改定不完全コレスキー前処理付CG法の大規模行列での特性

2023年7月12日
後　保範（ISCPC）

1. はじめに

- 拡散方程式における差分法の反復解法
- 不完全コレスキー(IC)にαを追加前処理CG法
- MICCG法の収束の良いパラメータαの自動推定
- MICCG法はICCG法より圧倒的に収束が速い
- 最良パラメータωのSSOR前処理付きCG法と比較

  ２次元：　反復回数が1/1.2～1/4.7に減少

  ３次元：　反復回数が1/1.1～1/3.1に減少

  大規模、複雑なほどMICCG法の効果が大

2. 前処理付き共役勾配法(CG)

- 共役勾配法(CG)は対称疎行列用の反復解法
- M行列(拡散方程式)は前処理で収束が良くなる
- 前処理はSSOR、IC(不完全コレスキー)が有名
- MIC(改定IC)はICにパラメータ(α)を追加
- MIC,SSOR,IC前処理ともCG法の反復計算は同じ
- αは分解誤差を対角で補正:1-αは対角との差
- 従来α=0.98前後。規模大で1-αは小さくなる
- パラメータαを分割数で自動作成を提案

2.1 前処理付きCG法の手順

初期値x=0,　行列Aを不完全LDL^T分解する

$r=b-Ax$,　$p=(LDL^T)^{-1}r$,　$\mu_1=(r,p)$

収束するまで以下を反復計算

$q=Ap$,　$\alpha=\mu_1/(p,q)$

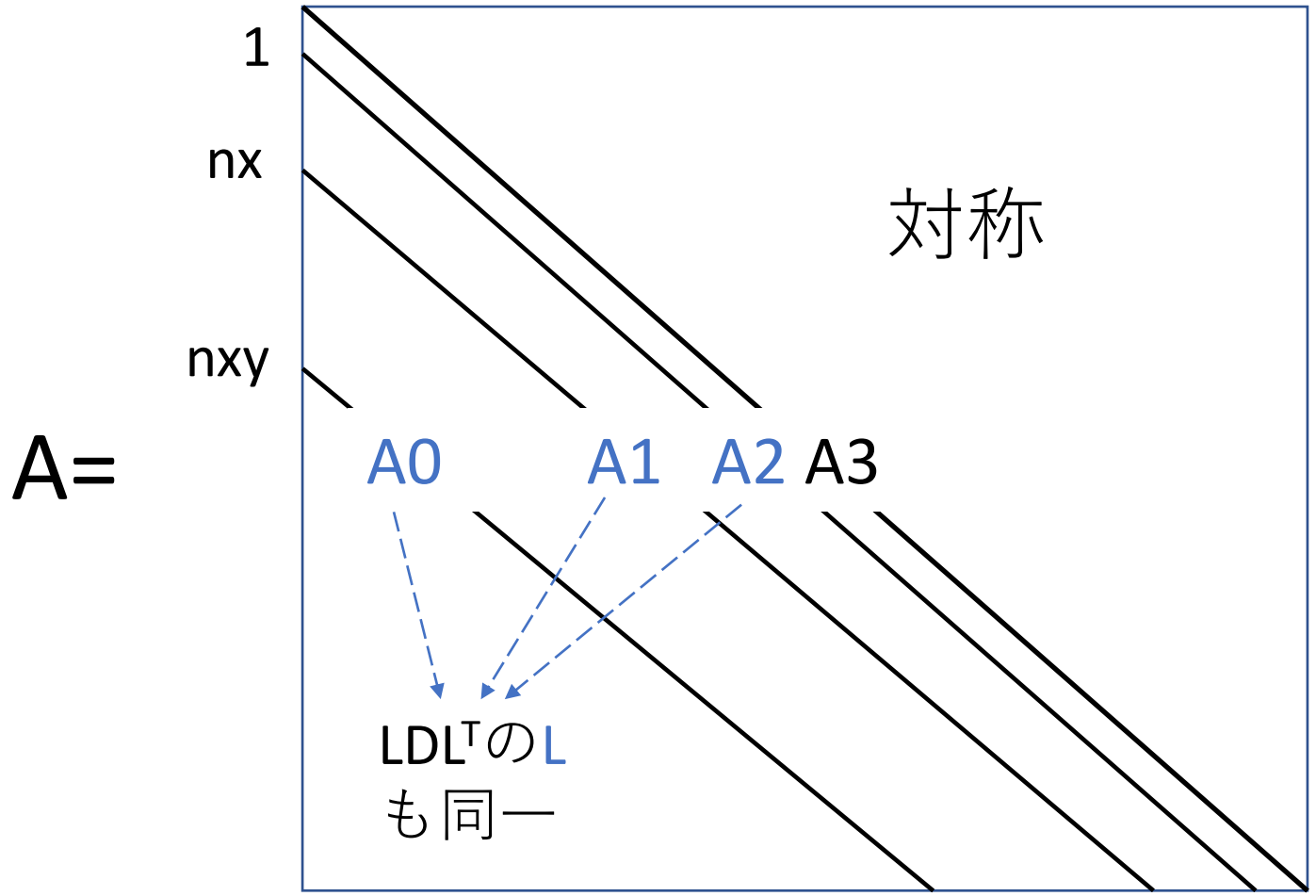
$x=x+\alpha p$,　$r=r-\alpha q$

$q=(LDL^T)^{-1}r$,　$\mu_2=(r,q)$

$\beta=\mu_2/\mu_1$,　$\mu_1=\mu_2$

$p=q+\beta p$

2.3 ３次元差分法による行列



2.4 $q=(LDL^T)^{-1}r$の計算(共通)

```
for (k=0; k<n; k++)
   { q[k] = D[k]*( r[k] - A2[k]*q[k-1]
                       - A1[k]*q[k-nx]
                       - A0[k]*q[k-nxy] ) ;
   }
for (k=n-1; k>=0; k--)
   { q[k] -= D[k]*( A2[k+1]*q[k+1]
                  + A1[k+nx]*q[k+nx]
                  + A0[k+nxy]*q[k+nxy] ) ;
   }
```

2.5 不完全LDL^{T}分解(SSOR)

- SSORの加速係数ω(<2.0)、Dを求める。

```
for (k=0; k<n; k++)
    { D[k] = ω / A3[k]; }
```

注)

不完全LDL^{T}分解では対角を除くLは

A0,A1,A2をそのまま使用(IC,MICも)

2.6 不完全LDL^{T}分解(IC)

- パラメータ無し、Dを求める。分解誤差無視。

```
for (k=0; k<n; k++)
  { dv = A3[k] - A2[k]*A2[k]*D[k-1]
               - A1[k]*A1[k]*D[k-nx]
               - A0[k]*A0[k]*D[k-nxy];
    D[k] = 1.0 / dv;
  }
```

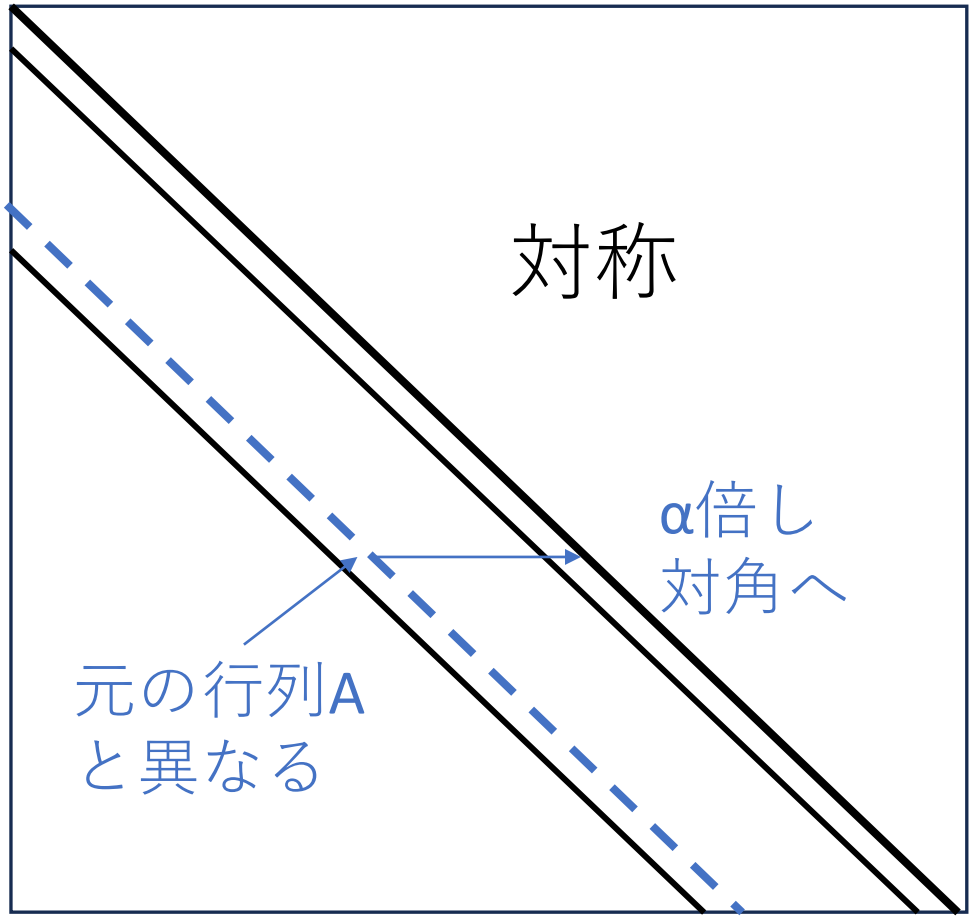
2.7 不完全LDL^{T}分解(MIC)

- パラメータα(<1.0)、Dを求める。α*誤差を対角に

```
for (k=nxy; k<n_A; k++)
  { dv = A3[k] - A2[k]*D[k-1]*( A2[k] + α *
                    ( A1[k+nx-1] + A0[k+nxy-1]) )
               - A1[k]*D[k-nx]*( A1[k] + α *
                    ( A2[k-nx+1] + A0[k+nxy-nx]) )
               - A0[k]*D[k-nxy]*( A0[k] + α *
                    ( A2[k-nxy+1] + A1[k-nxy+nx]) );
    D[k] = 1.0 / dv;
  }
```

2.8 MICの考え方(2次元の例)

$LDL^T=$



考え方

変数値が対角とα異なる

3 計算対象(環境)

- 計算機

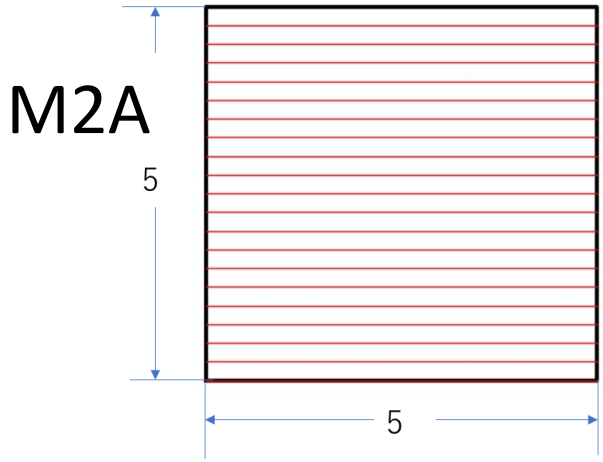
  Intel Core i7 6700k, 4Ghz, 8GBメモリ
- OS: Windows10
- コンパイラ

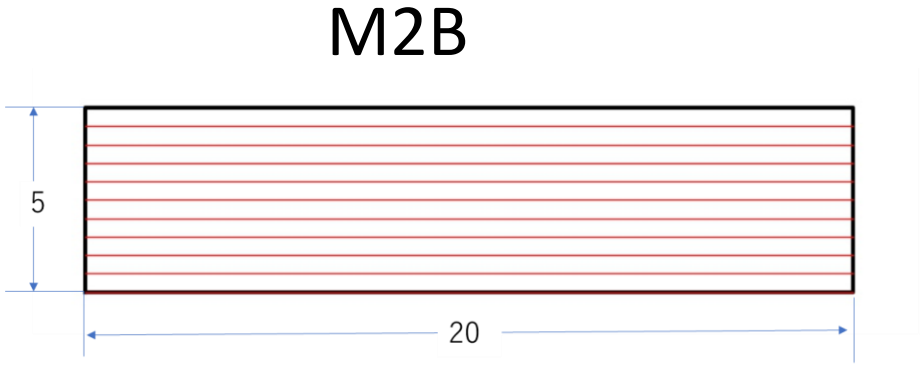
  Cygwin Ver. 2.7 gcc, -O3
- 拡散方程式(-div(k・grad(φ))=f)の差分法

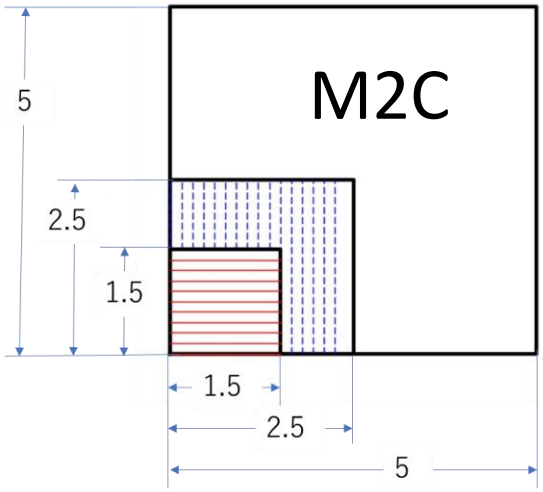
  離散化(2次元、3次元)
- 収束判定(下記となる反復回数)

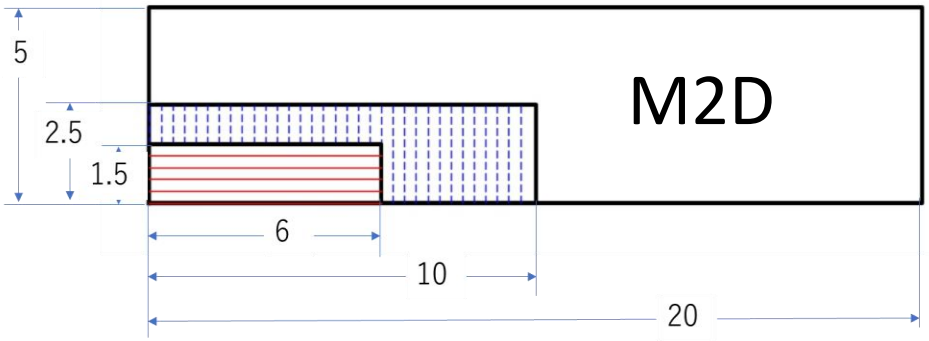
  $||Ax-b||_2 / ||b||_2 <= 1.0e-8$

3.1 ２次元対象モデル





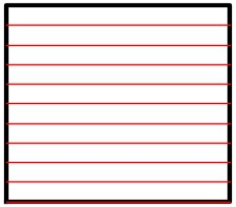
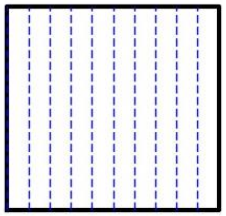
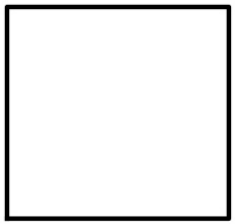




3.2 係数と境界条件

- 拡散方程式(-div(grad(k・φ))の係数

k = 0.5	k = 0.1	k = 1.0
f = 10	f = 0	f = 0

- 境界条件

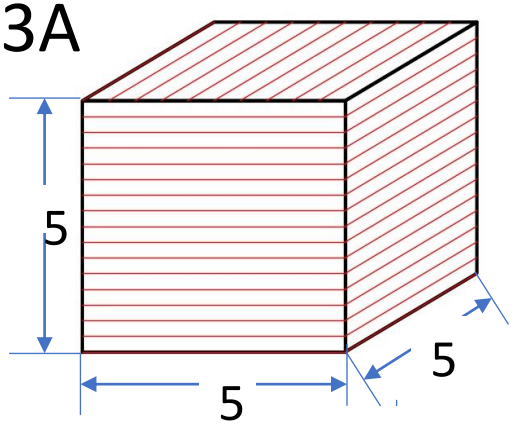
  B0: 全て固定境界

  B2: ２辺(低位)対称、２辺固定境界

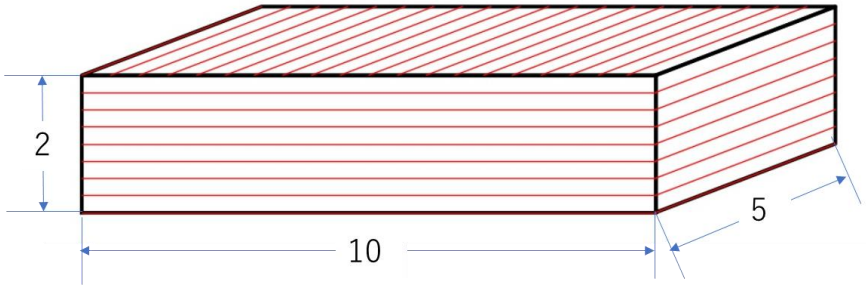
  B4: 各辺対称境界で、１点だけ固定境界

3.3 3次元対象モデル

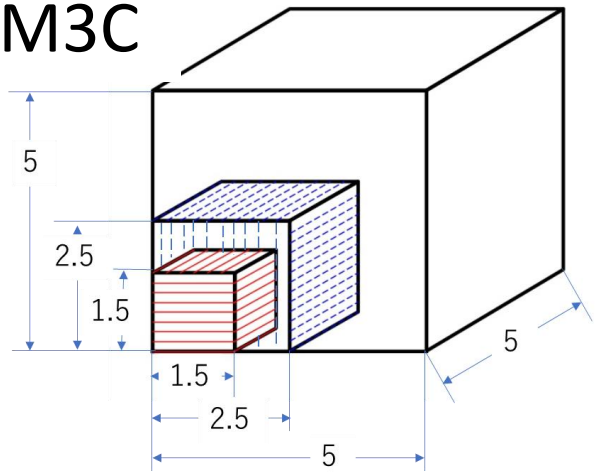
M3A



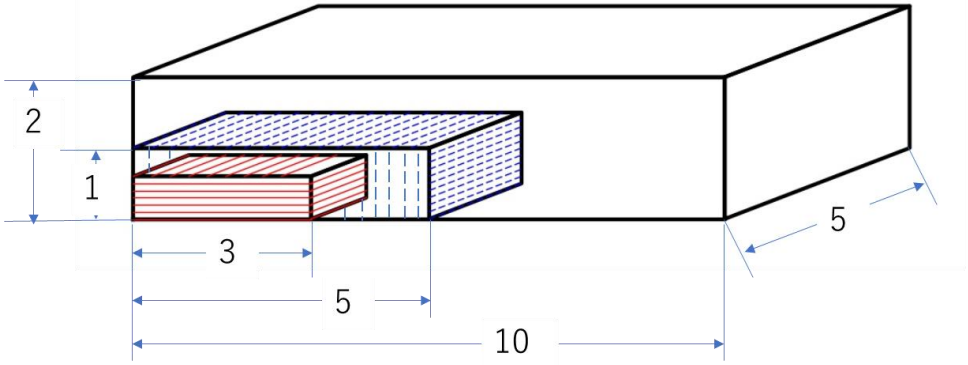
M3B



M3C



M3D

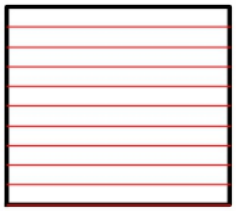
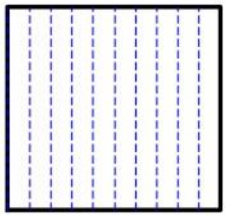
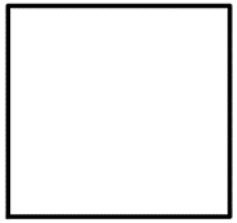


3.4 係数と境界条件

- 拡散方程式(-div(grad(k・φ))の係数

k = 0.5　　k = 0.1　　k = 1.0

f= 10　　f = 0　　f = 0

- 境界条件

B0: 全て固定境界

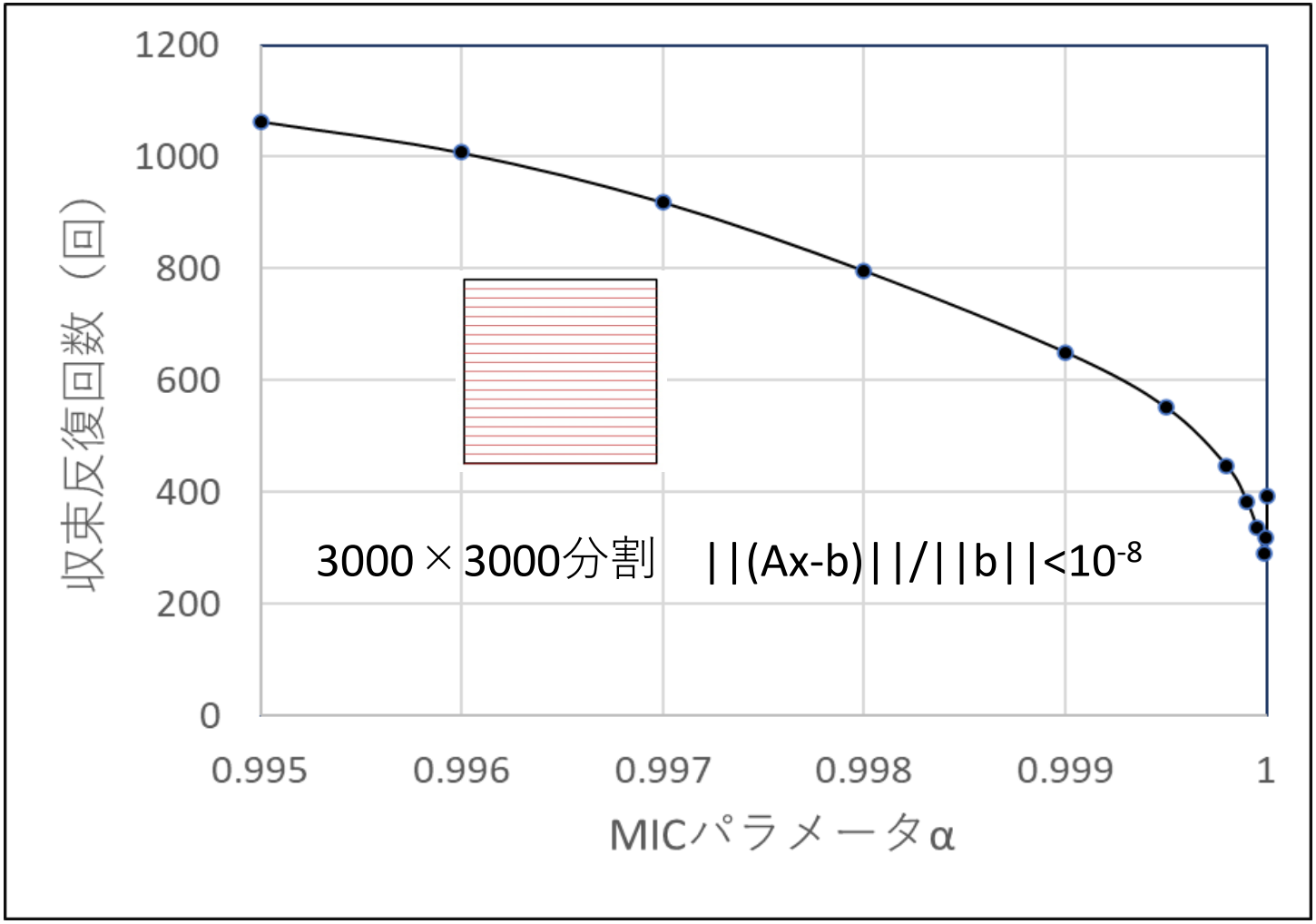
B3: ３面(低位)対称、３面固定境界

B6: 各面対称境界で、１点だけ固定境界

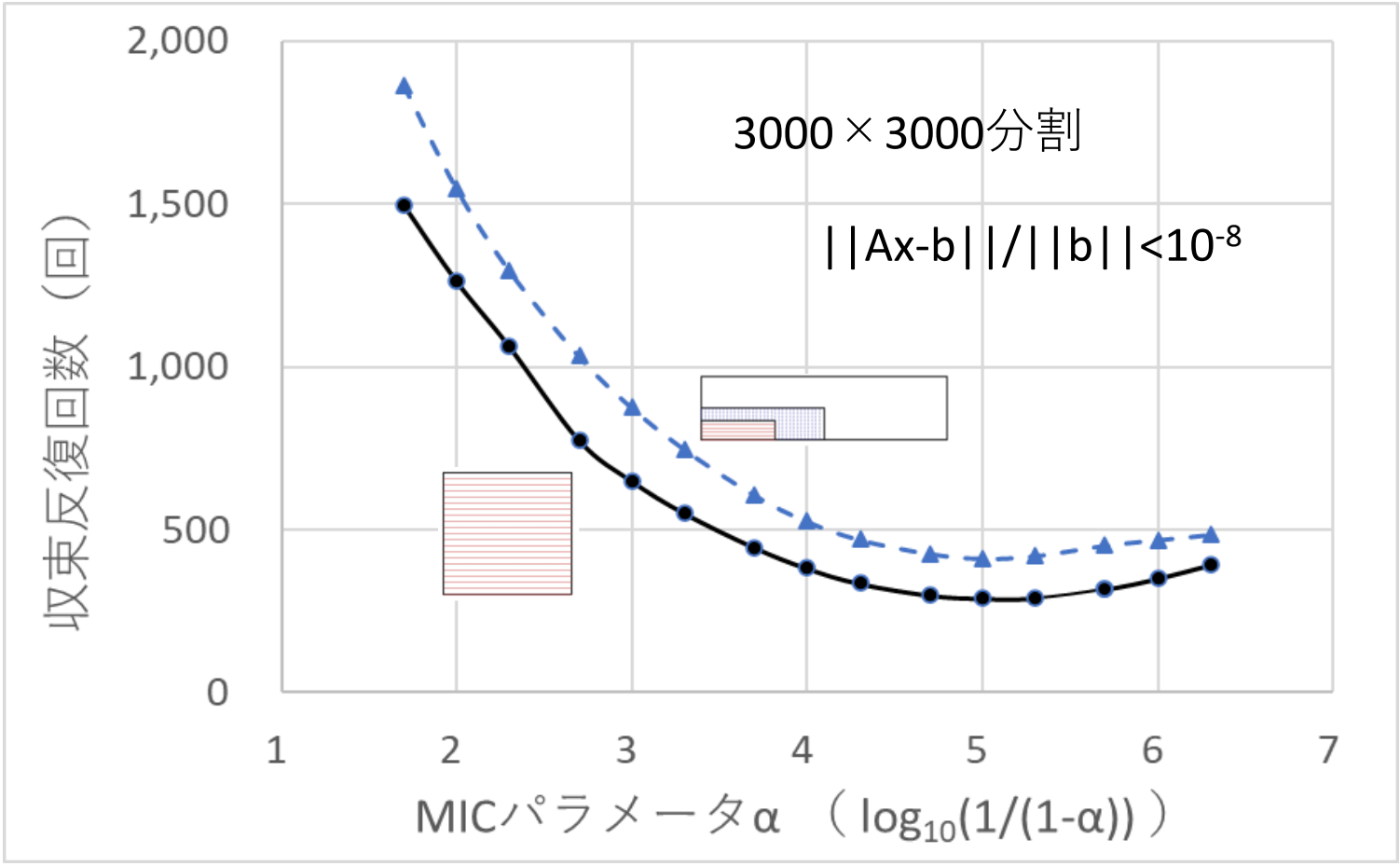
4 MICCG法のパラメータα

- 不完全コレスキー分解(IC)は対角だけ計算
- 非ゼロ要素の位置は元と分解後で一致する
- 非ゼロ要素以外の位置に誤差が発生する
- MICは非対角$R=LDL^T-A$を$D=D-\alpha R$として処理
- 収束が良いαの自動算出を計画
- 規模大で１の手前のαは収束が大きく変化
- $1-\alpha$と分割数はほぼ対数関係にあると判明
- 分割数で収束の良いαの算出に成功

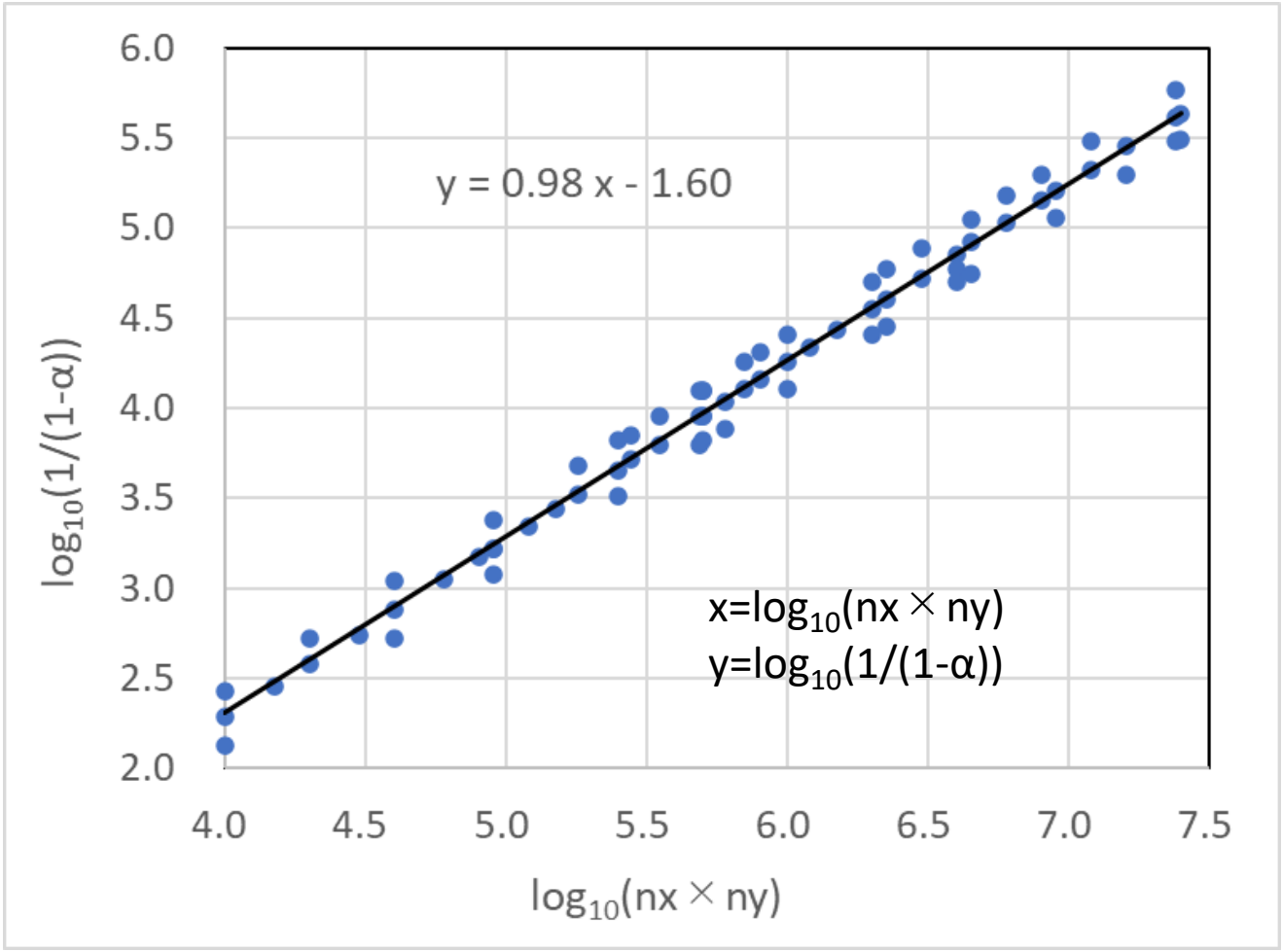
4.1 αとMICCG収束状況（２次元）



4.2 対数での収束状況（２次元）



4.3 分割数と最適α（２次元MIC)



全モデル
対象

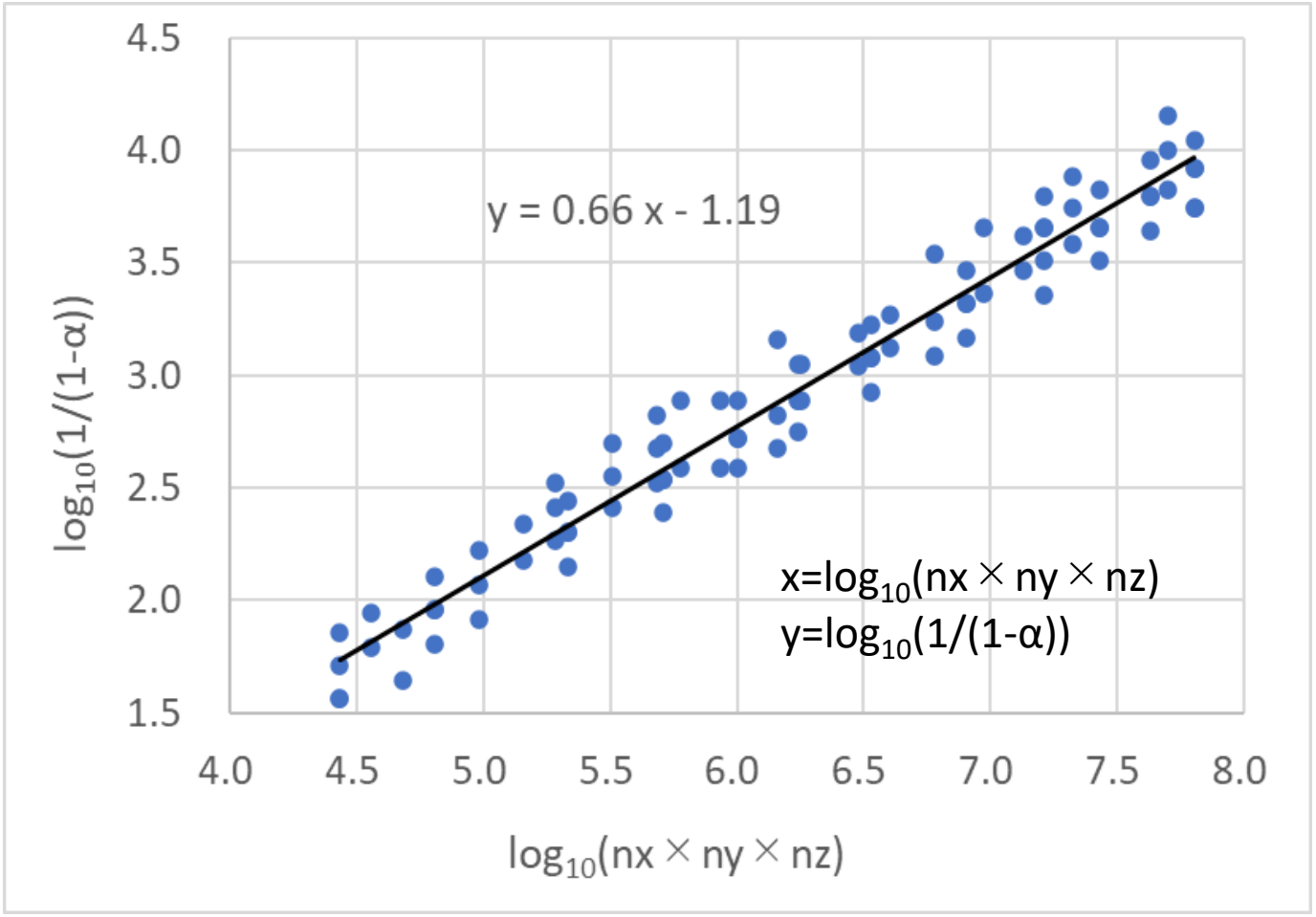
分割数
nx >= ny
nx <=2ny

4.4 適用αの計算(２次元)

- MICCGに適用するパラメータαの計算
- x, y方向の分割数をnx, nyとする
- $x=\log_{10}(nx \times ny)$, $y=\log_{10}(1/(1-\alpha))$とする
- y=0.98x-1.60の関係が前図から得られる

- nx, nyから$x=\log_{10}(nx \times ny)$を求める
- y=0.98x-1.60にxを与えyの値を求める
- $1/d=1/(1-\alpha)=10^y$からdを求める
- $\alpha=1-d$でMICCGのパラメータαを求める

4.5 分割数と最適α（３次元MIC)



全モデル
対象

分割数
nx <=ny<=nz
nx <= 2.5nz

4.6 適用αの計算(３次元)

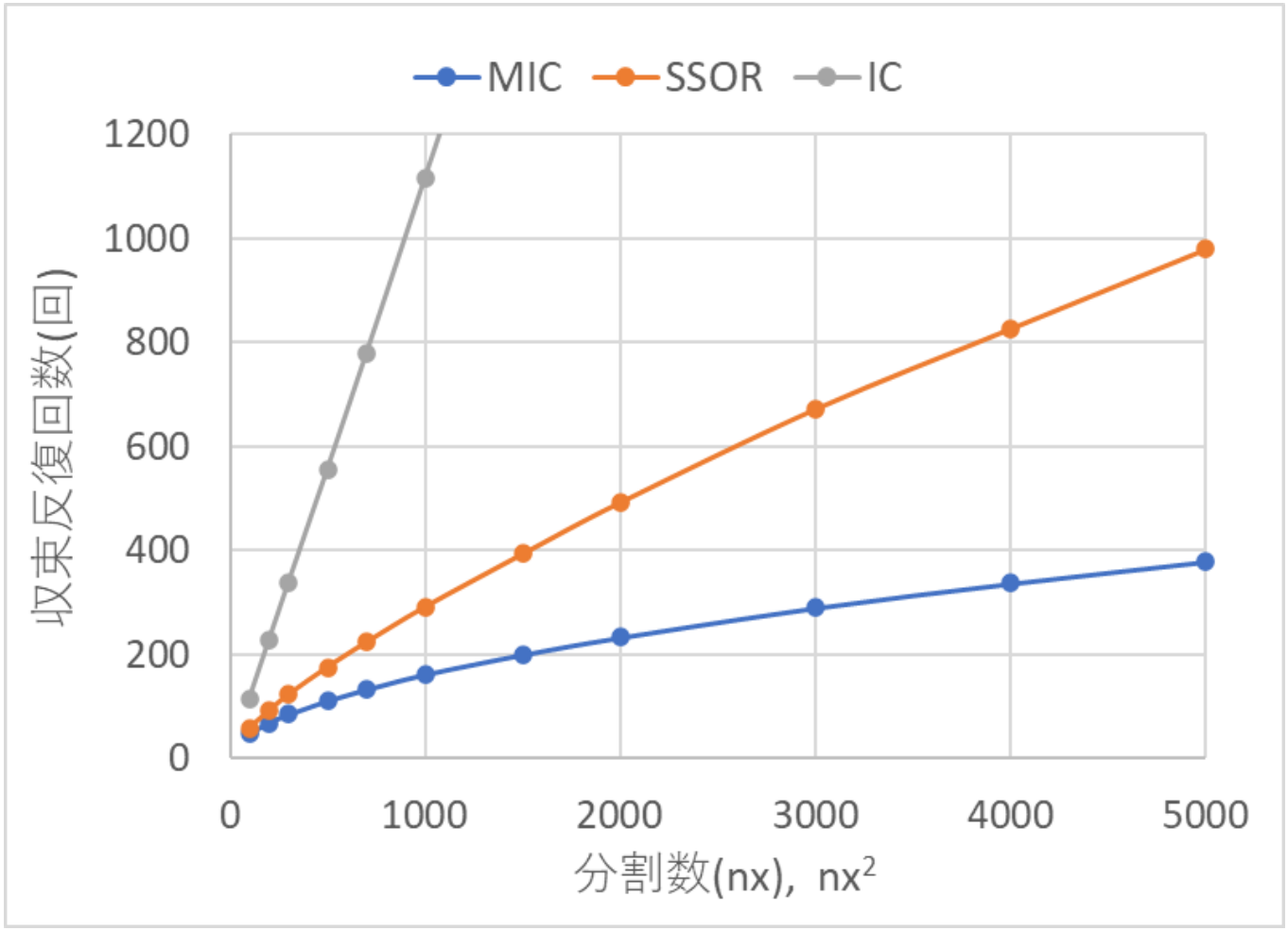
- MICCGに適用するパラメータαの計算
- x, y, z方向の分割数をnx, ny, nyとする
- $x=\log_{10}(nx\times ny\times nz)$, $y=\log_{10}(1/(1-\alpha))$とする
- y=0.66x-1.19の関係が前図から得られる

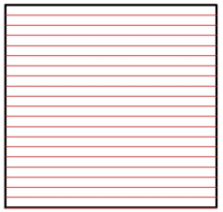
- nx, ny, nyから$x=\log_{10}(nx\times ny\times nz)$を求める
- y=0.66x-1.19にxを与えyの値を求める
- $1/d=1/(1-\alpha)=10^{y}$からdを求める
- $\alpha=1-d$でMICCGのパラメータαを求める

5 MIC,SSOR,IC前処理の比較

- 拡散方程式の差分法の反復法(Ax=b)で比較
- 比較は$||Ax-b||_2/||b||_2 < 10^{-8}$となる反復数
- MIC,SSOR,ICの前処理付きCG法で比較
- SSORは加速係数ωを1/200単位で変化させ、
  反復数が最も少ないものを使用
- MICはパラメータαを分割数で自動設定の
  反復数を使用
- ICはパラメータ無し

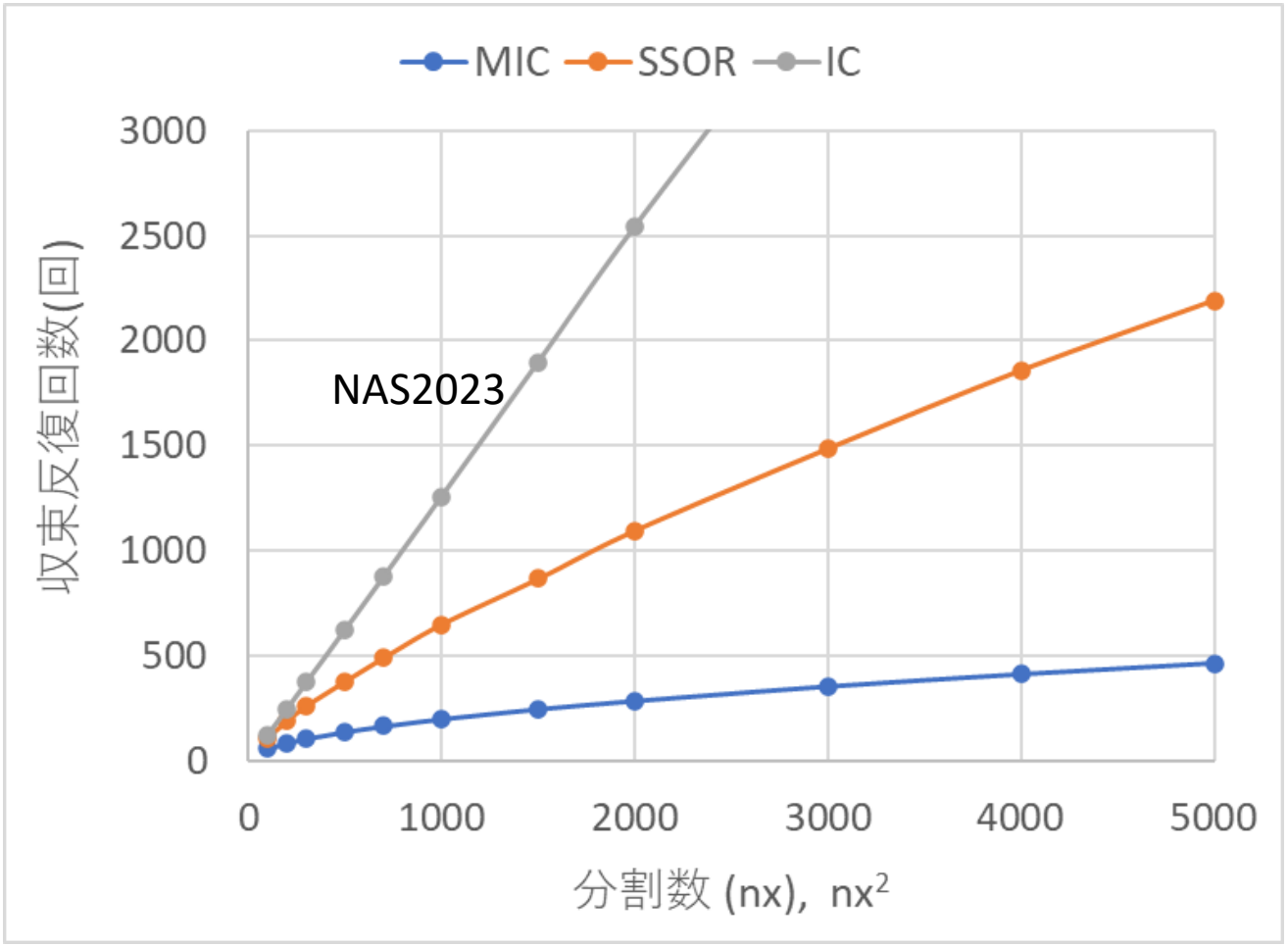
5.1 M2A-B2での比較結果





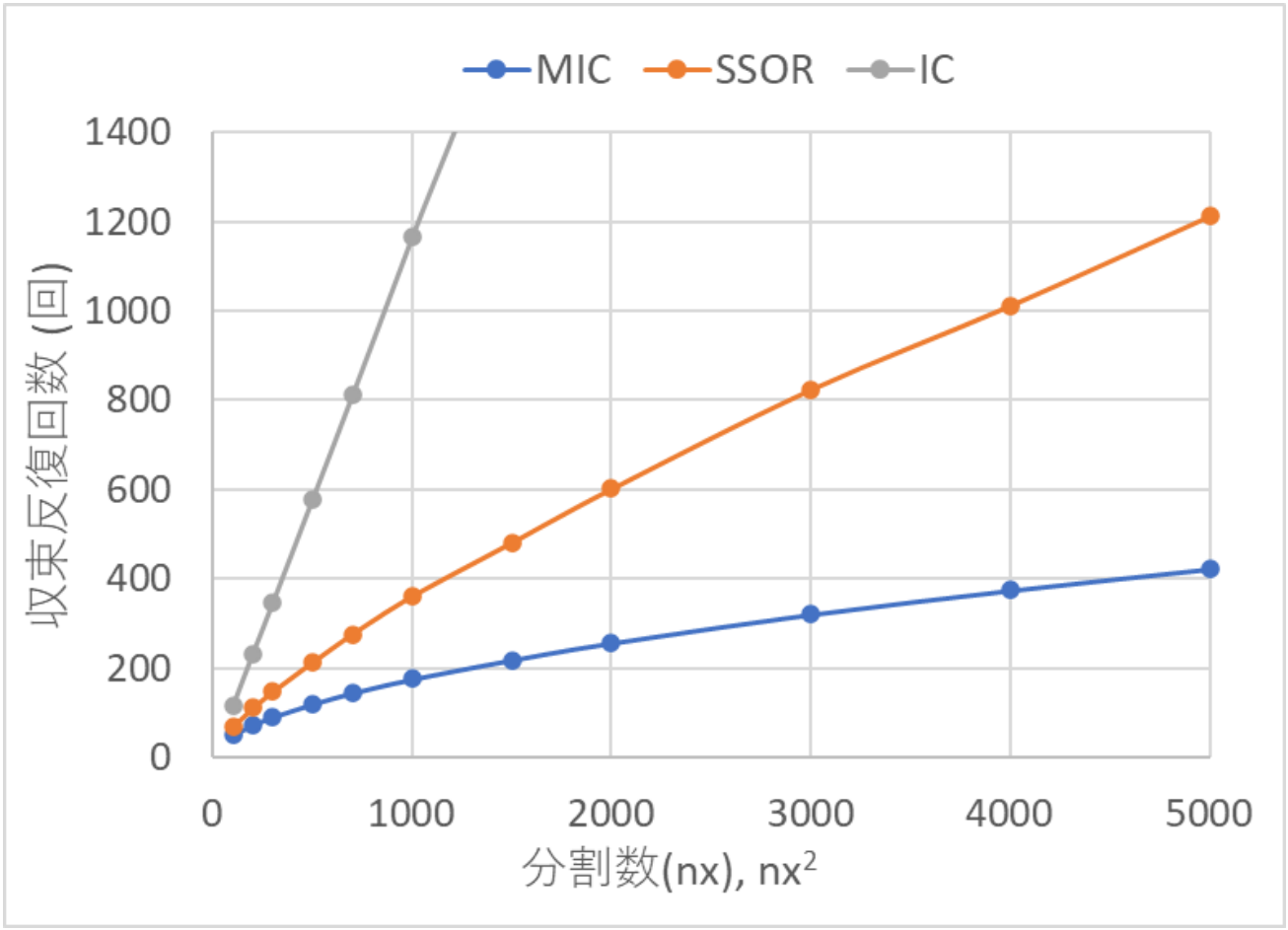
MIC: α
自動
SSOR: ω
最良

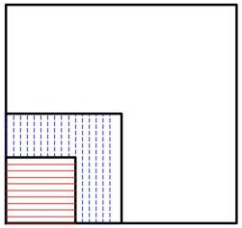
5.1 M2B-B2での比較結果



MIC: α
自動
SSOR: ω
最良

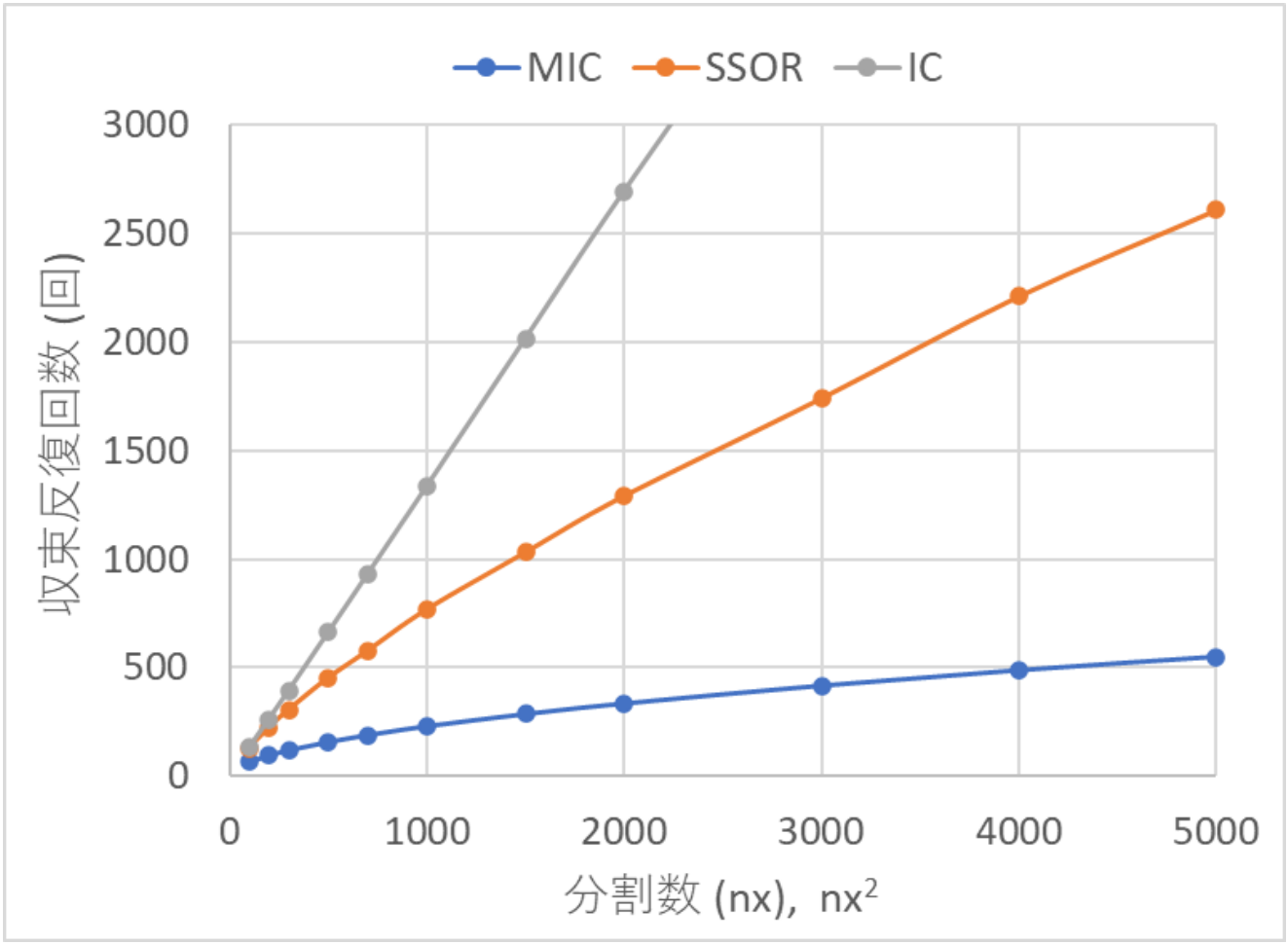
5.1 M2C-B2での比較結果

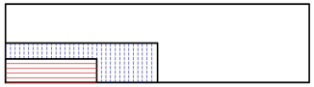




MIC: α
自動
SSOR: ω
最良

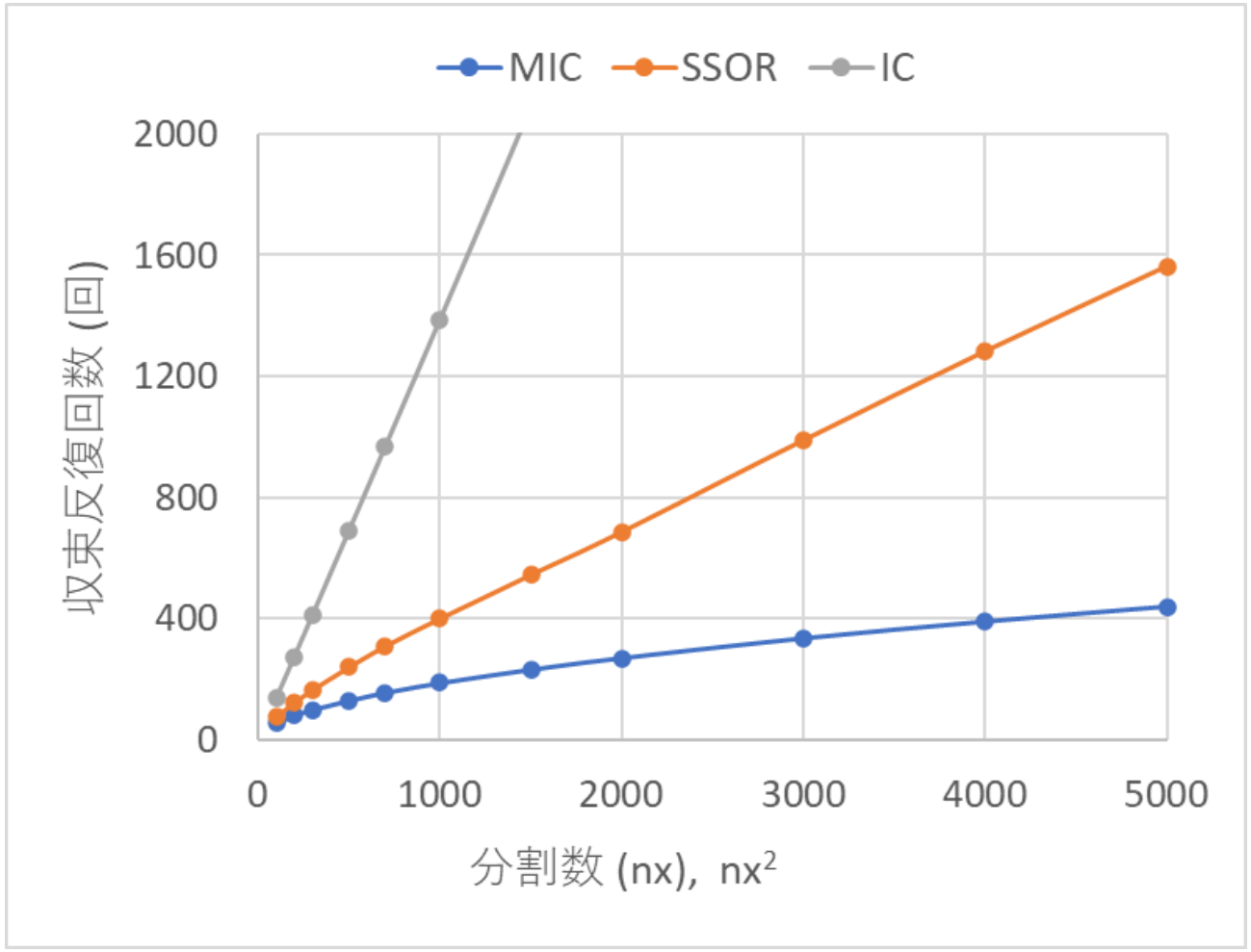
5.1 M2D-B2での比較結果



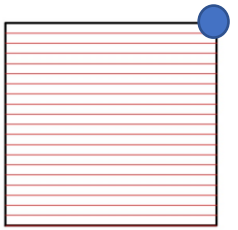


MIC: α
自動
SSOR: ω
最良

5.2 M2A-B4での比較結果

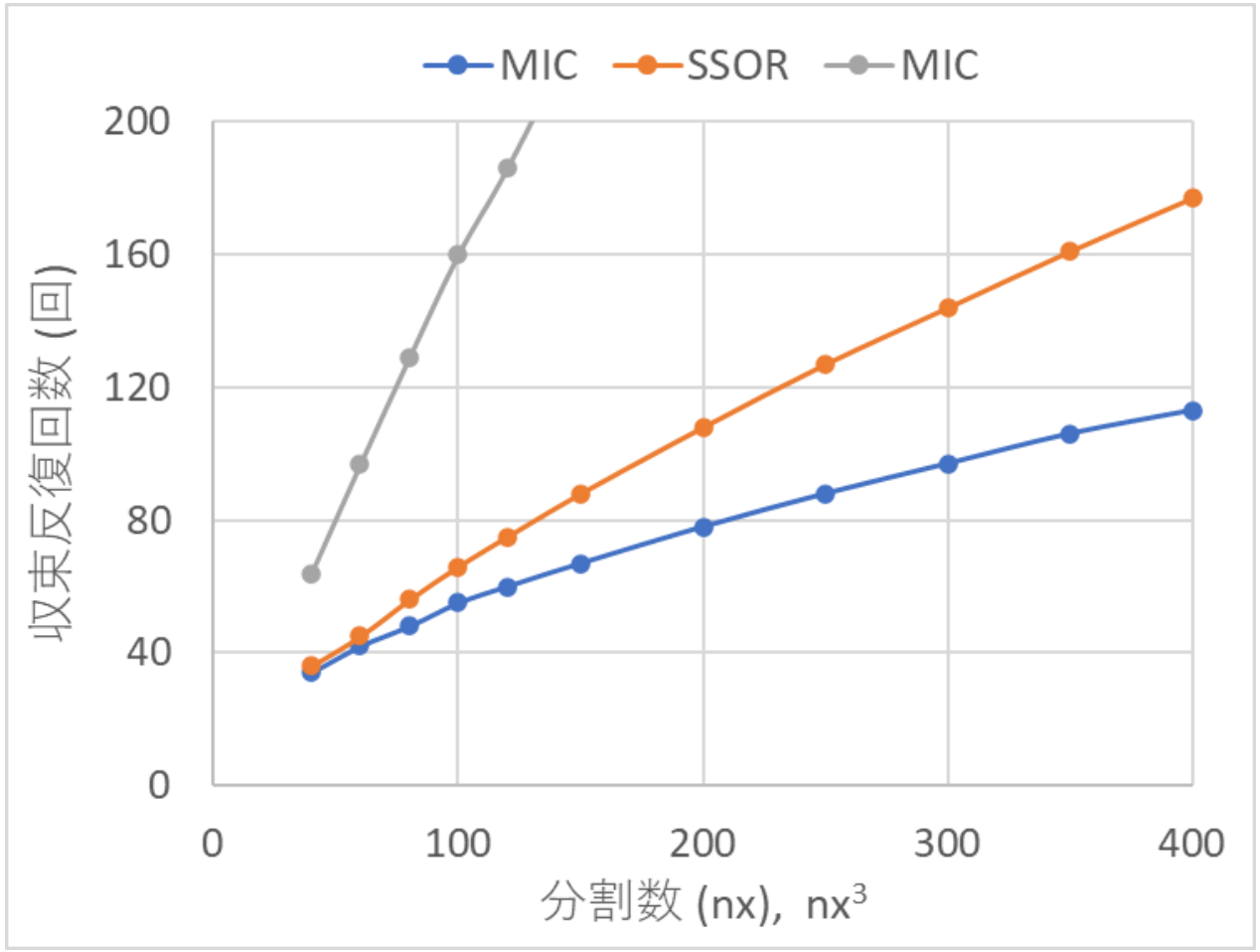


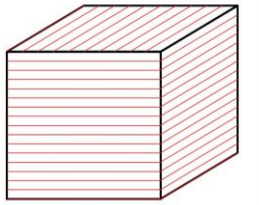
1点固定



MIC: α
自動
SSOR: ω
最良

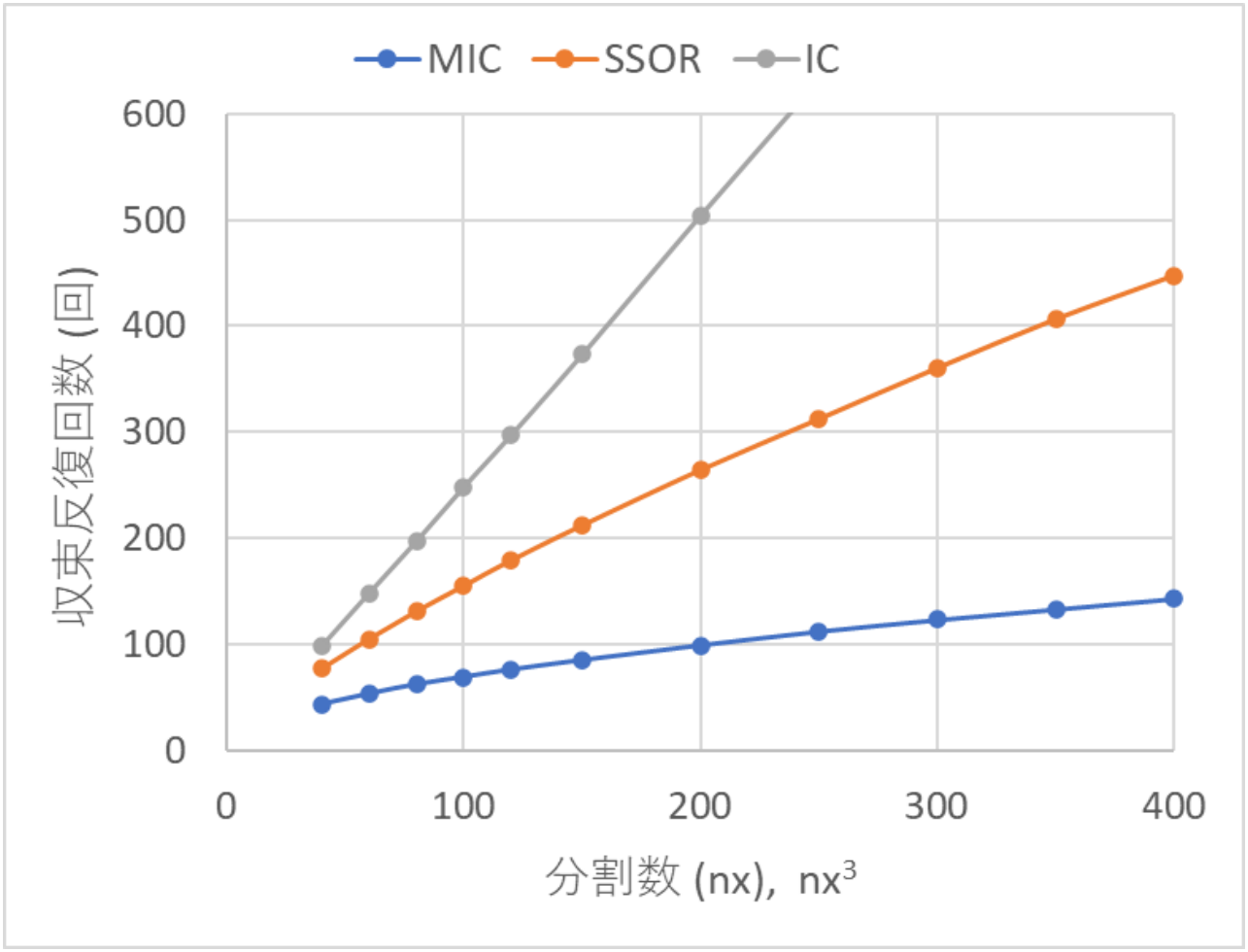
5.3 M3A-B3での比較結果

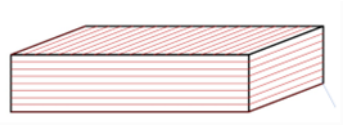




MIC: α
自動
SSOR: ω
最良

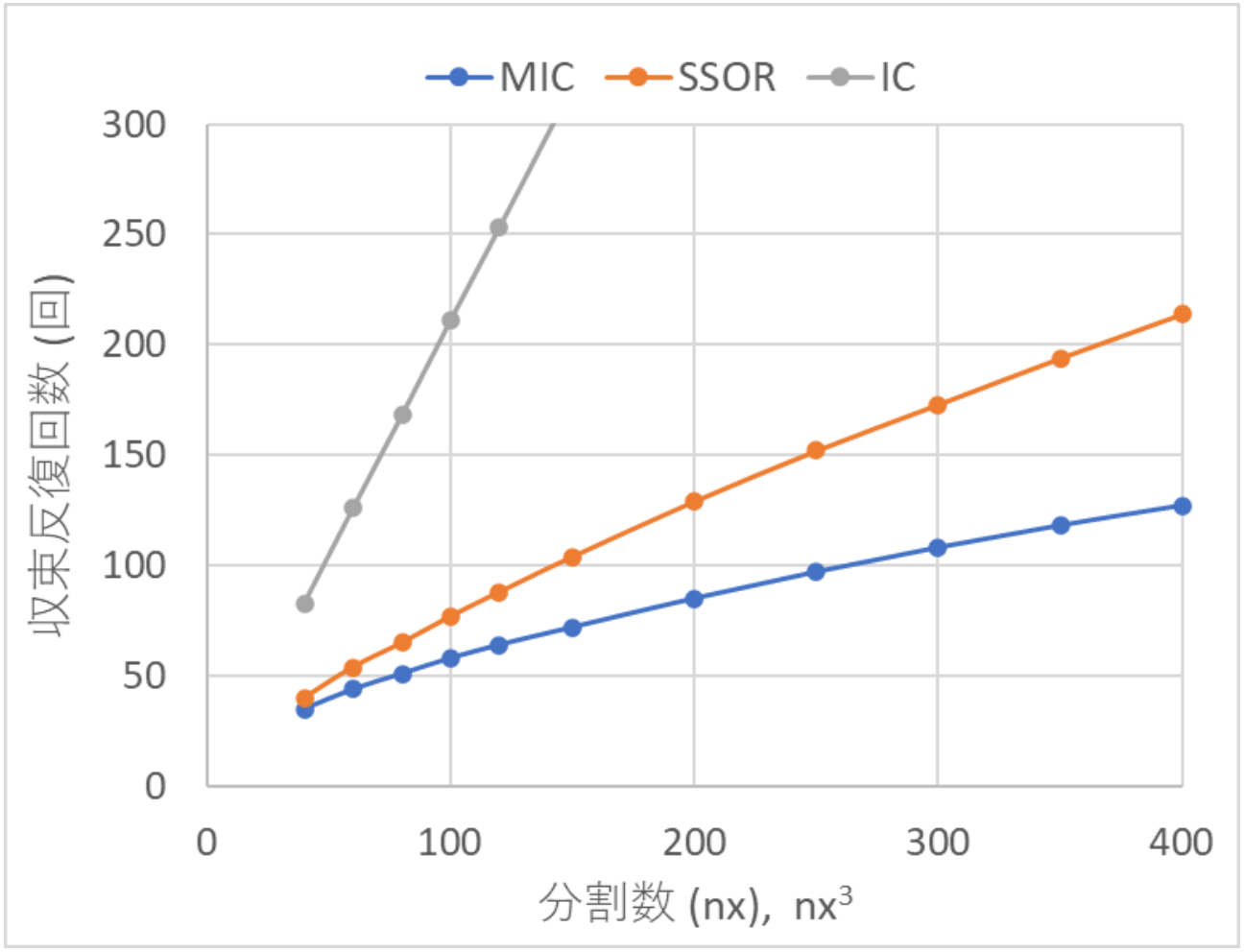
5.3 M3B-B3での比較結果

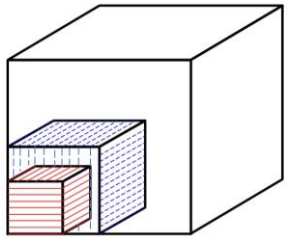




MIC: α
自動
SSOR: ω
最良

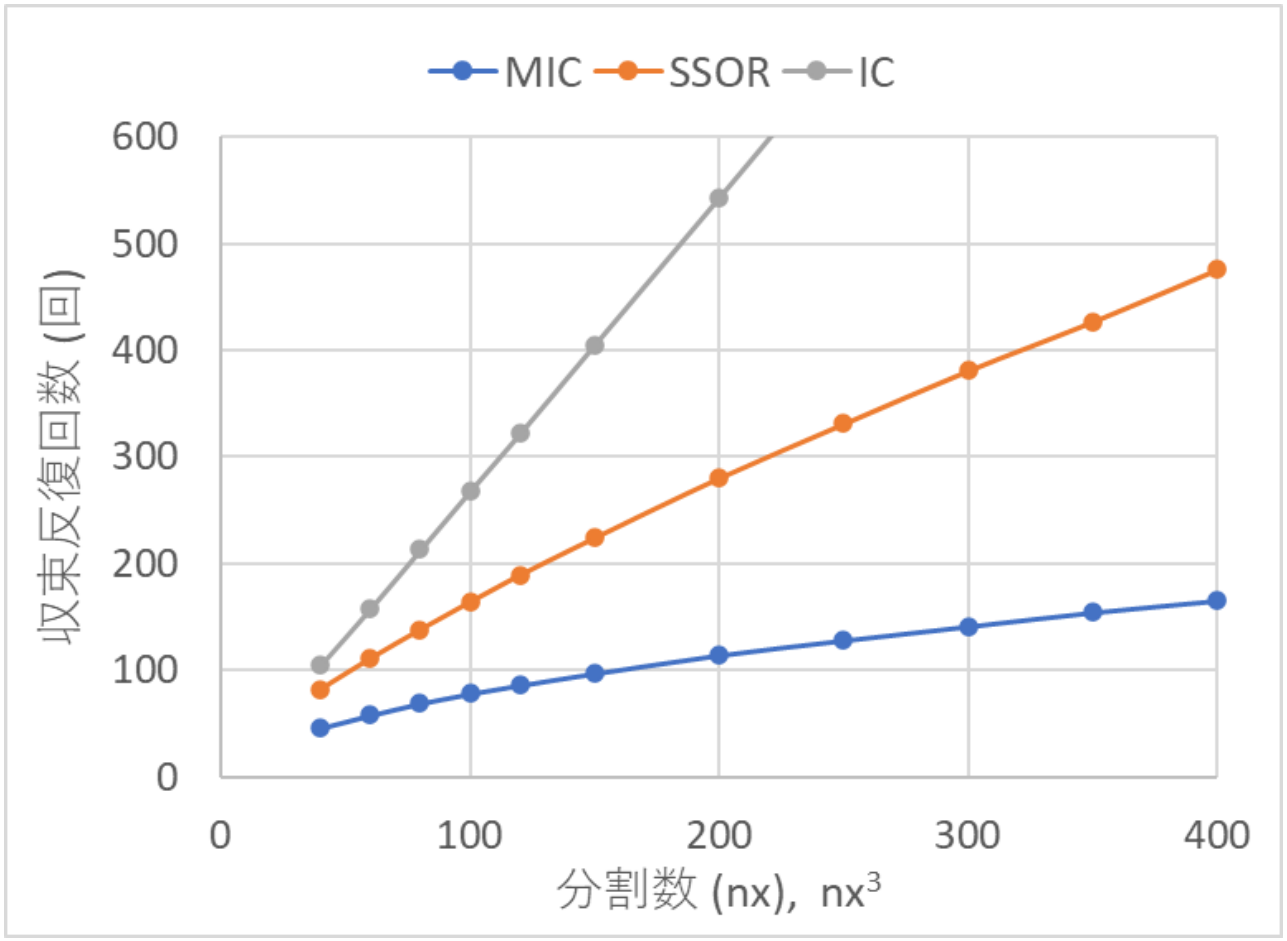
5.3 M3C-B3での比較結果

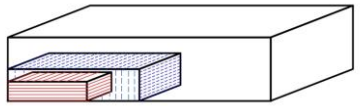




MIC: α
自動
SSOR: ω
最良

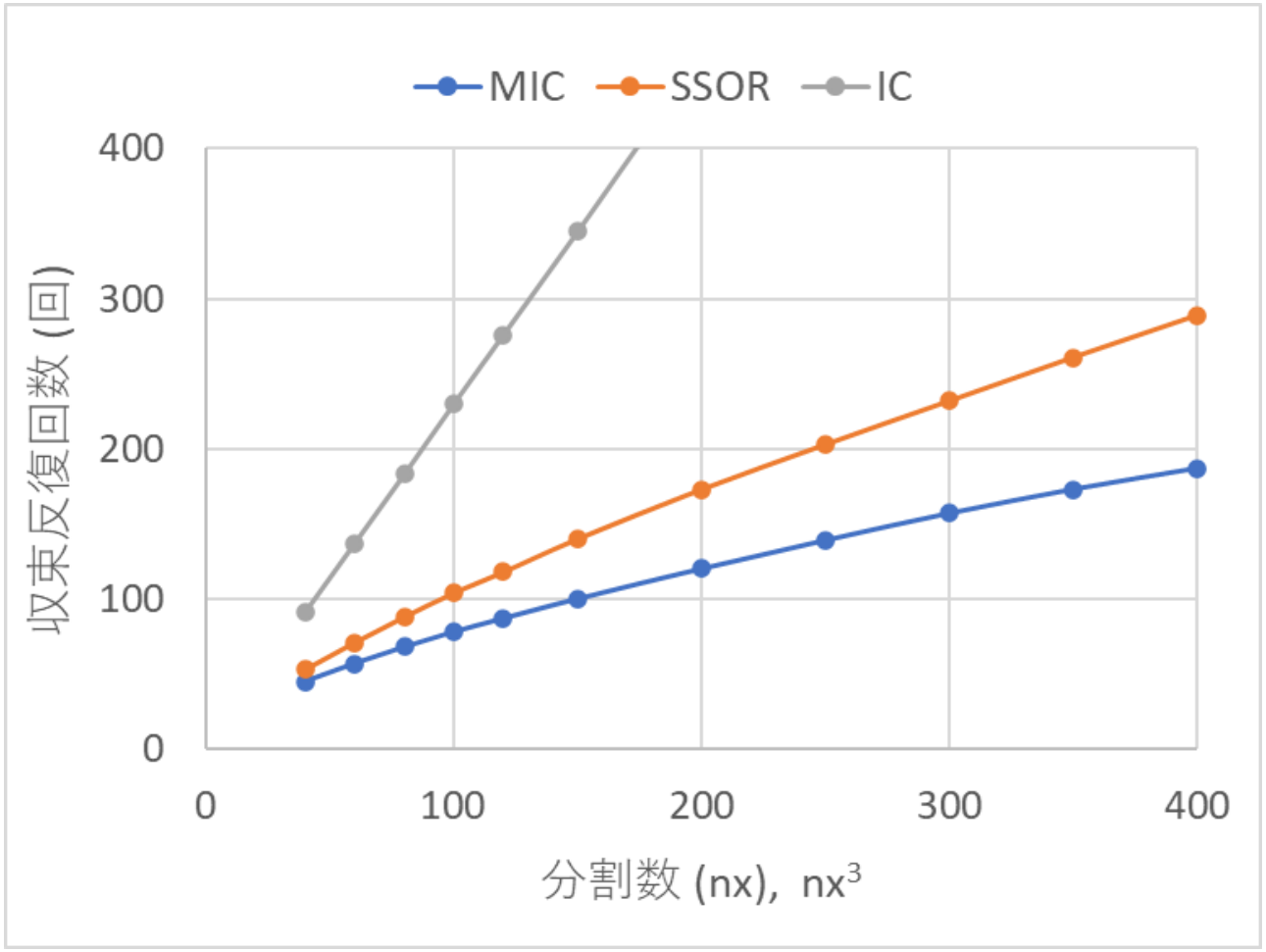
5.3 M3D-B3での比較結果



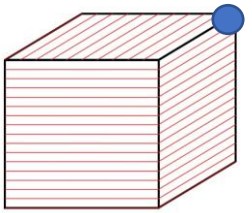


MIC: α
自動
SSOR: ω
最良

5.4 M3A-B6での比較結果



1点固定



MIC: α
自動
SSOR: ω
最良

6 おわりに

- MICCG法のパラメータαを自動設定できた
- 分割数と1−αは対数で比例と判明
- MICCG法はICCG法より圧倒的に有利
- MICCG法とSSOR前処理付きCG法の比較

  最良ωのSSORより自動αのMICが有利

  分割数が大きいほどMICCG法の効果が大

  収束が遅い(複雑)ものほどMICCG法の効果大

  2次元では最大で4.7倍の効果(5000×5000)

  3次元では最大で3.1倍の効果(400×400×400)