October 1, 2021
CCMS Web Hands-on

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
物質・材料研究機構 エネルギー・環境材料研究拠点
Youhei Yamaji
Center for Green Research on Energy and Environmental Materials,
National Institute for Materials Science

1. HΦとは
2. 何故対角化？
3. HΦが扱えるハミルトニアン
4. 並列化
5. アルゴリズム紹介

HΦとは

量子格子模型の数値厳密対角化法は、量子多体問題、とくに
強相関電子系の数値的研究を行う際の最も基本的な手法です。
西森秀稔教授(東京工業大学) が開発された量子スピン模型に
対する先駆的な数値対角化パッケージTITPACK[1]は、その公開
以来20年以上にわたって幅広いユーザーに利用されてきました。
　HΦはTITPACK に代わる並列計算機対応数値対角化パッケージ
を目指して開発されました。遍歴電子系を含む幅広い量子格子模型
に柔軟に適用でき、さらに高並列に対応するソフトウェアです。
2015/2016/2017年度東大物性研ソフトウェア開発・高度化支援[2]
を受け開発を進めています。

[1] http://www.stat.phys.titech.ac.jp/~nishimori/titpack2_new/index-e.html
[2] http://www.issp.u-tokyo.ac.jp/supercom/rsayh2/softwea-dev

定量物性予測から
オープンソースソフトウェアへ

量子格子模型ソルバーHΦ

実験と理論の直接比較を目指して

-広汎な多体量子系に対応
第一原理有効ハミルトニアン
-絶対零度[1,2]/有限温度[3]/有限周波数[1]の物理量計算
磁化, 比熱, 非弾性中性子/X線散乱スペクトル, …
-非平衡ダイナミクス
-大規模並列計算機対応
数理的基礎: 応用数理手法による情報圧縮

オープンソースソフトウェア (latest release: ver.3.5.0)
License: GNU GPL version3

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

[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).

HΦ開発チーム

三澤貴宏博士
北京量子信息科学研究院

藤堂眞治教授
東京大学大学院
理学系研究科物理学専攻

井戸康太博士
東京大学
物性研究所

河村光晶博士
東京大学
物性研究所

川島直輝教授
東京大学
物性研究所

本山裕一博士
東京大学
物性研究所

吉見一慶博士
東京大学
物性研究所

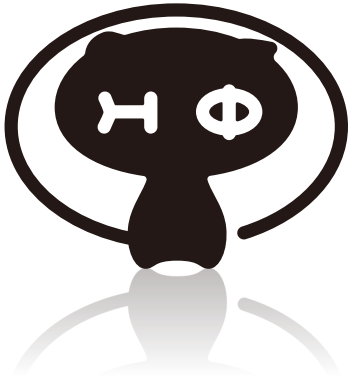
代表: 山地洋平
物質・材料研究機構

謝辞: 中野義典博士 (同志社大学)
星健夫准教授 (鳥取大学)
曽我部智広准教授 (名古屋大学)

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

T. Hoshi, M. Kawamura, K. Yoshimi, Y. Motoyama, T. Misawa, Y. Yamaji, S. Todo, N. Kawashima, and T. Sogabe, "Kω—Open-source library for the shifted Krylov subspace method," CPC 2021

HΦの普及状況



・全国10のスパコンにプリインストール

北海道大学 HOKKAIDO UNIVERSITY
JCAHPC
東京大学 物性研究所 THE INSTITUTE FOR SOLID STATE PHYSICS THE UNIVERSITY OF TOKYO
東北大学
東京大学 THE UNIVERSITY OF TOKYO
名古屋大学
東京工業大学
京都大学 KYOTO UNIVERSITY
大阪大学 OSAKA UNIVERSITY
九州大学 KYUSHU UNIVERSITY

・通算700ダウンロード以上
・HΦが使われた研究論文(2020年)

-H. Zhang, *et al.*, Phys. Rev. RESEARCH 2, 013214 (2020).
-H. Araki, *et al.*, Phys. Rev. RESEARCH 2, 012009(R) (2020).
-J. Yoshitake, *et al.*, Phys. Rev. B 101, 100408(R) (2020).
-P. Laurell and S. Okamoto, npj Quantum Mat. 5, 2 (2020).
-R. Martinazzoa and E. Pollakc, PNAS 117, 16181 (2020).
-Y. Nomura, arXiv:2009.14777.
-Y. Nomura and M. Imada, arXiv:2005.14142.
-T. Nakamura, Sci. Rep. 10, 14201 (2020).
-A. S. Patri, et al., Phys. Rev. RESEARCH 2, 023253 (2020).
-M. Charlebois and M. Imada, Phys. Rev. X 10, 041023 (2020).
-N. Aiba and K. Nomura, Phys. Rev. B 102, 134435 (2020).
-S. H. Jang, et al., Phys. Rev. Mat. 4, 104420 (2020).
-H. Kobayashi, et al., arXiv:2011.04303.
-G. Sala, *et al.*, arXiv:2003.01754.
-T. Yamada, T. Suzuki, and S. Suga, arXiv:2004.09622.

数値対角化が用いられてきた重要な例

分数量子ホール効果

分数量子ホール効果
D. C. Tsui, *et al.*, Phys. Rev. Lett. 48, 1559 (1982).
R. B. Laughlin, Phys. Rev. Lett. 50, 1395 (1983).

F. D. M. Haldane, Phys. Rev. Lett. 55, 2095 (1985).

ラフリンの波動関数の精度を検証

孤立量子系の熱化

M. Rigol, V. Dunjko, and M. Olshanii, Nature 452, 854 (2008).

孤立量子系の時間発展を長時間追っていった結果

厳密対角化に基づく量子格子模型ソルバー

TITPACK by Y. Taguchi & H. Nishimori (1985-)
Heisenberg & XXZ model

KOBEPACK by M. Kaburagi, T. Nishino, & T. Tonegawa (1992-)
-S=1 Heisenberg

SPINPACK by J. Schulenburg (1995-)
-MPI & PTHREAD
-XXZ, Hubbard, & t-J model
-Symmetries

ALPS lETL library by P. Dayal, M. Troyer, & R. Villiger

主に量子統計力学の標準的な模型
(Heisenberg, Hubbard)向け
cf.) Quspin, Quanty (不純物向け)

How Useful in Condensed Matter Physics

Exact diagonalization/full CI

-Lanczos/Arnoldi or CG-type algorithm
-Applicable to finite-size clusters
N (< 30) orbital systems
N (< 50) *spin* systems

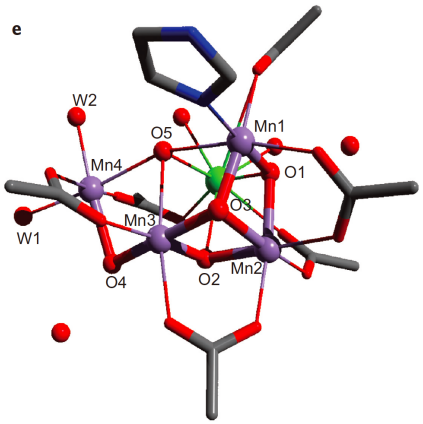
DMRG in chemistry

-A H_2O molecule: 5↑ & 5↓ electrons in 41 orbitals
→ 5.6×10^{11} dimensional

G. K.-L. Chan & M. Head-Gordon, J. Chem. Phys. 118, 8551 (2003).

-Manganese cluster in photosystem II:
44 electrons in 35 orbitals
→ 2×10^{8} dimensional

Y. Kurashige, G. K.-L. Ghan, & T. Yanai, Nat. Chem. 5, 660 (2013).



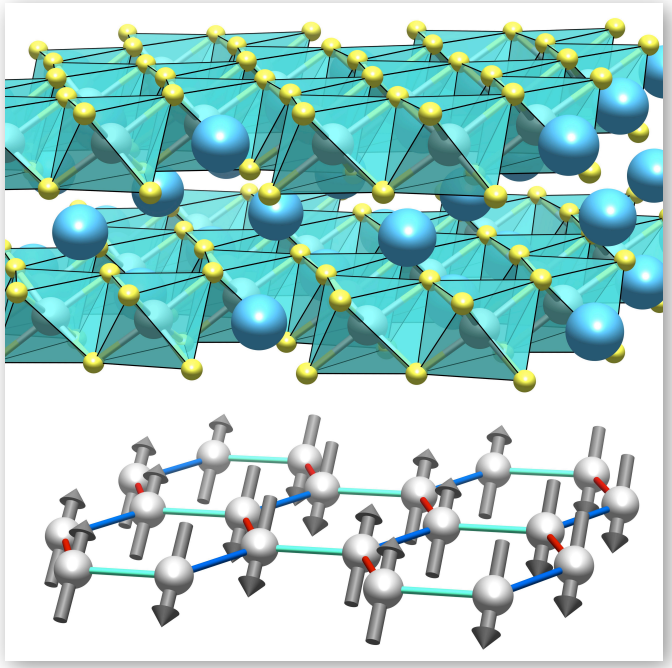
How Useful in Condensed Matter Physics

Exact diagonalization/full CI

-Lanczos/Arnoldi or CG-type algorithm
-Applicable to finite-size clusters
 N (< 30) orbital systems
 N (< 50) *spin* systems

Application to crystalline lattice (with periodic boundary)

-Emergent phenomena
 beyond perturbation theory
-Finite-temperature thermodynamic
 quantities and spectra



cf.) Nearsightedness
W. Kohn, Phys. Rev. Lett. 76, 3168 (1996).

lower temperature, larger size

Basis of Hamiltonians in HΦ: Spin-1/2 Fermions

Hamiltonian in 2nd quantization form

$$\hat{H} = \hat{H}_K + \hat{H}_I$$

$$\hat{H}_K = -\sum_{\ell_1,\ell_2}\sum_{\sigma_1,\sigma_2} t^{\sigma_1\sigma_2}_{\ell_1\ell_2}\hat{c}^\dagger_{\ell_1\sigma_1}\hat{c}_{\ell_2\sigma_2}$$

$$\hat{H}_I = \sum_{\ell_1,\ell_2,\ell_3,\ell_4}\sum_{\sigma_1,\sigma_2,\sigma_3,\sigma_4} I^{\sigma_1\sigma_2\sigma_3\sigma_4}_{\ell_1\ell_2\ell_3\ell_4}\hat{c}^\dagger_{\ell_1\sigma_1}\hat{c}_{\ell_2\sigma_2}\hat{c}^\dagger_{\ell_3\sigma_3}\hat{c}_{\ell_4\sigma_4}$$

Numerically exact eigenvalues and eigenvectors

$$\hat{H}|\phi\rangle = E|\phi\rangle$$

$$|\phi\rangle = \sum_{\{I_j,\overline{I}_j\}} C_{I_0\overline{I}_0I_1\overline{I}_1\cdots I_{N-1}\overline{I}_{N-1}} \left[\prod_{j=0}^{N-1}(\hat{c}^\dagger_{\ell_j\uparrow})^{I_j}(\hat{c}^\dagger_{\ell_j\downarrow})^{\overline{I}_j}\right]|0\rangle$$

CI coefficients

Pauli principle: $I_j, \overline{I}_j \in \{0,1\}$

Variety of Hamiltonian

Periodic (Standard or Expert)
-Hubbard model
-S=1/2 Heisenberg model with anisotropy
-Kondo lattice model
-S=1, 3/2, 2, … Heisenberg model

(Standard or Expert)
-*Ab initio* hamiltonian

Apriodic (Expert)
-Molecules/qubits

Example of Hamiltonian: 1D Hubbard model

$$\hat{H} = -t \sum_{i=0}^{L-1} \sum_{\sigma=\uparrow,\downarrow} \left[\hat{c}^{\dagger}_{i\sigma} \hat{c}_{\mathrm{mod}(i+1,L)\sigma} + \hat{c}^{\dagger}_{\mathrm{mod}(i+1,L)\sigma} \hat{c}_{i\sigma} \right]$$

$$+ U \sum_{i=0}^{L-1} \hat{c}^{\dagger}_{i\uparrow} \hat{c}_{i\uparrow} \hat{c}^{\dagger}_{i\downarrow} \hat{c}_{i\downarrow}$$

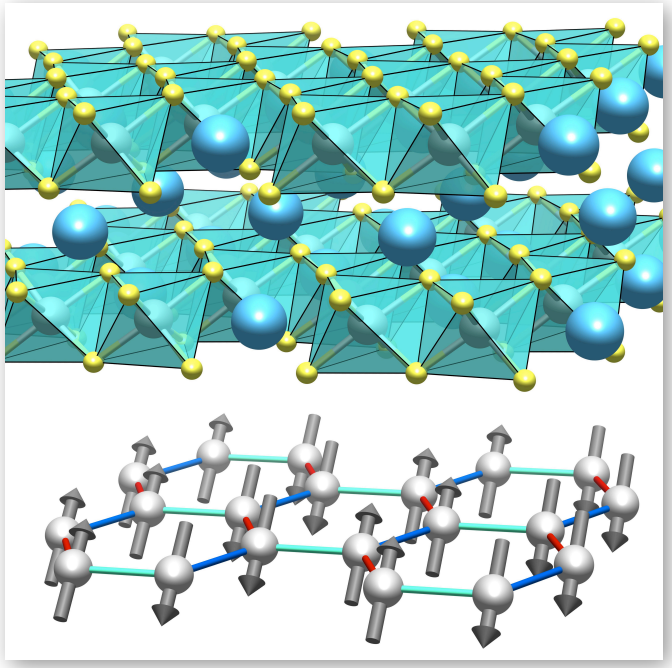
Standard input

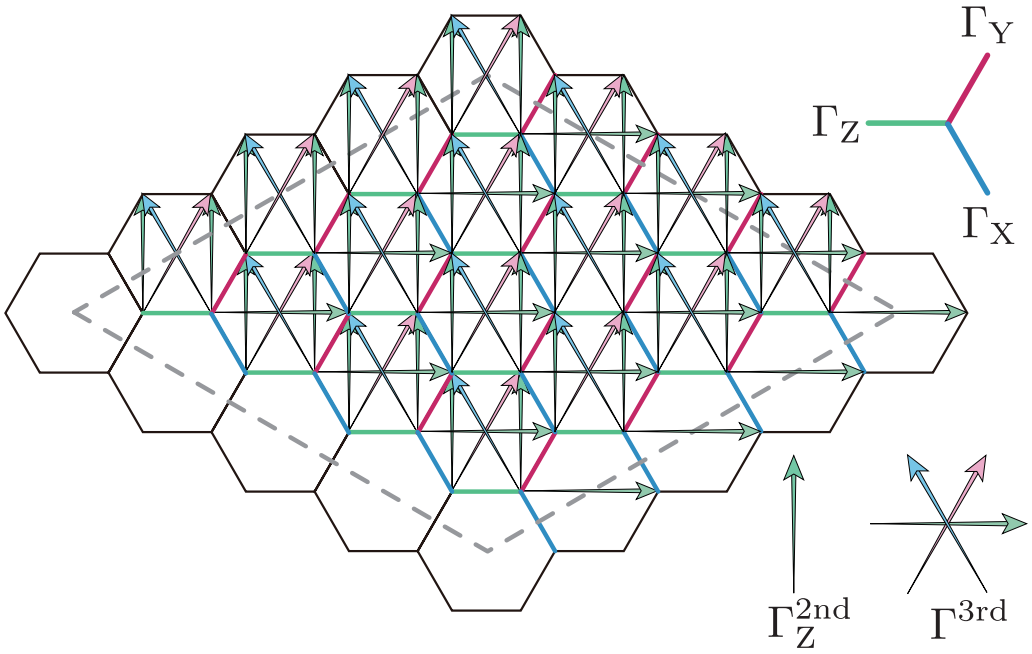
```
model = "Hubbard"
method = "CG"
lattice = "chain"
L = 8
t = 1.0
U = 8.0
nelec = 8
2Sz = 0
exct = 1
```

An Example of Complicated Hamiltonian: *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 Complicated Hamiltonian: *Ab Initio* Spin Hamiltonian

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

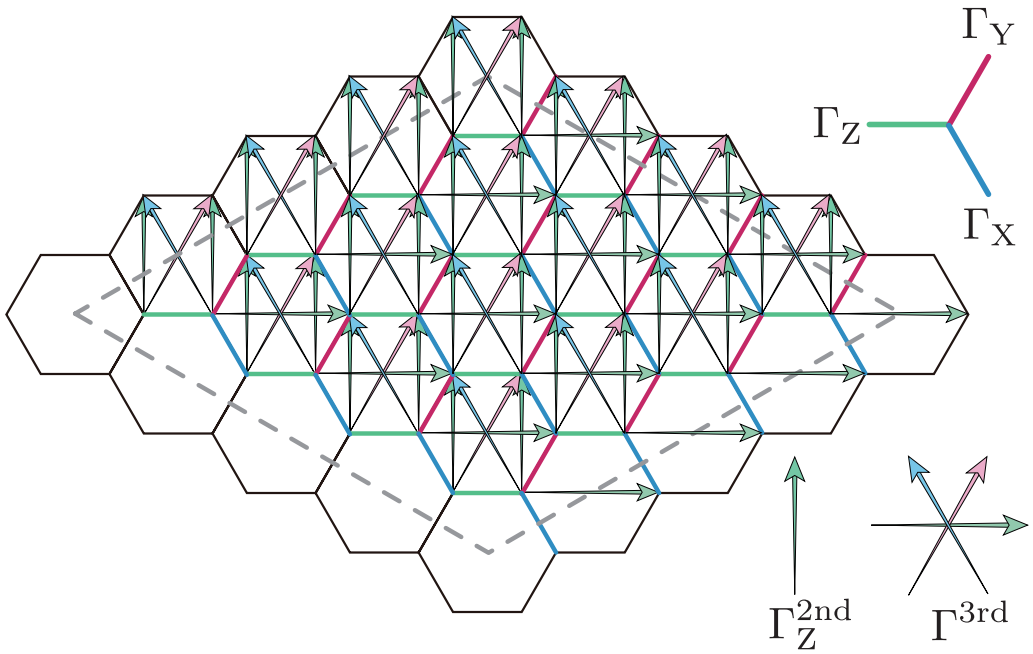
$$\hat{H} = \sum_{\Gamma=X,Y,Z,Z_{\rm 2nd},3} \sum_{\langle \ell,m\rangle \in \Gamma} \vec{\hat{S}}_\ell^T \mathcal{J}_\Gamma \vec{\hat{S}}_m$$

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

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

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

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



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

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

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 useful

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

↓

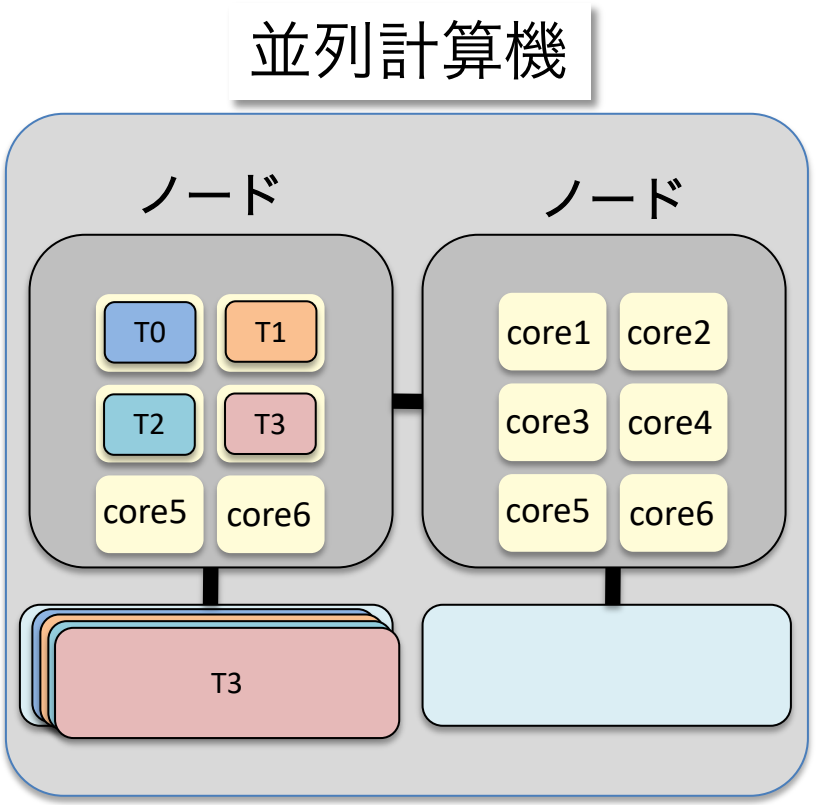
Subroutines:
-Lanczos
-CG
-TPQ
-TimeEvolution
-Full diag.
(LAPACK, Scalapack, MAGMA)

→ Standard output
Output files

HΦ Automatically Performs Parallelized Simulation

・Hybrid parallelization

-共有メモリ並列(OpenMP) スレッド
-分散メモリ並列(MPI) プロセス



HΦ Automatically Performs Parallelized Simulation

・Hybrid parallelization

-共有メモリ並列(OpenMP) スレッド
-分散メモリ並列(MPI) プロセス

・Hubbard/Kondo Lattice /HubabrdGC
→4^nプロセス
・Spin/SpinGC
→$(2S+1)^n$プロセス

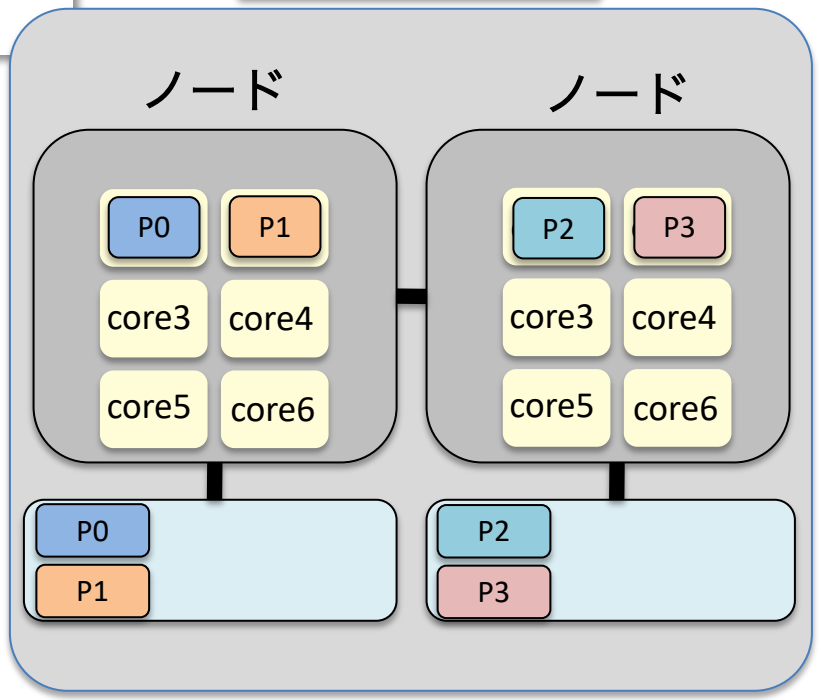
Decomposition of wave function by 2 processes

$$
\begin{aligned}
|0\rangle &= |\downarrow\downarrow\downarrow\rangle \\
|1\rangle &= |\uparrow\downarrow\downarrow\rangle \\
|2\rangle &= |\downarrow\uparrow\downarrow\rangle \\
|3\rangle &= |\uparrow\uparrow\downarrow\rangle \\
\hline
|4\rangle &= |\downarrow\downarrow\uparrow\rangle \\
|5\rangle &= |\uparrow\downarrow\uparrow\rangle \\
|6\rangle &= |\downarrow\uparrow\uparrow\rangle \\
|7\rangle &= |\uparrow\uparrow\uparrow\rangle
\end{aligned}
$$

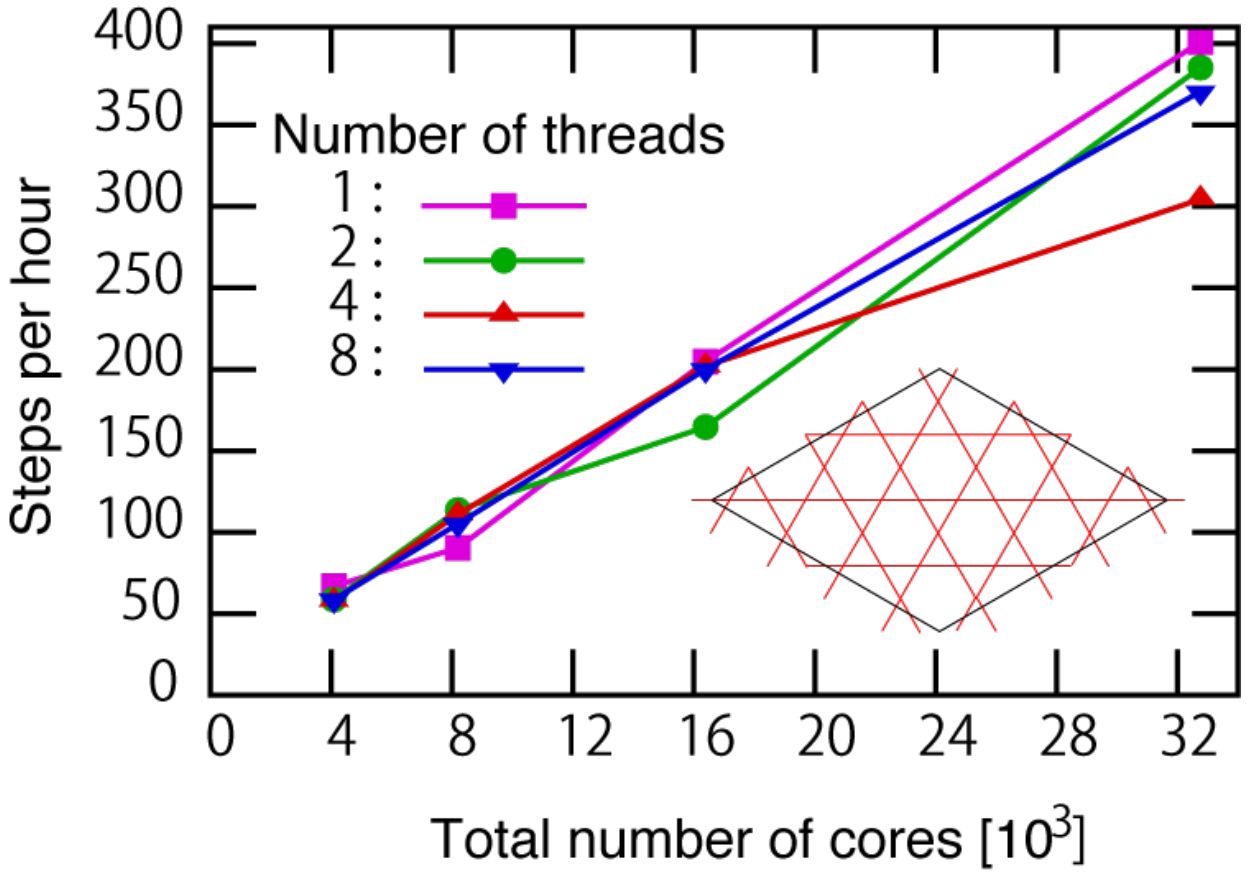
rank 0

rank 1

並列計算機



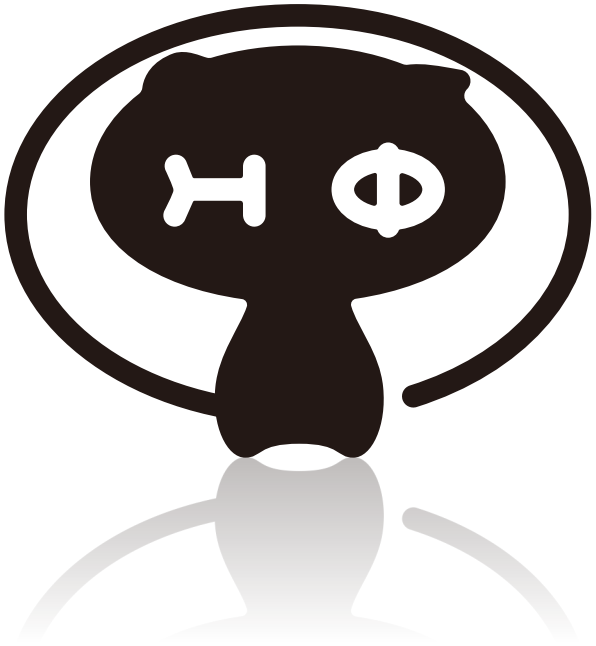
Speedup



Lanczos method: Up to 6.87×10^{10} dimension

@K computer & ISSP supercomputer

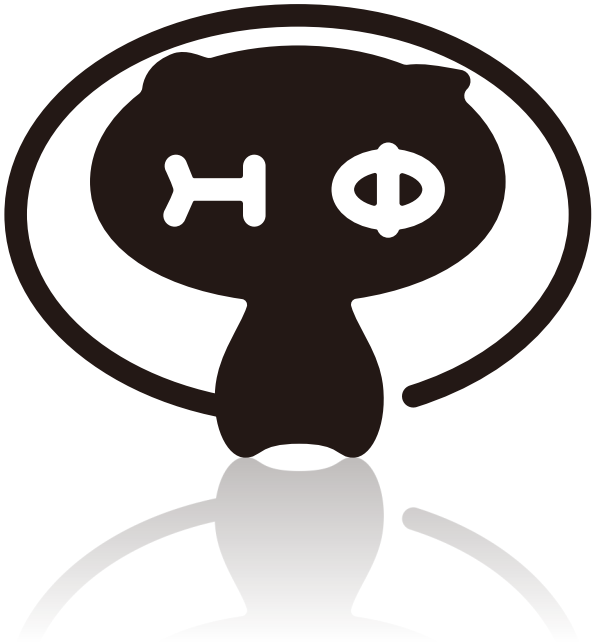
From 4096 32768 cores: Parallelization efficiency 80%



アルゴリズム紹介

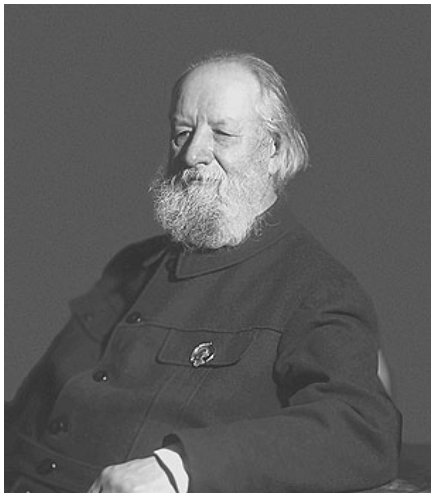
1. Zero temperature
2. Finite temperatures
3. Real-time evolution
4. Dynamical properties (linear response)

https://issp-center-dev.github.io/HPhi/manual/develop/tutorial/en/html/index.html



Algorithm Implemented in HΦ
1. Zero Temperature: 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

An Example of Sparse Matrix: TFIM

$$\hat{H} = J \sum_{i=0}^{L-1} \hat{S}_i^z \hat{S}_{i+1}^z - \Gamma \sum_{i=0}^{L-1} \hat{S}_i^x$$

-Non-commutative

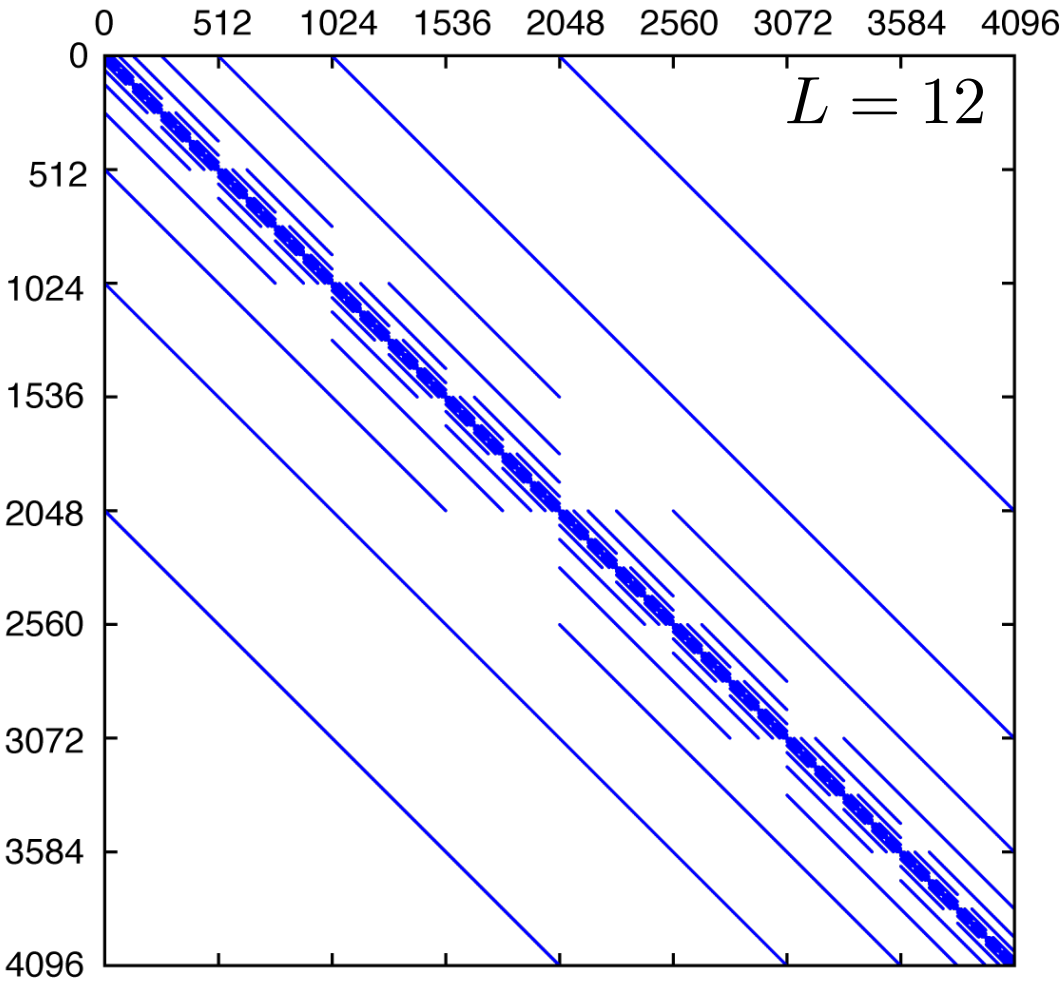
$$\left[\sum_{i=0}^{L-1} \hat{S}_i^z \hat{S}_{i+1}^z, \sum_{i=0}^{L-1} \hat{S}_i^x \right] \neq 0$$

→Quantum fluctuations
or Zero point motion

-Sparse
# of elements ∝ $O(2^L)$

-Solvable

-Hierarchical matrix?



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

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

$$\textbf{for } j = 1, 2, \ldots, m \textbf{ do}$$

$$|w_j\rangle \leftarrow \hat{H}|v_j\rangle - \beta_j |v_{j-1}\rangle$$

$$\alpha_j = \langle w_j | v_j \rangle$$

$$|w_j\rangle \leftarrow |w_j\rangle - \alpha_j |v_j\rangle$$

$$\beta_{j+1} = \sqrt{\langle w_j | w_j \rangle}$$

$$|v_{j+1}\rangle = |w_j\rangle / \beta_{j+1}$$

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

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

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

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

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

注意

Lanczos法

良い点
-実装が簡単
-メモリ・コストが少ない

悪い点
-固有ベクトルが直接求まらない

固有ベクトルはCG法で求められる:

逆反復法 $(\hat{H} - E_m)\vec{v}_{k+1} = \vec{v}_k$

$$\vec{v}_k \to |m\rangle$$

固有値・固有ベクトルを同時に求める方法

Locally optimal block CG (LOBCG):
A. V. Knyazev,
SIAM journal on scientic computing **23**, 517 (2001).

LOB(P)CG

A. V. Knyazev, SIAM journal on scientic computing **23**, 517 (2001).

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 ith 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 ith smallest eigenstate of $S_A \boldsymbol{v}^{(i)} = \mu^{(i)} S_B \boldsymbol{v}^{(i)}$, $\boldsymbol{v}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_m^{(i)}, \beta_1^{(i)}, \ldots, \beta_m^{(i)}, \gamma_1^{(i)}, \ldots, \gamma_m^{(i)})^T$ $(i = 1, \ldots, m)$

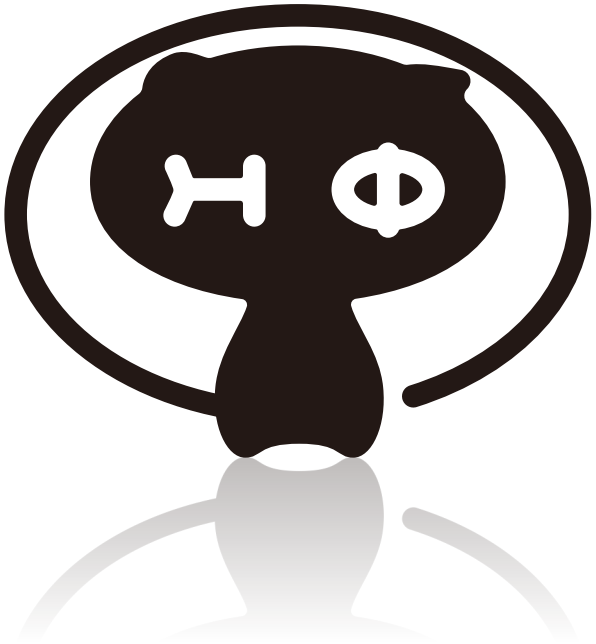
←m smallest eigenstates are chosen from $3m$ states

$$\boldsymbol{x}_{k+1}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} \boldsymbol{w}_k^{(j)} + \beta_j^{(i)} \boldsymbol{x}_k^{(j)} + \gamma_j^{(i)} \boldsymbol{p}_k^{(j)} \right)$$

$$\boldsymbol{p}_{k+1}^{(i)} = \sum_{j=1}^{m} \left(\alpha_j^{(i)} \boldsymbol{w}_k^{(j)} + \gamma_j^{(i)} \boldsymbol{p}_k^{(j)} \right)$$

Until convergence

解説と発展: 山田進, 今村俊幸, 町田昌彦,
日本計算工学会論文集**2006**, 20060027 (2006).



Algorithm Implemented in HΦ
2. Finite Temperatures
TPQ

Finite-Temperature Physical Quantity: Heat Capacity

Spread of energy distribution

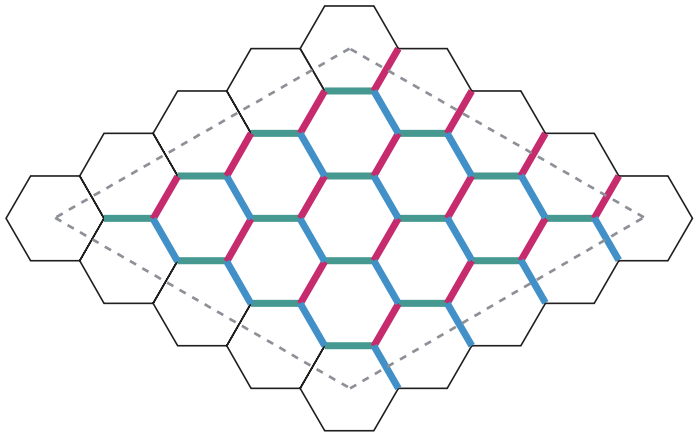
$$C = \frac{\langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2}{k_{\mathrm{B}} T^2}$$

-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 ~3x10^{8}TB!

$|n\rangle$ ~69GB

Typical Pure State Approach

Complexity $\mathcal{O}(N_{\rm H})$

Memory

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

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

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

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

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

Average over the distribution

At finite temperature

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

How large is the standard deviation?

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

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).

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

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

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 $T = +\infty$): $|\Phi_0\rangle = (\text{Random vector})$

If possible, taking random average

do k=1,N_{step}

$$|\Phi_k\rangle = (\ell - \hat{H}/N)\,|\Phi_{k-1}\rangle \,/\sqrt{\langle\Phi_{k-1}|\,(\ell - \hat{H}/N)^2\,|\Phi_{k-1}\rangle}$$

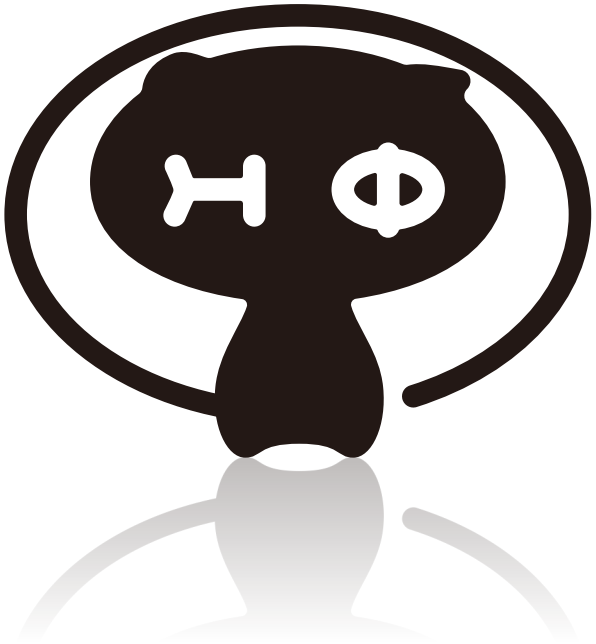
$$u_k = \langle\Phi_k|\,\hat{H}/N\,|\Phi_k\rangle$$

$$\beta = 2(k/N)/(\ell - u_k) \quad (\beta = 1/k_{\mathrm{B}}T)$$

$$\overline{O}(\beta) = \langle\Phi_k|\,\hat{O}\,|\Phi_k\rangle + \mathcal{O}(1/N)$$

enddo

Hamiltonian-wave function product is essential



Algorithm Implemented in HΦ
3. Real Time Evolution
4. Dynamical Properties

3. Real-Time Evolution

Real time evolution by Taylor expansion

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

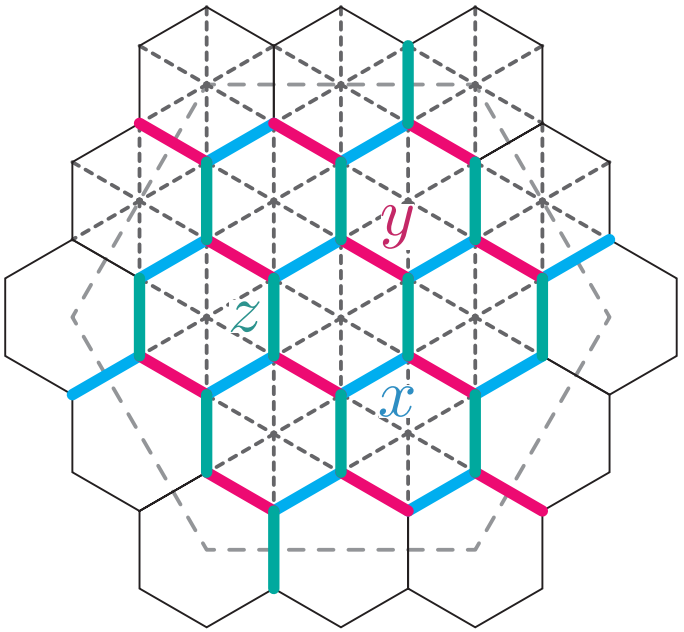
$$|\psi(t_{n+1})\rangle = e^{-i\mathcal{H}(t_n)\Delta t} |\psi(t_n)\rangle$$

Time dependent Hamiltonian
-Standard: Interaction quench or laser (pulse, AC, DC)
-Expert: Time dependent 1-body and 2-body terms

→ Tutoraial Sec. 3

Example of Excitation Spectra: Effective Hamiltonian of α-$RuCl_3$, K-Γ-J_3 Model

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



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

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

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

Nearest neighbor

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

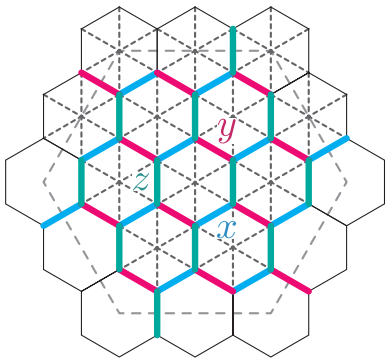
3rd neighbor

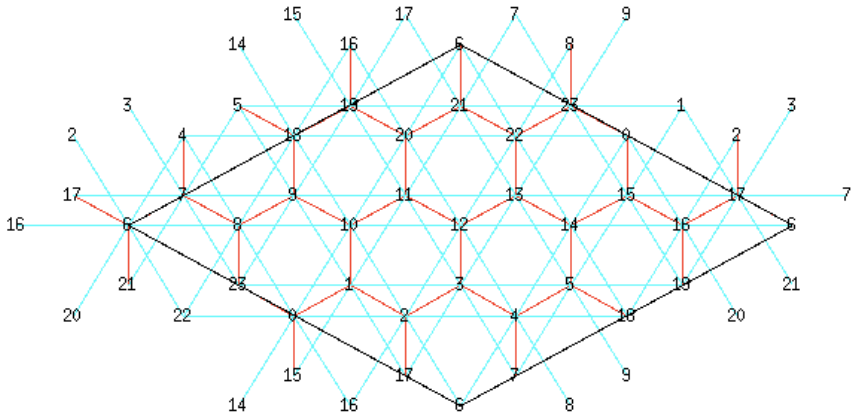
$$J_3[\hat{S}_\ell^x \hat{S}_m^x + \hat{S}_\ell^y \hat{S}_m^y + \hat{S}_\ell^z \hat{S}_m^z]$$

Excitation Spectra: 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
J'' = 0.05
h = 0.07071067811
Gamma = -0.07071067811
2S=1
```





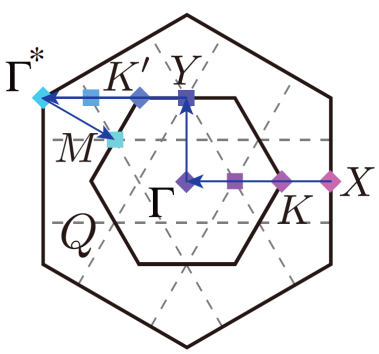
$$\mathcal{J}_X = \begin{bmatrix} -\cos\phi & 0 & 0 \\ 0 & 0 & \sin\phi \\ 0 & \sin\phi & 0 \end{bmatrix}$$

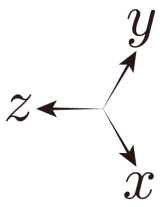
$$\mathcal{J}_Y = \begin{bmatrix} 0 & 0 & \sin\phi \\ 0 & -\cos\phi & 0 \\ \sin\phi & 0 & 0 \end{bmatrix}$$

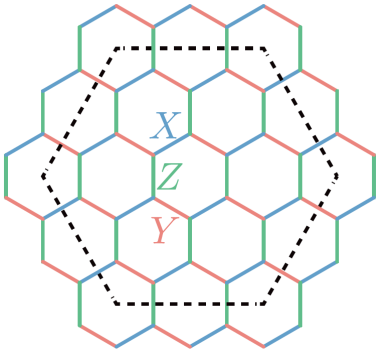
$$\mathcal{J}_Z = \begin{bmatrix} 0 & \sin\phi & 0 \\ \sin\phi & 0 & 0 \\ 0 & 0 & -\cos\phi \end{bmatrix}$$

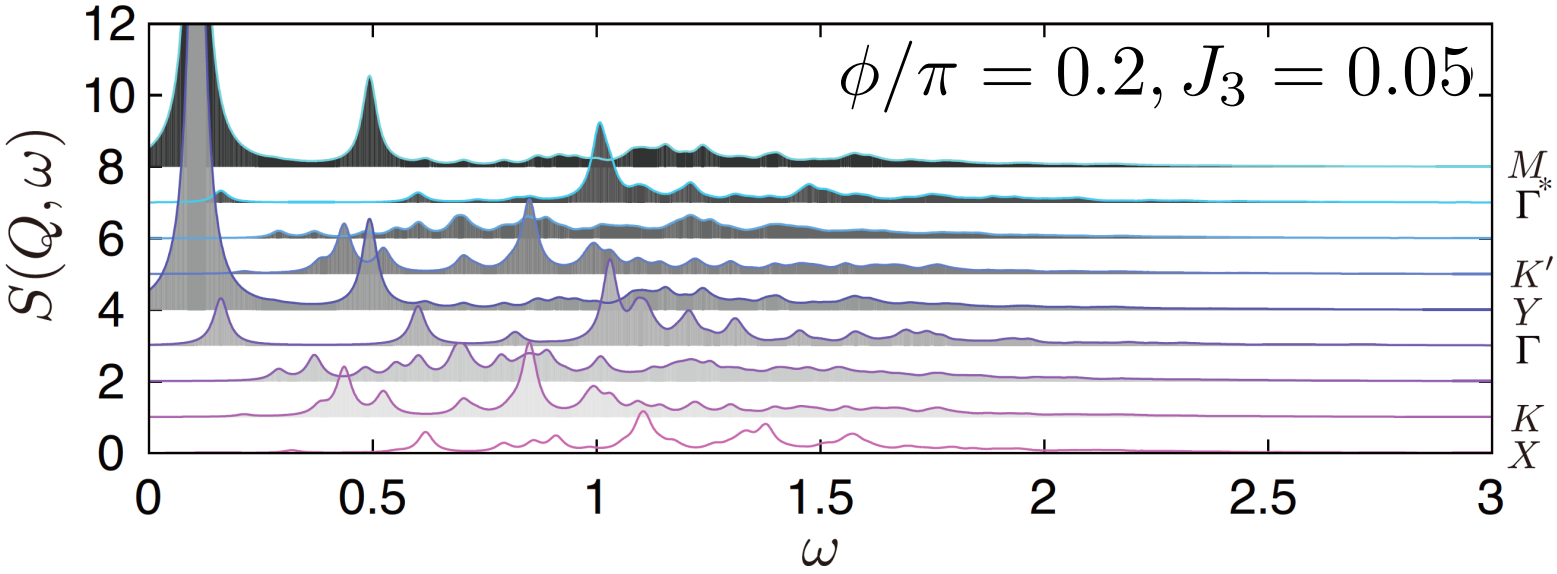
$$\vec{B} \propto (1, 0, -1)$$

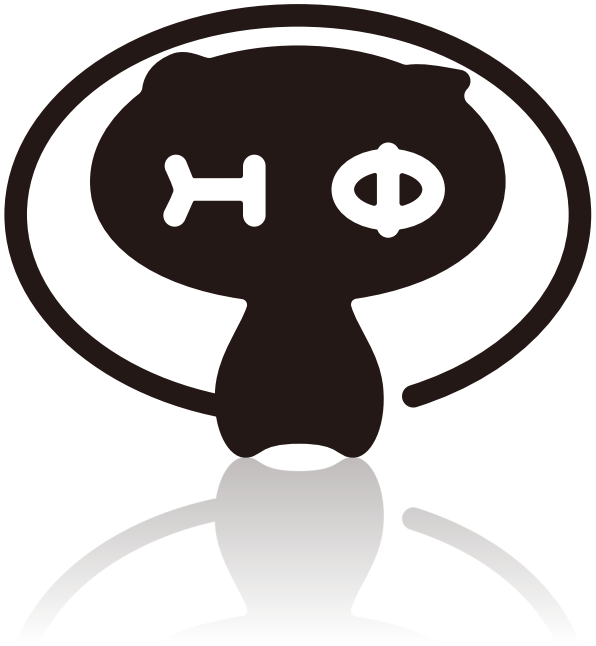
Excitation Spectra: K-Γ-J_3 Model











チュートリアルをお楽しみください!

1. Zero temperature
2. Finite temperatures
3. Real-time evolution
4. Dynamical properties (linear response)

https://issp-center-dev.github.io/HPhi/manual/develop/tutorial/en/html/index.html