \boldsymbol{p}_k^{(1)}, \ldots, \boldsymbol{p}_k^{(m)}\}^T A \{\boldsymbol{w}_k^{(1)}, \ldots, \boldsymbol{w}_k^{(m)}, \boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(m)}, \boldsymbol{p}_k^{(1)}, \ldots, \boldsymbol{p}_k^{(m)}\}$
    $S_B = \{\boldsymbol{w}_k^{(1)}, \ldots, \boldsymbol{w}_k^{(m)}, \boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(m)}, \boldsymbol{p}_k^{(1)}, \ldots, \boldsymbol{p}_k^{(m)}\}^T \{\boldsymbol{w}_k^{(1)}, \ldots, \boldsymbol{w}_k^{(m)}, \boldsymbol{x}_k^{(1)}, \ldots, \boldsymbol{x}_k^{(m)}, \boldsymbol{p}_k^{(1)}, \ldots, \boldsymbol{p}_k^{(m)}\}$
    Obtain $i$th smallest eigenstate of $S_A \boldsymbol{v}^{(i)} = \mu^{(i)} S_B \boldsymbol{v}^{(i)}$, $\boldsymbol{v}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_m^{(i)}, \beta_1^{(i)}, \ldots, \beta_m^{(i)}, \gamma_1^{(i)}, \ldots, \gamma_m^{(i)})^T$
    $(i = 1, \ldots, m)$
                ←m smallest eigenstates are chosen from 3m states
    $\boldsymbol{x}_{k+1}^{(i)} = \sum_{j=1}^{m} \left( \alpha_j^{(i)} \boldsymbol{w}_k^{(j)} + \beta_j^{(i)} \boldsymbol{x}_k^{(j)} + \gamma_j^{(i)} \boldsymbol{p}_k^{(j)} \right)$
    $\boldsymbol{p}_{k+1}^{(i)} = \sum_{j=1}^{m} \left( \alpha_j^{(i)} \boldsymbol{w}_k^{(j)} + \gamma_j^{(i)} \boldsymbol{p}_k^{(j)} \right)$
Until convergence
```

Algorithm Implemented in HΦ: TPQ

(Finite-Temperature Simulation)

Finite-Temperature Physical Quantity: Heat Capacity

Spread of energy distribution

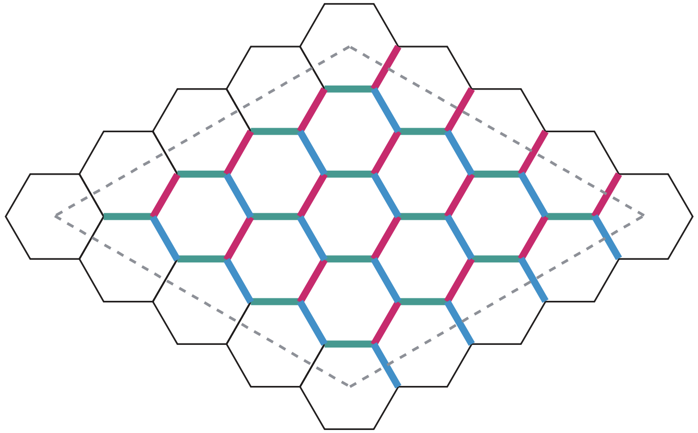
$$C = \frac{\langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2}{k_{\mathrm{B}} T^2}$$

-Average with Boltzmann distribution

$$\langle \hat{O}_{\beta}^{\mathrm{ens}} \rangle = \frac{\sum_n e^{-E_n/k_{\mathrm{B}}T} \langle n|\hat{O}|n\rangle}{\sum_n e^{-E_n/k_{\mathrm{B}}T}}$$

Complexity	$\mathcal{O}(N_{\mathrm{H}}^3)$
Memory	$\mathcal{O}(N_{\mathrm{H}}^2)$

32 site cluster of S=1/2 spin



$$N_{\mathrm{H}} = 2^{32}$$

Hamiltonian
∼3x10⁸TB!

$|n\rangle$ ∼69GB

Typical Pure State Approach

Complexity

Memory

$$\mathcal{O}(N_{\mathrm{H}})$$

Imada-Takahashi (1986)
Lloyd (1988)
Jacklic-Prelovsek (1994)
Hams-De Raedt (2000)
Sugiura-Shimizu (2012, 2013)

M. Imada & M. Takahashi, J. Phys. Soc. Jpn. 55, 3354 (1986).

$\beta = 0 \ (T \to +\infty)$ Typical state: Random vector

$$|\phi_0\rangle = \sum_x c_x |x\rangle \quad (\sum_x |c_x|^2 = 1)$$

$$\langle \hat{O} \rangle^{\mathrm{ens}}_{\beta=0} = \mathbb{E}[\langle \phi_0 | \hat{O} | \phi_0 \rangle]$$

N. Ullah, Nucl. Phys. 58, 65 (1964).
-Uniform distribution on
unit sphere in $\mathbb{R}^{2N_{\mathrm{H}}}$

$$\mathbb{E}[|c_x|^{2n}] = \frac{\Gamma(N_{\mathrm{H}})\Gamma(n+1)}{\Gamma(N_{\mathrm{H}}+n)}$$

Average over the distribution

At finite temperature

$$|\phi_\beta\rangle = e^{-\beta \hat{H}/2} |\phi_0\rangle$$

How large is the standard deviation?

$$\sigma_O^2 = \mathbb{E}\left[\left(\frac{\langle \phi_\beta | \hat{O} | \phi_\beta \rangle}{\langle \phi_\beta | \phi_\beta \rangle} - \langle \hat{O} \rangle^{\mathrm{ens}}_{\beta}\right)^2\right]$$

Typical Pure State Approach

Seth Lloyd, Ph.D. Thesis, Rockefeller University (1988); arXiv:1307.0378.
A. Hams & H. De Raedt, Phys. Rev. E 62, 4365 (2000).

A. Sugita, RIMS Kokyuroku (Kyoto) 1507, 147 (2006).
P. Reimann, Phys. Rev. Lett. 99, 160404 (2007).
S. Sugiura & A. Shimizu, Phys. Rev. Lett. 108, 240401 (2012).

S. Sugiura & A. Shimizu, Phys. Rev. Lett. 111, 010401 (2013).

$$\boxed{\sigma_O^2 \le \frac{\langle(\Delta O)^2\rangle_{2\beta}^{\mathrm{ens}} + (\langle O\rangle_{2\beta}^{\mathrm{ens}} - \langle O\rangle_{\beta}^{\mathrm{ens}})^2}{\exp[2\beta\{F(2\beta) - F(\beta)\}]}}$$

$$\propto \exp[-S(\beta^*)/2] \ (\beta < \beta^* < 2\beta)$$

Exponetially small when system size increases

Construction of Typical Pure State

Thermal Pure Quantum (TPQ) States $|\phi_\beta\rangle = |\Phi_k\rangle$

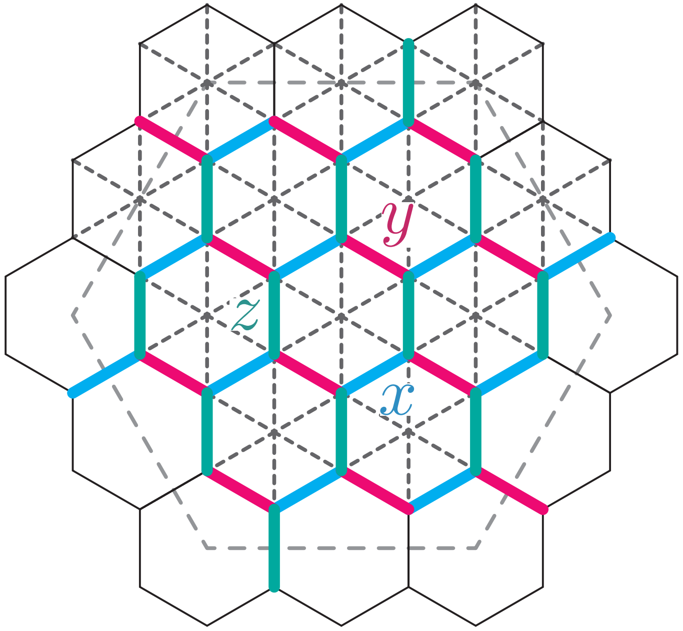
Sugiura & Shimizu, Phys. Rev. Lett. 108, 240401 (2012)

```
Initial state ( at $T$ = +∞ ): $|\Phi_0\rangle = (\mathrm{Random\ vector})$
do $k$=1,$N_{\mathrm{step}}$                    If possible, taking  random average
    $|\Phi_k\rangle = (\ell - \hat{H}/N)\,|\Phi_{k-1}\rangle \,/ \sqrt{\langle\Phi_{k-1}|\,(\ell - \hat{H}/N)^2\,|\Phi_{k-1}\rangle}$
    $u_k = \langle\Phi_k|\,\hat{H}/N\,|\Phi_k\rangle$
    $\beta = 2(k/N)/(\ell - u_k) \quad (\beta = 1/k_{\mathrm{B}}T)$
    $\overline{O}(\beta) = \langle\Phi_k|\,\hat{O}\,|\Phi_k\rangle + \mathcal{O}(1/N)$
enddo
```

Hamiltonian-wave function product is essential

Example of TPQ: Effective Hamiltonian of α-RuCl$_3$, K-Γ-J_3 Model

$$\hat{H} = \sum_{\Gamma=X,Y,Z,3} \sum_{\langle \ell,m\rangle\in\Gamma} \hat{\vec{S}}_{\ell}^{T} \mathcal{J}_{\Gamma} \hat{\vec{S}}_{m} \qquad \hat{\vec{S}}_{\ell}^{T} = (\hat{S}_{\ell}^{x}, \hat{S}_{\ell}^{y}, \hat{S}_{\ell}^{z})$$



$$\left.\begin{aligned}
\mathcal{J}_X &= \begin{bmatrix} -\cos\phi & 0 & 0 \\ 0 & 0 & \sin\phi \\ 0 & \sin\phi & 0 \end{bmatrix} \\
\mathcal{J}_Y &= \begin{bmatrix} 0 & 0 & \sin\phi \\ 0 & -\cos\phi & 0 \\ \sin\phi & 0 & 0 \end{bmatrix} \\
\mathcal{J}_Z &= \begin{bmatrix} 0 & \sin\phi & 0 \\ \sin\phi & 0 & 0 \\ 0 & 0 & -\cos\phi \end{bmatrix}
\end{aligned}\right\}$$

Nearest neighbor

$$\mathcal{J}_3 = \begin{bmatrix} J_3 & 0 & 0 \\ 0 & J_3 & 0 \\ 0 & 0 & J_3 \end{bmatrix}$$

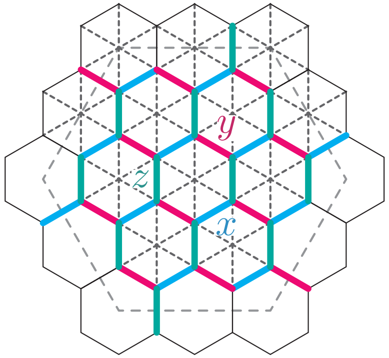
3rd neighbor

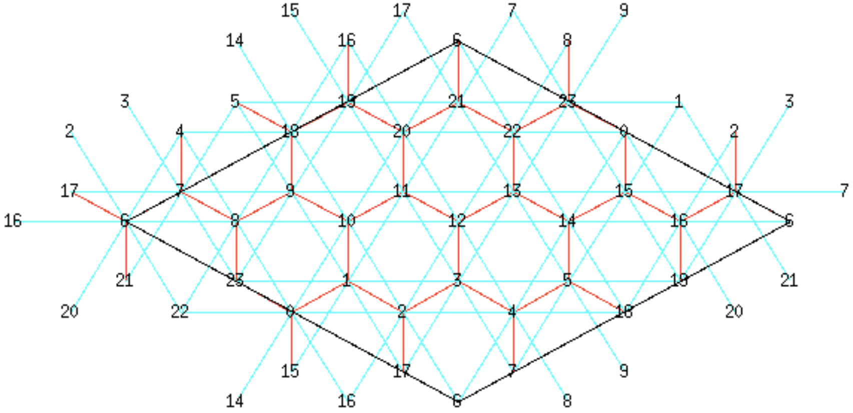
$$J_3[\hat{S}_{\ell}^{x}\hat{S}_{m}^{x} + \hat{S}_{\ell}^{y}\hat{S}_{m}^{y} + \hat{S}_{\ell}^{z}\hat{S}_{m}^{z}]$$

How to Simulate K-Γ-J_3 Model

$$\phi/\pi = 0.2$$

```
model = "SpinGC"
method = "TPQ"
lattice = "Honeycomb"
a0w = 2
a0l = 2
a1w = 4
a1l = -2
J0x = -0.80901699437
J0yz = 0.58778525229
J0zy = 0.58778525229
J1zx = 0.58778525229
J1y = -0.80901699437
J1xz = 0.58778525229
J2xy = 0.58778525229
J2yx = 0.58778525229
J2z = -0.80901699437
h = 0.07071067811
Gamma = -0.07071067811
2S=1
```





$$\mathcal{J}_X = \begin{bmatrix} -\cos\phi & 0 & 0 \\ 0 & 0 & \sin\phi \\ 0 & \sin\phi & 0 \end{bmatrix}$$

$$\mathcal{J}_Y = \begin{bmatrix} 0 & 0 & \sin\phi \\ 0 & -\cos\phi & 0 \\ \sin\phi & 0 & 0 \end{bmatrix}$$

$$\mathcal{J}_Z = \begin{bmatrix} 0 & \sin\phi & 0 \\ \sin\phi & 0 & 0 \\ 0 & 0 & -\cos\phi \end{bmatrix}$$

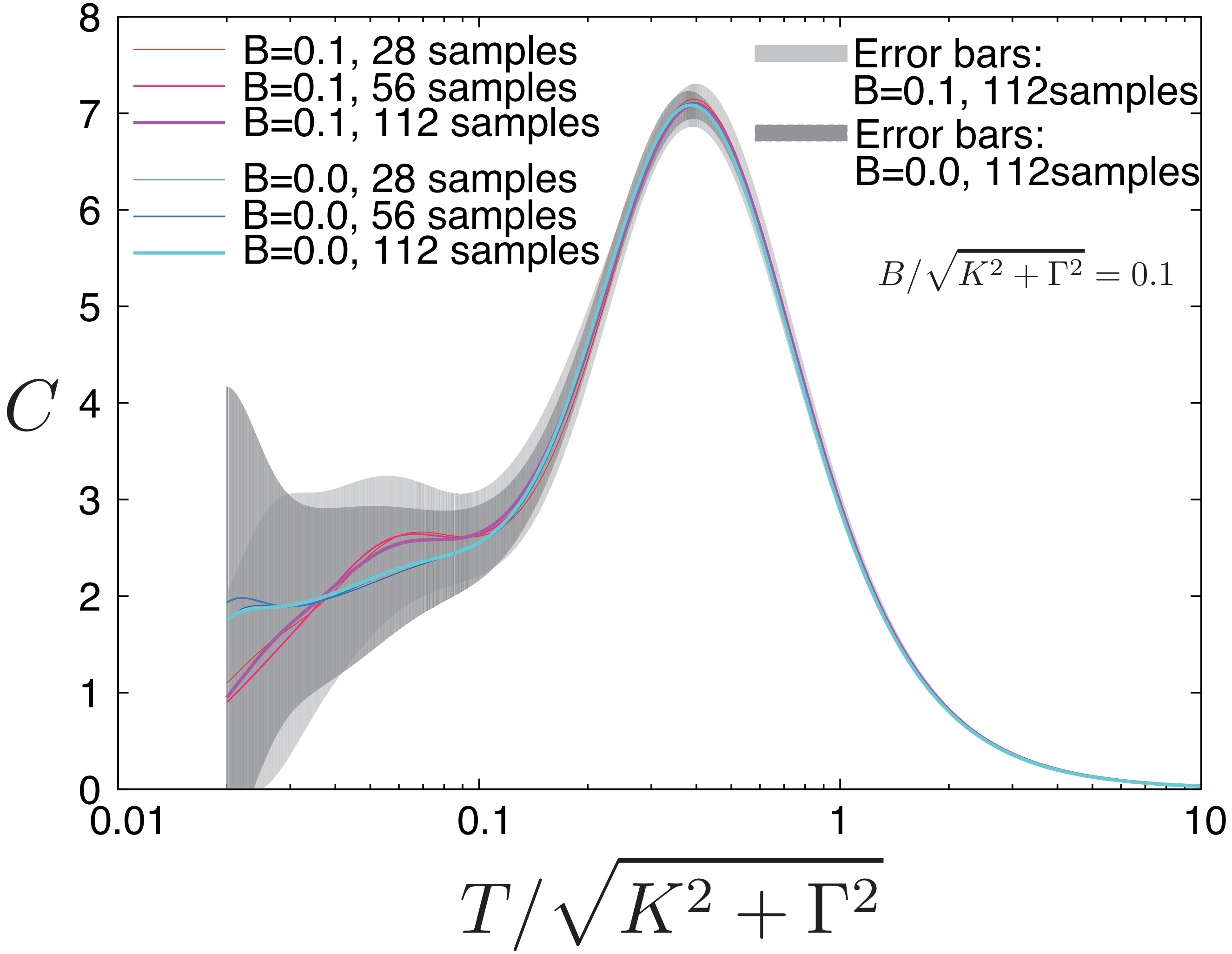
$$\vec{B} \propto (1, 0, -1)$$

$$\mathcal{J}_3 = \begin{bmatrix} J_3 & 0 & 0 \\ 0 & J_3 & 0 \\ 0 & 0 & J_3 \end{bmatrix}$$

Add `Exchange` and `Ising` by Expert mode

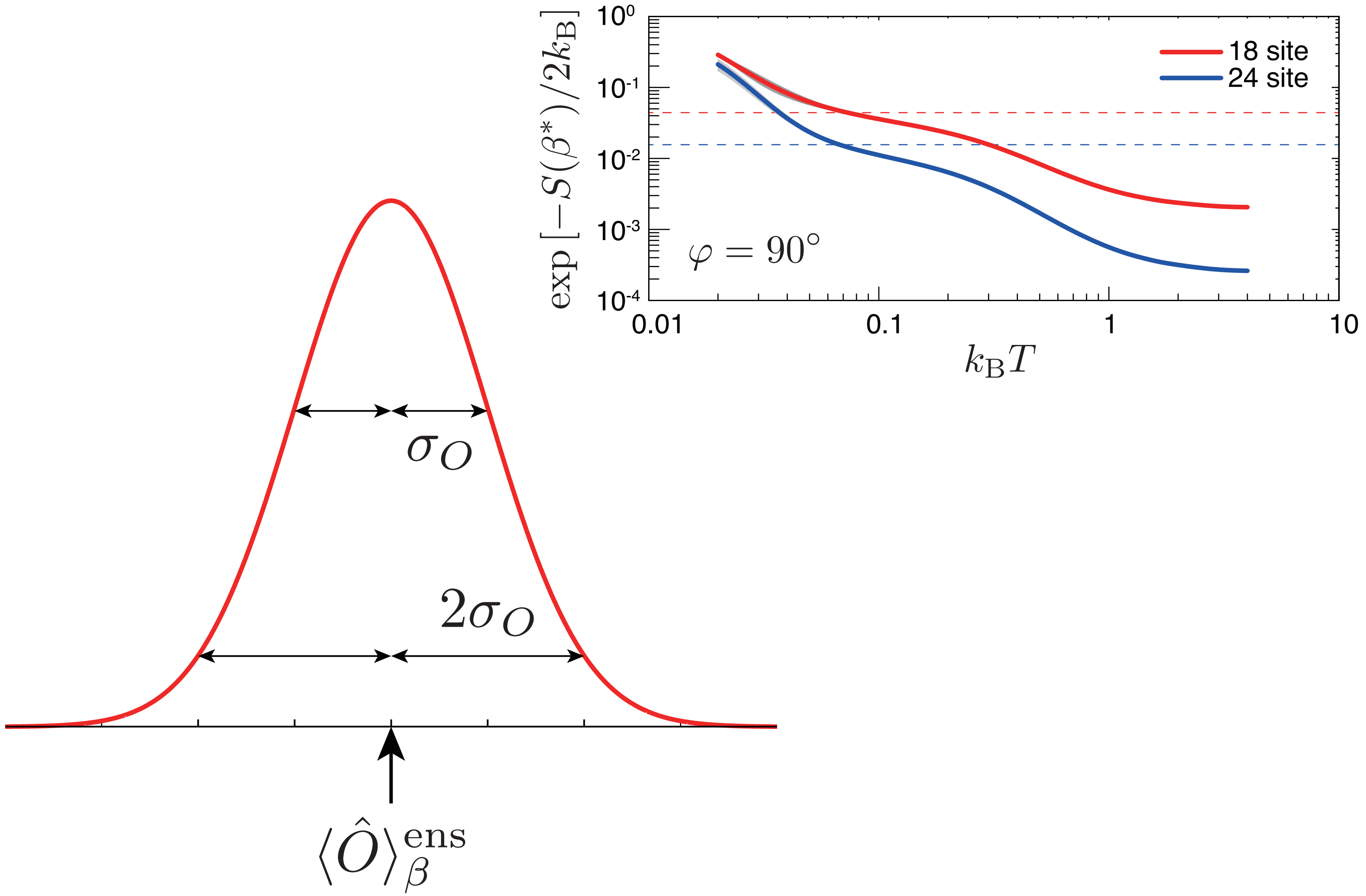
Heat Capacity of K-Γ-J_3 Model

$$\phi/\pi = 0.2,\ \ J_3/\sqrt{K^2+\Gamma^2} = 0.05$$



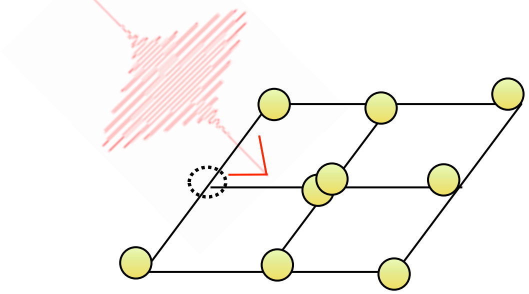
Standard Deviation

Standard deviation in TPQ



$\propto \sigma_O$

$$i\frac{\partial}{\partial t}\left|\psi\right\rangle = \mathcal{H}\left|\psi\right\rangle$$



Algorithm Implemented in HΦ: Real time evolution

Univ. of Tokyo, ISSP
Kota Ido

Quantum Dynamics

“強相関電子系”

多彩な現象が出現

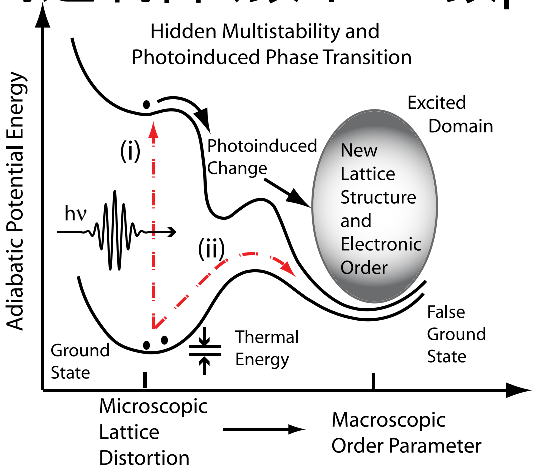
・金属絶縁体転移　・高温超伝導

・磁気秩序　　　　・・・

近年におけるレーザー技術の発達により、強相関電子系における非平衡状態に注目されている

ex1)光誘起相転移

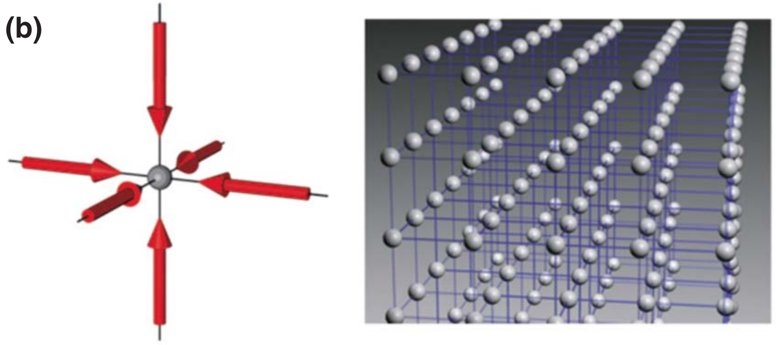
光照射による相転移の超高速制御（数十fs~数ps）



Basov et al. , Rev. Mod. Phys.**83**,431(2011).

ex2)光格子における冷却原子のダイナミクス

非平衡ダイナミクスの基礎的性質



Bloch et al. , Rev. Mod. Phys.**80**,885431(2008).

HΦへの量子ダイナミクス機能実装

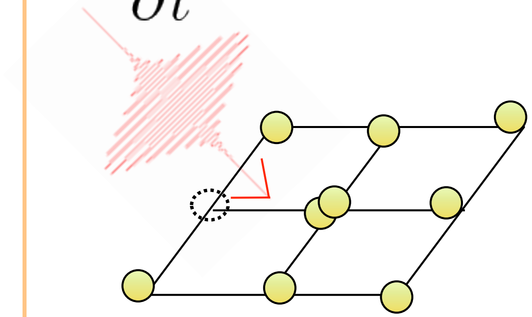
- テイラー展開に基づく実時間発展計算

$$e^{-i\mathcal{H}(t_n)\Delta t} \approx \sum_{l=0}^{M} \frac{1}{l!} \left(-i\mathcal{H}(t_n)\Delta t\right)^l$$

$$|\psi(t_{n+1})\rangle = e^{-i\mathcal{H}(t_n)\Delta t} |\psi(t_n)\rangle$$

Nonequilibrium

$$i\frac{\partial}{\partial t}|\psi\rangle = \mathcal{H}|\psi\rangle$$



- スタンダードモード: 相互作用クエンチ、光照射(パルス、AC、DC)
- エキスパートモード: 各時刻での一体・二体相互作用を指定

- 出力

各時刻でのエネルギー、二重占有度、ノルム、
同時刻グリーン関数

スタンダードモードを使った全体の流れ

- 1st. Step : 基底状態の計算
  Lanczos法による初期状態(固有ベクトル)の計算・出力

- 2nd. Step : 実時間発展演算
  Method= “Time-Evolution”に指定
  時間発展の仕方を指定(キーワードPumpType)
  各種パラメータを指定

※必要に応じて、光学伝導度などのスペクトル計算やTPQなどの有限温度計算を行う。

スタンダードモードを使った基底状態の計算

```
model = ″Hubbard″
lattice = ″Square″
a0W = 2
a0L = 2
a1W = 2
a1L = −2
method = ″CG″
t = 1.0
U = 10.0
nelec = 8
2Sz = 0
EigenvecIO = ″out″
```

zvo_eigenvec_%%_rank_$$.dat (%%: 固有値の番号、$$: プロセス番号)がoutputディレクトリに出力される

スタンダードモードを使った時間発展演算

関連キーワード

method: “Time-Evolution”を指定

lanczos_max: 時間発展のステップ数

dt : 時間刻み幅

PumpType = “Quench”, “AC Laser”, “DC Laser”, “Pulse Laser”

スタンダードモードを使った時間発展演算

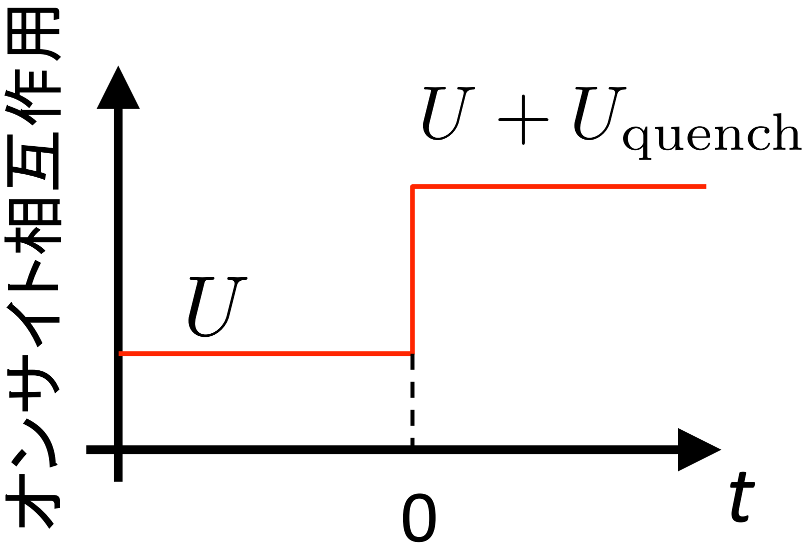
関連キーワード

method: “Time-Evolution”を指定

lanczos_max: 時間発展のステップ数

dt : 時間刻み幅, ExpandCoef:テイラー展開次数

PumpType = “Quench”, “AC Laser”, “DC Laser”, “Pulse Laser”



キーワードUquench (実数)

$$U_{\text{quench}} \sum_i n_{i\uparrow} n_{i\downarrow}$$

が初期時刻で加えられる

スタンダードモードを使った時間発展演算

関連キーワード

method: “Time−Evolution”を指定

lanczos_max: 時間発展のステップ数

dt : 時間刻み幅, ExpandCoef:テイラー展開次数

PumpType = “Quench”, “AC Laser”, “DC Laser”, “Pulse Laser”

パイエルス位相により電場効果を導入

$$t_{ij} \rightarrow t_{ij} e^{-i\boldsymbol{A}(t)\cdot(\boldsymbol{R}_i - \boldsymbol{R}_j)/(2\pi)}$$

スタンダードモードを使った時間発展演算

各モードごとのベクトルポテンシャル

- “AC Laser” $\boldsymbol{A}(t) = \boldsymbol{A}_0 \sin\left[\omega(t - t_0)\right]$

- “DC Laser” $\boldsymbol{A}(t) = \boldsymbol{A}_0 t$

- “Pulse Laser”

$$\boldsymbol{A}(t) = \boldsymbol{A}_0 \exp\left[-(t - t_0)^2/(2t_{\mathrm{dump}}^2)\right] \cos\left[\omega(t - t_0)\right]$$

関連キーワード

freq: ω	tdump: t_{dump}
tshift: t_0	VecPotW, L: $\boldsymbol{A}_0$の強度

時間発展演算での出力ファイル

Outputディレクトリ

Norm.dat

各時刻ごとのノルム絶対値を出力
→ユニタリー時間発展がどれだけうまくいっているかの指針

SS.dat

各時刻ごとのエネルギー、二重占有度などを出力

##_cisajs_step%%.dat

各時刻ごとのOneBodyGで指定した一体グリーン関数の計算結果を出力

など

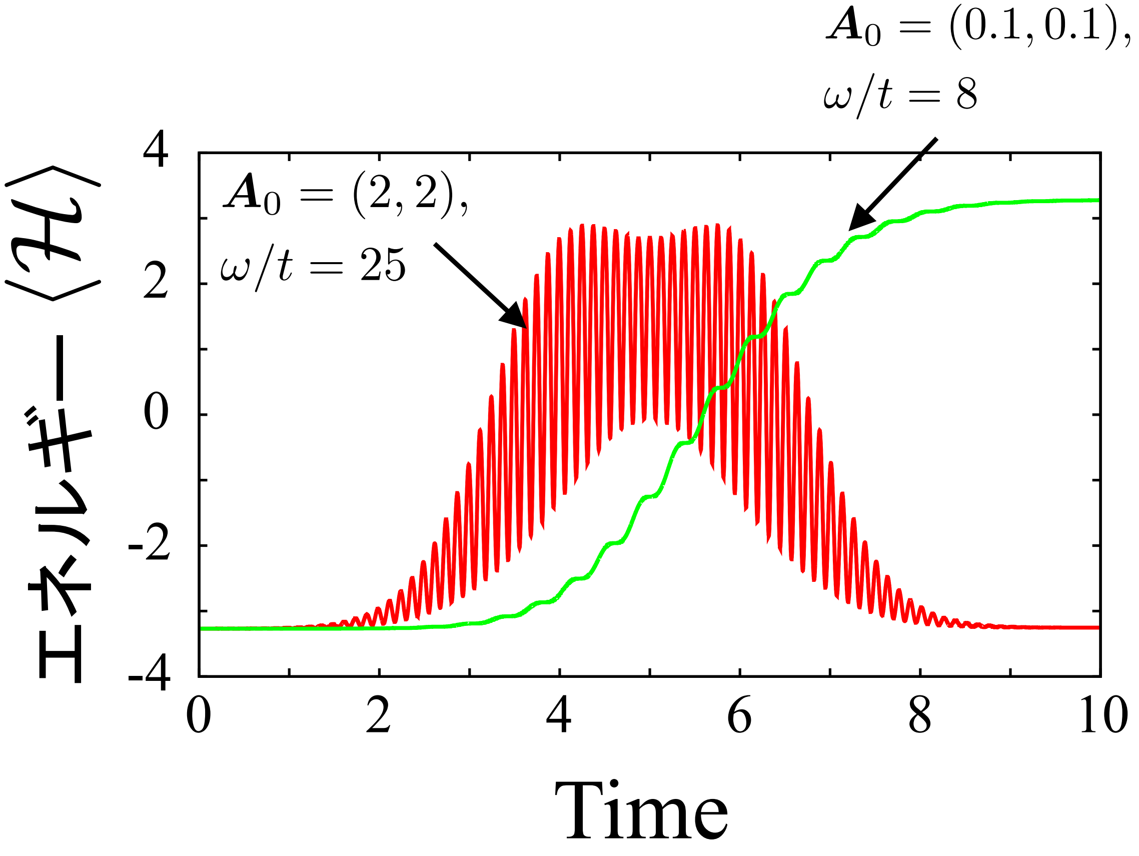
※実行ディレクトリ直下には、各時刻でのベクトルポテンシャルと電場を図示するためのファイル potential.dat が出力

実例: Gaussianパルス照射

```
model = ″Hubbard″
method = ″Time−Evolution″
lattice = ″Square″
a0W = 2
a0L = 2
a1W = 2
a1L = −2
lanczos_max = 5000
dt = 0.001
PumpType = ″Pulse Laser″

 t = 1.0
 U = 10.0
 nelec = 8
 2Sz = 0
 EigenvecIO = ″in″
 tshift = 5.0
 tdump = 1.5
 freq = 8.0
 VecPotW = 0.1
 VecPotL = 0.1
```

$$\boldsymbol{A}(t) = \boldsymbol{A}_0 \exp\left[-(t-t_0)^2/(2t_{\rm dump}^2)\right] \cos\left[\omega(t-t_0)\right]$$



その他

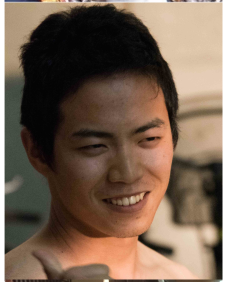
- エキスパートモードではOneBodyTEやTwoBodyTEを使うことで、任意の一体・二体相互作用作用をハミルトニアンに導入できます。

- 詳細はマニュアルをご覧ください

http://issp-center-dev.github.io/HPhi/manual/userguide_HPhi_ja.pdf

HΦ Developers

Dr. Takahiro Misawa
The Institute for Solid State Physics,
The University of Tokyo



Dr. Mitsuaki Kawamura
The Institute for Solid State Physics,
The University of Tokyo

Dr. Yoshikazu Yoshimi
The Institute for Solid State Physics,
The University of Tokyo

Prof. Synge Todo
Department of Physics,
The University of Tokyo

Prof. Naoki Kawashima
The Institute for Solid State Physics,
The University of Tokyo

Kota Ido
The Institute for Solid State Physics,
The University of Tokyo

Acknowledgement:
Dr. Yoshinori Nakanishi-Ohno (Univ. of Tokyo)
Prof. Takeo Hoshi (Tottori Univ.)
Prof. Tomohiro Sogabe (Nagoya Univ.)
“*Project for advancement of software usability in materials science*” by ISSP

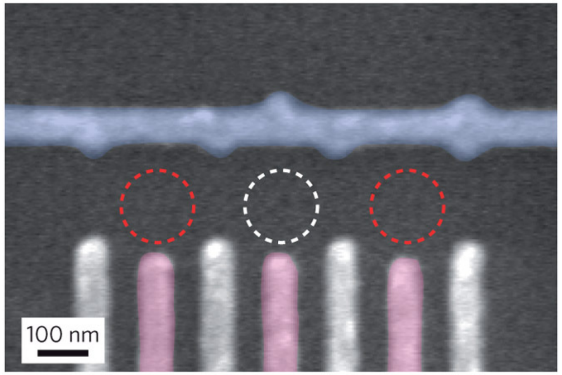
Youhei Yamaji
Department of Applied Physics,
The University of Tokyo

Appendix: Formulation of Quantum Many-Body Problem

Quantum Many-Body Problems

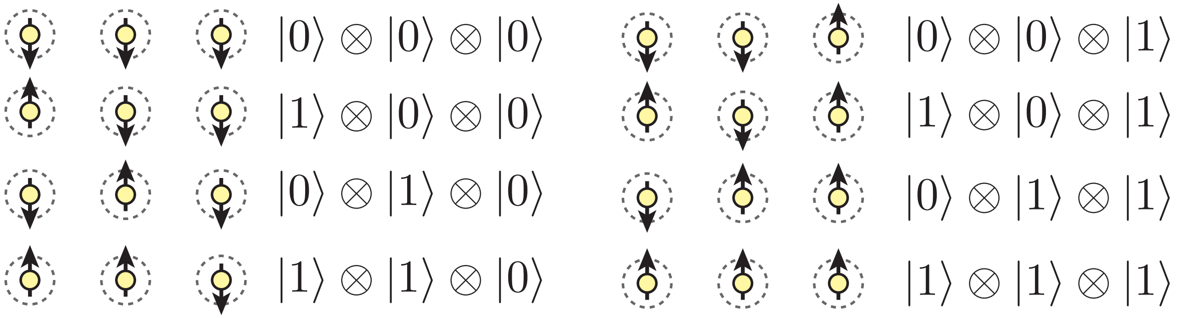
An example: 3 Quantum dots

F. R. Braakman, et al., Nat. Nano. 8, 432 (2013)



Quantum dot:
-A quantum box can confine a single electron
-Utilized for single electron transistor, quantum computers

Three-body problem:
→ Number of states = 2^3 (factor 2 from spin)



$|0\rangle \otimes |0\rangle \otimes |0\rangle$ $|0\rangle \otimes |0\rangle \otimes |1\rangle$

$|1\rangle \otimes |0\rangle \otimes |0\rangle$ $|1\rangle \otimes |0\rangle \otimes |1\rangle$

$|0\rangle \otimes |1\rangle \otimes |0\rangle$ $|0\rangle \otimes |1\rangle \otimes |1\rangle$

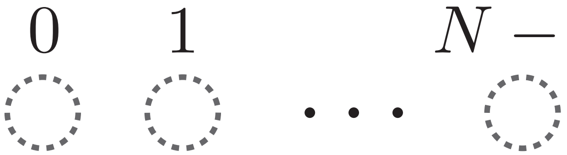
$|1\rangle \otimes |1\rangle \otimes |0\rangle$ $|1\rangle \otimes |1\rangle \otimes |1\rangle$

States represented by superposition

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \sum_{n_2=0,1} C_{n_0 n_1 n_2} |n_0\rangle \otimes |n_1\rangle \otimes |n_2\rangle : C_{n_0 n_1 n_2} \in \mathbb{C} \}$$

Quantum Many-Body Problems

Mutual Interactions



Operators acting on
a single qubit

A two dimensional representation
of Lie algebra SU(2)

$$[\hat{S}_j^x, \hat{S}_j^y] = i\hat{S}_j^z$$
$$[\hat{S}_j^y, \hat{S}_j^z] = i\hat{S}_j^x$$
$$[\hat{S}_j^z, \hat{S}_j^x] = i\hat{S}_j^y$$

$$\hat{S}_j^x|0\rangle = \frac{1}{2}|1\rangle$$
$$\hat{S}_j^x|1\rangle = \frac{1}{2}|0\rangle$$
$$\hat{S}_j^y|0\rangle = \frac{i}{2}|1\rangle$$
$$\hat{S}_j^y|1\rangle = -\frac{i}{2}|0\rangle$$
$$\hat{S}_j^z|1\rangle = \frac{1}{2}|1\rangle$$
$$\hat{S}_j^z|0\rangle = -\frac{1}{2}|0\rangle$$

Vectors in Fock Space

Correspondence between spin and bit

$$|\uparrow\rangle = |1\rangle$$
$$|\downarrow\rangle = |0\rangle$$

2^N-dimensional Fock space:

$$\mathcal{F} = \{ \sum_{n_0=0,1} \sum_{n_1=0,1} \cdots \sum_{n_{N-1}=0,1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle \}$$

$$(C_{n_0 n_1 \cdots n_{N-1}} \in \mathbb{C})$$

Decimal representation of orthonormalized basis

$$|I\rangle_{\mathrm{d}} = |n_0\rangle \otimes |n_1\rangle \otimes |n_2\rangle \otimes \cdots \otimes |n_{N-1}\rangle \qquad I = \sum_{\nu=0}^{N-1} n_\nu \cdot 2^\nu$$

Wave function as a vector

$$|\phi\rangle = \sum_{n_0=0}^{1} \sum_{n_1=0}^{1} \cdots \sum_{n_{N-1}=0}^{1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

$$v(I) = C_{n_0 n_1 \cdots n_{N-1}} \qquad v(0 : 2^N - 1)$$

Vectors and Matrices in Fock Space

Inner product of vectors

$$\left(\langle n_0| \otimes \langle n_1| \otimes \cdots \otimes \langle n_{N-1}|\right) \times \left(|n'_0\rangle \otimes |n'_1\rangle \otimes \cdots \otimes |n'_{N-1}\rangle\right)$$
$$= \langle n_0|n'_0\rangle \times \langle n_1|n'_1\rangle \times \cdots \times \langle n_{N-1}|n'_{N-1}\rangle$$

$$\langle n| \times |n'\rangle = \langle n|n'\rangle = \delta_{n,n'}$$

$$\langle \phi'|\phi\rangle = \sum_{n_0=0}^{1}\sum_{n_1=0}^{1}\cdots\sum_{n_{N-1}=0}^{1} {C'}^{*}_{n_0 n_1 \cdots n_{N-1}} C_{n_0 n_1 \cdots n_{N-1}}$$

$$|\phi'\rangle = \sum_{n_0=0}^{1}\sum_{n_1=0}^{1}\cdots\sum_{n_{N-1}=0}^{1} C'_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

$$|\phi\rangle = \sum_{n_0=0}^{1}\sum_{n_1=0}^{1}\cdots\sum_{n_{N-1}=0}^{1} C_{n_0 n_1 \cdots n_{N-1}} |n_0\rangle \otimes |n_1\rangle \otimes \cdots \otimes |n_{N-1}\rangle$$

Hamiltonian matrix

$$H_{II'} = \langle I|\hat{H}|I'\rangle$$

Orthonomalized basis: $|I\rangle, |I'\rangle \in \mathcal{F}$ $\quad \langle I|I'\rangle = \delta_{I,I'}$

Example: Two Spins

Decimal representation of orthonormalized basis

		0 th site		1 st site
$\vert 0\rangle_{\mathrm{d}}$	$=$	$\vert\downarrow\rangle$	$\otimes$	$\vert\downarrow\rangle$
$\vert 1\rangle_{\mathrm{d}}$	$=$	$\vert\uparrow\rangle$	$\otimes$	$\vert\downarrow\rangle$
$\vert 2\rangle_{\mathrm{d}}$	$=$	$\vert\downarrow\rangle$	$\otimes$	$\vert\uparrow\rangle$
$\vert 3\rangle_{\mathrm{d}}$	$=$	$\vert\uparrow\rangle$	$\otimes$	$\vert\uparrow\rangle$

Problem: Find 4 by 4 Hamiltonian matrix that describes

$$\hat{H}/J = \hat{S}_0^x\hat{S}_1^x + \hat{S}_0^y\hat{S}_1^y + \hat{S}_0^z\hat{S}_1^z$$

$$= \frac{1}{2}\left(\hat{S}_0^+\hat{S}_1^- + \hat{S}_0^-\hat{S}_1^+\right) + \hat{S}_0^z\hat{S}_1^z$$

Useful transformation:
Ladder operators

$$\hat{S}_j^+ = \hat{S}_j^x + i\hat{S}_j^y$$

$$\hat{S}_j^- = \hat{S}_j^x - i\hat{S}_j^y$$

$$\hat{S}_j^+|\downarrow\rangle = |\uparrow\rangle$$

$$\hat{S}_j^+|\uparrow\rangle = 0$$

$$\hat{S}_j^-|\downarrow\rangle = 0$$

$$\hat{S}_j^-|\uparrow\rangle = |\downarrow\rangle$$

Answer of the Problem

$$\hat{H} = J\left(\hat{S}_0^x\hat{S}_1^x + \hat{S}_0^y\hat{S}_1^y + \hat{S}_0^z\hat{S}_1^z\right)$$

Matrix element $\quad {}_{\mathrm{d}}\langle I|\hat{H}|J\rangle_{\mathrm{d}} \quad (I, J = 0, 1, 2, 3)$

4 by 4 Hamiltonian matrix

$$\hat{H} \doteq J \begin{bmatrix} +1/4 & 0 & 0 & 0 \\ 0 & -1/4 & +1/2 & 0 \\ 0 & +1/2 & -1/4 & 0 \\ 0 & 0 & 0 & +1/4 \end{bmatrix}$$

Answer of the Problem 2: Energy Spectrum of the Two Spins

$$\hat{H} \doteq J \begin{bmatrix} +1/4 & 0 & 0 & 0 \\ 0 & -1/4 & +1/2 & 0 \\ 0 & +1/2 & -1/4 & 0 \\ 0 & 0 & 0 & +1/4 \end{bmatrix}$$

$$E = -\frac{3J}{4}, +\frac{J}{4}, +\frac{J}{4}, +\frac{J}{4}$$