# Hierarchical Parallelization of Coloring Procedures for ILU Preconditioner



Masatoshi Kawai

Supercomputing Research Division, ITC, Research Associate

SPPEXA Workshop @ Akihabara
Apr/6/2017

# Index

- Backgrounds and objective
- Regularized IC preconditioner
- Parallelization method for IC preconditioner
- Parallelization method for multi-coloring
- Results
- Summary and future works

# Index

- Backgrounds and objective
- Regularized IC preconditioner
- Parallelization method for IC preconditioner
- Parallelization method for multi-coloring
- Results
- Summary and future works

# Target applications 1/2

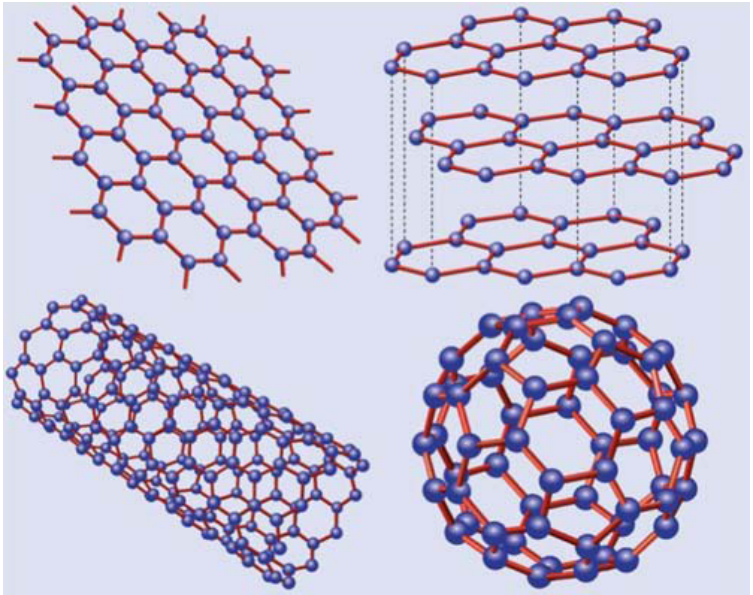
Quantum system simulations are target applications of ESSEX.

Generalized eigenvalue problems provided from the quantum systems.

■ *Graphene*

Simulation of electrical properties.

- Number of DOF is 128～1,000,000
- Number of non-zero elements per row : 13 or 4



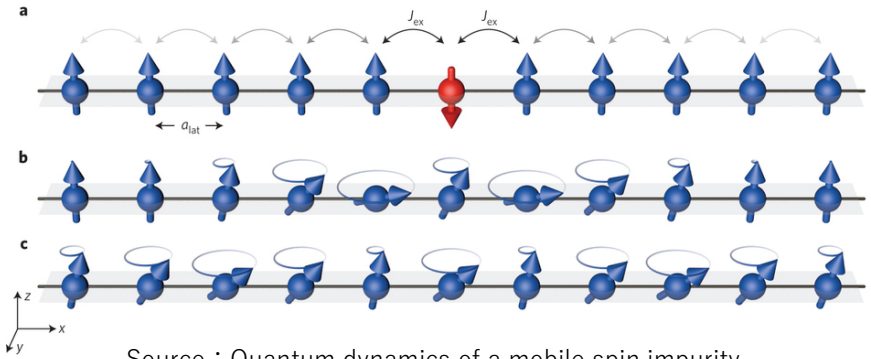
Source : The electronic properties of graphene

# Target applications 2/2

■ *Spinz*

Simulation of considering the XXZ chain and Hamilton operator

- Number of DOF is 252～2,704,156
- Number of non-zero elements per row : 6~12 on average

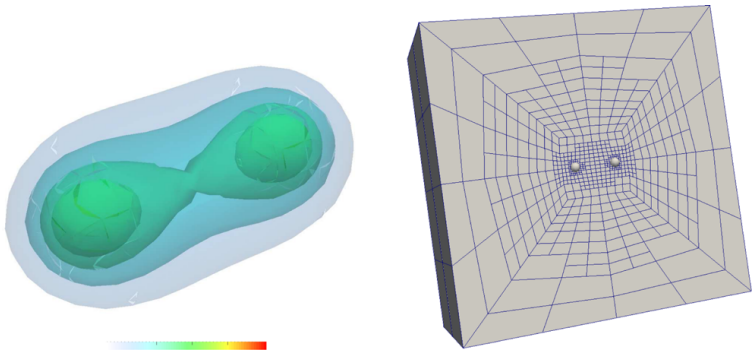



Source : Quantum dynamics of a mobile spin impurity

■ *Kohn-Sham*

Simulation of the status of any molecules.

- Number of DOF is 57,575～76,163
- Number of non-zero elements per row : 20~24 on average



Source : On the adaptive finite element analysis of the Kohn–Sham equations : methods, algorithms, and implementation

# Target problem

## Challenging problem for iterative linear solvers

■**Consider $A_z x = b$ arising when solving $A\tilde{x} = \lambda B\tilde{x}$**

**For FEAST and Sakurai-Sugiura method,**

$A_z := zB - A$

**For Blocked Jacobi-Davidson Method,**

$A_z := \left(I - V \cdot V^{\mathrm{H}}\right)(A - zI)\left(V \cdot V^{\mathrm{H}} - I\right)$

■**Suppose properties of the matrix $A_z$**

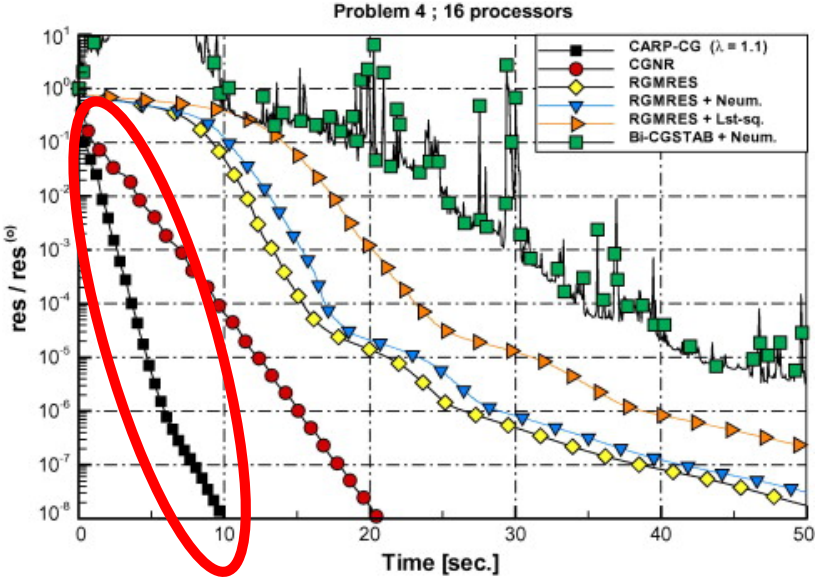
- Indefinite
- Small diagonal entries $O\left(\left|A_{z\_ii}\right|\right) : 10^{-4} \sim 10^{0}$
- Positive and negative diagonal entries
- ill-conditioned: high condition number

# Current study of target equations

**Krylov Subspace methods with special preconditioner have promise.**

■Carp-CG : One of preconditioned Krylov subspace(KS) methods

- Special preconditioner based on Kaczmarz method
- **Carp-CG shows good convergence on graphene model.**




Source: D. Gordon, R. Gordon CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated PDEs, Parallel Computing, Volume 36, Issue 9, September 2010, Pages 495-515

# Objective

■ **To develop special preconditioner of KS method**
- **High convergence ratio**
- **Massive parallelism**

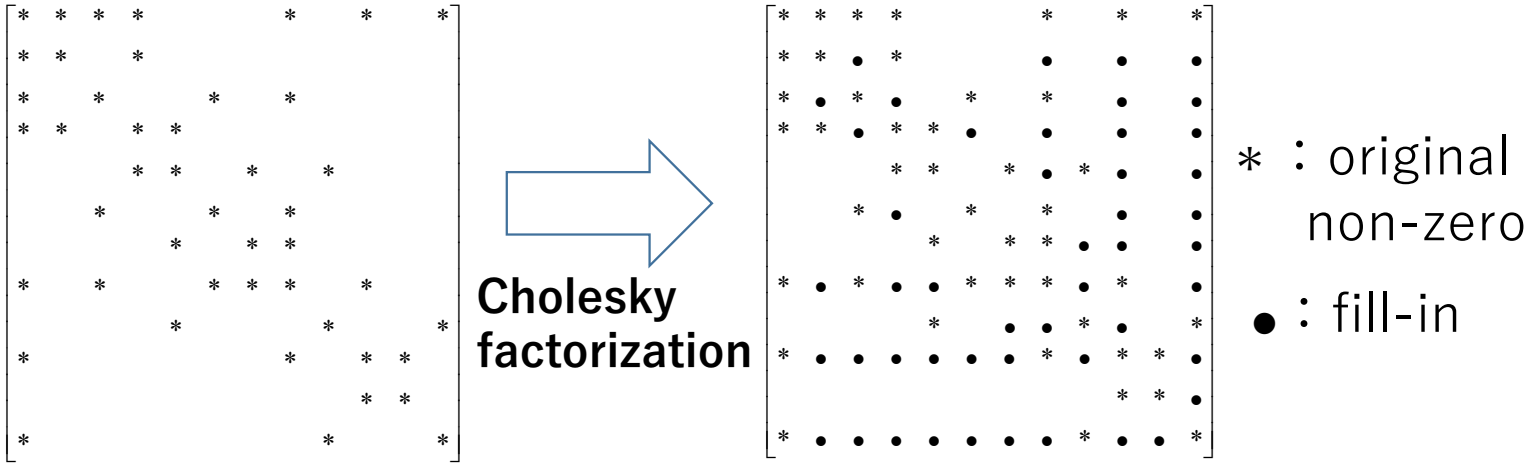
①AMG with Carp-CG smoother
②H-matrices based approximate inverse matrix
③**Regularized IC with hierarchical multi-coloring parallelization**
Convergence ratio : Regularization
Parallelization for IC : Multi-coloring
Parallelization for Multi-coloring: Hierarchical approach (Just idea in this talk)

# Index

- Backgrounds and objective
- **Regularized IC preconditioner**
- Parallelization method for IC preconditioner
- Parallelization method for multi-coloring
- Results
- Summary and future works

# IC(Incomplete Cholesky) preconditioning

- IC preconditioning matrix is constructed through the incomplete version of Cholesky factorization $A = LL^{\mathrm{T}}$.
- “Incomplete” means that a part of fill-ins are ignored in the factorization process.
- On ill-conditioned problem, it increases calculation error.



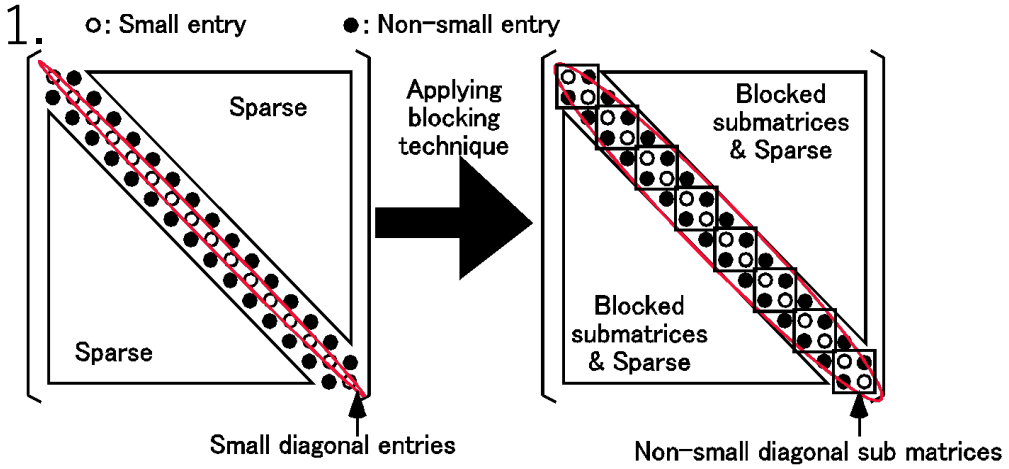

$A_z$ $L + L^{\mathrm{T}}$

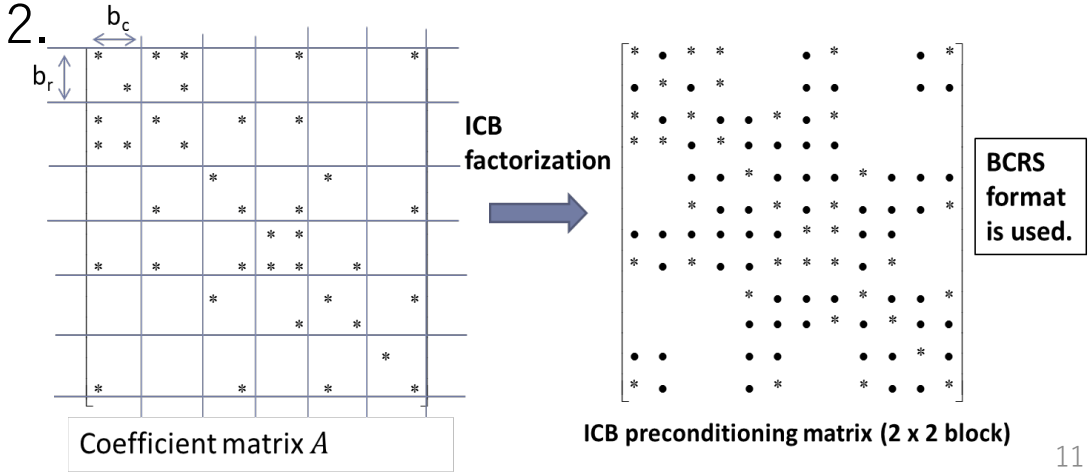
# Regularized IC preconditioner. (1/2)

Applying 2 regularization methods for IC preconditioner
→For robustness and improving convergence

■ Blocking technique

- Applying the incomplete decomposition to a block matrix
  1. More robustness because of including non-smalloff-diagonals
  2. Better convergence ratio because of allowing more fill-ins



# Regularized IC preconditioner. (2/2)

Applying 2 regularization methods for IC preconditioner
→For robustness and improving convergence

■ Blocking technique(Regularization①)
- Applying the incomplete decomposition to a block matrix
  1. More robustness because of including non-small off-diagonals
  2. Better convergence ratio because of allowing more fill-ins

■ Diagonal transformation(Regularization②)
- Adding constant value $\alpha$ to the diagonal elements
- Directly method to make the diagonally dominant matrix

$$\widetilde{A_z} = A_z + \alpha I \quad I = \text{identity matrix}$$

# Index

- Backgrounds and objective
- Regularized IC preconditioner
- **Parallelization method for IC preconditioner**
- Parallelization method for multi-coloring
- Results
- Summary and future works

# Parallelization of ICCG

Focus on the IC preconditioner part to parallelize the ICCG

- Dominant part in the whole computational time
- Hard to parallelize

Flow of the preconditioned CG

Block ILU decomposition

do itr = 1, DoF

dot_product

$q = \tilde{A}^{-1} r$

Matrix Vector Multiplication

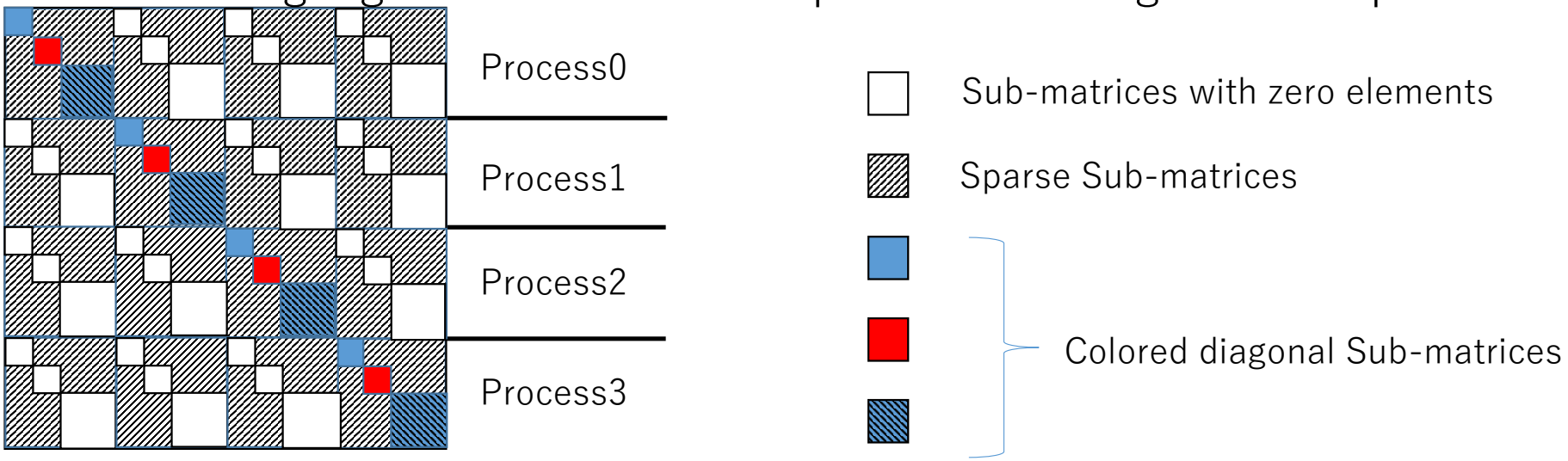
enddo

Hard part to prallelize

# Multi-coloring method

**Applying the multi-coloring method for massively parallelization**

- The elements colored same color has any relationship
  → Calculate them in parallel
- There are many coloring algorithm (Greedy, Algebraic MC……)
  →Multi coloring algorithm has much impact for convergence and performance




Example of 4 processes and 3 colors

# Algebraic multi-coloring

## Applying algebraic multi-coloring(AMC) method

Sample code of AMC

```
ncolor=some value ! Set number of used colors
color(1:n)=0 ! Initialize the array color
icolor=1
do i=1,n
    j=1
    do while(j <= lnz(i))
        if (color(lnzc(i, j))==icolor) then
            icolor=mod(icolor, ncolor)+1 !To next color
            j=0
        endif
        j=j+1
    enddo
    color(i)=icolor ! Assignment of color
    icolor=mod(icolor, ncolor)+1 !To next color
enddo
```

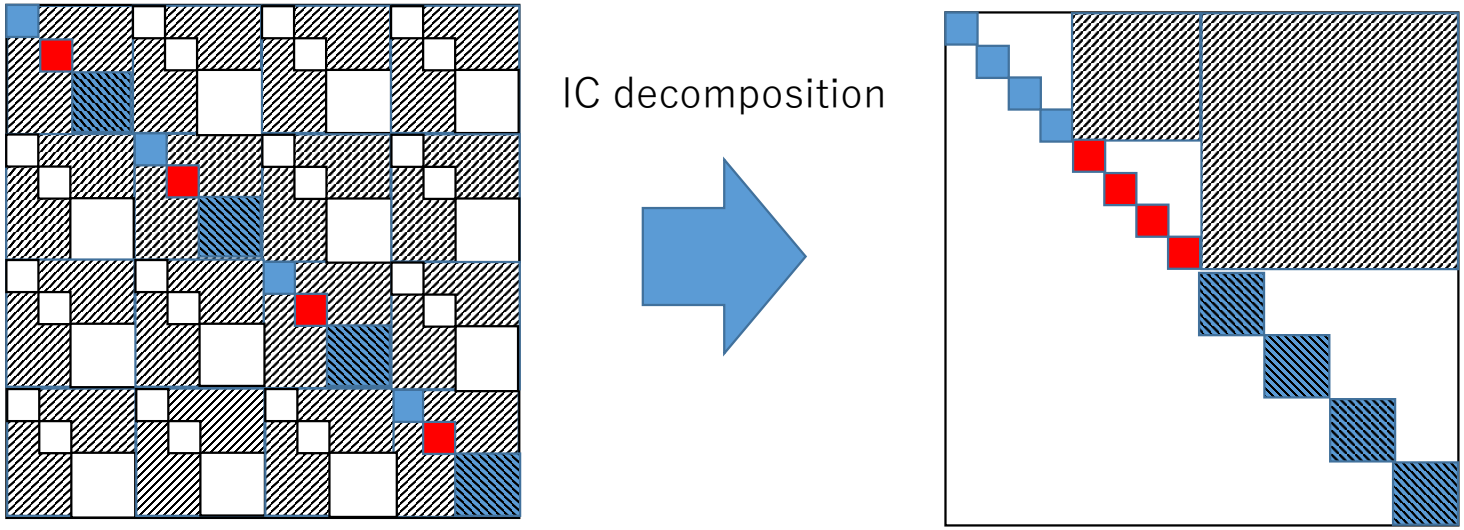
- We can control the number of colors.
  → The convergence ratio and computation time have dependency on the number of colors.
- The number of unknowns are nearly even in each colors, relatively.
  → Load is evenly distributed.

Ref: T. Iwashita. et al. “Algebraic Multi-Color Ordering Method for Parallelized ICCG Solver in Unstructured Finite Element Analyses”

# IC decomposition with multi-coloring

The decomposition must be calculated in the coloring order.

→Forward and backward substitution is calculated in the coloring order.



# Index

- Backgrounds and objective
- Regularized IC preconditioner
- Parallelization method for IC preconditioner
- **Parallelization method for multi-coloring**
- Results
- Summary and future works

# Parallelization of multi-coloring

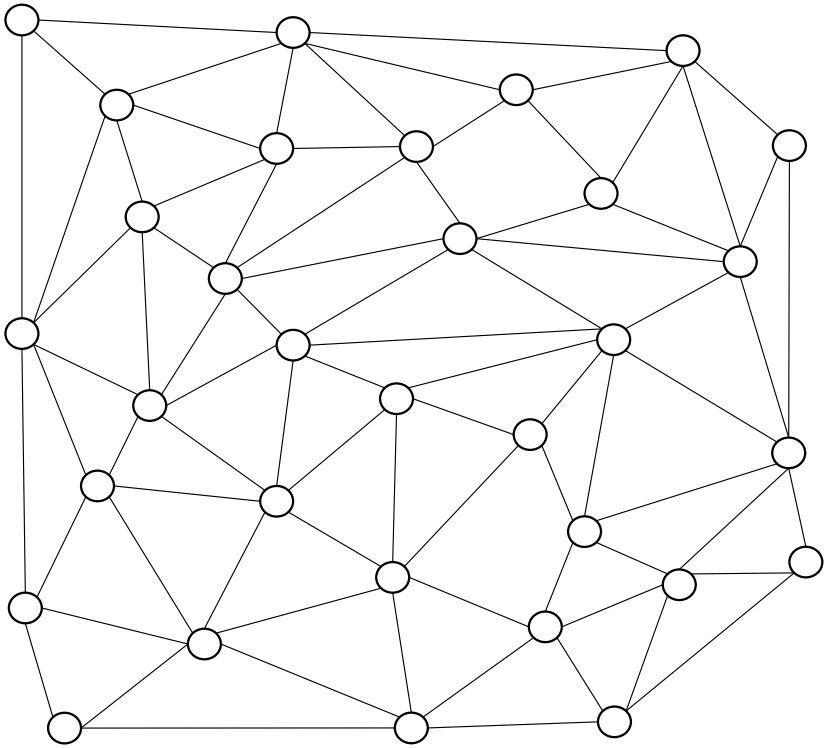
Propose hierarchical parallelization method for multi-coloring algorithms.

- General coloring algorithm is sequential.
  → Propose special parallelization for each process?→No
- Coloring algorithm has much impact for convergence and performance
  → Versatile algorithm is better

On the hierarchical approach, we can parallelize almost coloring algorithms.

# Sequential multi-coloring 1/2

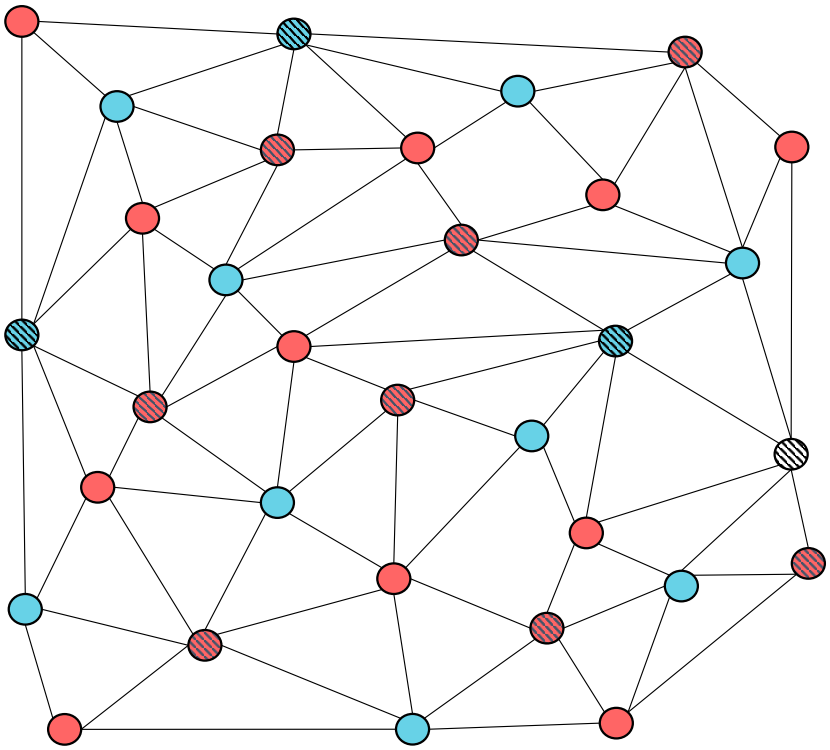
## Objective of multi-coloring



Example of the graph between block or elements

# Sequential multi-coloring 2/2

## Objective of multi-coloring



- Color1
- Color2
- Color3
- Color4
- Color5

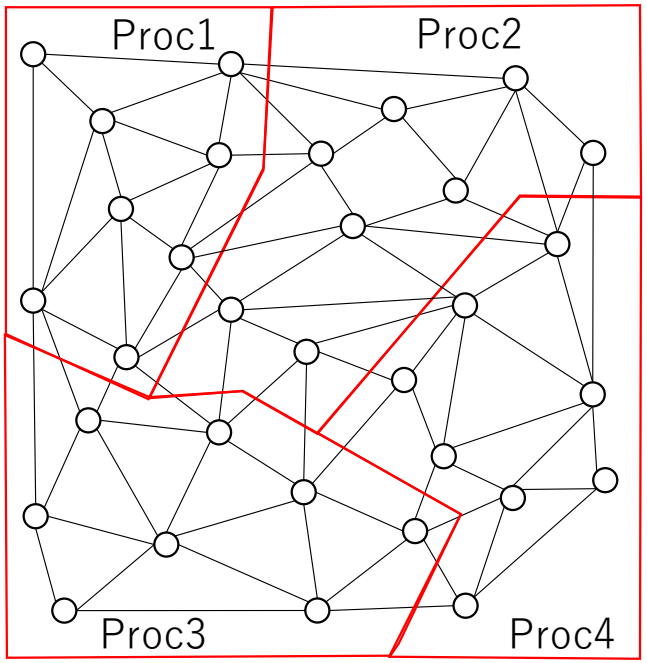
Example of the graph between block or elements

Coloring nodes with any algorithm

Almost algorithms are sequential.

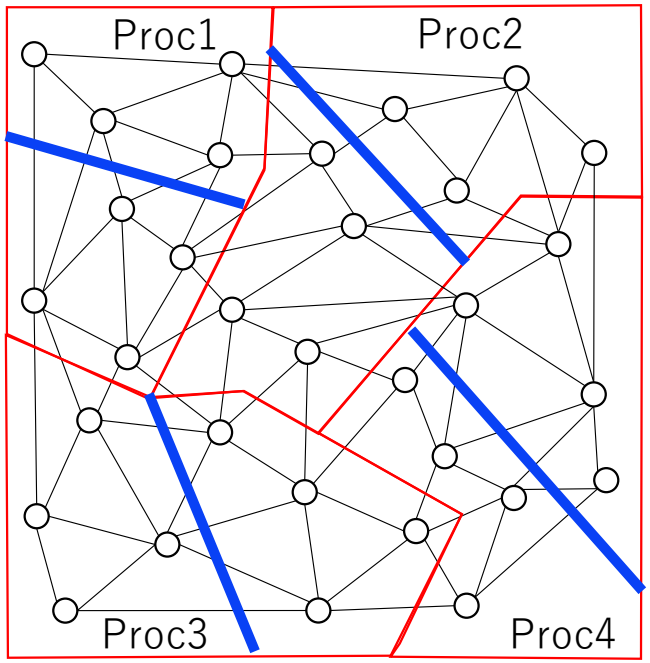
# Hierarchical parallelization 1/9

## Initial conditions

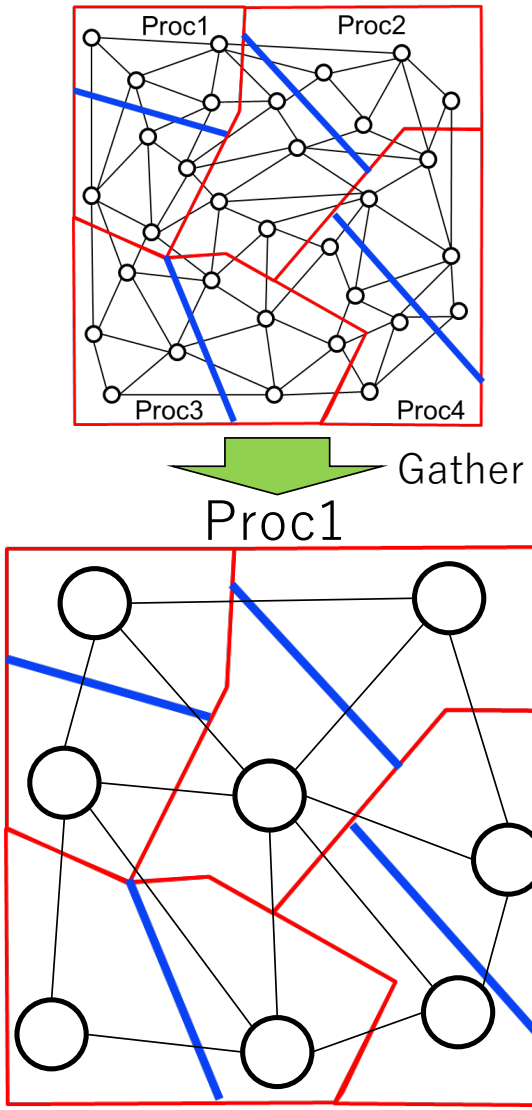


# Hierarchical parallelization 2/9

Initial conditions
1.Each process separates calculation area.



# Hierarchical parallelization 3/9

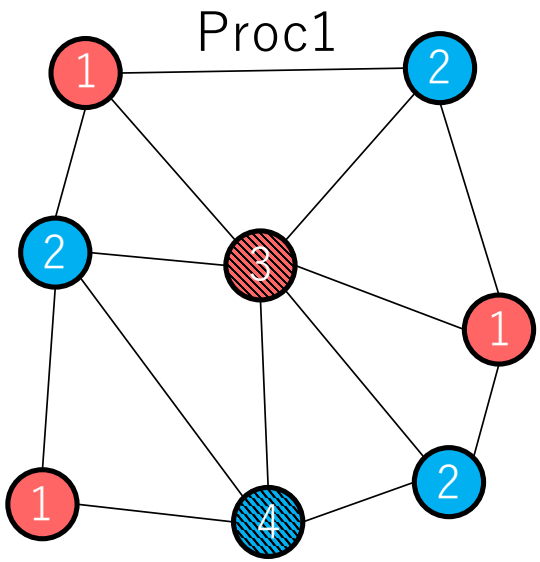



Initial conditions
1.Each process separates calculation area.
2.Gather the graph structures.

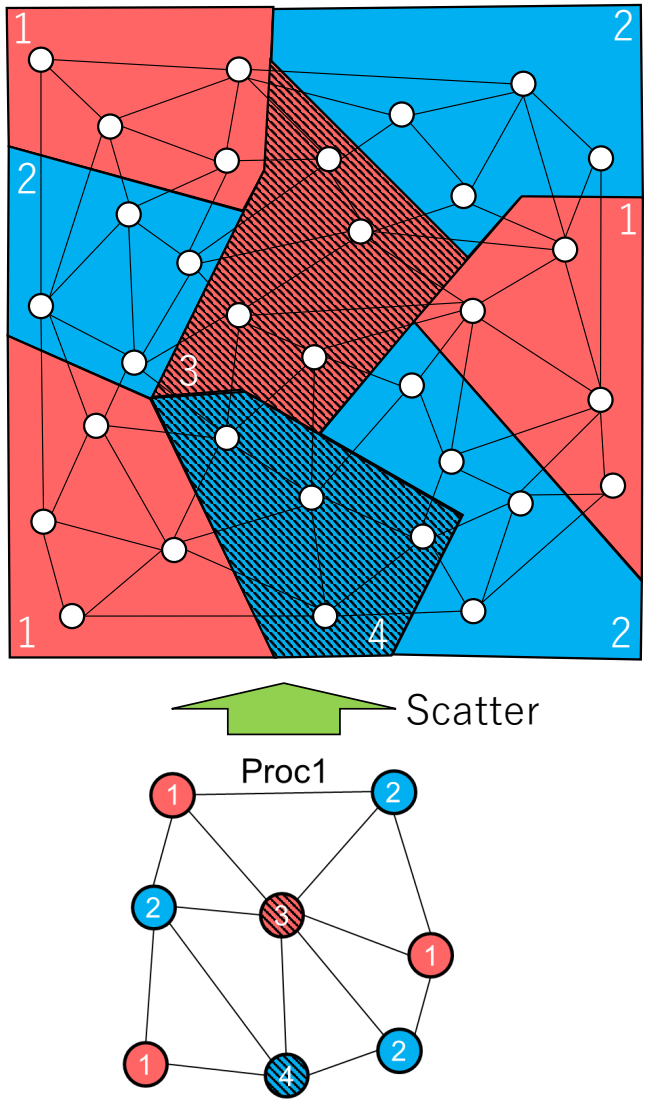
# Hierarchical parallelization 4/9

Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.



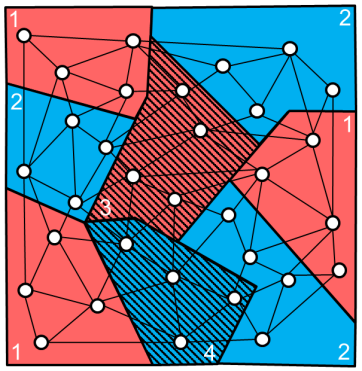
# Hierarchical parallelization 5/9

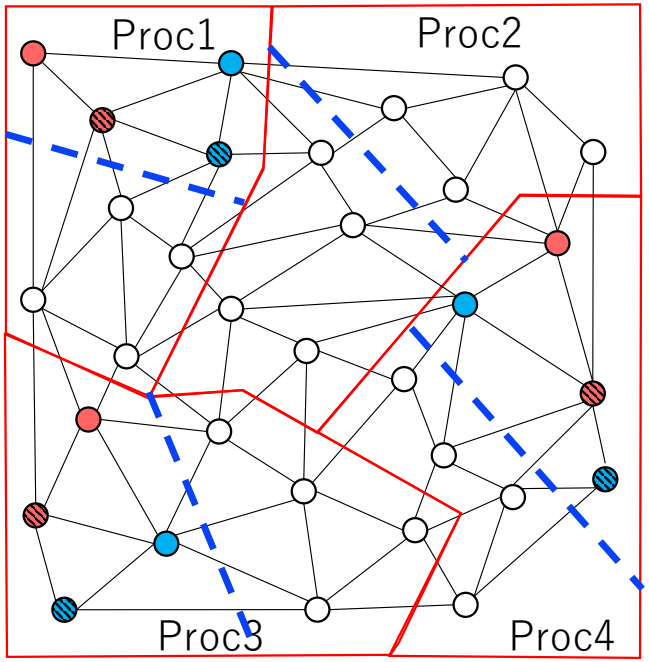



Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.
4. Broadcast coloring result.

# Hierarchical parallelization 6/9

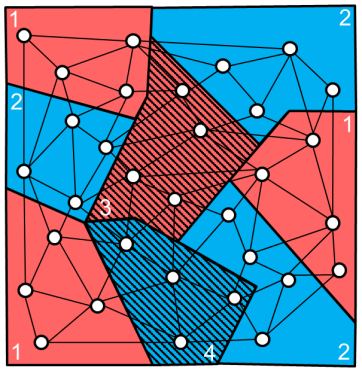


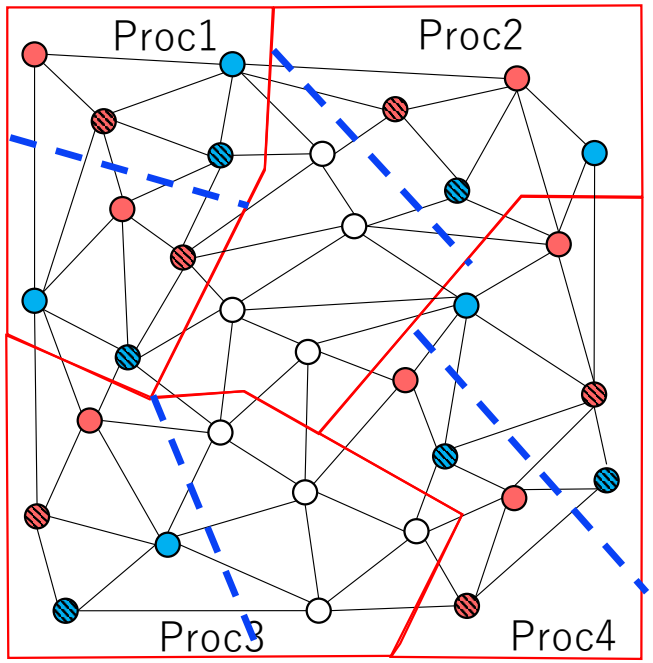




Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.
4. Broadcast coloring result.
5. Each process colors all nodes with any algorithm in parallel.

# Hierarchical algorithm 7/9

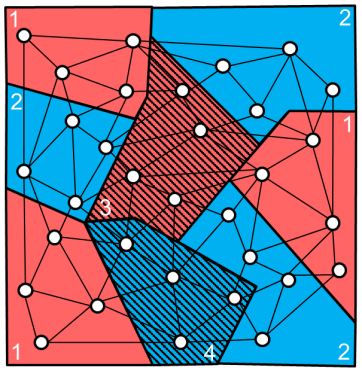


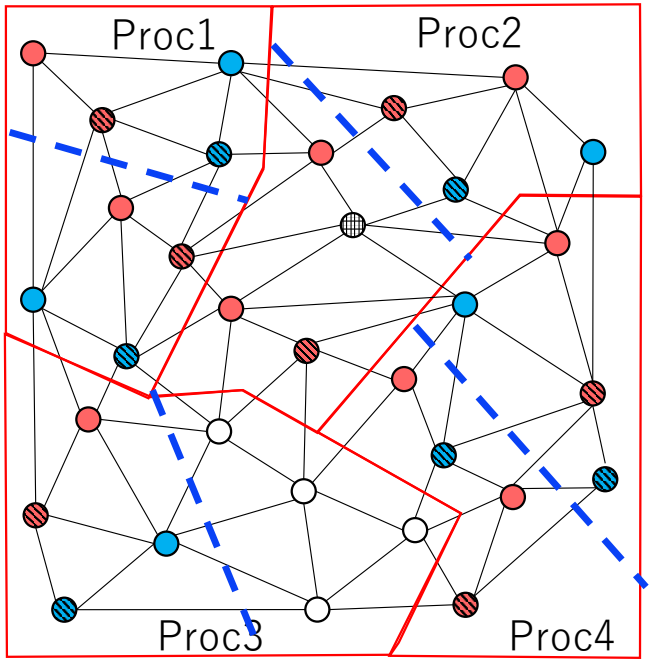




Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.
4. Broadcast coloring result.
5. Each process colors all nodes with any algorithm in parallel.

# Hierarchical algorithm 8/9

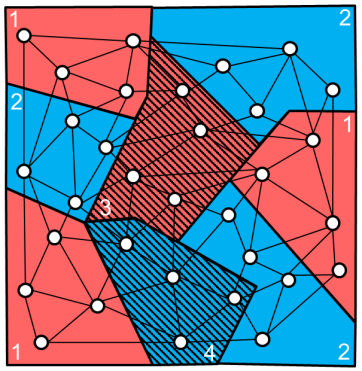


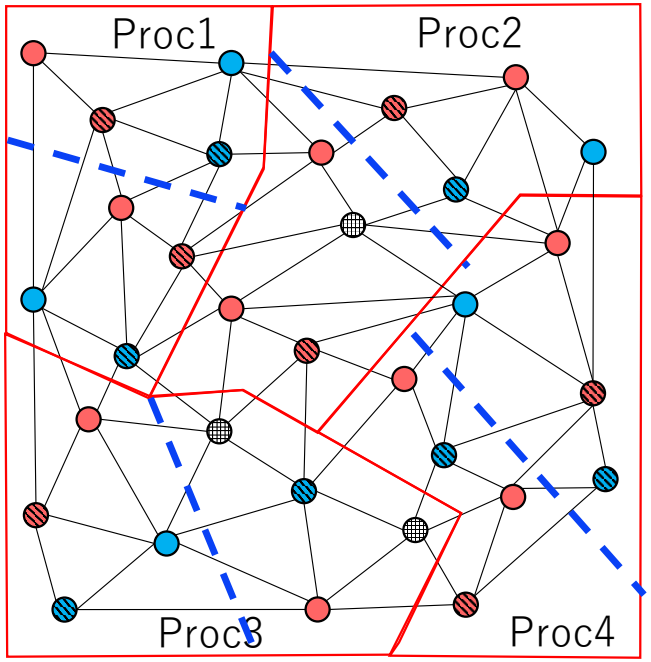




Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.
4. Broadcast coloring result.
5. Each process colors all nodes with any algorithm in parallel.

# Hierarchical algorithm 9/9




Initial conditions

1. Each process separates calculation area.
2. Gather the graph structures.
3. Master process colors the nodes with any method.
4. Broadcast coloring result.
5. Each process colors all nodes with any algorithm in parallel.

Finish

# Index

- Backgrounds and objective
- Regularized IC preconditioner
- Parallelization method for IC preconditioner
- Parallelization method for multi-coloring
- **Results**
- Summary and future works

# Test data sets

■Graphene
- The simulation of the electrical properties.
- Number of DOF is 128～1,000,000
- 9 data sets
- Number of non-zero elements per row : 13 or 4

■Spinz
- Simulation of considering the XXZ chain and Hamilton operator
- Number of DOF is 252～705,432
- 8 data sets
- Number of non-zero elements per row : 6~12 on average

■Kohn-Sham
- Simulation of the status of any molecules.
- Number of DOF is 57,575～76,163
- 6 data sets
- Number of non-zero elements per row : 20~24 on average

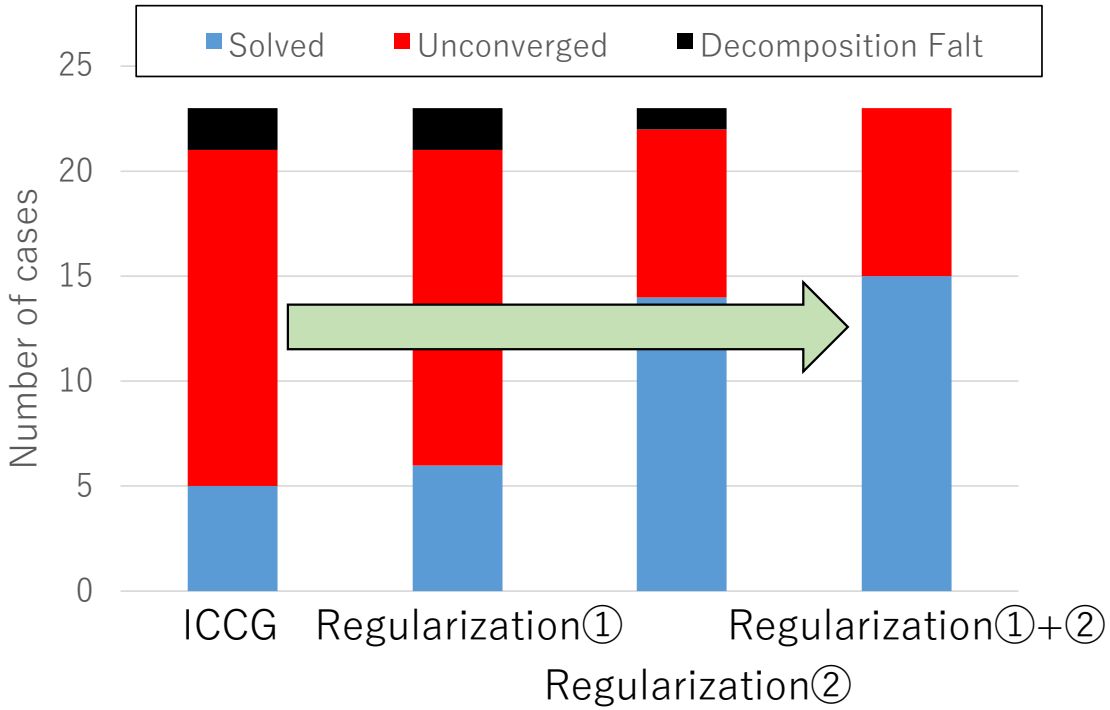
# Configurations of the tests (Regularizations)

- Right hand vector $b$ is calculated as following.
- $b = A_Z x \quad x^T = randam(min = 1, max = 10)$
- Iteration is stop if the number of iteration reach to DoF

  or relative residual fills the requirement $\left\| \frac{r^k}{r^0} \right\|_2 \leq 10^{-7}$
- Define regularization① with blocks size s as ICCG(S)
  - Blocking technique for ICCG
- Define regularization② with the constant value $\alpha$ as t = $\alpha$
  - Adding the constant value $\alpha$ to the diagonals.
- 18 test data sets

# Effect of the regularization techniques

**We can solve 15 test data sets by applying the regularization① and ②**




Unsolved data sets are SpinSZ.

# Configurations of numerical analysis (Parallelization)

- Data sets:Graph-1M、Graphene-512x256、Kohn-Sham
- Regularization①(Size of block) : 4
- Regularization②
  (Adding the constant value to the diagonal elements) : 100.0
- Number of color : Graph-1M and Graphene-512x256 are 10 colors
  Kohn-Sham is 30 colors

| | DOF |
|---|---|
| Graph-1M | 1,000,000 |
| Graphene-512x256 | 131,072 |
| Kohn-Sham | 57,575 |

# Evaluations of the numerical analysis

■Reedbush-u 1～2 nodes

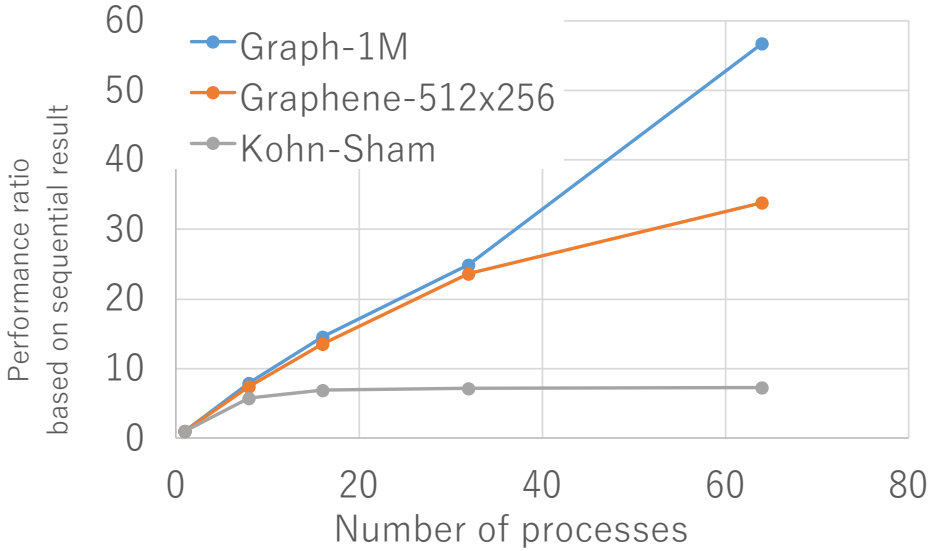
- New system of the ITC, U-Tokyo
- Node specifications
  - ✓Intel Xeon E5-2695v4 x 2socket (1.210 TF)
    Broadwell-EP, 2.1GHz 18core
    →36 Cores per node
  - ✓256 GiB (153.6GB/sec)
- Network specifications
  - ✓InfiniBand EDR


■1、8、16、32、64 processes

- 1process-1core, Flat MPI

# Result of numerical analysis

**Proposed method showed good performance on the large scale graphene data.**




| | DOF |
|---|---|
| Graph-1M | 1,000,000 |
| Graphene-512x256 | 131,072 |
| Kohn-Sham | 57,575 |

| Computational Time [s] | Number of processes | | | | |
|---|---|---|---|---|---|
| | 1 | 8 | 16 | 32 | 64 |
| Graph-1M | 87.86 | 11.10 | 6.05 | 3.53 | 1.55 |
| Graphene-512x256 | 10.04 | 1.36 | 0.74 | 0.43 | 0.30 |
| Kohn-Sham | 30.37 | 5.26 | 4.38 | 4.26 | 4.18 |

| Number of iterations | Seq | 1 | 8 | 16 | 32 | 64 |
|---|---|---|---|---|---|---|
| Graph-1M | 406 | 410 | 410 | 410 | 410 | 410 |
| Graphene-512x256 | 1131 | 1131 | 1131 | 1131 | 1131 | 1131 |
| Kohn-Sham | 1117 | 1303 | 1193 | 1287 | 1285 | 1285 |

# Index

- Backgrounds and objective
- Regularized IC preconditioner
- Parallelization method for IC preconditioner
- Parallelization method for multi-coloring
- Results
- Summary and future works

# Summary

■The target equations are provided from the generalized eigenvalue problems of the quantum system

■Proposed the 2 regularization techniques

- The blocking technique
- Adding constant value to the diagonal elements.

→**Proposed preconditioner solved 15 data sets**

■Applying the multi-coloring for parallelization

- The algebraic multi-color ordering

→**Proposed method shows good performance on the large scale graphene data.**

■Proposed hierarchical approach for parallelize MC

# Future works

■To improving convergence
Better regularization②
Relationship between Regularization① and ②
Applying Low-rank-approximation for IC preconditioner.

■Parallelization
Investigating the impact of coloring algorithm and hierarchical approach

■Optimizations
SIMDization

■Others
Supporting complex equations

# Thank you!

# Objective of this study

**To develop parallel preconditioner for Krylov subspace(KS) method to solve linear system of equations $A_z x = b$**
**Target equations are derived from large-scale generalized eigenvalue problems of quantum systems.**

$$Ax = \lambda Bx \quad A, B \in \mathbb{C}^{n \times n}, \lambda: eigenvalue, x: eigenvector$$

**The equations we focus on are derived from *FEAST* and *Sakurai-Sugiura* method.**

Backgrounds
To solve large scale equations are needed in the quantum system fields

Direct solver is generally used because of unfitted conditions for iterative solvers.

# Equations derived from SSM and FEAST

**The Linear system of equations derived from *Sakurai-Sugiura* method (SSM) and *FEAST***

$$A_z x = b \leftarrow (zB - A)V = BY$$

$$A_z = (zB - A), x = \text{column vector of } V, b = \text{column vector of } BY$$

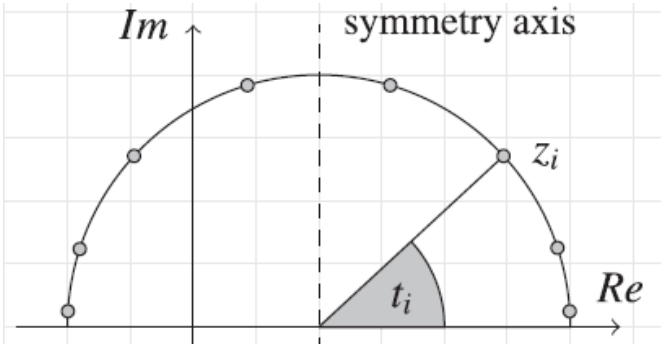
■ Certain eigenvalues and eigenvectors, that lie in the given domain, are determined.

- Contour integral of the given domain
- The equations

  $$(zB - A)V = BY, \qquad V, Y \in \mathbb{C}^{n \times m}$$

  is defined on each integral points $z_i$.
- $z = 0$ in this presentation

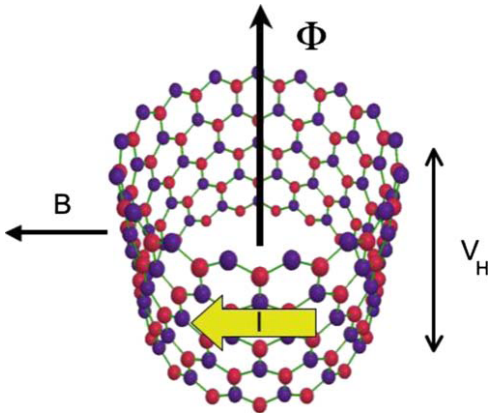
  → Only real eigenvalues are calculated.



# Problems 1/2

■Test data sets "*Graphene*"
Simulation of electrical properties.

- Number of DOF is 128～1,000,000
- 9 data sets
- Number of non-zero elements per row : 13 or 4



Source : The electronic properties of graphene

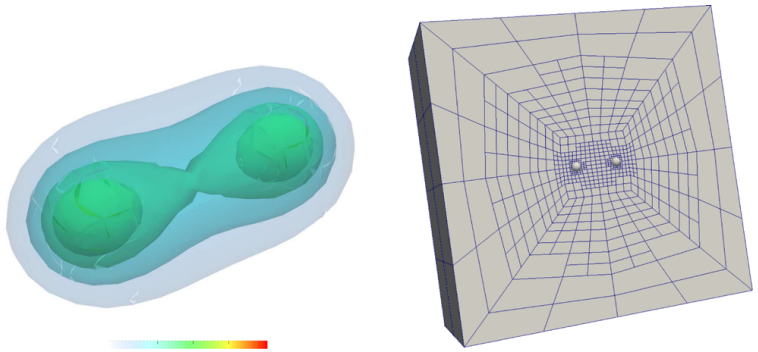
■Test data sets "*Spinz*"

- Simulation of considering the XXZ chain and Hamilton operator
- Number of DOF is 252～705,432
- 5 data sets(3 symmetric data sets)
- Number of non-zero elements per row : 6~12 on average

# Problems 2/2

■Test data sets "*Kohn-Sham*"
Simulation of the status of any molecu

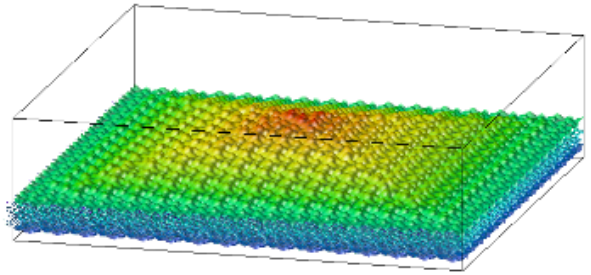
- Number of DOF is 57,575～76,163
- 6 data sets
- Number of non-zero elements per row : 20~24 on average



Source : On the adaptive finite element analysis of the Kohn–Sham equations : methods, algorithms, and implementation

■Test data sets "Conquest"
Simulation of charge density of many qua

- Number of DOF is 94948 and 213633.
- 2 data sets
- Number of non-zero elements per row : 70 or 135 on average



Source : Recent progress with large-scale ab initio calculations: the CONQUEST code

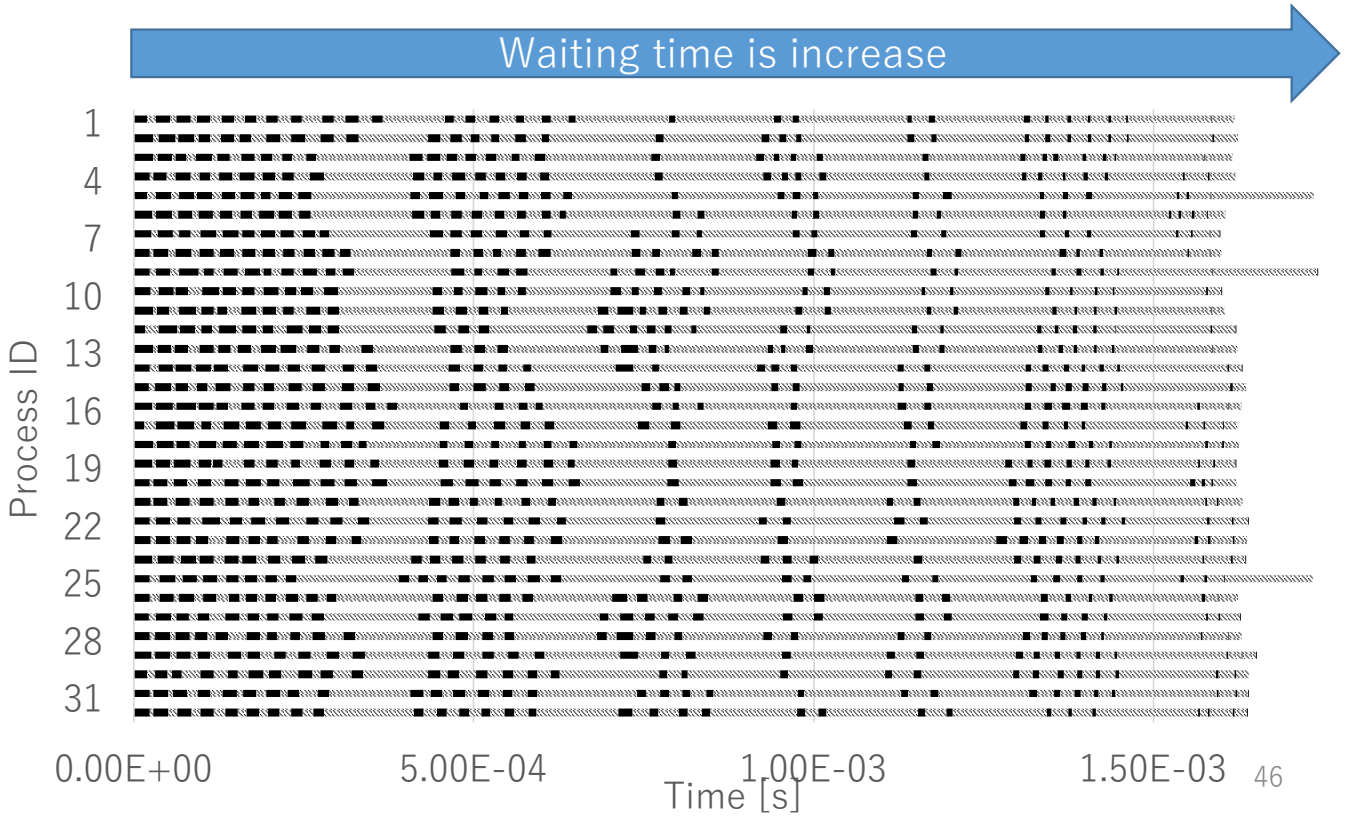
# Detail of the computatinal time  on Kohn-Sham

Increasing waiting time because of getting out of step with the other processes.

If increasing the number of colors, the time of waiting communicates are increase.

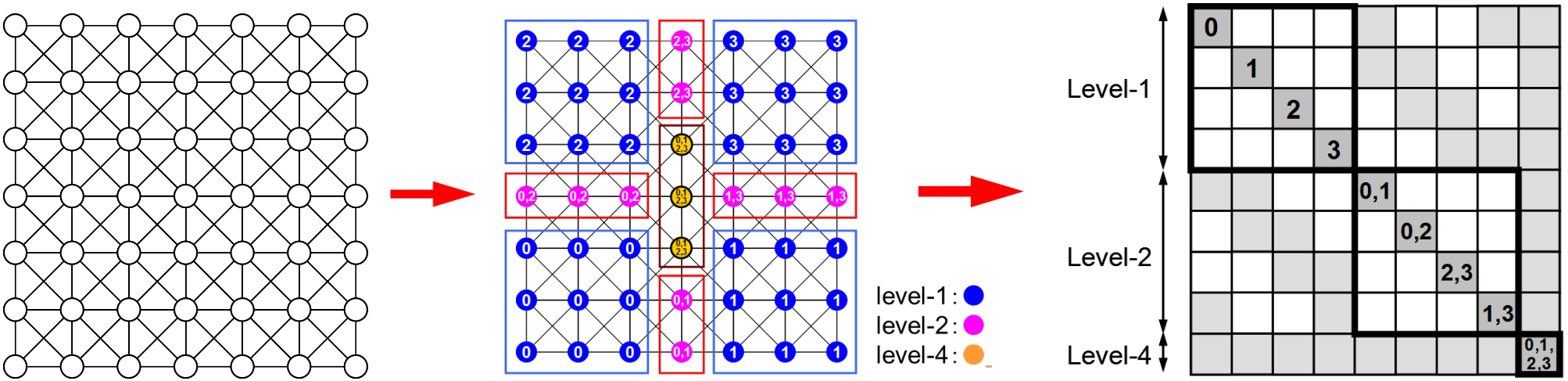
We should decrease the number of colors to decrease waiting times.

Communication and computational time of forward substitution
Black parts are computations and dilute parts are communications.



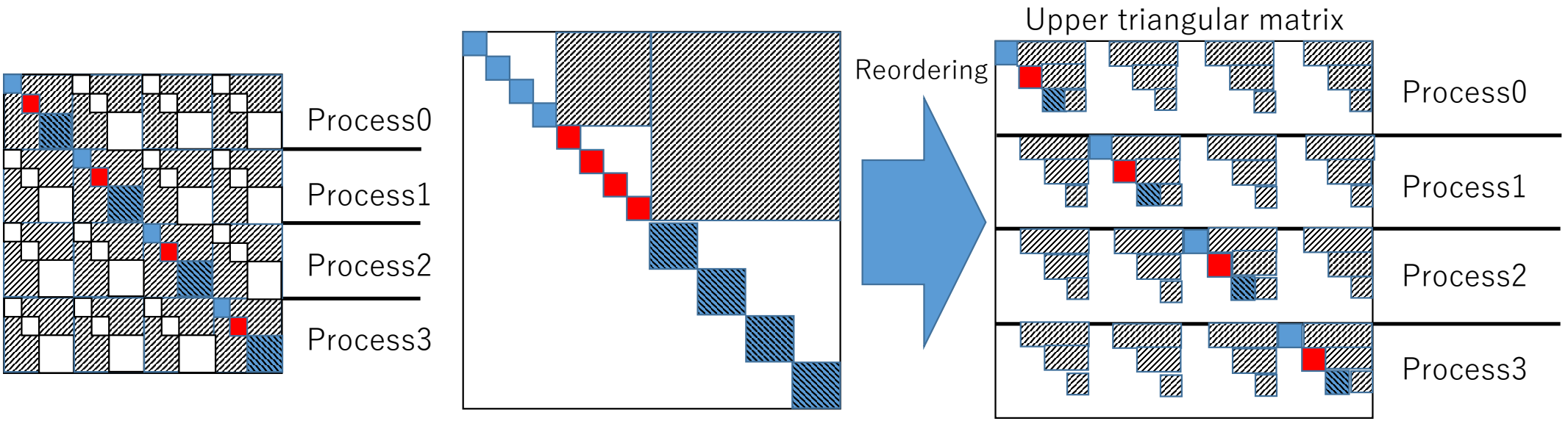
# HID: Hierarchical Interface Decomposition [Henon & Saad 2007]

- Multilevel Domain Decomposition
  - Extension of Nested Dissection
- Non-overlapping at each level: Connectors, Separators
- Suitable for Parallel Preconditioning Method

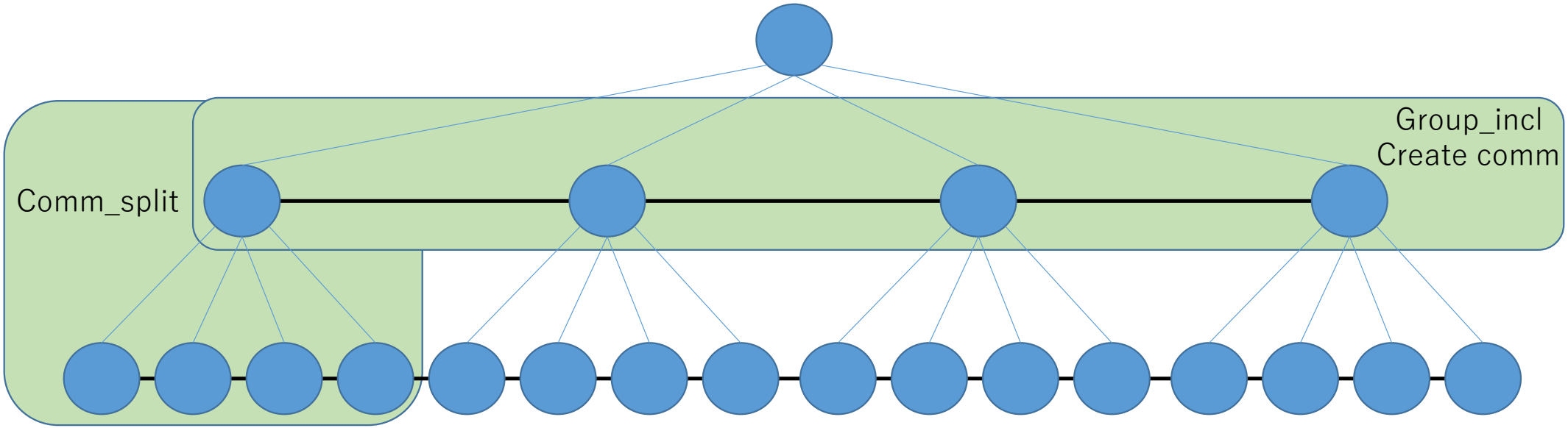


# IC decomposition with multi-coloring

In the program, we calculate the upper triangular matrix directly from the colored base matrix.

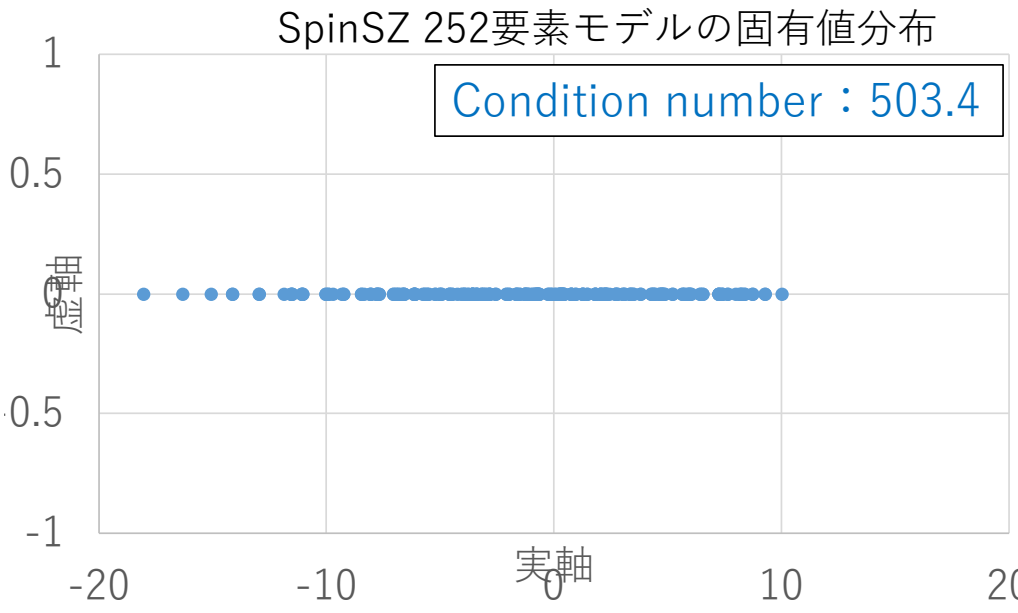


# Implementation of hierarchical approach

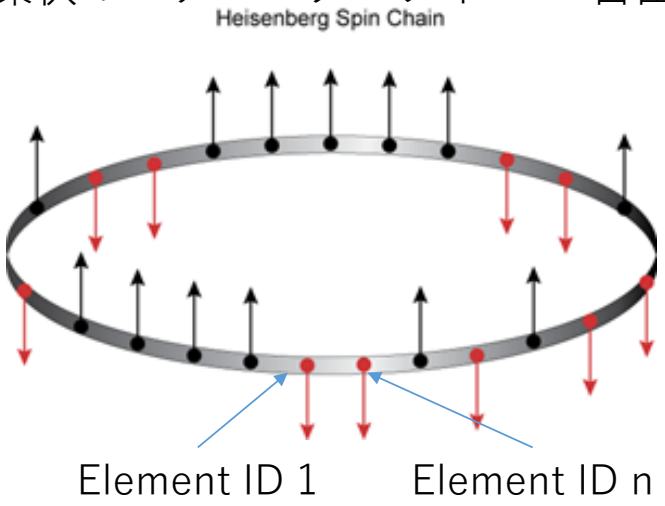


# 解けなかった問題に対する考察 1/2

Why we don’t solve spinSZ?
→High-condition number and wide pro-fills.




円環状のモデル→プロフィル*＝自由度

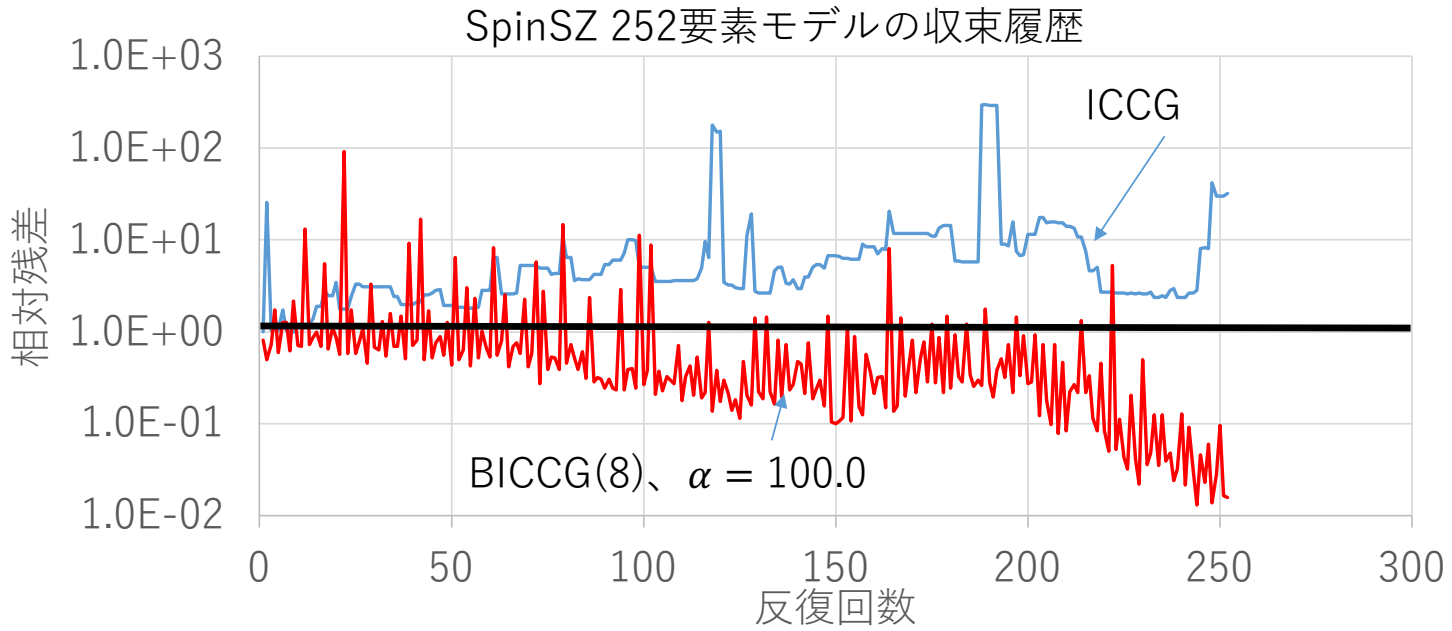



出展：http://physics.aps.org/articles/v2/79
Viewpoint: The weak and strong ends of a theory

＊プロフィル：1～自由度までの値を取る行列の評価値
小さいほどIC前処理の効果大

# 解けなかった問題に対する考察 2/2

正則化の適用で収束性改善は確認できている。




さらに収束性を改善させるために、

- Reordering
- Adaptive cross approximationによるfill-inを許可していない部分の近似

# Example of eigenvalue