INFORMATION TECHNOLOGY CENTER
THE UNIVERSITY OF TOKYO

JCAHPC

$\mathcal{H}$ACApK library
with low-rank structured matrices

Akihiro Ida
(Information Technology Center, University of Tokyo)

THE UNIVERSITY OF TOKYO

Overview of low-rank structured matrices

■Approximation technique for dense matrices from Integral operator.

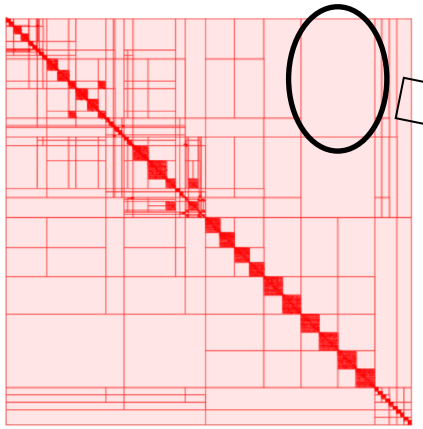
$$g[u](x) = \int_{\Omega} g(x,y)\,u(y)\mathrm{d}y$$

Discretization

$\mathcal{H}$-matrices

Full rank dense matrix

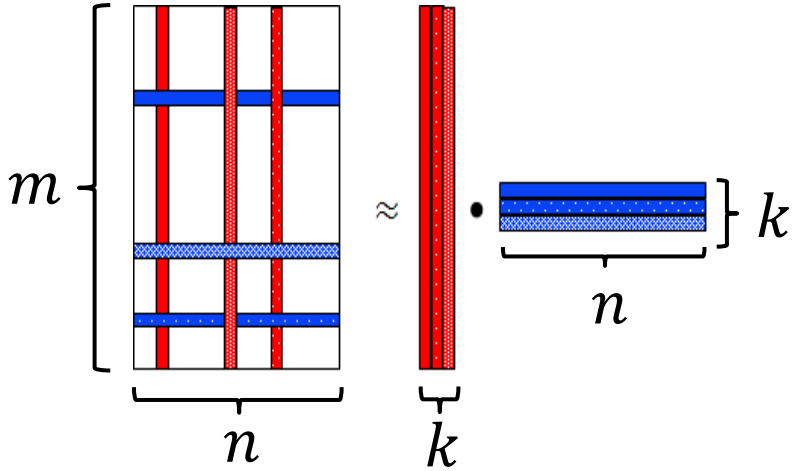
Permutation
Partition



Full-Rank

Low-Rank

Low-rank
approximation



- $O(mn) \Rightarrow O(k(m+n))$
- Accuracy control : # of k

$\mathcal{H}$ACApK library for low-rank structured matrices

■ Library for simulations using BEM(boundary element method)

➢ Application

Integral operator

$$g[u](x) = \int_{\Omega} g(x,y)\,u(y)\mathrm{d}y$$

Degenerate kernel:

$$g(x,y) \cong \sum_{\nu=1}^{r} g_1^{\nu}(x) g_2^{\nu}(y)$$

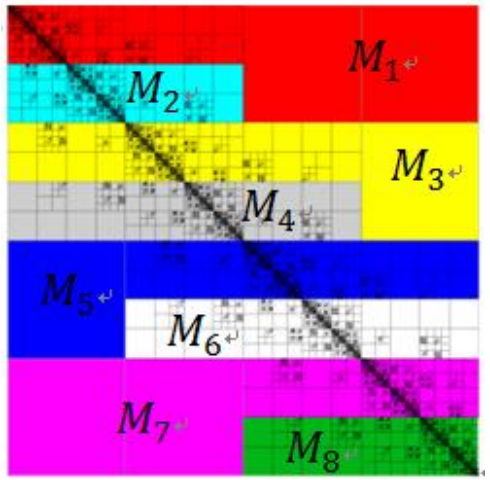
,for far remote $\boldsymbol{x}, \boldsymbol{y}$

e.g. $g(x,y) \propto |x-y|^{-1}$

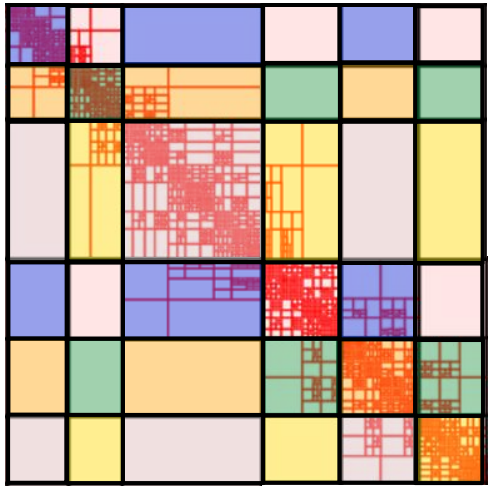
➢ (Lattice) $\mathcal{H}$-matrices : $\boldsymbol{O(N^2) \Rightarrow O(N \log N)}$

➢ Parallel computing

For CPU cluster：MPI+OpenMP (partially GPU)



Assignment for $\mathcal{H}$



Assigmment for lattice $\mathcal{H}$

➢ Download site :

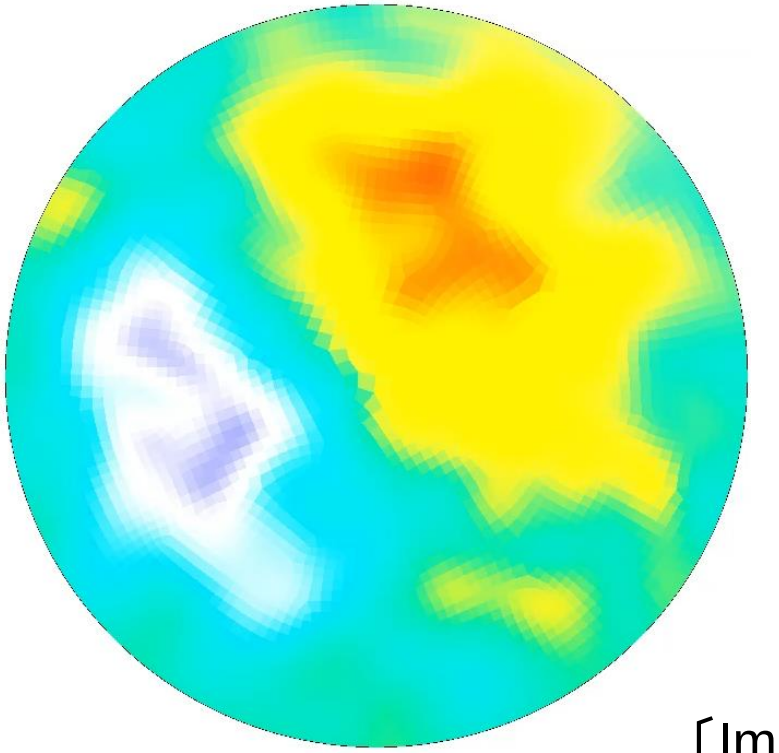
- http://ppopenhpc.cc.u-tokyo.ac.jp
- https://github.com/Post-Peta-Crest/ppOpenHPC/tree/MATH/HACApK

Example analyses using $\mathcal{H}$ACApK

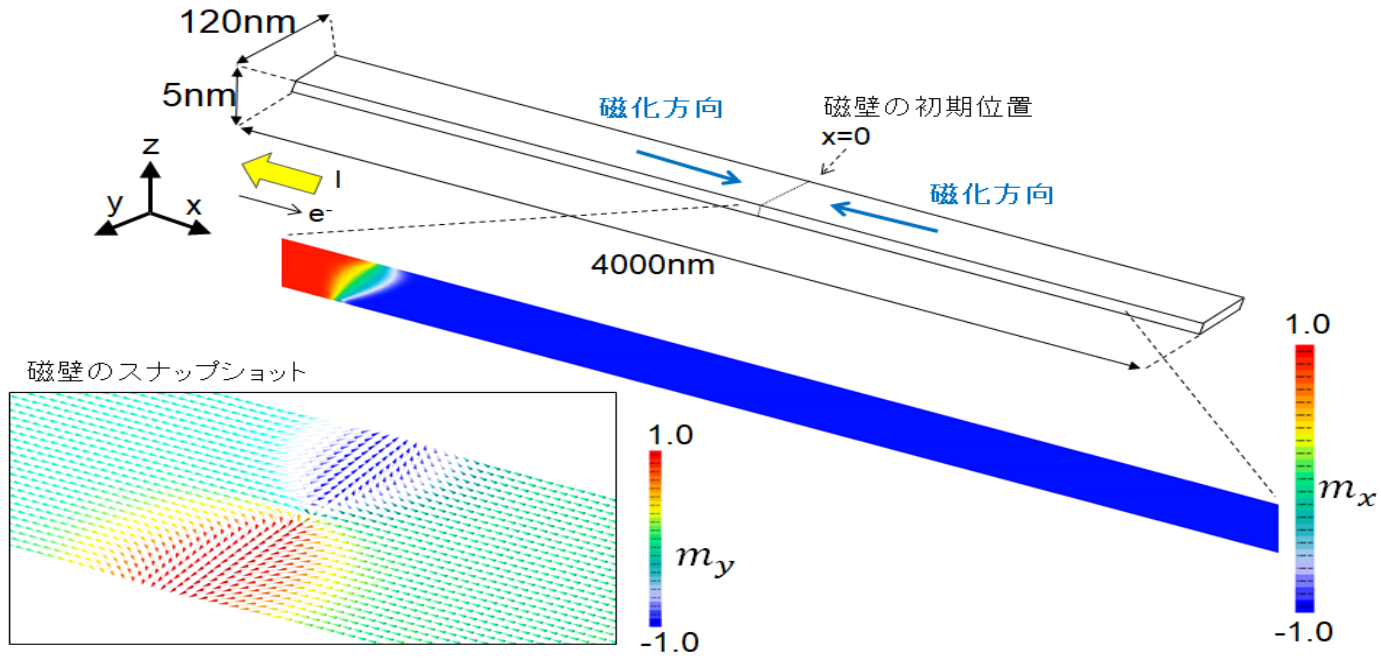
■ Micromagnetic Simulation

$$g(x,y) = \frac{1}{4\pi}\int_{\partial V} \varphi_1(\boldsymbol{y})\nabla|\boldsymbol{x}-\boldsymbol{y}|^{-1}\cdot \mathrm{d}\boldsymbol{S}$$

➢ Spin-torque oscillator



➢ Current-induced domain wall Motion



〔Image courtesy：T. Ataka（Fujitsu Ltd.）〕

Example analyses using $\mathcal{H}$ACApK

■ Earthquake cycle

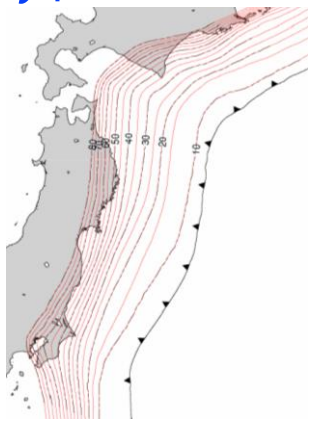
Integral operator with $g(x,y) \propto |x-y|^{-3}$

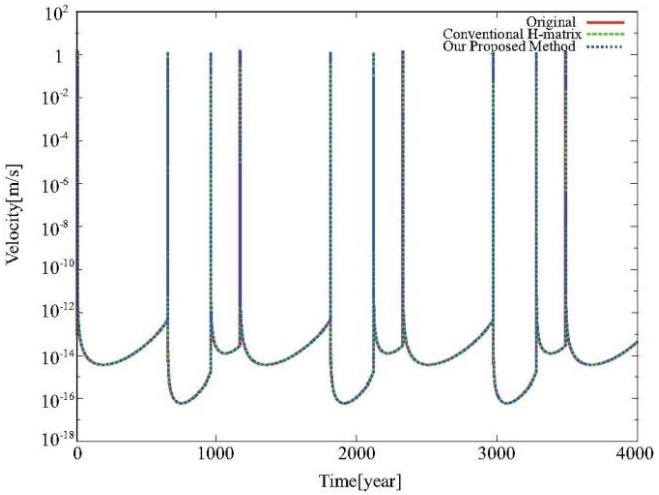
➢eq. motion $\tau_i = \sum_{i=1}^{N} K_{ij}(u_j - V_{pl}t) - \frac{\xi G}{2V_s} V_i$

➢friction law

$$\tau_i = \mu\sigma_n^{eff} = \tau_* + A_i \ln(V_i / V_*) + B_i \ln(V_* \theta_i / L_i)$$

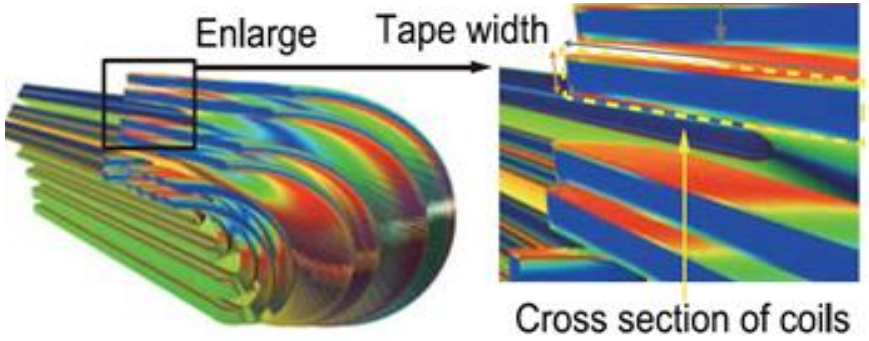
$$\frac{d\theta_i}{dt} = \exp(-V_i / V_c) - V_i \theta_i / L_i \ln(V_i \theta_i / L_i)$$





■ Superconductor Analyses

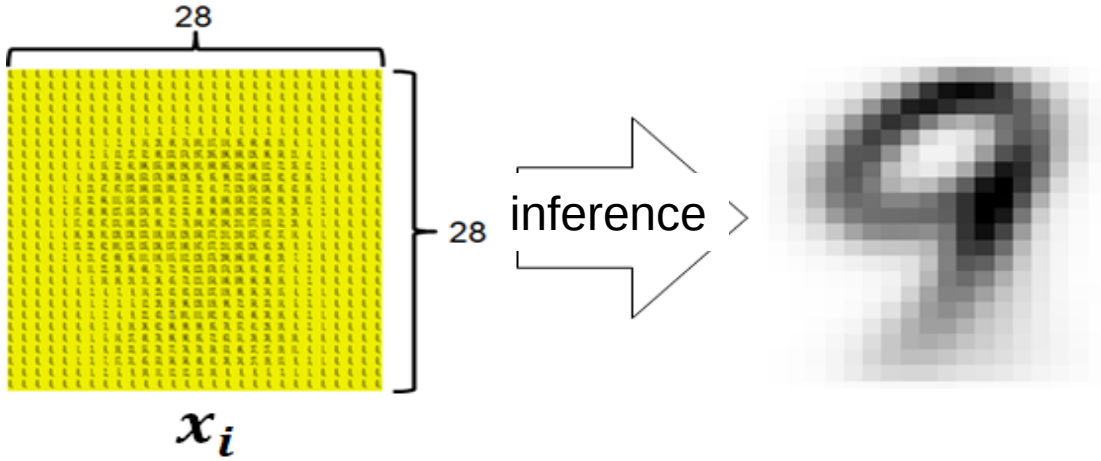
$$g(\mathsf{r}) = \frac{\mu_0 t_s}{4\pi} \int_{S'} \frac{(\nabla \times \boldsymbol{n}' T') \times \boldsymbol{r} \cdot \boldsymbol{n}}{r^3} \mathrm{d}S'$$



〔Image courtesy：Amemiya Lab.（Kyoto Univ.）〕

■ Machine learning

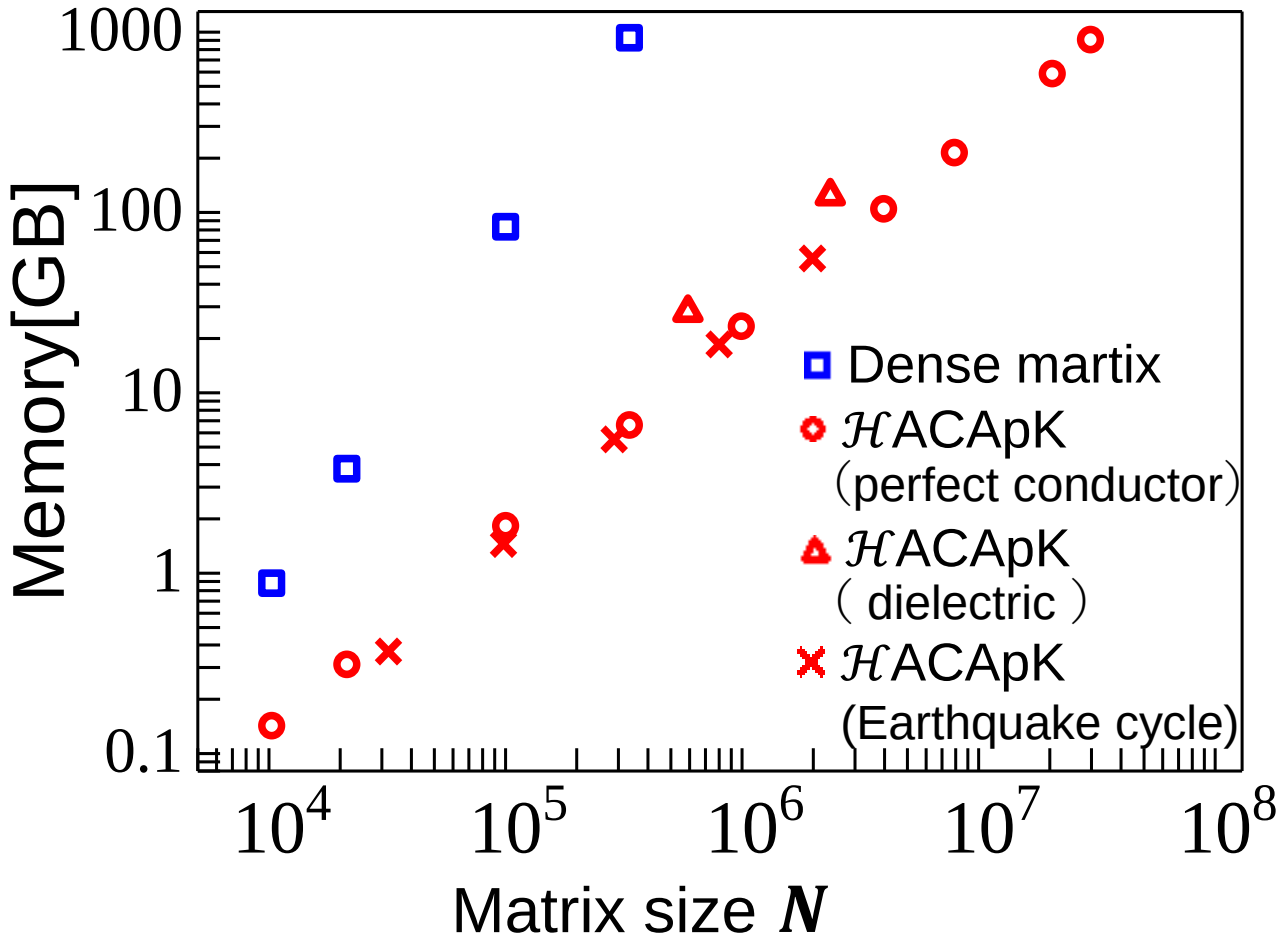
$$A_{ij} := \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 / 2\sigma^2\right)$$



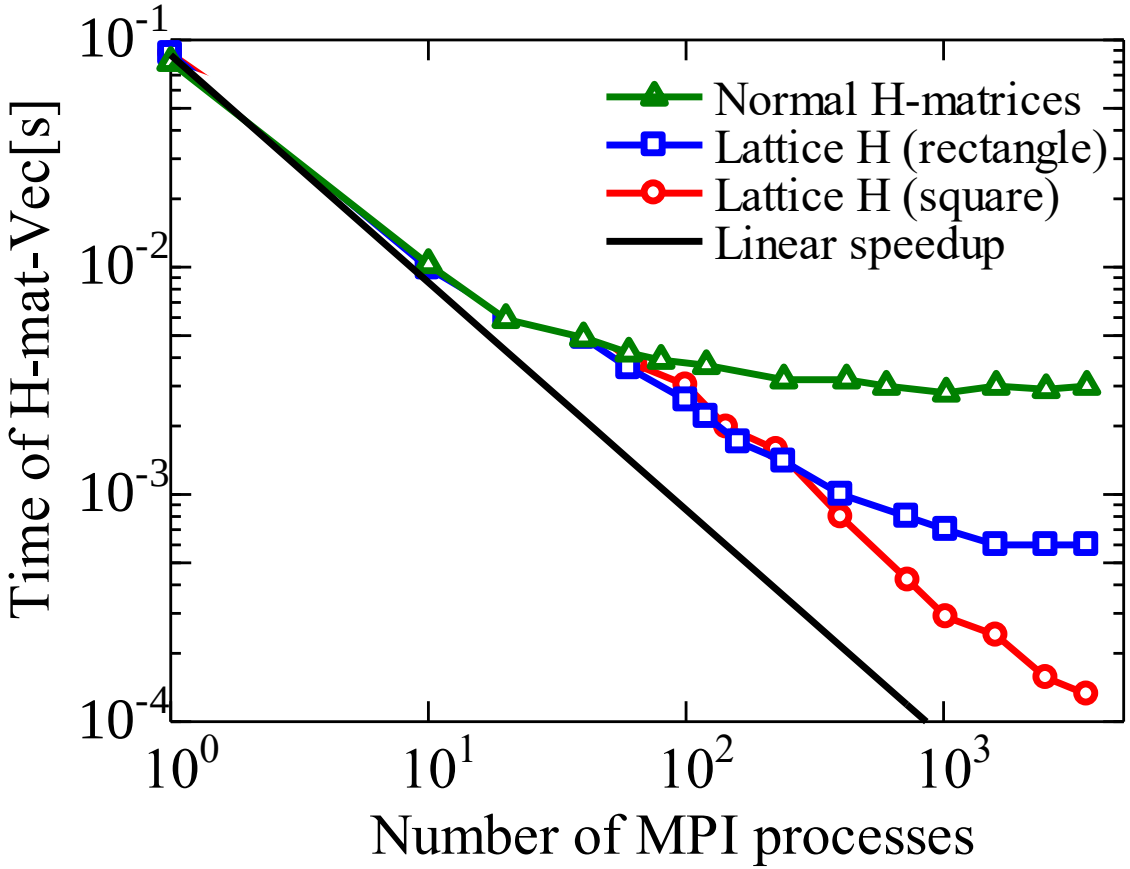
Performance of $\mathcal{H}$ACApK library

■Memory usage

➢ Dense matrix:$\boldsymbol{O(N^2)}$ ➢ $\mathcal{H}$ACApK: $\boldsymbol{O(N\log N)}$



■Parallel scalability



➢ Calculation contidition
- Micromagnetic Simulation： $N = 20{,}910$
- Computer: Fujitsu PRIMERGY CX2550

Conclusion

- $\mathcal{H}$**ACApK**: Library for low-rank structured matrices
- Integral operator to which low-rank structured matrices can be applied
  - ► $g[u] = f$, where $g[u](x) := \int_\Omega \kappa(x,y)\,u(y)\mathrm{d}y$
    - Degenerate kernel: $\kappa(x,y) \cong \sum_{\nu=1}^{r} \kappa_1^\nu(x)\kappa_2^\nu(y)$,for far remote $\boldsymbol{x}, \boldsymbol{y}$
    - Naïve discretization yields dense matrices : $\boldsymbol{O(N^2)}$
- For large-scaled simulations
  - ► Approximation technique for dense matrices
    - (Lattice) $\mathcal{H}$-matrices : $\boldsymbol{O(N^2)} \Rightarrow \boldsymbol{O(N\log N)}$
  - ► Parallel computing
    - Hybrid MPI+OpenMP for CPU clusters (partially ported to GPU)
- Download site : https://github.com/Post-PetaCrest/ppOpenHPC/tree/MATH/HACApK
  - Open source
  - MIT license

Thank you for watching