

第3章 連立一次方程式の解法

3.1 GMRES 法

3.1.1 GMRES 法概説

第 A.2 章で紹介した Krylov 部分空間法のひとつとして GMRES 法がある。GMRES 法とは、The Generalized Minimum Residual Method の略である。日本語では「一般化された最小残差法」となる。第 A.2 章では、Krylov 部分空間法とは Krylov 部分空間 $K_m(A, r_0)$ の原点を x_0 にとった空間 $x_0 + K_m(A, r_0)$ から（何らかの意味で）最適な解を探す方法であると述べたが GMRES 法では「最適な解」というのは「残差の最も小さい解」を意味する。すなわち GMRES 法は空間 $x_0 + K_m(A, r_0)$ から、最も残差ノルムを小さくする解を探すという方法になる。

3.1.2 GMRES(m) 法

GMRES(m) 法とは、適当な自然数 m を与え、Krylov 部分空間の次元が m になったら、一度空間を初期化して、再び 1 から Krylov 部分空間を構成していく方法である。これは、初期解 x_0 から反復して得られた m 回目の近似解 x_m を再び初期解 x_0 として計算を始める方法と言い換えることもできる。そのため、GMRES(m) 法は「 m 回でリスタートする GMRES 法、Restarted GMRES 法」と呼ばれる。

一般に GMRES 法より GMRES(m) 法の方が、Krylov 部分空間の次元が増えすぎないためメモリ効率及び計算効率などが良いと言われている。また GMRES 法では、もし誤差がほとんど減らない場合になると無限に計算を行うため、現実の実装では GMRES(m) が用いられる。本論文でも以降は GMRES(m) 法について議論していく。また、特に断りが無い限り、GMRES(m) 法を GMRES 法と書く。

3.1.3 GMRES 法の収束性

ここでは、GMRES 法に関する非常に重要な定理を証明する。

定理 3.1.1 (GMRES 法の収束性) 連立一次方程式の係数行列 A に対して $\frac{1}{2}(A + A^T)$ が正定値であるとする。このとき GMRES 法の残差の列 \mathbf{r}_m のノルム $\|\mathbf{r}_m\|$ は反復ごとに必ず減少する。更に、残差ノルムの減少比は以下の式で評価できる。

$$\|\mathbf{r}_k\| \leq \rho^k \|\mathbf{r}_0\| \quad \text{where } \rho = \left(1 - \frac{\underline{a}^2}{\|A\|^2}\right)^{\frac{1}{2}}$$

但し \underline{a} は $\frac{1}{2}(A + A^T)$ の最小固有値とする。

■

$\frac{1}{2}(A + A^T)$ は正定値対称なので最小固有値は正の実数になることに注意。すなわちこの定理は、 A の対称部分さえ正定値であれば、GMRES 法は必ず収束していくことを表す非常に強力な定理である。更に、 A 自体が対称であれば、 \underline{a} は A の最小固有値となり、また補題 A.1.7 から $\|A\| = \rho(A)$ となる ($\|A\|$ が A の最大固有値となる) ので残差の減り具合は条件数 $\text{cond}(A)$ を用いて

$$\|\mathbf{r}_k\| \leq \rho^k \|\mathbf{r}_0\| \quad \text{where } \rho = \left(1 - \frac{1}{\text{cond}(A)}\right)^{\frac{1}{2}}$$

と評価することが出来る。このため、条件数が大きいと収束に時間がかかるということが容易にわかる。以下、この定理 3.1.1 を証明していく。

[証明] ● 証明の方針

ここで注意しておくが、GMRES(m) によって得られる残差の列を $\{\mathbf{r}_k\}$ と定義し、1 回ごとにリスタートをして Krylov 部分空間を次元分しか取らない GMRES(1) の残差の列を $\tilde{\mathbf{r}}_k$ と定義している。明らかに

$$\mathbf{r}_k \leq \tilde{\mathbf{r}}_k \tag{3.1.1}$$

が成り立つ。

以降の論法はこの GMRES(1) の残差 $\tilde{\mathbf{r}}_k$ について議論し、この残差が減少することを示し、式 (3.1.1) を用いて GMRES(m) の残差 \mathbf{r}_k も減少することを示す。本来は GMRES(m) について直接評価すれば、より残差が減少することを保証できる可能性もあ

るが、現実的には難しいのでここでは簡単な GMRES(1) の評価にとどめる。また、初期残差 \mathbf{r}_0 はすべての方法で同じものを用いることとする。

● step1

定理 A.2.3 から、 $\forall \hat{\mathbf{x}} \in \mathbf{x}_0 + \text{span}(\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{n-1}\mathbf{r}_0, A^n\mathbf{r}_0)$ に対する残差 $\hat{\mathbf{r}}$ は適当な k 次の多項式 P_k を用いて

$$\hat{\mathbf{r}} = P_k(A)\mathbf{r}_0, \quad \text{where } P_k(0) = 1 \quad (3.1.2)$$

と書ける。但し、 $\mathbf{r}_0 = \hat{\mathbf{r}}_0$ としている。つまり、Krylov 部分空間法で計算される任意の残差 $\hat{\mathbf{r}}$ は式 (3.1.2) の形で表せるということになる。さて、GMRES 法でもとまる残差の列 $\{\mathbf{r}_k\}$ はその中から特に残差ノルムを最小にするようなものを選び出してくる方法であるので明らかに

$$\|\mathbf{r}_k\| \leq \|\hat{\mathbf{r}}_k\|, \quad \text{for } \forall \hat{\mathbf{r}}_k$$

が成り立つ。つまり

$$\|\mathbf{r}_k\| \leq \|P_k(A)\mathbf{r}_0\|, \quad \forall P_k(0) = 1$$

が成立つ。

さて、先ほどから述べているように GMRES 法で用いる残差は

$$\mathbf{r}_k = P_k(A)\mathbf{r}_0, \quad \text{where } P_k(0) = 1$$

の形で書けることが分かっているので scalar の列 $\{\alpha_k\}$ を用いて

$$\begin{aligned} \mathbf{r}_1 = \hat{\mathbf{r}}_1 &= P_1(A)\mathbf{r}_0 \\ &= (I - \alpha_1 A)\mathbf{r}_0 \end{aligned}$$

と書ける。ここで、 $A\mathbf{r}_0$ で張られる直線を l とし、直線 l への \mathbf{r}_0 の射影を \mathbf{c}_0 とすれば、残差ノルムが最小になる \mathbf{r}_1 は明らかに $\mathbf{r}_1 = \mathbf{r}_0 - \mathbf{c}_0$ でなければならない事が分かる（つまり直線 $\alpha_1 A\mathbf{r}_0$ と \mathbf{r}_1 が直交する）。

よって

$$\alpha_1 = \frac{(\hat{\mathbf{r}}_0, A\hat{\mathbf{r}}_0)}{(A\hat{\mathbf{r}}_0, A\hat{\mathbf{r}}_0)}$$

となることが分かる。更に、このようにして求めた \mathbf{r}_1 を \mathbf{r}_0 として同じ論法を用いていけば、アルゴリズム的に GMRES(1) を用いていることになるので

$$\hat{\mathbf{r}}_k = (I - \alpha_1 A) \cdots (I - \alpha_k A)\mathbf{r}_0$$

となる。但し

$$\alpha_k = \frac{(\tilde{\mathbf{r}}_{k-1}, A\tilde{\mathbf{r}}_{k-1})}{(A\tilde{\mathbf{r}}_{k-1}, A\tilde{\mathbf{r}}_{k-1})}$$

である。また,

$$\frac{\|\tilde{\mathbf{r}}_1\|^2}{\|\mathbf{r}_0\|^2} = \frac{(\tilde{\mathbf{r}}_0 - \frac{(\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)}{(A\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)}A\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0 - \frac{(\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)}{(A\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)}A\tilde{\mathbf{r}}_0)}{(\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0)} \quad (3.1.3)$$

$$= \frac{((\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0) - \frac{(\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)^2}{(A\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)}A\tilde{\mathbf{r}}_0)}{(\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0)} \quad (3.1.4)$$

$$= 1 - \frac{(\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)^2}{(\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0)(A\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)} \quad (3.1.5)$$

が成り立つ。ここで

$$\sigma\left(\frac{1}{2}(A + A^T)\right) \geq \underline{a}$$

が成り立つことを用いれば

$$(\mathbf{r}_0, A\mathbf{r}_0)^2 = \left(\mathbf{r}_0, \frac{1}{2}(A + A^T)\mathbf{r}_0\right)^2 \quad (3.1.6)$$

$$\geq (\mathbf{r}_0, \mathbf{r}_0)^2 \underline{a}^2 \quad (3.1.7)$$

と書ける。一方, $\|A\|$ の定義から

$$(A\mathbf{r}_0, A\mathbf{r}_0) \leq \|A\|^2(\mathbf{r}_0, \mathbf{r}_0) \quad (3.1.8)$$

が成り立つので, 式 (3.1.5) ~ (3.1.8) と合わせれば

$$\frac{\|\tilde{\mathbf{r}}_1\|^2}{\|\mathbf{r}_0\|^2} = 1 - \frac{(\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)^2}{(\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_0)(A\tilde{\mathbf{r}}_0, A\tilde{\mathbf{r}}_0)} \leq 1 - \frac{\underline{a}^2}{\|A\|^2}$$

が示せる。これを繰り返し用いれば

$$\|\mathbf{r}_k\| \leq \rho^k \|\mathbf{r}_0\| \quad \text{where } \rho = \left(1 - \frac{\underline{a}^2}{\|A\|^2}\right)^{\frac{1}{2}}$$

となり, 題意が示せた。

□

3.1.4 Hessenberg 行列と Harmonic Ritz Value

まずは GMRES 法のアルゴリズムの中で登場する Hessenberg 行列について紹介する。

定義 3.1.2 (Hessenberg 行列) Hessenberg 行列 H_m 並びにそれを元に作られる \bar{H}_m を以下の式で定義する.

$$H_m = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ 0 & h_{3,2} & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & h_{m-1,m-1} & \vdots \\ 0 & \cdots & 0 & h_{m,m-1} & h_{m,m} \end{pmatrix}$$

$$\bar{H}_m = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ 0 & h_{3,2} & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & h_{m-1,m-1} & \vdots \\ 0 & \cdots & 0 & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \cdots & 0 & h_{m+1,m} \end{pmatrix}$$

■

すなわち H_m の最下段に 1 行分付け加えられた行列が \bar{H}_m である.

Hessenberg 行列は GMRES 法のアルゴリズムの中で計算される行列である (第 3.1.5 小節参照). 実はこの H_m 並びに \bar{H}_m を用いて計算した

$$F \equiv H_m^{-1} \bar{H}_m^{-1} \bar{H}_m$$

の固有値が元の連立一次方程式の係数行列 A の固有値の特徴と非常に似ている (最大固有値や最小固有値などの値がほぼ同じ) ということが知られている. この詳しい話については片桐^[26]を参照していただきたい. F の固有値は Harmonic Ritz Value と呼ばれる.

係数行列 A の次元が大きくなると固有値の個数も膨大になり, 実際に固有値分布を求めるのは計算量の面から考えて非常に難しい (理論的には時間はかかるが求まるが). そこで今回は, GMRES 法を使用した際に生成された F の固有値分布 (固有値の個数は高々反復階数の個数) を係数行列 A の固有値分布とみなして結果を考察している.

3.1.5 GMRES 法の具体的なアルゴリズム

まず、初期解 \mathbf{x}_0 を適当に定める。次に、 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ を計算する。この \mathbf{r}_0 と A を用いて K_M という空間を以下のように定義する。

$$K_m = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}$$

定義の仕方からもわかるとおり、 K_m は m 次元の空間である。

Krylov 部分空間法というのは、 $\mathbf{x}_0 + K_m$ という空間の中から最も最適な解（残差ノルムが最小になる解）を探してくるという方法である。

以下ではこの空間 $\mathbf{x}_0 + K_m$ のなかでの最適解（残差最小解）を発見するアルゴリズムを説明する。

まず、空間 $\mathbf{x}_0 + K_m$ の正規直交基底を \mathbf{v}_i , $i = 1, \dots, m$ とする。するとこの空間の中のすべての vector \mathbf{x} は

$$\mathbf{x} = \mathbf{x}_0 + y_1\mathbf{v}_1 + y_2\mathbf{v}_2 + \dots + y_m\mathbf{v}_m, \quad y_i \in R, \quad \mathbf{v} \in R^n$$

とかける。ここで

$$V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m], \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

と定義すれば、

$$\mathbf{x} = \mathbf{x}_0 + V_m\mathbf{y}$$

とも書くことができる。また、 $AV_m = V_{m+1}\bar{H}_m$ が成り立つことが知られているので、これらを用いれば、残差ノルムは以下のように変形出来る。

$$\begin{aligned} \|\mathbf{b} - A\mathbf{x}\|_2 &= \|\mathbf{b} - A(\mathbf{x}_0 + V_m\mathbf{y})\|_2 \\ &= \|\mathbf{r}_0 - AV_m\mathbf{y}\|_2 \\ &= \|\beta\mathbf{v}_1 - V_{m+1}\bar{H}_m\mathbf{y}\|_2 \\ &= \|V_{m+1}(\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y})\|_2 \\ &= \|\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y}\|_2 \end{aligned}$$

なお、途中で $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$ と $\|V_{m+1}\|_2 = 1$ を用いた。この、 $\|\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y}\|_2$ を $J(\mathbf{y})$ と定義する。つまり、「 $\mathbf{x}_0 + K_m$ の中から残差ノルムを最小にするような \mathbf{x} を探し出す」ということは、「 $J(\mathbf{y})$ を最小にするような \mathbf{y} を探し出してくる」ということと全く同値なわけである。

もし、このような \mathbf{y} が見つければ、 $\mathbf{x} = \mathbf{x}_0 + V_m\mathbf{y}$ に代入して、残差最小の近似解 \mathbf{x} が無事求まることになる。

ここまでで説明してきた GMRES を簡単にまとめると以下のようなになる。

- $\mathbf{x}_0 + K_m$ のなかでの最適解（残差最小解）を見つける \Leftrightarrow
 - $\mathbf{x} = \mathbf{x}_0 + V_m\mathbf{y}$ と表示される \mathbf{x} に対して $\|\mathbf{b} - A\mathbf{x}\|_2$ が最小になるものを探す。 \Leftrightarrow
 - $J(\mathbf{y}) = \|\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y}\|_2$ を最小にするような \mathbf{y} を探し出してくる。
- これらを含めた GMRES のすべての流れを書くと、以下のようなになる。

GMRES の algorithm

1. Compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\beta}$
2. Define the $(m+1) \times m$ matrix H_m ; Set $H_m = 0$
3. For $j = 1, m$ Do:
4. Compute $\mathbf{w}_j = A\mathbf{v}_j$
5. For $i = 1, j$ Do:
6. $h_{ij} = (\mathbf{w}_j, \mathbf{v}_i)$
7. $\mathbf{w}_j = \mathbf{w}_j - h_{ij}\mathbf{v}_i$
8. EndDo
9. Compute $h_{j+1,j} = \|\mathbf{w}_j\|_2$ If $h_{j+1,j} = 0$ set $m = j$ and Goto 12
10. Compute $\mathbf{v}_{j+1} = \frac{\mathbf{w}_j}{h_{j+1,j}}$
11. End Do
12. Compute \mathbf{y}_m the minimizer of $\|\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m$

3.1.6 最小値問題の解法

ここまでの話で、

$J(\mathbf{y}) = \|\beta\mathbf{e}_1 - \bar{H}_m\mathbf{y}\|_2$ を最小にするような \mathbf{y} を探し出せばよい。

ということはわかった。ではそのような y (以下では y_m と書く) をどのようにして探していくかを以下で紹介していく。まず,

$$\beta e_1 - \bar{H}_m y = \begin{pmatrix} \beta \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ 0 & h_{3,2} & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & h_{m-1,m-1} & \vdots \\ 0 & \cdots & 0 & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \cdots & 0 & h_{m+1,m} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

である。ここで、現時点では \bar{H}_m の部分が上三角行列に近い形 (対角成分が 1 行分はみ出している。) になっているが、これを完全に上三角行列にすることによって、計算がスムーズに進むようになる。

\bar{H}_m の上三角行列化

ここでは簡単のために、 $m=5$ として計算を行う。まず、以下のような回転行列 Ω_1 を考える。

$$\Omega_1 = \begin{pmatrix} c_1 & s_1 & 0 & 0 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

ちなみに,

$$s_1 = \frac{h_{2,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}}, \quad c_1 = \frac{h_{1,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}}$$

である（対角成分の1つ下が0になっていくようにうまく取っておく.）. これを H_5 に左からかける. かけたものを $\bar{H}_5^{(1)}$ と書くことにする.

$$\begin{aligned} \bar{H}_5^{(1)} &= \Omega_1 \bar{H}_5 = \begin{pmatrix} c_1 & s_1 & 0 & 0 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} \\ 0 & 0 & h_{4,3} & h_{4,4} & h_{4,5} \\ 0 & 0 & 0 & h_{5,4} & h_{5,5} \\ 0 & 0 & 0 & 0 & h_{6,6} \end{pmatrix} \\ &= \begin{pmatrix} h_{1,1}^{(1)} & h_{1,2}^{(1)} & h_{1,3}^{(1)} & h_{1,4}^{(1)} & h_{1,5}^{(1)} \\ 0 & h_{2,2}^{(1)} & h_{2,3}^{(1)} & h_{2,4}^{(1)} & h_{2,5}^{(1)} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} \\ 0 & 0 & h_{4,3} & h_{4,4} & h_{4,5} \\ 0 & 0 & 0 & h_{5,4} & h_{5,5} \\ 0 & 0 & 0 & 0 & h_{6,6} \end{pmatrix} \end{aligned}$$

次に

$$s_2 = \frac{h_{3,2}}{\sqrt{h_{2,2}^2 + h_{3,2}^2}}, \quad c_2 = \frac{h_{2,2}}{\sqrt{h_{2,2}^2 + h_{3,2}^2}}$$

として

$$\Omega_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & s_2 & 0 & 0 & 0 \\ 0 & -s_2 & c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

を定義し, $\bar{H}_5^{(1)}$ に左からかける. 以下, この作業を繰り返していくと対角成分の1つ下がすべて0になっていくことがわかる. 具体的には以下のようなになる.

$$\bar{H}_5^{(5)} = \Omega_5 \Omega_4 \Omega_3 \Omega_2 \Omega_1 \bar{H}_5 = \begin{pmatrix} h_{1,1}^{(5)} & h_{1,2}^{(5)} & h_{1,3}^{(5)} & h_{1,4}^{(5)} & h_{1,5}^{(5)} \\ 0 & h_{2,2}^{(5)} & h_{2,3}^{(5)} & h_{2,4}^{(5)} & h_{2,5}^{(5)} \\ 0 & 0 & h_{3,3}^{(5)} & h_{3,4}^{(5)} & h_{3,5}^{(5)} \\ 0 & 0 & 0 & h_{4,4}^{(5)} & h_{4,5}^{(5)} \\ 0 & 0 & 0 & 0 & h_{5,5}^{(5)} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

ちなみに, 以下のように $\bar{\mathbf{g}}_5$ も定義しておく.

$$\bar{\mathbf{g}}_5 = \Omega_5 \Omega_4 \Omega_3 \Omega_2 \Omega_1 (\beta \mathbf{e}_1) = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{pmatrix}$$

よって, $\|\Omega_i\|_2 = 1$ に注意すれば,

$$\begin{aligned} \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2 &= \|\Omega_5\|_2 \|\Omega_4\|_2 \|\Omega_3\|_2 \|\Omega_2\|_2 \|\Omega_1\|_2 \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2 \\ &= \|\Omega_5 \Omega_4 \Omega_3 \Omega_2 \Omega_1 (\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y})\|_2 \\ &= \|\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y}\|_2 \end{aligned}$$

と変形できた. つまり,

● $J(\mathbf{y}) = \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{y}\|_2$ を最小にするような \mathbf{y} を探し出してくる. \Leftrightarrow

● $\|\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y}\|_2$ を最小にするような \mathbf{y} を探し出してくる.

ということがわかった.

$\|\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y}\|_2$ を最小にする y の求め方

$\|\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y}\|_2$ を最小にする y の求めるのは非常に簡単である。まずは、 $\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y}$ を成分で書くと

$$\begin{aligned} \bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y} &= \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \\ \gamma_6 \end{pmatrix} - \begin{pmatrix} h_{1,1}^{(5)} & h_{1,2}^{(5)} & h_{1,3}^{(5)} & h_{1,4}^{(5)} & h_{1,5}^{(5)} \\ 0 & h_{2,2}^{(5)} & h_{2,3}^{(5)} & h_{2,4}^{(5)} & h_{2,5}^{(5)} \\ 0 & 0 & h_{3,3}^{(5)} & h_{3,4}^{(5)} & h_{3,5}^{(5)} \\ 0 & 0 & 0 & h_{4,4}^{(5)} & h_{4,5}^{(5)} \\ 0 & 0 & 0 & 0 & h_{5,5}^{(5)} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1 - h_{1,1}y_1 - h_{1,2}y_2 - h_{1,3}y_3 - h_{1,4}y_4 - h_{1,5}y_5 \\ \gamma_2 - h_{2,2}y_2 - h_{2,3}y_3 - h_{2,4}y_4 - h_{2,5}y_5 \\ \gamma_3 - h_{3,3}y_3 - h_{3,4}y_4 - h_{3,5}y_5 \\ \gamma_4 - h_{4,4}y_4 - h_{4,5}y_5 \\ \gamma_5 - h_{5,5}y_5 \\ \gamma_6 \end{pmatrix} \end{aligned}$$

よってこのノルムを取れば,

$$\begin{aligned} \|(\bar{\mathbf{g}}_5 - \bar{H}_5^{(5)} \mathbf{y})\|_2^2 &= (\gamma_1 - h_{1,1}y_1 - h_{1,2}y_2 - h_{1,3}y_3 - h_{1,4}y_4 - h_{1,5}y_5)^2 \\ &+ (\gamma_2 - h_{2,2}y_2 - h_{2,3}y_3 - h_{2,4}y_4 - h_{2,5}y_5)^2 \\ &+ (\gamma_3 - h_{3,3}y_3 - h_{3,4}y_4 - h_{3,5}y_5)^2 \\ &+ (\gamma_4 - h_{4,4}y_4 - h_{4,5}y_5)^2 \\ &+ (\gamma_5 - h_{5,5}y_5)^2 \\ &+ \gamma_6^2 \end{aligned}$$

ここで、 γ や h は定数であることに注意すれば、 $\|\bar{g}_5 - \bar{H}_5^{(5)}y\|_2$ の値を最小にする y は明らかに以下のようになる (y_5 から順々に決まっていく.)

$$\begin{aligned} y_5 &= \frac{1}{h_{5,5}} \times \gamma_5 \\ y_4 &= \frac{1}{h_{4,4}} \times (\gamma_4 - h_{4,5}y_5) \\ y_3 &= \frac{1}{h_{3,3}} \times (\gamma_3 - h_{3,4}y_4 - h_{3,5}y_5) \\ y_2 &= \frac{1}{h_{2,2}} \times (\gamma_2 - h_{2,3}y_3 - h_{2,4}y_4 - h_{2,5}y_5) \\ y_1 &= \frac{1}{h_{1,1}} \times (\gamma_1 - h_{1,2}y_2 - h_{1,3}y_3 - h_{1,4}y_4 - h_{1,5}y_5) \end{aligned}$$

またそのときの $\|\bar{g}_5 - \bar{H}_5^{(5)}y\|_2$ の値は明らかに γ_6 であることもわかる。ちなみに、この γ_6 の値 (一般には γ_{m+1}) は収束判定にも利用できる貴重な値である。

あとは、ここでもとまった y_i を縦に並べた vectory(= y_m) を

$$x = x_0 + V_m y$$

に代入して計算をすれば、近似解が求まる。以上が、基本的な GMRES の説明である。

3.2 前処理

3.2.1 前処理

前処理とは解くべき連立一次方程式

$$Ax = b$$

を前処理行列 M を用いて

$$M^{-1}Ax = M^{-1}b$$

と変形し計算を行う方法である。すなわち、係数行列として元の A ではなく $M^{-1}A$ を用いた問題を解くことになる。

前の節までの議論で、係数行列 A の対称部が正定値でありさえすれば GMRES 法は理論上は収束することが言えた。だが、実際にはそれだけでは収束のスピードは上がらず、現実的に収束させることは困難なことも多い。また、たとえ前処理無しで収束したとしても、前処理によって固有値分布・条件数を改善し、収束性を劇的に上昇させることもできる。

また、もし前処理により係数行列 A の固有値を実軸を中心とした半径 1 の円に入れることができれば、いわゆる古典的の反復法でも連立一次方程式を解くことができる。

以上のような理由から、係数行列 A に対する前処理の適用並びに前処理の研究というものは非常に重要であるということがわかる。

一般的には基本反復法ソルバー単独ではなく、前処理と基本反復ソルバーを組み合わせ使用していくことがほとんどである。そのため、前処理と基本反復ソルバーを合わせて「ソルバー」と呼ぶこともある。本論文においても「ソルバー」は前処理までを含めた基本反復ソルバーとして呼ぶことにする。従って、本論文は表題のとおり前処理の研究ではあるものの、新たなソルバーを開発したなどの表現を用いることもある。しかしながら基本反復ソルバーとしては既に開発されている GMRES 法を用いている。本論文のテーマはあくまでも前処理の開発であり、基本反復ソルバーの開発ではないことに注意してほしい。

3.2.2 ILU 前処理

ここでは、本論文において使用している混合型 FEM 向き ILU(不完全) 前処理行列について述べる。まずはじめに、通常の (完全)LU 分解について解説する。次に、予め定められた非ゼロパターン上のみフィルインを許す ILU 分解について説明する。最後に、混合型 FEM で生じる問題用に有用な ILU 前処理を説明する。そこでは、物理的な情報(節点の種類)を利用し、行列の並びを定め、また上記非ゼロパターンも節点の種類の情報をもとに決められる。

LU 分解

LU 分解は狭義下三角行列 L 、狭義上三角行列 U 、並びに対角行列 D を用いて行列 A を

$$A = (L + D)D^{-1}(D + U) \quad (3.2.1)$$

と分解する方法である。 $A = (a_{i,j})$, $L = \text{low}(q_{i,j})$, $D = \text{diag}(q_{i,i})$, $U = \text{upp}(q_{j,i})$ として右辺ファクタ L, D, U 内のすべての成分を $Q = (q_{ij})$ という正方行列の一部として表すことにする。すなわち、ここで low は Q の狭義下三角部のみを取り出しその他の成分をゼロとすること、 diag は対角部のみを取り出しその他の成分をゼロとすること、 upp

は狭義上三角部のみを取り出しその他の成分をゼロとすることを意味するものとする。このとき式 (3.2.1) は、以下の式がすべての (i, j) に対して成立することと等価となる。

$$q_{i,j} = a_{i,j} - \sum_{k < \min(i,j)} q_{i,k} q_{k,k}^{-1} q_{k,j} \quad (3.2.2)$$

上記の等式からわかるように、 $a_{i,j} = 0$ であっても $q_{i,j} \neq 0$ となる可能性もあり、このようにして L, U 内に新たに生じる非ゼロ成分はフィルイン (fill-in) と呼ばれている。通常疎行列の LU 分解では、そのオーダリング (行と列の並び) をいかに工夫してももとの行列 A の非ゼロ成分の数を上回るオーダーの非ゼロ成分が L, U 内に生成される。これが、直接法が大規模な疎行列問題に対して適さない理由である。上記 $Q = L + D + U$ の成分は具体的には、例えば以下のアルゴリズムにより計算できる。

LU 分解アルゴリズム

```

for  $i = 1, \dots, n$ ;
  for  $j = 1, \dots, n$ ;
     $q_{i,j} := a_{i,j}$ ;
  end  $j$ ;
end  $i$ ;
for  $i = 1, \dots, n$ ;
  for  $k = 1, \dots, i - 1$ ;
    for  $j = k + 1, \dots, n$ ;
       $q_{i,j} := q_{i,j} - q_{i,k} q_{k,k}^{-1} q_{k,j}$ ;
    end  $j$ ;
  end  $k$ ;
end  $i$ ;

```

一旦行列 A が上記のように LU 分解されると連立一次方程式 $Ax = b$ は、次の二つの方程式の解を求めることにより解かれる。

$$(L + D)s = b, \quad (3.2.3)$$

$$D^{-1}(D + U)x = (I + D^{-1}U)x = s. \quad (3.2.4)$$

上記方程式は、以下のような前進代入 (式 (3.2.3) を解く部分) および後退代入 (式 (3.2.4) を解く部分) を実行することにより解かれる。

LU 分解後の解法

```
for  $i = 1, \dots, n$ ;  
     $w_i := b_i$ ;  
    for  $j = 1, \dots, i - 1$ ;  
         $w_i := w_i - q_{i,j}s_j$ ;  
    end  $j$ ;  
     $s_i := q_{i,i}^{-1}w_i$ ;  
end  $i$ ;  
for  $i = n, \dots, 1$ ;  
     $w_i := 0$ ;  
    for  $j = i + 1, \dots, n$ ;  
         $w_i := w_i + q_{i,j}x_j$ ;  
    end  $k$ ;  
     $x_i := s_i - q_{i,i}^{-1}w_i$ ;  
end  $i$ ;
```

ILU 分解前処理

ILU(不完全 LU) 分解では、 A の近似行列を作ることを目標に $\{(i, j) | 1 \leq i, j \leq n\}$ 内のある部分集合 Ω 内の成分に対してのみフィルインを許して ILU 分解を行い、それに対して前進後退代入を行うことにより方程式を近似的に解く。特に k の前進後退代入演算は ILU 前処理と呼ばれる。通常 Ω はもとの行列の非ゼロ成分全体を含む。以下に ILU 分解と前進後退代入のアルゴリズムを示す。前小節の LU 分解と異なるところは計算すべき成分が Ω に限られていることである。

ILU 分解

```
for  $i = 1, \dots, n$ ;  
    for  $j = 1, \dots, n$  only for  $(i, j) \in \Omega$ ;  
         $q_{i,j} := a_{i,j}$ ;  
    end  $j$ ;  
end  $i$ ;
```

```

for  $i = 1, \dots, n$ ;
  for  $k = 1, \dots, i - 1$  only for  $(i, k) \in \Omega$ ;
    for  $j = k + 1, \dots, n$  only for  $(k, j) \in \Omega$ ;
      if  $((i, j) \in \Omega)$   $q_{i,j} := a_{i,j} - q_{i,k}q_{k,k}^{-1}q_{k,j}$ ;
    end  $j$ ;
  end  $k$ ;
end  $i$ ;

```

ILU 前処理

```

for  $i = 1, \dots, n$ ;
   $w_i := b_i$ ;
  for  $j = 1, \dots, i - 1$  only for  $(i, j) \in \Omega$ ;
     $w_i := w_i - q_{i,j}s_j$ ;
  end  $j$ ;
   $s_i := q_{i,i}^{-1}w_i$ ;
end  $i$ ;
for  $i = n, \dots, 1$ ;
   $w_i := 0$ ;
  for  $j = i + 1, \dots, n$  only for  $(i, j) \in \Omega$ ;
     $w_i := w_i + q_{i,j}x_j$ ;
  end  $k$ ;
   $x_i := s_i - q_{i,i}^{-1}w_i$ ;
end  $i$ ;

```

定式化の上では、式(3.2.2)の代わりに

$$\begin{cases} q_{i,j} = a_{i,j} - \sum_{k < \min(i,j)} q_{i,k}q_{k,k}^{-1}q_{k,j} & \text{if } (i, j) \in \Omega \\ q_{i,j} = 0 & \text{if } (i, j) \notin \Omega \end{cases} \quad (3.2.5)$$

という条件のもと L, D, U の成分 $q_{i,j}$ は一意的に定まる。上式は $(L + D)D^{-1}(D + U)$ の Ω 上の各成分が A の成分と等しくなること、および Ω 以外では、 L, U の成分がゼロとなることを要求するものである。ここで Ω はいつもすべての対角成分を含むものとする。

混合型 FEM 用 ILU 前処理

ここでは混合型 FEM にも有効な、物理的な情報を考慮した ILU 前処理行列について述べる。添え字 i, j などは節点に対応するものとし、節点単位の ILU 分解を考える。したがって行列の成分 $q_{i,j}$ などは小行列となる。ここで、以下の特徴を持つ。

1. ILU 分解時に行列 A を節点単位で表示する際に、節点の種類(流速, 変位, 圧力)ごとにまとめて並べる。
2. 節点の組み合わせにより、フィルインの制御方法を定める。

まず、 A の成分表示 $A = (a_{i,j})$ において、それを 2×2 のブロックに分け、第一ブロックに流速または変位を、第二ブロックに圧力を並べることにし、さらに列についても、行の並べ替えと同じもの適用するものとする。また、 kind という節点上の整数値をとる関数を以下のように定める。

$$\text{kind}(i) = \begin{cases} 1 & i \text{ が流速または変位節点のとき} \\ 2 & i \text{ が圧力節点のとき} \end{cases} \quad (3.2.6)$$

式 (3.2.2) を見ればわかるが、フィルインは、 $i, k, j (k < \min(i, j))$ の組み合わせから定まる。本論文では、文献 [1] の方法に従い以下のようなアルゴリズムにしたがってフィルインを定めている。

Ω の定め方

$\Omega := \Omega_0 (= A \text{ の非ゼロパターン});$

for $i = 1, \dots, n; k_1 = \text{kind}(i);$

 for $k = 1, \dots, i - 1$ only $(i, k) \in \Omega; k_2 = \text{kind}(k);$

 for $j = k + 1, \dots, n$ only $(k, j) \in \Omega; k_3 = \text{kind}(j);$

 if $((k_1, k_2, k_3) = (2, 1, 2)) \Omega := \Omega \cup (i, j);$

 end $j;$

 end $k;$

end $i;$

上記アルゴリズムの Ω の決定法を以下のように言い表すこともできる。

1. (2,2) ブロック上に (2,1),(1,2) ブロックの組み合わせから生成されるフィルインをいつでも許す。

2. 上記以外のフィルインを許さない。特に (1,1),(2,2) ブロックの非ゼロパターンはもとの行列の非ゼロパターンと一致する。

混合型 FEM 用 ILU 前処理の比較

ここでは鷺尾らが開発した心臓シミュレータ向け前処理付反復解法で採用している ILU 前処理^[1] 並びに Wille が開発した混合型 FEM 用 ILU 前処理^[11] の特徴を比較したい。

鷺尾の ILU 前処理の特徴を挙げると以下のようなになる。

1. 必要なデータは行列データ，各節点と未知数の対応，並びに節点の種類である。
2. 圧力部に生じる fill-in をすべて取り込んでいる。
3. どのような補間要素にも適用可能である。
4. 実装が非常に容易である。
5. 線形要素にも適用できる。

Wille の ILU 前処理の特徴を挙げると以下のようなになる。

1. 行列データ，各節点と未知数の対応，並びに節点の種類の他に要素の情報が必要。
2. 流速を消去した際に圧力対角ブロックに生じる fill-in が要素外になる場合は無視される。すなわち，要素内の Fill-in をすべて取り込む。
3. 特徴 2 のため 8-1d 要素並びに 4-1d 要素などには適用できない。
4. 高次の有限要素に適用されている。

例えば鷺尾の特徴 2 から，鷺尾の ILU 前処理に用いる近似行列 Q は Wille のそれよりも密になることがわかる。

図 3.1 が 4/1d 要素であり，図 3.2 が 4/4e 要素である。●が流速節点を表し，×が圧力節点を表す。4/4e 要素においては流速節点及び圧力節点が重なっている。

続いて 4/1d 要素に対して鷺尾の ILU 前処理を行った場合に発生する fill-in は図 3.3 で与えられ，Wille の ILU 前処理を行った場合に発生する fill-in は図 3.4 であたえられる。各図においては，赤色の●で表された流速節点を消去した際に圧力-圧力ブロックに発生する fill-in を赤線で表している。Wille の前処理は要素外 fill-in を無視するため，4/1d のような圧力の定数補間を行う要素では圧力-圧力ブロックに fill-in が発生

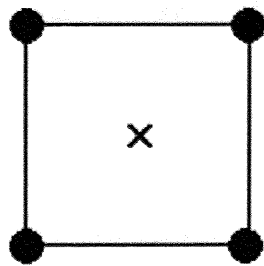


図 3.1: 4/1d 要素

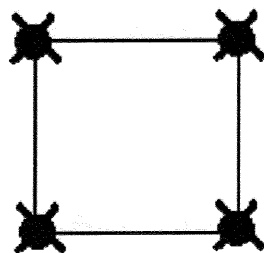


図 3.2: 4/4c 要素

しないため、このような要素に対しては適用することができない。

4/4c 要素に関して同様のことを行った結果は以下ようになる。各図においては、赤色の●で表された流速節点を消去した際に圧力-圧力ブロックに発生する fill-in を赤線（要素内のつながり）及び青線（要素外のつながり）で表している。

3.2.3 前処理付き GMRES 法の収束性

定理 3.2.1 (前処理付き GMRES 法の収束性) Q を任意の正則行列とする。その Q を用いて A, \hat{x}_0, \hat{b}_0 をそれぞれ以下のように定義する。

$$\hat{A} = Q\tilde{A}Q^{-1}, \quad \hat{x}_0 = Q\tilde{x}_0, \quad \hat{b}_0 = Q\tilde{b}_0$$

また、 \tilde{r}_k を \tilde{A} に対する GMRES 法の残差、 \hat{r}_k を \hat{A} に対する GMRES 法の残差とすれば以下の不等式が成り立つ。

$$\frac{\|\tilde{r}_k\|}{\|\tilde{r}_0\|} \leq \|Q^{-1}\| \|Q\| \frac{\|\hat{r}_k\|}{\|\hat{r}_0\|} \quad (3.2.7)$$

■

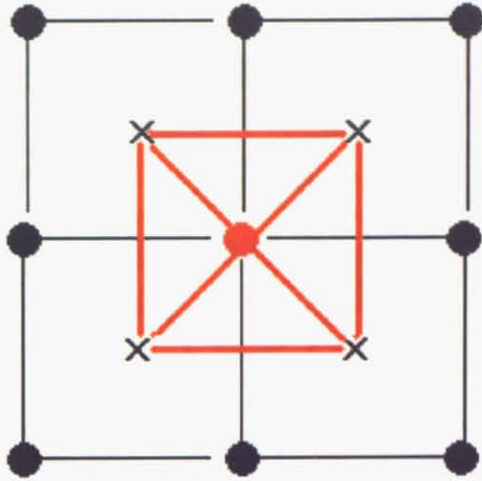


図 3.3: 鷲尾の ILU 前処理を行った場合に発生する fill-in(4/1d 要素)

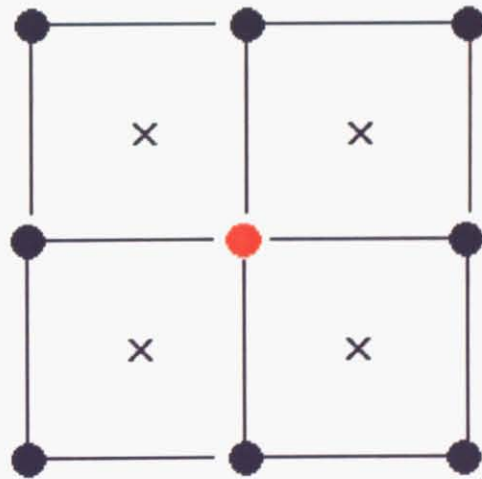


図 3.4: Wille の ILU 前処理を行った場合に発生する fill-in(4/1d 要素)

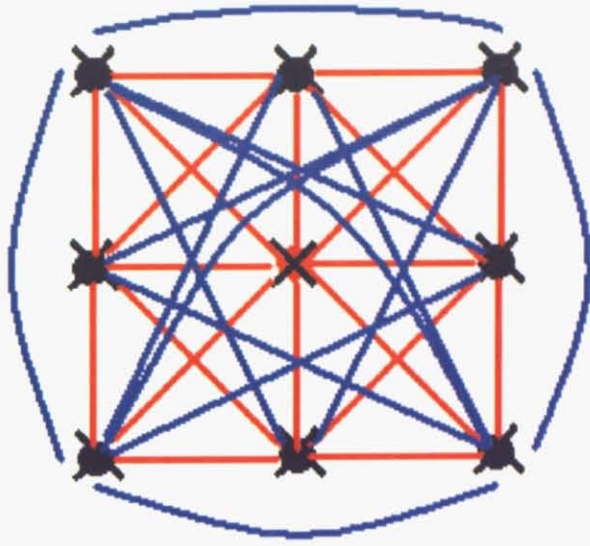


図 3.5: 鷲尾の ILU 前処理を行った場合に発生する fill-in(4/4c 要素)

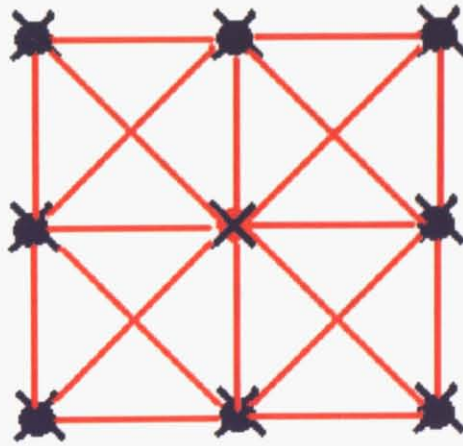


図 3.6: Wille の ILU 前処理を行った場合に発生する fill-in(4/4c 要素)

[証明] $\tilde{\mathbf{r}}_k$ は \tilde{A} に対する GMRES 法の残差なので定理 A.2.4 から $P_k(0) = 1$ を満たす任意の k 次の多項式 $P_k(x)$ に対し

$$\|\tilde{\mathbf{r}}_k\| \leq \|P_k(\tilde{A})\tilde{\mathbf{r}}_0\|$$

が成り立つ。また、

$$\begin{aligned} \|P_k(\tilde{A})\tilde{\mathbf{r}}_0\| &= \|Q^{-1}P_k(Q\tilde{A}Q^{-1})Q\tilde{\mathbf{r}}_0\| \\ &\leq \|Q^{-1}\| \|P_k(\tilde{A})\tilde{\mathbf{r}}_0\| \end{aligned}$$

が成り立つ。よって

$$\|\tilde{\mathbf{r}}_k\| \leq \|Q^{-1}\| \|\tilde{\mathbf{r}}_k\| \quad (3.2.8)$$

となる。一方

$$\begin{aligned} \|\tilde{\mathbf{r}}_0\| &= \|Q\tilde{\mathbf{r}}_0\| \\ &\leq \|Q\| \|\tilde{\mathbf{r}}_0\| \end{aligned} \quad (3.2.9)$$

なので式 (3.2.8) と (3.2.9) を合わせれば

$$\frac{\|\tilde{\mathbf{r}}_k\|}{\|\tilde{\mathbf{r}}_0\|} \leq \|Q^{-1}\| \|Q\| \frac{\|\tilde{\mathbf{r}}_k\|}{\|\tilde{\mathbf{r}}_0\|}$$

となり題意は示せた。

□

つまり、 $\text{sym}(Q\tilde{A}Q^{-1})$ が正定値になり、且つ条件数がそれほど大きくない Q が存在することを示せば、 $\tilde{A} = M^{-1}A$ に対する GMRES 法も $Q\tilde{A}Q^{-1}$ と同程度の収束性で収束すると言えるのである。逆にいえば、良い前処理行列 M を作るということは、 \tilde{A} の条件数並びに Q の条件数がそれぞれ大きくならないように考慮する必要があるといえる。

3.2.4 心臓シミュレータ向け前処理付反復解法

心臓シミュレータ^[4] 向けの連立一次方程式解法として、以前は skyline 法や multi-frontal 法などの直接法ソルバーを用いていた。しかし現在では扱う問題の自由度が非常に大きくなっているため反復法を用いるようになってきている。心臓シミュレータ用の前処理付き反復法として前処理は前小節で述べた混合型 FEM 用 ILU 前処理、反復法は第 3.1.3 小節で述べた GMRES 法を採用した「混合型 FEM 用 ILU 前処理付き GMRES 法」と呼ばれるものを使用している。

3.3 Multigrid 法

3.3.1 Multigrid 法概説

Multigrid 法とは初めに与えられた最も細かいメッシュを元にして複数の粗いメッシュを作り出し、様々な幅のメッシュを用いて、様々な種類の波長の誤差を同時に減らしていく方法である。この方法を用いれば、Poisson 方程式においては反復数は問題サイズに比例せず大規模問題にも適用できることが知られている [23]。

Multigrid 法には幾何学的 Multigrid 法と代数的 Multigrid 法の 2 種類が存在する。幾何学的 Multigrid 法は粗さが段階的に変化するメッシュを外部から与え計算を行う方法である。比較的実装が容易ではあるが、本来の細かいメッシュ以外の余分なデータを準備するため手間がかかることや、複雑な領域を持つメッシュには適用しづらいなどの問題点がある。一方代数的 Multigrid 法は与えられた細かいメッシュ上で離散化して得られた係数行列を元に数種類の係数行列を作成し、代数的な階層構造を用いる方法である。入力するデータは他のソルバーと変わらないため、ユーザー側は扱いやすいが、アルゴリズムは複雑で実装は難しいといえる。

3.3.2 Two grid cycle

Two grid cycle とは細かいメッシュと粗いメッシュを用いて異なる波長を持つ誤差を同時に減衰させていく方法である。このアルゴリズムを大きく分けると pre-smoothing, coarse grid correction, そして post-smoothing の 3 つから構成される。具体的には以下のアルゴリズムを用いて現在の (細かいグリッド上における) 解 \mathbf{u}_F^i から \mathbf{u}_F^{i+1} を求める方法である。細かいメッシュにおいて離散化された係数行列を A_F 、粗いメッシュにおいて離散化された係数行列を A_C とする。本論文では A_F から A_C を生成するのではなく、独立したメッシュ上で行列を作成している。

1. 線形化された方程式を細かいメッシュ上で緩和し、結果を $\hat{\mathbf{u}}_F^i = S_F(A_F, \mathbf{f}, \mathbf{u}_F^i)$ とする。演算子 S_F (例えば Gauss-Seidel) は緩和演算子 (smoothing operator) と呼ばれる。この操作を pre-smoothing と呼ぶ。
2. 細かいメッシュ上で残差 $\mathbf{r}_F = \mathbf{f} - A_F \hat{\mathbf{u}}_F^i$ を求める。

3. 制限写像 R を用いて, 細かいメッシュ上における残差を粗いメッシュ上に $\mathbf{r}_C = R\mathbf{r}_F$ の式で制限する.
4. 方程式 $A_C\Delta\mathbf{u}_C = \mathbf{r}_C$ を粗いメッシュ上で解く.
5. 粗いメッシュ上での修正量 $\Delta\mathbf{u}_C$ を補間写像 P を用いて細かいメッシュに $\Delta\mathbf{u}_F = P\Delta\mathbf{u}_C$ の式で補間する.
6. 細かいメッシュ上における解を求めた修正量 $\Delta\mathbf{u}_F$ を用いて $\hat{\mathbf{u}}_F^{i+1} = \hat{\mathbf{u}}_F^i + \Delta\mathbf{u}_F$ のように更新する.
7. 修正された解 $\hat{\mathbf{u}}_F^{i+1}$ に smoother をかける. すなわち $\mathbf{u}_F^{i+1} = S_F(A_F, \mathbf{f}, \hat{\mathbf{u}}_F^{i+1})$ として \mathbf{u}_F^{i+1} を求める. この操作を post-smoothing と呼ぶ.
8. 相対残差 $\frac{\|\mathbf{f} - A_F\mathbf{u}_F^{i+1}\|}{\|\mathbf{f}\|}$ の値を計算し, 基準値以下 (本論文においては 10^{-8} 以下) になるまで 1~8 のプロセスを繰り返す.

上記のアルゴリズム 1~7 を用いて \mathbf{u}_F^i から \mathbf{u}_F^{i+1} を求める演算子 TG を

$$\mathbf{u}_F^{i+1} = TG(A_F, \mathbf{f}, \mathbf{u}_F^i) \quad (3.3.1)$$

と定義することにする.

なお, このアルゴリズムにおいては pre-smoother 及び post-smoother をそれぞれ 1 回ずつつけているが, これを数回ずつ行う方法もある^[18]. しかしながら, smoother を多くかけることは計算量の増加をまねくため本論文では pre-smoother 及び post-smoother をそれぞれ 1 回のみかける方法を推奨する.

Multigrid cycle

Multigrid cycle とは, Two grid cycle を再帰的に適用したものである. すなわち, Two grid cycle の中で粗いメッシュ上で与えられた方程式 $A_C\Delta\mathbf{u}_C = \mathbf{r}_C$ を解く際に, 再び Two grid cycle を用いて解を求める方法である. 2 階層の Multigrid cycle は Two grid cycle そのものである. k 階層の Multigrid cycle は, k 段目 (最も粗いメッシュ上) では $A_C\Delta\mathbf{u}_C = \mathbf{r}_C$ を直接法などで解き, それ以外の階層では Two grid cycle を再帰的に呼ぶという方法である. 具体的なアルゴリズムは式 (3.3.1) を用いれば以下のように書くことができる.

1. 線形化された方程式を細かいメッシュ上で緩和し, 結果を $\tilde{\mathbf{u}}_F^i = S_F(A_F, \mathbf{f}, \mathbf{u}_F^i)$ とする. 演算子 S_F (例えば Gauss-Seidel) は緩和演算子 (smoothing operator) と呼ばれる. この操作を pre-smoothing と呼ぶ.
2. 細かいメッシュ上で残差 $\mathbf{r}_F = \mathbf{f} - A_F \tilde{\mathbf{u}}_F^i$ を求める.
3. 制限写像 R を用いて, 細かいメッシュ上における残差を粗いメッシュ上に $\mathbf{r}_C = R\mathbf{r}_F$ の式で制限する.
4. 方程式 $A_C \Delta \mathbf{u}_C = \mathbf{r}_C$ を最も粗いメッシュ上であれば直接法などで解き, それ以外のメッシュ上であれば $\Delta \mathbf{u}_C = TG(A_C, \mathbf{r}_C, \mathbf{0})$ を用いて $\Delta \mathbf{u}_C$ の求解を次に粗い階層に引き継ぐ.
5. 粗いメッシュ上での修正量 $\Delta \mathbf{u}_C$ を補間写像 P を用いて細かいメッシュに $\Delta \mathbf{u}_F = P\Delta \mathbf{u}_C$ の式で補間する.
6. 細かいメッシュ上における解を求めた修正量 $\Delta \mathbf{u}_F$ を用いて $\tilde{\mathbf{u}}_F^{i+1} = \tilde{\mathbf{u}}_F^i + \Delta \mathbf{u}_F$ のように更新する.
7. 修正された解 $\tilde{\mathbf{u}}_F^{i+1}$ に smoother をかける. すなわち $\mathbf{u}_F^{i+1} = S_F(A_F, \mathbf{f}, \tilde{\mathbf{u}}_F^{i+1})$ として \mathbf{u}_F^{i+1} を求める. この操作を post-smoothing と呼ぶ.
8. 相対残差 $\frac{\|\mathbf{f} - A_F \mathbf{u}_F^{i+1}\|}{\|\mathbf{f}\|}$ の値を計算し, 基準値以下 (本論文においては 10^{-8} 以下) になるまで 1~8 のプロセスを繰り返す.

上記のアルゴリズム 1~7 を用いて \mathbf{u}_F^i から \mathbf{u}_F^{i+1} を求める演算子 MG を

$$\mathbf{u}_F^{i+1} = MG(A_F, \mathbf{f}, \mathbf{u}_F^i) \quad (3.3.2)$$

と定義することにする.

上記のような Multigrid cycle を V-cycle と呼ぶ. 他に W-cycle や F-cycle など存在する [23]. 本論文では, Multigrid cycle としては V-cycle のみを扱う. これは, 本論文では Multigrid 法を反復ソルバーとして収束させるのではなく, 最終的には Multigrid 法を前処理とし反復ソルバーを GMRES 法として用いるため, V-cycle で十分と考えるためである.

ここまで述べてきたような Multigrid cycle を用いて解を求める方法を Multigrid 法と呼ぶ.

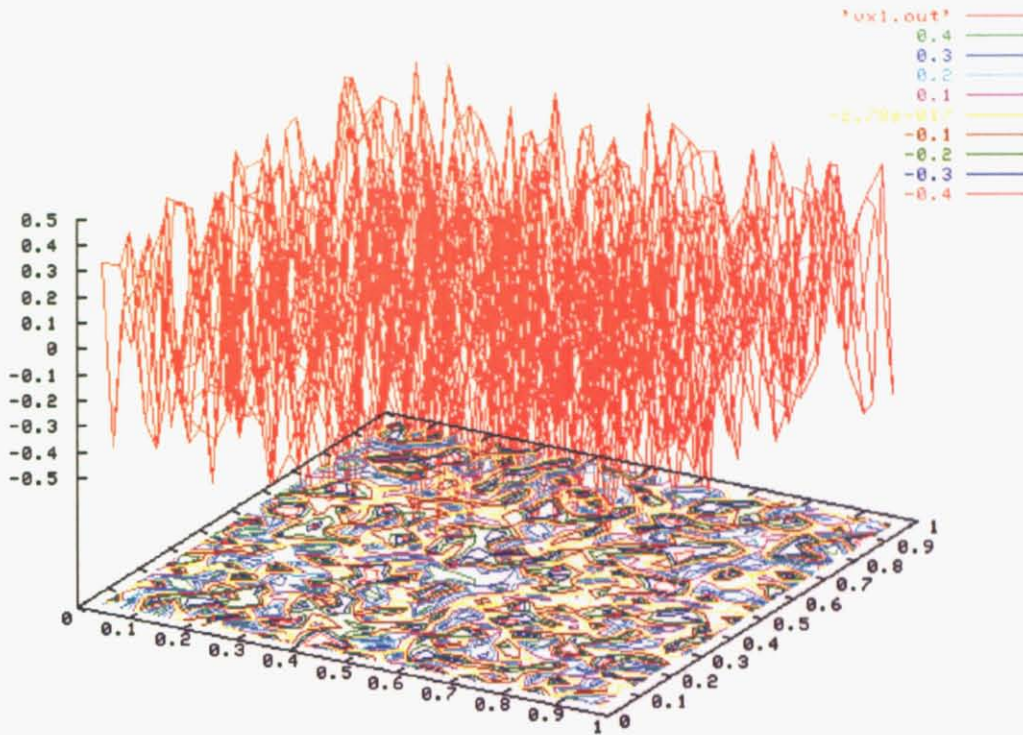


図 3.7: smoother 適用前の初期誤差

3.3.3 smoothing

smoothing の説明

smoothing とは現在与えられている誤差を滑らかにする方法である。例えば図 3.7 のような初期誤差を与える。そしてその誤差に対して本論文の第 5 章において提案する (EBEMP+ILUPS)PS-smoother を適用したものが図 3.8 で与えられる。適当な smoother を適用することにより、誤差が滑らかになっていることがわかる。

smoother は一般に式 (3.3.6) で与えられるような行列分離によって定義される。行列 M を用いて A を以下のように行列分離する。

$$A = M + (A - M) \tag{3.3.3}$$

M の選び方によって名称や収束性などが大きく異なる。例えば行列 A を、以下のよ

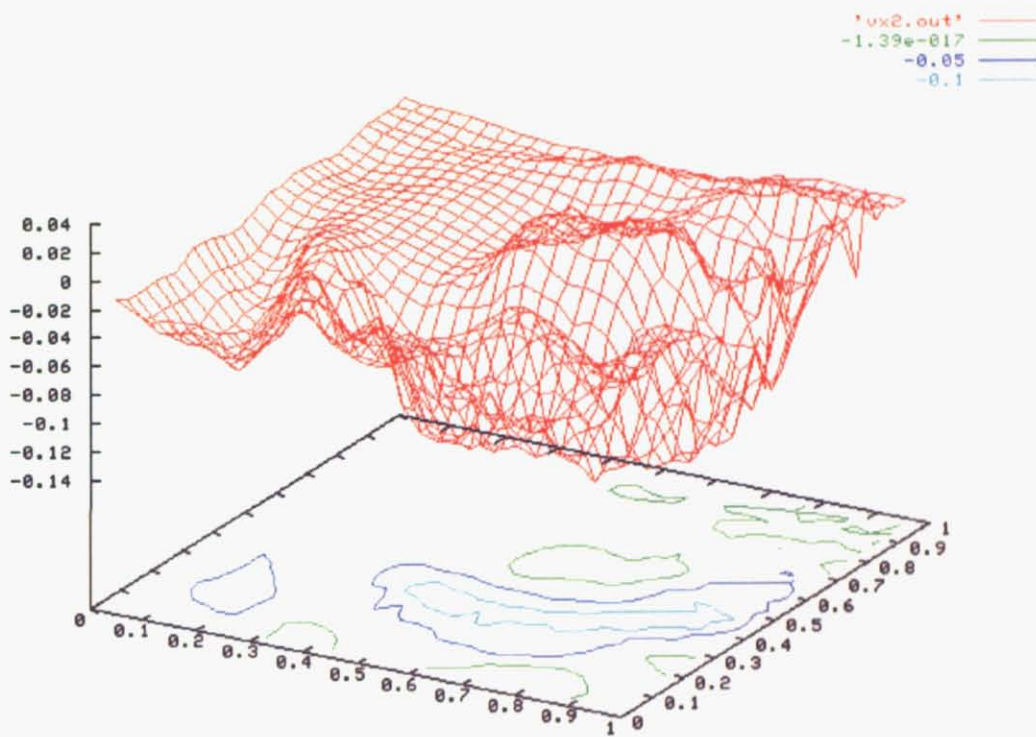


図 3.8: (EBEMP+ILUPS)PS-smoother 適用後の誤差

うに対角成分 D , 狭義下三角行列 E , 狭義上三角行列 F に分離して表現したとする.

$$A = D + E + F$$

このとき, 行列 M として A の対角成分 D を用いるものを Jacobi smoother と呼ぶ. また, 行列 M として $D + E$ を用いるものを Gauss-Seidel smoother と呼ぶ. smoother として用いるためには ρ をスペクトル半径とした場合

$$\rho(M^{-1}(A - M)) \leq 1 \quad (3.3.4)$$

を満たす必要がある. A が正定値であれば Gauss-Seidel smoother は式 (3.3.4) の条件を満たすことが知られている^[23]. そのため, 正定値問題に対しては Gauss-Seidel smoother が用いられることが多い. しかしながら本研究で扱うような流速及び圧力の混合要素を用いて離散化した不定値問題に対して Gauss-Seidel smoother を用いてもこのような条件は満たされないことに注意する. 本論文の課題は, このような不定値問題に対して, 式 (3.3.4) の条件を満たし, 有効な smoother を構築することである.

また, 本論文第 5 章で詳しく述べるが, Gauss-Seidel smoother のような点緩和法は異方性の強い問題に弱い. 例えばアスペクト比が大きい場合や移流拡散方程式において移流項が強い場合などがそれにあたる. 本論文では異方性が強い問題については, 2 次元問題においては ILU smoother が比較的有効であることを示す.

本節では前節において記述した緩和演算子 (smoothing operator) S_F について詳しく説明していく.

簡単な説明のために Poisson 方程式に差分法を用いた内容で話を進める. メッシュ幅を x 方向, y 方向共に h としメッシュ点 (x_i, y_i) 上における m 回目の反復時の解を $u_h^m(x_i, y_i)$ とおくことにする. メッシュ点 (x_i, y_i) 上における収束解を $u_h(x_i, y_i)$ とおくことにする. このとき, 各メッシュ点における誤差を

$$e_h^m(x_i, y_i) = u_h(x_i, y_i) - u_h^m(x_i, y_i)$$

で定義する.

2 次元 Poisson 方程式の差分離散化を行うと, テイラー展開の 2 次項の展開までを行って

$$\frac{\partial^2 u}{\partial x^2}(x, y) = \frac{u(x + dx, y) - 2u(x, y) + u(x - dx, y)}{dx^2}$$

$$\frac{\partial^2 u}{\partial y^2}(x, y) = \frac{u(x, y + dy) - 2u(x, y) + u(x, y - dy)}{dy^2}$$

となる。これを2次元 Poisson 方程式

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = -f(x, y)$$

に代入すれば

$$\frac{u(x + dx, y) - 2u(x, y) + u(x - dx, y)}{dx^2} + \frac{u(x, y + dy) - 2u(x, y) + u(x, y - dy)}{dy^2} = -f(x, y)$$

となる。 $dx = dy = h$ として離散化すれば

$$\begin{aligned} 4u_h(x_i, y_i) &= u_h(x_i + h, y_i) + u_h(x_i - h, y_i) + u_h(x_i, y_i + h) + u_h(x_i, y_i - h) \\ &+ h^2 f_h(x_i, y_i) \end{aligned} \quad (3.3.5)$$

が成り立つ。この式に Gauss-Seidel 型の行列分離

$$(D + E)\mathbf{x}_{m+1} + F\mathbf{x}_m = \mathbf{b} \quad (3.3.6)$$

を適用すれば

$$\begin{aligned} 4u_h^{m+1}(x_i, y_i) &= u_h^m(x_i + h, y_i) + u_h^{m+1}(x_i - h, y_i) + u_h^m(x_i, y_i + h) + u_h^{m+1}(x_i, y_i - h) \\ &+ h^2 f_h(x_i, y_i) \end{aligned} \quad (3.3.7)$$

が成り立つ。式 (3.3.5)-式 (3.3.7) を計算すると

$$\begin{aligned} 4e_h^{m+1}(x_i, y_i) &= e_h^m(x_i + h, y_i) + e_h^{m+1}(x_i - h, y_i) \\ &+ e_h^m(x_i, y_i + h) + e_h^{m+1}(x_i, y_i - h) \end{aligned} \quad (3.3.8)$$

となる。式 (3.3.8) は点 (x_i, y_i) 上における誤差 $e_h^{m+1}(x_i, y_i)$ の値が前後左右の誤差の平均で与えられることを意味する。すなわち、Gauss-Seidel法を行うことにより、誤差の平均化、すなわち誤差の smoothing が行われることを意味する（スムーズにはなるが、誤差が減る保障はないことに注意する）。

smoothing とフーリエ級数展開

smoothing についてももう少し詳しく見ていく。ここで、誤差 $e_h^m(x_i, y_i)$ をフーリエ級数展開して以下のようにあらわす。

$$e_h^m(x_i, y_i) = \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} \alpha_{k,l} \sin k\pi x \sin l\pi y = \sum_{k=1}^{n-1} \sum_{l=1}^{n-1} \alpha_{k,l} \varphi_h^{k,l}(x, y) \quad (3.3.9)$$

ここで、フーリエ級数の各成分に対して低周波及び高周波成分を以下のように定義する。

$$\max(l, k) < \frac{2}{n} \quad \rightarrow \varphi_h^{k,l}(x, y) \text{ を低周波成分と呼ぶ} \quad (3.3.10)$$

$$\frac{2}{n} \leq \max(l, k) < n \quad \rightarrow \varphi_h^{k,l}(x, y) \text{ を高周波成分と呼ぶ} \quad (3.3.11)$$

誤差がスムーズになるということは、高周波成分が減衰し、低周波成分はほとんど変化しないということを意味する。メッシュ幅を $2h$ に粗くした場合、式 (3.3.11) で与えられる高周波はこのメッシュでは検知できなくなり、減衰させることができない。しかしながら、先ほどの低周波成分の中で、長い波長を持つものは（このメッシュにとっては高周波であるため）、ここまで述べてきた理由により減衰させやすいことがわかる。

よってメッシュ幅を粗く変化させ複数用いることにより、細かいメッシュにとっては減らしづらい低周波を素早くすることができるのである。

なお、ここであげた例題の場合、（パラメータ調節にもよるが）Jacobi-type の緩和法を用いると高周波成分が 1 回の smoothing によりそれぞれ $\frac{2}{3}$ 倍ずつ、Gauss-Seidel-type の緩和法を用いると 1 回の smoothing によりそれぞれ $\frac{1}{2}$ 倍ずつ減衰することが知られている [23]。

一般の反復法（Gauss-Seidel 法、SOR 法）を用いると、基本的にメッシュサイズと同じ波長を持った誤差成分が効率よく減衰する。しかしながら、長波長（低周波）の誤差成分はなかなか減衰しない [23]。

大規模問題ではメッシュが細かくなり、従って長波長の誤差が落ちずに反復数が増大する。例えば CG 法などで反復数は問題サイズに比例する。一般の反復法は大規模問題になるにつれて反復数、ひいては計算時間が膨大になり大きな問題となる。これらの問題点を解決するのが Multigrid 法である。

3.3.4 FEM における形状関数と Multigrid の補間写像・制限写像の関係

ここでは FEM を用いた場合の Multigrid 法に用いる制限写像 R の与え方について考える。FEM を用いた場合、Multigrid 法に用いる補間写像 P と形状関数 N の間には以

下の関係がある.

$$(Pe_H)_{x_h} = [N_H]e_H(x_h) \quad (3.3.12)$$

ここで x_h は細かいメッシュ上の任意の節点とする.

P を FEM における補間写像とし, 細かいメッシュ上における誤差を e_h , 粗いメッシュ上における誤差を e_H とする. また, その変化量を同様に $\delta e_h, \delta e_H$ とする. また, 細かいメッシュ上における形状関数を $[N_h]$, 粗いメッシュ上における形状関数を $[N_H]$ とする. 続いて, 細かいメッシュ上における係数行列を A_h , 粗いメッシュ上における係数行列を A_H とする. 誤差 e_h を e_H の補間により与えられると仮定する. すなわち以下の式が成り立つと仮定する.

$$e_h = Pe_H \quad (3.3.13)$$

$$\delta e_h = P(\delta e_H) \quad (3.3.14)$$

まず, 粗いメッシュ上における FEM を用いた離散化の式の左辺は以下のように表すことができる.

$$\delta e_H^T A_H e_H = \int_V \delta e_H^T [N_H]^T L [N_H] e_H dV \quad (3.3.15)$$

ここで L は係数行列を定める適当な作用素である. また, 関数の同値関係として以下の式が成り立つ.

$$[N_H]e_H = [N_h]Pe_H \quad (3.3.16)$$

これを式 (3.3.15) に代入して,

$$\delta e_H^T A_H e_H = \int_V \delta e_H^T P^T [N_h]^T L [N_h] Pe_H dV \quad (3.3.17)$$

$$= \delta e_H^T P^T A_h Pe_H \quad (3.3.18)$$

が任意の e_H に対して成り立つ. 従って

$$A_H = P^T A_h P \quad (3.3.19)$$

である必要がある.

以上の考察から

$$R = P^T \quad (3.3.20)$$

を用いるのが適当と考える.

3.3.5 Multigrid 法と前処理の関係

Multigrid 法を GMRES 法などの反復解法の前処理として用いることもできる。まず, Multigrid cycle MG に対応する以下のような係数行列 MG を定義する。この係数行列の具体的な値を求めることは一般にはできない。

$$\Delta \mathbf{x}^k = MG(A, \mathbf{r}^k, \mathbf{0}) \leftrightarrow MG \Delta \mathbf{x}^k = \mathbf{r}^k$$

Multigrid 法は以下のアルゴリズムを $k = 0$ から始め, 適当なところ (1.における相対残差が 10^{-8} を下回った時など) で計算を打ち切る手法である。

1. Calculate $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$
2. Solve $MG \Delta \mathbf{x}^k = \mathbf{r}^k$
3. Update $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}^k$

特に $k = 0$ かつ $\mathbf{x}^0 = \mathbf{0}$ の場合を考える。

1. $\mathbf{r}^0 = \mathbf{b}$
2. $MG \Delta \mathbf{x}^0 = \mathbf{b}$
3. $\mathbf{x}^1 = \Delta \mathbf{x}^0 = MG^{-1} \mathbf{b}$

すなわち, Multigrid 法のアルゴリズムに初期値 $\mathbf{x}_0 = \mathbf{0}$ を代入し, 前処理を行いたいベクトルを \mathbf{b} として代入する。そしてアルゴリズムを 1 回だけ行えば, 出力された解 \mathbf{x}_1 は $MG^{-1} \mathbf{b}$ となる。これは \mathbf{b} を Multigrid cycle で前処理したものに他ならないのである。

Multigrid 法は反復ソルバーとしても高い性能を誇る^[23]。それと同時に, 非常に優秀な前処理として用いることができる。現在では, Multigrid 法は多くの場合 Krylov 部分空間法の前処理として用いられている。

3.4 Poisson 問題への適用

3.4.1 緒言

本節では, ここまで述べてきた各反復解法のうち, 古典的反復法の代表である Gauss-Seidel 法 (以下 GS 法), 前処理付き Krylov 部分空間法の代表である ILU-GMRES 法

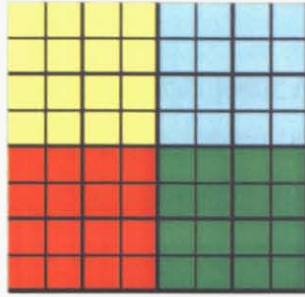


図 3.9: 解析領域とメッシュ

(ILU 分解を前処理とした GMRES 法), そして Multigrid 法の代表である GSMG 法 (Gauss-Seidel 法を smoother とした Multigrid 法) の 3 解法を 2 階の楕円型微分方程式の代表ともいえる Poisson 方程式に適用し, 収束性を比較していく.

Poisson 方程式は温度の伝わりなどを表す方程式であり, 2 階の楕円型方程式の基礎となる非常に重要な式である. 複雑な 2 階の楕円型方程式を考える際には, Poisson 方程式に帰着させることも多い. Poisson 方程式は解析的にも理論解が求まる非常に簡単な式である. また, 数値計算的に見ても離散化も容易であり, 導出された連立一次方程式を解くことも非常に容易である. 数値計算的な容易さは導出された連立一次方程式の優対角性および正定値性に起因する [25].

Poisson 方程式は Gauss-Seidel 法のような非常に古典的なソルバーでも解くことはできるものの, 収束性は決して良くない. 前処理付き GMRES 法などを用いればかなり速く収束させることができるが, 自由度数が大きくなると反復数が徐々に増加していく. Multigrid 法は, 自由度が増加しても反復数が増加しないところが大きな特徴である.

3.4.2 問題設定

離散化等の過程は省略する. FEM を用いて Poisson 方程式の離散化を行った. 解析領域は 2 次元正方形領域であり, x 軸, y 軸方向に等間隔で図 3.9 のようにメッシュを張った. この図では 3 階層のメッシュを表している. 細い黒線が最も細かいメッシュ, 黒い太線が中間のメッシュ, そして色別に塗り分けられた要素が最も粗いメッシュを表す. 境界条件は y 座標の値に比例した数値を Dirichlet 境界条件により与えた. また, > 500 と書いてあるものは反復回数が 500 回を超えることを意味する.

3.4.3 計算結果

このようにして生成される連立一次方程式の求解にかかる反復数及び計算時間を以下に示す。連立一次方程式のソルバーとしては古典的反復法の代表として Gauss-Seidel 法を、Krylov 部分空間法の代表として ILU 前処理付き GMRES 法 (ILU-GMRES 法) を、Multigrid 法の代表として Gauss-Seidel smoother を用いた Multigrid 法 (GSMG 法) を採用した。結果を表 3.1, 3.2 及び図 3.10, 3.11 に示す。なお、 n は各方向のメッシュ分割数である。また、計算時間はの単位は秒であり、0 秒は 0 秒以上 1 秒未満で計測不能であることを示す。計算時間 100 秒は 100 秒以上を表す。反復回数 500 回は 500 回以上を表す。なお、GSMG 法においてはメッシュの階層を可能な限り多く取るようにした。

表 3.1: Poisson 問題における GS 法, ILU-GMRES 法及び GSMG 法の反復数の比較

n	自由度数	GS 法	ILU-GMRES 法	GSMG 法
4	18	20	5	5
8	98	78	8	9
16	450	298	13	9
32	1922	> 500	23	9
64	7938	> 500	42	9
128	32258	> 500	77	9
256	130050	> 500	132	9

表 3.2: Poisson 問題における GS 法, ILU-GMRES 法及び GSMG 法の計算時間の比較

n	自由度数	GS 法	ILU-GMRES 法	GSMG 法
32	1922	1	0	0
64	7938	19	1	0
128	32258	> 100	10	1
256	130050	> 100	76	3

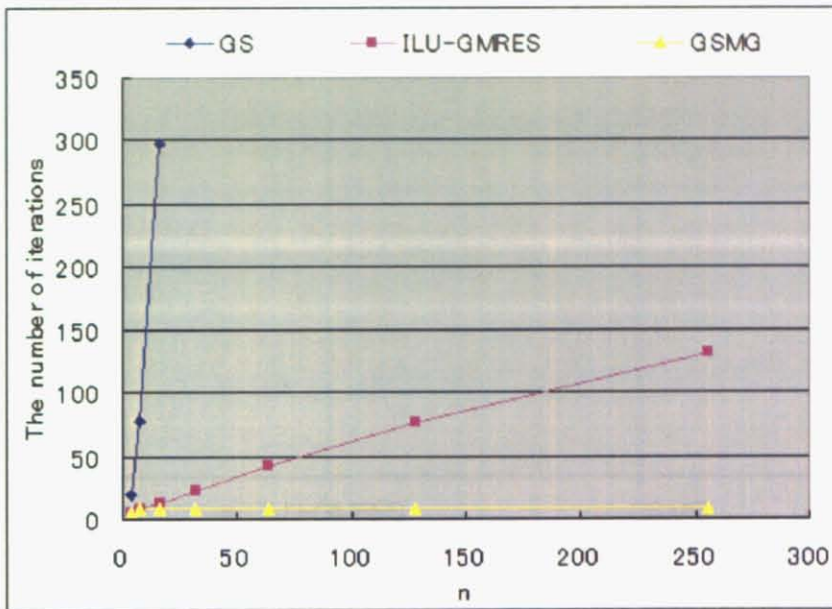


図 3.10: Poisson 問題における GS 法, ILU-GMRES 法及び GSMG 法の反復数の比較

古典的反復法である GS 法を用いた場合, 自由度が増加するにつれて反復数が激増してしまう. 現在, 当研究室の心臓シミュレーター解析^[4]に使われている ILU-GMRES 法を用いた場合, 反復数の増大はある程度抑えられているものの自由度が増えるに従って反復数が増加してしまっている. ところが GSMG 法を用いた場合, 反復数がほぼ一定に抑えられている. GSMG 法は大規模問題に対して非常に強力な連立一次方程式ソルバーといえる.

3.4.4 結言

ここまでで述べてきたとおり, Multigrid 法は Poisson 方程式のような正定値行列に対しては自由度に応じて反復数が増加せず, 極めて有効であるといえる. しかしながら後に述べる Stokes 問題等には, ここで用いたような GSMG 法など既存の Multigrid 法は有効ではない. 以後は, Stokes 問題に対して有効な Multigrid 法を説明する. 本研究の最終的な目標は「非圧縮拘束条件付 Stokes, Navier-Stokes 問題など, 離散化した際に不定値の係数行列が現れるような問題に対して, 反復数が自由度に依存しない Multigrid ソルバー, 又は Multigrid を前処理とした反復ソルバーの開発」といえる.

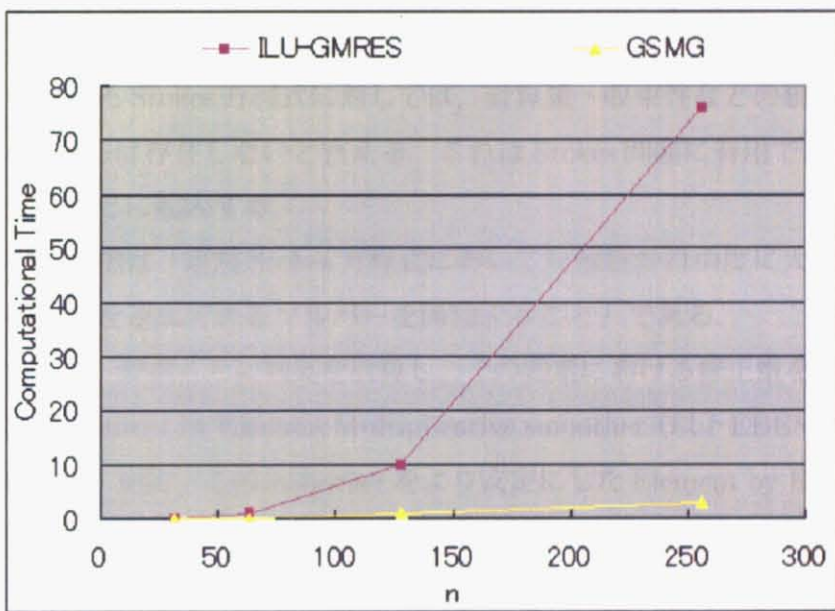


図 3.11: Poisson 問題における GS 法, ILU-GMRES 法及び GSMG 法の計算時間の比較

第4章 非圧縮拘束条件付定常 Stokes 問題

4.1 緒言

本章では定常の Stokes 問題に対して Multigrid 法をどのように適用していくかを述べる。現在のところ Stokes 方程式に対しては、計算量・収束性などの観点から考えて有効な Multigrid 法は存在しないと言える。これは Stokes 問題に有用である smoother が存在しないことに起因する。

従って本章の目標は「定常 Stokes 方程式において反復数が自由度に大きく依存しない理想的な収束性を達成できるソルバーを開発すること」である。

本章においては、要素ごとに残差を計算し、その要素における修正量を全体の解に足しこむタイプの Element by Element Multiplicative smoother (以下 EBEMP-smoother) を新たに提唱する。更に、この smoother をより安定にした Element by Element Multiplicative Pressure Stabilized smoother (以下 EBEMP-smoother) 更に、この smoother を用いた Element by Element Multiplicative Multigrid 法 (以下 EBEMPMG 法) を前処理に用いた GMRES 法を用いることによって、定常 Stokes 方程式においても反復数が自由度に大きく依存しない理想的な収束性を達成できることを示していく。

4.2 非圧縮拘束条件付定常 Stokes 方程式

4.2.1 非圧縮拘束条件付定常 Stokes 方程式の離散化

定常 Stokes 方程式は 2.2 節で導かれた

$$\rho \frac{\partial \mathbf{v}}{\partial t} \Big|_x + \rho (\mathbf{v} \otimes \nabla_x) \cdot \mathbf{v} = \rho \mathbf{g} + -\nabla_x p + \mu \nabla^2 \mathbf{v} \quad (4.2.1)$$

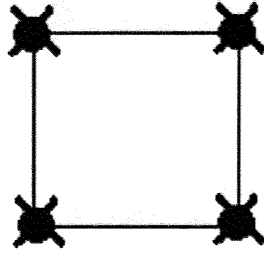


図 4.1: 4/4c 要素

の左辺第一項（慣性項）及び左辺第二項（移流項）を無視したもので与えられる。また重力も無視すれば、定常 Stokes 方程式は以下の形で与えられる。

$$\mu \nabla^2 \mathbf{v} - \nabla_x p = \mathbf{0} \quad (4.2.2)$$

また、非圧縮拘束条件の連続の式は以下のように与えられる。

$$\nabla_x \cdot \mathbf{v} = 0 \quad (4.2.3)$$

本章では式 (4.2.2) 及び式 (4.2.3) を連立して解いていく。まず、この方程式を弱形式化し、離散化を行う。離散化には各頂点に流速及び圧力節点がそれぞれ存在する 2 次元 4/4c 要素を用いて有限要素を用いる。なお図 4.1 においては●が流速節点を表し、×が圧力節点を表す。4/4c 要素においては流速節点及び圧力節点が重なっている。第 2 章の結果を参照すると、離散化された定常 Stokes 方程式に対応する連立一次方程式 $A\mathbf{x} = \mathbf{b}$ は以下のような形になる。ここで一行目は Stokes 方程式、二行目は連続の式に対応する。

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \quad (4.2.4)$$

式 (4.2.2) から \mathbf{A} は対称正定値行列であることがわかる。また、 \mathbf{B} は divergence オペレータに対応する項で $\ker(\mathbf{B}^T) = 0$ を満たす。 \mathbf{C} は安定化項などが付与されれば対称正定値行列、付与されない場合は零行列とする。ここで式 (4.2.2) をそのまま離散化すると、式 (4.2.4) の係数行列の (1,2) ブロック \mathbf{B}^T の部分は $-\mathbf{B}^T$ となる。しかしながら、圧力部分の符号を反転させることにより、 \mathbf{B}^T と解釈することが可能である。係数行列に対称性を持たせるために、議論を行う際は式 (4.2.4) のように表記することが多いことに注意しておく。

4.2.2 Reynolds 数について

移流項が存在しない Stokes 問題においては Reynolds 数 Re は本質的に意味を持たないが、ここでは本質的に意味を持たない理由の説明、及び次章以降における説明の省略のために Reynolds 数について触れておく。

Reynolds 数 Re は以下のように定義される。

$$Re = \frac{LV}{\nu}$$

ここで、 L は代表長さ、 V は代表速度を表す。また ν は動粘性係数であり、流体の密度 ρ 及び粘性係数 μ を用いて以下のように表すことができる。

$$\nu = \frac{\mu}{\rho}$$

代表長さや代表速度の定め方には様々な方法があるが、本論文のシミュレーション（キャピティーフロー問題）においては代表長さは正方領域における一辺の幅、代表速度は境界条件として上部に与える流速とした。

移流項が存在しない Stokes 問題においては Reynolds 数 Re は本質的に意味を持たない。その理由を以下に示す。領域の代表長さが L で一定であり、流体の密度が一様であるとした場合流速 V は以下のように表すことができる。

$$\begin{aligned} V &= Re \times \frac{\nu}{L} \\ &= Re \times \frac{\mu}{\rho L} \end{aligned}$$

したがって一般には Re が高い流れになると流速も比例して上昇する。しかしながら、移流項が存在しない Stokes 問題では方程式は V に関して線形であり、従って $V = \alpha v$ に対する連立一次方程式が

$$Ax = b \tag{4.2.5}$$

$$\tag{4.2.6}$$

で表せるとすると、 $V = \alpha v$ （但し、 α は実定数）となった場合でも

$$\alpha Ax = b \tag{4.2.7}$$

$$\tag{4.2.8}$$

となり，両辺を α で割れば式 (4.2.5) と式 (4.2.7) は係数行列が同じになる．従って本質的には同じ問題を解いているということになるからである．

Navier-Stokes 問題のように移流項がある場合，方程式は V に関して非線形（2次）であり従って， Re の値によって問題の性質，ひいては行列の性質が大きく変化することに注意する．例えば，水の動粘性係数は摂氏 20 度のとき

$$\mu = 1.0 \times 10^{-3}$$

$$\rho = 1.0 \times 10^3$$

程度で与えられるため，

$$\nu = 1.0 \times 10^{-6}$$

となる．

4.2.3 心臓問題との比較

現在本研究室で開発している心臓シミュレータに用いられている密度 ρ 及び静粘性係数 μ の値は以下のようになる．

$$\mu = 1.0$$

$$\rho = 1.06 \times 10^3$$

$$\mu = 4.71 \times 10^{-3}$$

従って，動粘性係数 ν は以下のように計算できる．

$$\nu = 4.44 \times 10^{-6}$$

また，心臓を立方体と考えると一辺の長さはおよそ 0.05m 程度になる．よって Reynolds 数を 5000 とし，代表速度を v とすれば

$$5000 = \frac{0.05v}{4.44 \times 10^{-6}}$$

が成り立つ．よって

$$v = 4.44 \times 10^{-1}$$

となる．従って本節で扱うシミュレーションは，心臓問題において流速 $v = 4.44 \times 10^{-1}$ を与えた場合と同様となる．

4.3 安定化手法の説明および本章の目的

4.3.1 安定化手法を付加した際の係数行列

定常 Stokes 問題に対して、安定化手法を用いない場合の連立一次方程式は以下のよう
にあらわすことができる。

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \quad (4.3.1)$$

この方程式に反復法を用いた場合、このまま解くことはできない。なぜならば、係数行列の圧力部の対角成分が存在せず解くことが不可能だからである。また、用いる要素によっては圧力振動などが発生し、正確な解を得ることが出来ない^[7]。そのため、Stokes 問題及び Navier-Stokes 問題においては安定化手法と呼ばれる方法を用いるのが一般的である。定常 Stokes 問題に対する安定化手法としては、(2,2) ブロック (圧力-圧力ブロック) の安定化を行うために以下のような微小な項を付け加える。

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\varepsilon \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} \quad (4.3.2)$$

\mathbf{C} はラプラシアンを離散化した項に対応する。直接法であれば ε が (計算機上で) 0 ではない値であれば LU 分解が可能になるため、 ε の値は解き易さに影響を与えない。しかしながら反復法を用いた場合、 ε によって解きやすさが大きく異なる。これは、一般には ε の値が大きくなるほど優対角性が強まるため、反復法を用いて解きやすくなるからであるといえる。次小節においては今後非定常 Navier-Stokes 方程式に拡張することを踏まえた場合、本章において解くべき ε の値を考察していく。

なお、式 (4.2.2) から μ の値は \mathbf{A} と \mathbf{B}^T の比を決めるといえる。 μ の値が大きければ \mathbf{A} が大きくなり、定常 Stokes 問題においては優対角性が強くなり問題が解きやすくなる。すなわち、一般には μ の値により問題の解きやすさが変化すると考えられる。

しかしながら、式 (4.2.3) に圧力安定化項を加えた式

$$\nabla_x \cdot \mathbf{v} + \varepsilon \Delta p = 0 \quad (4.3.3)$$

を μ 倍した

$$\mu \nabla_x \cdot \mathbf{v} + \mu \varepsilon \Delta p = 0 \quad (4.3.4)$$

を離散化すれば、 μ に応じて流速が反比例するだけであり、行列の本質は変わらないため、問題の解きやすさには影響を及ぼさないことがわかる。この際、安定化係数は $\mu\varepsilon$ で与えられることに注意する。

ここまでの議論から、 μ の値をどのようにとっても問題の解きやすさには影響を与えないため、本章においては

$$\mu = 1.0$$

として議論を進めていく。

4.3.2 安定化パラメーターについて

本小節では Stokes 方程式を付加する安定化パラメーター ε として妥当な値を考察していく。非定常 Navier-Stokes 方程式の PSPG 安定化手法に用いられるパラメーター τ_{PSPG} は以下の式で与えられる^{[24][31]}。

$$\tau_{PSPG} = \left[\left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|c\|}{h^\sharp} \right)^2 + \left\{ \frac{4\nu}{(h^\sharp)^2} \right\}^2 \right]^{-\frac{1}{2}} \quad (4.3.5)$$

ここで、式(4.3.5)の右辺第1項目は慣性項の安定化パラメーターに対応する。 Δt は時間刻み幅をあらわす。この項は時間発展問題（非定常問題）を考えるには現れるが、定常問題を考える際は考慮しなくて良い。式(4.3.5)の右辺第2項目は移流項の安定化パラメーターに対応する。 c は局所移流速度であり、 (h^\sharp) は2次元の場合は要素と同じ面積を持つ円の直径として与えられる^{[24][31]}。すなわち正方形メッシュの1辺の長さを h とすれば、

$$h^\sharp = \frac{h}{\sqrt{\pi}}$$

として与えられる。但し、 h^\sharp の定め方については特に理論的な解釈が存在するわけではないため、オーダーが変わるようであれば適当に定めてよいといえる。今回の実装においては $h^\sharp = h$ として定めた。この項は Navier-Stokes 問題を考える際には必要だが、Stokes 問題を考える際には考慮しなくて良い。式(4.3.5)の右辺第3項目は粘性項の安定化パラメーターに対応する。 ν は流体の動粘性係数である。定常 Stokes 問題を考える際は、第3項目のみを考慮すればよい。本章においてはパラメーターとして

$$\mu = 1.0$$

$$\rho = 1.0 \times 10^3$$

を用いているため

$$\nu = 1.0 \times 10^{-3}$$

となる。また、非定常 Navier-Stokes 方程式の ALE 表示の PSPG 法に関する部分は以下のように表すことができる [24]。

$$\int_{V_x} \tau_{PSPG} \frac{1}{\rho} \nabla \delta p \cdot \left\{ \rho \frac{\partial \mathbf{v}}{\partial t} \Big|_{\mathbf{x}} + \rho \mathbf{c} \cdot (\nabla_{\mathbf{x}} \otimes \mathbf{v}) + \nabla_{\mathbf{x}} p - \mu \Delta_{\mathbf{x}} \mathbf{v} - \rho \mathbf{g} \right\} dV_x \quad (4.3.6)$$

よって、圧力-圧力ブロックの安定化項は

$$\int_{V_x} \tau_{PSPG} \frac{1}{\rho} \nabla \delta p \cdot \nabla_{\mathbf{x}} p dV_x$$

で与えられる。よって圧力-圧力ブロックの安定化項の係数は

$$\frac{\tau_{PSPG}}{\rho}$$

で与えられる。すべての値を代入すれば式 (4.3.5) から

$$\begin{aligned} \frac{\tau_{PSPG}}{\rho} &= \frac{1}{\rho} \times \frac{(h)^2}{4\nu} \\ &= \frac{1}{4} h^2 \end{aligned}$$

となる。

ここで定常及び非定常 Navier-Stokes 問題に拡張した際の τ_{PSPG} の値を考える。式 (4.3.5) を書き換えると以下のようにあらわすことができる。

$$\tau_{PSPG} = \frac{1}{\sqrt{\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{2\|\mathbf{c}\|}{h^2}\right)^2 + \left\{\frac{4\nu}{(h^2)^2}\right\}^2}} \quad (4.3.7)$$

この式は以下のように評価できる。

$$\frac{1}{\sqrt{\left(\frac{2}{\Delta t}\right)^2 + \left(\frac{2\|\mathbf{c}\|}{h^2}\right)^2 + \left\{\frac{4\nu}{(h^2)^2}\right\}^2}} < \frac{1}{\sqrt{\left\{\frac{4\nu}{(h^2)^2}\right\}^2}} \quad (4.3.8)$$

となる。右辺は仮に PSPG 法を定常 Stokes 問題に適用した場合の安定化パラメータである。従って移流項や慣性項に対する安定化項が加わった場合の τ_{PSPG} の値は移流項や慣性項に関する安定化項が無い場合の τ_{PSPG} に比べ値が小さくなるのがわかる。

以上の考察から，本章定常 Stokes 問題においてはパラメータ

$$\varepsilon = \frac{1}{4}h^2 \quad (4.3.9)$$

に対して有効であるソルバーを開発することができれば，次章以降の定常及び非定常 Navier-Stokes 問題に対しても，少なくとも圧力部分の安定化に関しては十分強固なものになると考えられる。

4.3.3 本章の目的

以上の考察から，本章の目的は「式 (4.3.9) で与えられるような ε に対する定常 Stokes 問題に対して有効な Multigrid ソルバーを開発する。」こととする。

なお，Stokes 方程式は Navier-Stokes 方程式において粘性が無限大，すなわち質量が 0 であり移流が無視できる流体をあらわしている。

4.4 問題設定

4.4.1 計算条件

本章においては正方形領域のキャビティフロー問題を考えることとする。図 4.2 においては，赤い矢印が存在する上部境界には一定の流速を与え，それ以外の流速境界条件は 0 とする。また境界条件のすべての点において圧力の値は与えない。特に断りが無い限り，本章においては以下に述べる条件を適用する。収束判定条件は相対残差が 1.0×10^{-8} を下回った時とする。メッシュは正方形領域を x 軸，y 軸方向に 128 等分したものとする。Multigrid の層数は 5 階層とする。要素は $4/4c$ を用いた。

4.4.2 結果の表示法について

本章においては表・図中の反復回数の中に > 1000 という表示があるものは，反復回数 1000 回以上を意味する（すなわち発散はせずに収束していることを意味する）。また，表・図中の反復回数の中に発散という表示があるものは，計算が発散してしまったことを意味する。いずれの場合もそれに対応する計算時間は \times とあrawす。GMRES 法のリスタートは 200 回。ILU は鷺尾らが開発したフィルインを制御した混合型 FEM

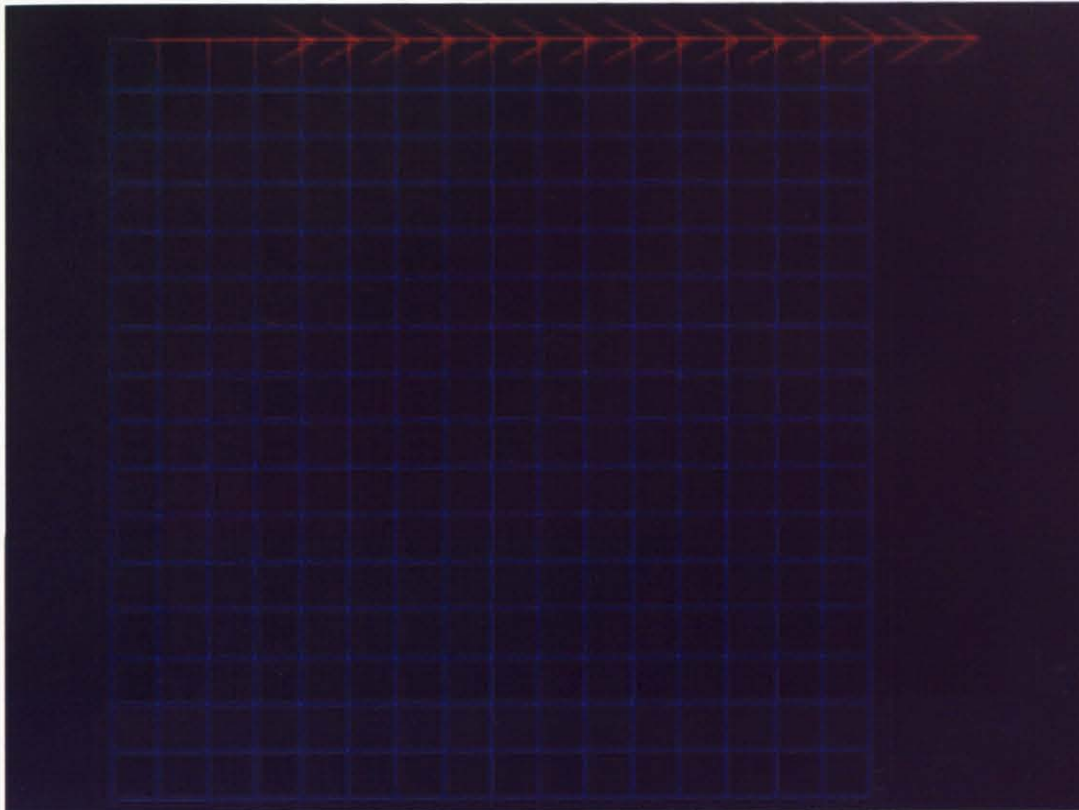


図 4.2: 本章 (定常 Stokes 方程式) における問題設定

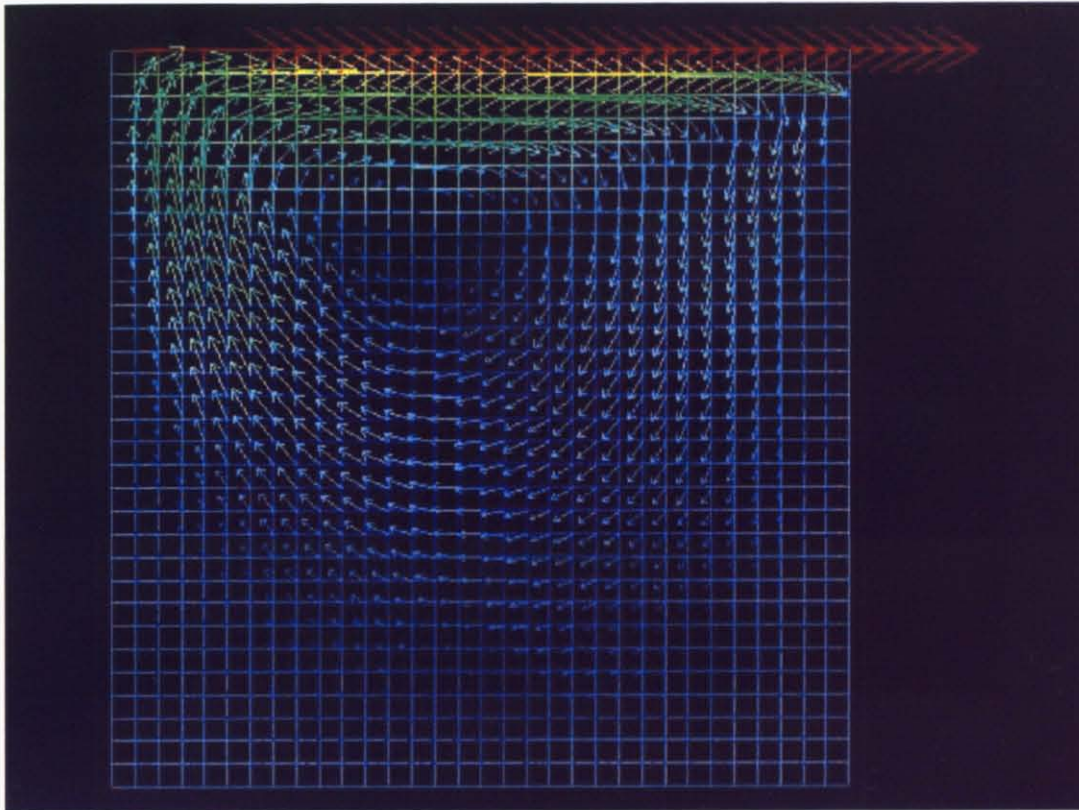


図 4.3: 定常 Stokes 問題 (キャビティーフロー) の流れの様子

用のILU分解・前処理^[1]を用いている。GMRES法のリスタート回数は大きくとった方が反復回数は少なくなるが、メモリの観点から500程度が限界である。また、本問題においてはリスタート200回と500回では反復回数に大きな差が無いことや、リスタートを大きく取ると計算に時間がかかることなどから200回で固定とした。以下の章でも同様とする。

また n を各軸方向に対する分割数としたとき、 $h = \frac{1}{n}$ とする。また、ILU前処理やILU-smootherに関しては、fill-levelの制御は以下のように行った。

1. (2,2)ブロック上に(2,1),(1,2)ブロックの組み合わせから生成されるフィルインをいつでも許す。
2. 上記以外のフィルインを許さない。

以後、本論文においては特に断りが無い場合はこのフィルインパターンを用いることとする。

4.4.3 要素の選定について

本論文においては4/4c要素を用いて計算を行っている。これは4/1d要素だと圧力振動が発生するためである。また、4/3c要素のようなバブル節点を持つ要素を用いることも考えられる。しかしながら心臓問題でこのような要素を取り扱う場合には、自由度の減少及び圧力の安定化を目的としてソルバー内でバブル節点を消去し、行列的には3/3c要素(3次元においては4/4c要素)として扱っていることとなる。従って、バブル節点が無い要素で安定した解法を構築できれば今後心臓問題にも応用することができるため、今回は4/4c要素を選定した。

4.4.4 各パラメータについて

4.3節で述べたように、本章において各パラメータは以下のものを使用する。

$$\mu = 1.0$$

$$\rho = 1.0 \times 10^3$$

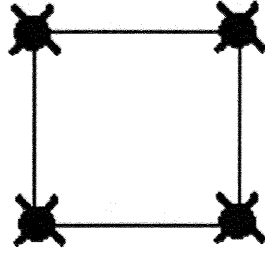


図 4.4: 4/1d 要素

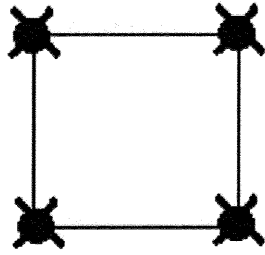


図 4.5: 4/4c 要素

$$\nu = 1.0 \times 10^{-3}$$

$$\varepsilon = \frac{1}{4}h^2$$

4.4.5 計算機性能

本論文の数値実験においては CPU が Pentium4 3.2GHz, Memory が 2GB である計算機を用いた.

4.5 計算結果 (既存の代表的なソルバーを用いた場合の収束性の比較)

4.5.1 計算結果 ($\varepsilon = 1.0 \times 10^2 h^2$ の場合)

まず, ε を大きめ取った $\varepsilon = 1.0 \times 10^2 h^2$ の場合の結果は表 4.1 のようになる. 定常 Stokes 問題を扱う場合, ε をかなり大きめにすれば, MG 系統の解法は ILU-GMRES

表 4.1: GSMG 法, ILUMG 法, ILU-GMRES 法の比較 ($\varepsilon = 1.0 \times 10^2 h^2$)

手法名	反復回数	計算時間 (秒)
GSMG	114	9
ILUMG	19	7
ILU-GMRES	152	17

法に比べて反復回数が少ないことがわかる。すなわち、MG 系統の解法は ε が大きい問題に対しては既存のソルバーである ILU-GMRES 法に比べて収束性が良いといえる。しかしながら、小さな ε に対しては MG 系統の解法は発散してしまうことを次小節において示す。

4.5.2 計算結果 ($\varepsilon = 2.5 \times 10^{-1} h^2$ の場合)

続いて、 ε を本章で目標とする式 (4.3.9) で与えられる $2.5 \times 10^{-1} h^2$ とした時の結果は以下のようになる。

表 4.2: GSMG 法, ILUMG 法, ILU-GMRES 法の比較 ($\varepsilon = 2.5 \times 10^{-1} h^2$)

手法名	反復回数	計算時間 (秒)
GSMG	発散	×
ILUMG	発散	×
ILU-GMRES	290	49

この場合、既存の MG 法である GSMG 法や ILUMG 法では発散してしまうことがわかる。現在の研究室のソルバーである ILU-GMRES 法を用いれば安定して解くことができている。この結果から MG 系統の解法は小さな ε に対する問題には不安定といえる。

4.5.3 4.5 節結言

本章で目標とする $\varepsilon = 2.5 \times 10^{-1} h^2$ に対する問題を解く場合、既存の Multigrid ソルバーを用いると収束させることができない。これは Multigrid 法に用いる現在の

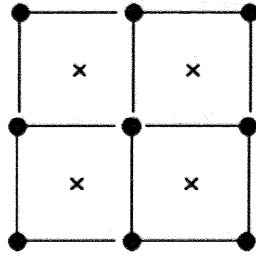


図 4.6: 全体領域

smoother (GS-smoother 及び ILU-smoother) が不安定であるからと考えられる. 次節ではこのような問題にも適用できる, 本論文にて開発した EBEMP-smoother について紹介していく.

4.6 Element by Element Multiplicative Multigrid 法

4.6.1 EBEMPMG 法概説

ここでは Element by Element Multiplicative Multigrid 法 (EBEMPMG 法) を提唱し, 説明する. これは一言で言うと, 「smoother をかける際に要素ごとに更新して足し合わせていく」というものである. 例えば図 4.6 のような全体領域が与えられたとする. まずは左下の要素に関して更新を行う (図 4.7). 続いて右下の要素 (図 4.8), 続いて左上の要素 (図 4.9), 最後に右上の要素 (図 4.10) というように左下から右上に関して順々に更新を行う手法である. 更新の順序に関しては特に制約はないが, 本論文においてはこのように右下から左上に向かう順序とした.

このように要素単位の更新を用いるのは以下の理由からである. 圧力-圧力ブロックの値は 0 もしくは非常に小さな値であり, 節点単位の更新を用いる場合, 緩和は難しいと言える. そこで, ブロック内 (本論文では要素内) の流速を消去し, 圧力-圧力ブロックに fill-in を生じさせることにより緩和が可能になるのである.

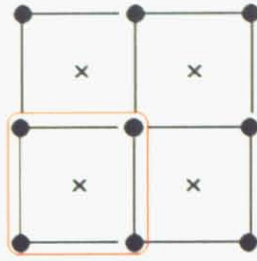


図 4.7: 1 番目の要素における更新

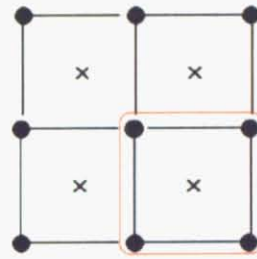


図 4.8: 2 番目の要素における更新

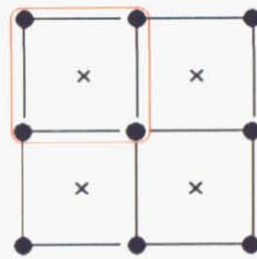


図 4.9: 3 番目の要素における更新

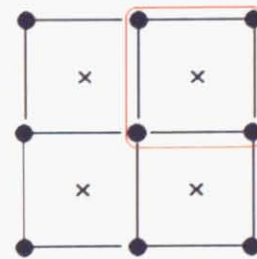


図 4.10: 4 番目の要素における更新

4.6.2 EBEMP-smoother

領域全体から、各要素 e への制限を R_e で定義し、各要素上における行列をそれぞれ A_e とすれば、全体行列を A として

$$A_e = R_e A R_e^T$$

とあわせることができる。ここで A_e は一般に FEM で用いられる要素剛性行列とはことなることに注意する。要素剛性行列は行列がフルランクではないため、要素ごとに更新量を求めることができないためである。続いて、この A_e を (完全) LU 分解する。すなわち $A_e = L_e D_e^{-1} U_e$ と分解する。ここで L_e は狭義下三角行列、 U_e は狭義上三角行列、 D_e は対角行列である。また、各要素 e における残差を $r_e = R_e(b - Ax)$ と定める。各要素 e に対して順番に

$$L_e D_e^{-1} U_e \Delta x_e = r_e$$

を解き Δx_e を求め

$$x_{new} = x + \Delta x_e$$

を足しこんでいく。現在の方法では更新は要素番号順に行っている。このように要素ごとに更新を行い、全体領域の解に足しこんでいく smoother を Element by Element Multiplicative smoother (以下 EBEMP-smoother) と呼ぶことにする。EBEMP-smoother を用いた Multigrid 法を Element by Element Multiplicative Multigrid 法 (以下 EBEMPMG 法) と呼ぶことにする。

4.6.3 EBEMPMG 法を用いた場合の誤差の考察

定義 4.6.1 (誤差) 真の解を \tilde{x} 、現在の解 x とすればその時点での誤差 e は

$$e = \tilde{x} - x \tag{4.6.1}$$

で定義される。



誤差を利用すれば解の誤差ノルムのひとつとして以下のものが定義出来る。

定義 4.6.2 (誤差ノルム) 連立一次方程式 $Ax = b$ の係数行列 A が正定値対称ならば、誤差ノルムのひとつとして

$$E(A) = \frac{1}{2} e^T A e \quad (4.6.2)$$

を考えることができる。



さて、全体領域 D からその部分領域 D_k への制限行列を R_k とする。また、 D_k において update を行う直前の解を x_k とし D_k において update を行った解 x_{k+1} を

$$x_{k+1} = x_k + \delta x_k$$

とする。すると明らかに

$$\delta x_k = A(x_k + \delta x_k) = b \text{ on } D \quad (4.6.3)$$

$$\delta x_k = 0 \text{ on } D^C \quad (4.6.4)$$

が成り立つ。更に

$$\delta x_k = -\delta e_k \quad (4.6.5)$$

が成り立つ事にも注意しておく。これらを用いれば D_k 上における誤差 e_k に対して次のことが成り立たなければならない事がわかる。

$$A(e_k + \delta e_k) = 0 \text{ on } D_k \quad (4.6.6)$$

$$\delta e_k = 0 \text{ on } D_k^C \quad (4.6.7)$$

定理 4.6.3 (誤差の評価) 全体領域 D の部分領域 D_k において残差が 0 となるように解の update が行なわれたとする。このとき式 (4.6.2) で与えられる誤差ノルム E は必ず減少し、その減少量は以下の式で与えられる。

$$E(e_k + \delta e_k) - E(e_k) = -\frac{1}{2} \delta e_k^T A \delta e_k < 0 \quad (4.6.8)$$



[証明]

$$E(\mathbf{e}_k + \delta\mathbf{e}_k) = E(\mathbf{e}_k) + \delta\mathbf{e}_k^T A\mathbf{e}_k + \frac{1}{2}\delta\mathbf{e}_k^T A\delta\mathbf{e}_k \quad (4.6.9)$$

$$= E(\mathbf{e}_k) + \delta\mathbf{e}_k^T A(\mathbf{e}_k + \delta\mathbf{e}_k - \delta\mathbf{e}_k) + \frac{1}{2}\delta\mathbf{e}_k^T A\delta\mathbf{e}_k \quad (4.6.10)$$

$$= E(\mathbf{e}_k) + \delta\mathbf{e}_k^T A(\mathbf{e}_k + \delta\mathbf{e}_k) - \frac{1}{2}\delta\mathbf{e}_k^T A\delta\mathbf{e}_k \quad (4.6.11)$$

$$(4.6.12)$$

よって式(4.6.6),(4.6.7)を考慮すれば

$$E(\mathbf{e}_k + \delta\mathbf{e}_k) - E(\mathbf{e}_k) = -\frac{1}{2}\delta\mathbf{e}_k^T A\delta\mathbf{e}_k < 0 \quad (4.6.13)$$

□

定理4.6.3は、係数行列 A が正定値であれば定義4.6.2で与えられるような誤差ノルムは単調に減少していくことを示している。 A が不定値であるときは $\delta\mathbf{e}_k^T A\delta\mathbf{e}_k > 0$ とは言えず、式(4.6.13)が成り立たないため単調減少であることが言えない。ここに不定値行列の難しさがある。

4.7 計算結果 (EBEMPMG法と既存のソルバーを用いた場合の収束性の比較)

4.7.1 計算結果 ($\varepsilon = 1.0 \times 10^2 h^2$ の場合)

4.5節と全く同様に考えていく。 $\varepsilon = 1.0 \times 10^2 h^2$ の時の結果は以下のようになる。

表 4.3: GSMG法, ILUMG法, EBEMPMG法, ILU-GMRES法の比較 ($\varepsilon = 1.0 \times 10^2 h^2$)

手法名	反復回数	計算時間(秒)
GSMG	114	9
ILUMG	19	7
ILU-GMRES	152	17
EBEMPMG	6	1

EBEMPMG 法が最も計算時間が短い。また、EBEMPMG 法は Multigrid 的解法であるため、自由度数が増加しても反復数は増加しない傾向にある。従って、 $\varepsilon = 1.0 \times 10^2 h^2$ の場合には、(大規模問題に対しては特に) EBEMPMG 法を用いるのが良いといえよう。

4.7.2 計算結果 ($\varepsilon = 2.5 \times 10^{-1} h^2$ の場合)

$\varepsilon = 2.5 \times 10^{-1} h^2$ とした時の結果は以下ようになる。

表 4.4: GSMG 法, ILUMG 法, EBEMPMG 法及び ILU-GMRES 法の比較 ($\varepsilon = 2.5 \times 10^{-1} h^2$)

手法名	反復回数	計算時間 (秒)
GSMG	発散	×
ILUMG	発散	×
EBEMPMG	137	41
ILU-GMRES	290	49

EBEMPMG 法を用いれば反復回数 137 回と、実用的な範囲で収束させることができている。計算時間も最も短い。現在の研究室心臓シミュレータのソルバーである ILU-GMRES 法でも安定して解くことができている。また、EBEMPMG 法は Multigrid 的解法であるため、自由度数が増加しても反復数は増加しない傾向にある。従って、 $\varepsilon = 2.5 \times 10^{-1} h^2$ の場合は (大規模問題に対しては特に) EBEMPMG 法を用いるのが良いといえよう。

4.7.3 結言

EBEMPMG 法は既存の Multigrid 法では発散していたような定常 Stokes 問題でも収束させることができる。問題点は、実用上重要と考えられる $\varepsilon = 2.5 \times 10^{-1} h^2$ では計算に時間がかかってしまっていることであろう。後に詳しく述べるが、同様の例題で $\varepsilon = 1.0 \times 10^{-1} h^2$ と、更に小さな ε を用いた場合 EBEMPMG 法をもってしても発散してしまう。一方 ILU-GMRES 法はこのような小さな ε に対して問題なく収束する。すなわち、 ε が大きな問題に対しては EBEMPMG 法は ILU-GMRES 法に比べて有用で

あるが、 ε が小さな問題に対しては用いることができない。圧力部の安定性の面から EBEMPMG 法の改良が必要といえる。

4.8 Element by Element Multiplicative Pressure Stabilized Multigrid 法

4.8.1 Element by Element Multiplicative Pressure Stabilized Multigrid 法の概説

続いて、Element by Element Multiplicative Pressure Stabilized Multigrid 法（以下 EBEMPPSMG 法）について説明する。これは、一言で言うと「Multigrid 法の中で用いる安定化手法のパラメータ ε_{MG} の値と元の係数行列に用いる安定化手法のパラメータ ε に対して違う値を用いる」という手法である。このような手法を Pressure Stabilized Multigrid 法（以下 PSMG 法）と呼ぶことにし、EBEMP-smoother をこの手法に用いた際に EBEMPPSMG 法と呼ぶことにする。普通は $\varepsilon_{MG} > \varepsilon$ であり $\varepsilon_{MG} = \varepsilon$ とすれば EBEMPMG 法に帰着する。その意味で EBEMPPSMG 法はここまで述べてきた EBEMPMG 法を拡張したものといえる。以下 EBEMPPSMG 法を説明する。

安定化手法のパラメータ ε として ε_{MG} を用いた時、その係数行列 \mathbf{A} を特に $\mathbf{A}(\varepsilon_{MG})$ 、マルチグリッドサイクル MG を特に $MG(\varepsilon_{MG})$ と書くことにする。 ε として ε を用いた際は同様に $\mathbf{A}(\varepsilon)$ 、 $MG(\varepsilon)$ とあらわす。

解くべき連立一次方程式は

$$\mathbf{A}(\varepsilon)\mathbf{x} = \mathbf{b}$$

であるとする。 ε_{MG} は ε より大きな値であるとする。EBEMPPSMG 法とは以下の方法である。

1. Calculate $\mathbf{r}^k = \mathbf{b} - \mathbf{A}(\varepsilon)\mathbf{x}^k$
2. Solve $\Delta\mathbf{x}^k = MG(\mathbf{A}(\varepsilon_{MG}), \mathbf{r}^k, \mathbf{0})$
3. Update $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k$

ここで関数 $MG(\mathbf{A}(\varepsilon_{MG}), \mathbf{r}^k, \mathbf{0})$ は式 (3.3.2) で与えられるものとする。

ε が小さく、 $A(\varepsilon)$ に対して普通に Multigrid 法を適用すると発散してしまう場合でも、Multigrid 法では $A(\varepsilon_{MG})$ を用い更新量を求めることにより、発散を抑える効果があると期待できる。第 4.7 節で述べてきたように、Multigrid 法は ε の値によって収束性を左右されやすい。そこで、Multigrid サイクルで用いる ε_{MG} を Multigrid 法によって解きやすい値にすることによって、収束性を向上させるという考え方である。

この方法を用いると、Multigrid サイクルの収束性は非常に高くなる。しかしながら ε からあまりにかけ離れた $A(\varepsilon_{MG})$ を用いると、Multigrid サイクルは簡単に収束するものの、そのサイクルで求めた修正量 Δx^k が元の係数行列 $A(\varepsilon)$ に対する修正量とは大きく乖離してしまう。以降では EBEMPPSMG 法の収束性評価に加えて ε_{MG} の定め方について検討していく。

4.8.2 計算結果

各 ε に対する GSMG 法、ILUMG 法、EBEMPMG 法及び EBEMPPSMG 法の収束性の比較は表 4.5 で与えられる。なお、 ε_{MG} の値は各々の ε に対して適当に定めた（詳しい定め方は第 4.10 小節にて述べる）。また、ILUMG 法は各反復において前処理行列を作り直しているため、アルゴリズムを改良すれば反復数は変わらないものの計算時間の短縮は可能である。

4.8.3 考察

小節 4.3.2 の考察から $\varepsilon = 2.5 \times 10^{-1} h^2$ で収束することが望ましい。古典的な GSMG 法、及び smoother を ILU に変更した ILUMG 法を用いた場合では $\varepsilon = 1.0 \times 10^2 h^2$ 程度までしか解くことができず、十分ではない。続いて EBEMPMG 法を用いた場合 $\varepsilon = 2.5 \times 10^{-1} h^2$ に対しては収束させることができるものの $\varepsilon = 1.0 \times 10^{-1} h^2$ 以下になると発散してしまう。EBEMPMG 法を用いた場合は $\varepsilon = 2.5 \times 10^{-1} h^2$ 程度の大きさがぎりぎりのラインであるといえる。最後に EBEMPPSMG 法を用いた場合 $\varepsilon = 1.0 \times 10^{-3} h^2$ と、MG 系統の解法としてはかなり小さな ε に対して収束させることができおり、非常に安定性が高い。また、 $\varepsilon = 2.5 \times 10^{-1} h^2$ の時も、EBEMPMG 法に対して計算時間が 2 分の 1 になっており、EBEMPPSMG 法は安定性・収束性の両面からみて非常に強力な Multigrid 法であるといえよう。

表 4.5: 各 ε に対する GSMG 法, ILUMG 法, EBEMPMG 法及び EBEMPPSMG 法の収束性の比較

手法名	ε	反復回数	計算時間 (秒)	ε_{MG}
GSMG	$1.0 \times 10^2 h^2$	114	9	—
GSMG	$1.0 \times 10^1 h^2$	発散	×	—
GSMG	$1.0 h^2$	発散	×	—
GSMG	$2.5 \times 10^{-1} h^2$	発散	×	—
GSMG	$1.0 \times 10^{-1} h^2$	発散	×	—
GSMG	$1.0 \times 10^{-2} h^2$	発散	×	—
GSMG	$1.0 \times 10^{-3} h^2$	発散	×	—
ILUMG	$1.0 \times 10^2 h^2$	19	124	—
ILUMG	$1.0 \times 10^1 h^2$	発散	×	—
ILUMG	$1.0 h^2$	発散	×	—
ILUMG	$2.5 \times 10^{-1} h^2$	発散	×	—
ILUMG	$1.0 \times 10^{-1} h^2$	発散	×	—
ILUMG	$1.0 \times 10^{-2} h^2$	発散	×	—
ILUMG	$1.0 \times 10^{-3} h^2$	発散	×	—
EBEMPMG	$1.0 \times 10^1 h^2$	11	3	—
EBEMPMG	$1.0 h^2$	41	12	—
EBEMPMG	$2.5 \times 10^{-1} h^2$	137	43	—
EBEMPMG	$1.0 \times 10^{-1} h^2$	発散	×	—
EBEMPMG	$1.0 \times 10^{-2} h^2$	発散	×	—
EBEMPMG	$1.0 \times 10^{-3} h^2$	発散	×	—
EBEMPPSMG	$1.0 \times 10^1 h^2$	11	4	$1.1 \times 10^2 h^2$
EBEMPPSMG	$1.0 h^2$	57	18	$5.0 h^2$
EBEMPPSMG	$2.5 \times 10^{-1} h^2$	59	18	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-1} h^2$	76	23	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-2} h^2$	706	216	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-3} h^2$	1000+	×	$5.0 \times 10^{-1} h^2$

4.8.4 結言

EBEMPMG 法及び EBEMPPSMG 法を用いた場合、古典的な GSMG 法などで収束させることができないような $\varepsilon = 1.0 \times 10^{-1}h^2$ 以下の問題でも収束させることができる。また、EBEMPMG 法を用いれば現実的に使われるであろう $\varepsilon = 2.5 \times 10^{-1}h^2$ に対しても収束させることが可能であるが、収束性がやや弱い。またそれよりやや小さい値 $\varepsilon = 1.0 \times 10^{-1}h^2$ に対しては発散してしまい、不安定性が付きまとう。EBEMPPSMG 法を用いれば $\varepsilon = 2.5 \times 10^{-1}h^2$ に対する収束性も EBEMPMG 法より 2 倍程度良く、それ以下の ε に対しても十分な安定性を達成している。

なお、EBEMPPS-smoother というものは存在しないことに注意する。本節では新しい smoother を作り出したわけではなく、EBEMP-smoother を安定化された MG 法 (PSMG 法) の中で用いているだけである。

EBEMPPSMG 法の問題点は、 $\varepsilon = 1.0 \times 10^{-2}h^2$ 以下では、収束はするものの計算に時間がかかってしまうことである。また、与えられた問題と ε に対して、どのような ε_{MG} を設定するかが難しい。今後考察していく。

4.9 MG-GMRES 法

第 3 章 3.3.5 節で述べたように、MG を前処理として利用して GMRES 法を用いることができる。このような方法を MG-GMRES 法と呼ぶことにする。例えば GSMG 法を前処理とした GMRES 法を GSMG-GMRES 法と呼ぶことにする。同様に ILUMG-GMRES 法、EBEMPMG-GMRES 法および EBEMPPSMG-GMRES 法などを考えることができる。本節では EBEMPPSMG 法と EBEMPPSMG-GMRES 法および ILU-GMRES 法の性能比較を行う。

4.9.1 計算結果

結果は表 4.9 及び図 4.11, 図 4.12 のようになった。

これらの表・図から、EBEMPPSMG-GMRES 法を用いれば、ILU-GMRES 法に比べて収束性が良いことがわかる。また、EBEMPPSMG 法を単体で用いるより、EBEMPPSMG 法を前処理とした GMRES 法 (EBEMPPSMG-GMRES 法) の方が優れた

表 4.6: 各 ε に対する ILU-GMRES 法, EBEMPMG 法及び EBEMPPSMG 法の収束性の比較

手法名	ε	反復回数	計算時間 (秒)	ε_{MG}
ILU-GMRES	$1.0 \times 10^1 h^2$	212	30	—
ILU-GMRES	$1.0 h^2$	271	44	—
ILU-GMRES	$2.5 \times 10^{-1} h^2$	290	52	—
ILU-GMRES	$1.0 \times 10^{-1} h^2$	296	54	—
ILU-GMRES	$1.0 \times 10^{-2} h^2$	296	52	—
ILU-GMRES	$1.0 \times 10^{-3} h^2$	305	56	—
EBEMPPSMG	$1.0 \times 10^1 h^2$	11	4	$1.1 \times 10^1 h^2$
EBEMPPSMG	$1.0 h^2$	57	18	$5.0 h^2$
EBEMPPSMG	$2.5 \times 10^{-1} h^2$	59	18	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-1} h^2$	76	23	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-2} h^2$	706	216	$5.0 \times 10^{-1} h^2$
EBEMPPSMG	$1.0 \times 10^{-3} h^2$	1000+	×	$5.0 \times 10^{-1} h^2$
EBEMPPSMG-GMRES	$1.0 \times 10^1 h^2$	8	2	$1.1 \times 10^1 h^2$
EBEMPPSMG-GMRES	$1.0 h^2$	21	5	$5.0 h^2$
EBEMPPSMG-GMRES	$2.5 \times 10^{-1} h^2$	37	9	$5.0 \times 10^{-1} h^2$
EBEMPPSMG-GMRES	$1.0 \times 10^{-1} h^2$	44	10	$5.0 \times 10^{-1} h^2$
EBEMPPSMG-GMRES	$1.0 \times 10^{-2} h^2$	88	31	$5.0 \times 10^{-1} h^2$
EBEMPPSMG-GMRES	$1.0 \times 10^{-3} h^2$	224	85	$5.0 \times 10^{-1} h^2$

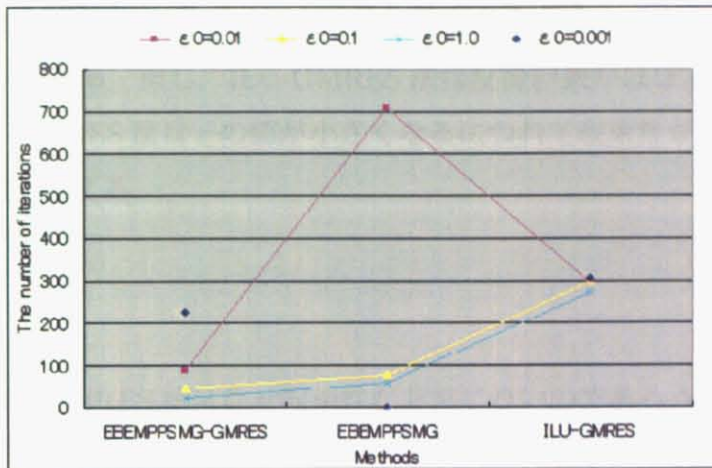


図 4.11: それぞれの手法における反復数の比較

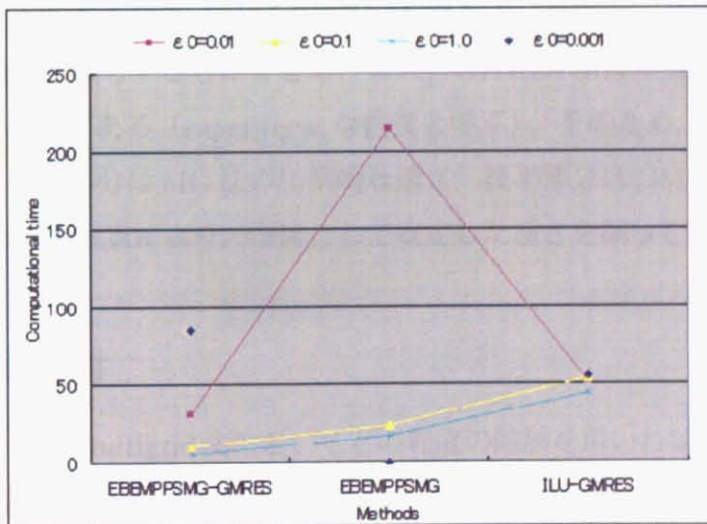


図 4.12: それぞれの手法における計算時間の比較

収束性を持っていることがわかる。また EBEMPPSMG-GMRES 法は安定性も非常に高く、ここまでの MG 法システムのソルバーを用いて収束させることができなかつた $\varepsilon = 1.0 \times 10^{-2}h^2, \varepsilon = 1.0 \times 10^{-3}h^2$ などの問題も収束させることができる非常に強力なソルバーである。但し、ILU-GMRES 法は反復回数がほぼ ε に依存しないが、EBEMPPSMG-GMRES 法は ε の値が小さくなるにつれて収束性が悪くなっていることも大きな特徴といえる。

4.9.2 考察

EBEMPPSMG-GMRES 法はなぜ収束性が非常に良いのであろうか。MG 法システム解法の特徴としては、序盤はすごい勢いで残差が落ちる。しかしながら、終盤には course grid collection の効果がなくなり、収束性がなまる。これは、MG 法は階層構造であるため、最も細かいメッシュにとってはそれほどでない ε の値でも、最も粗いメッシュにとってはかなりの値になるであるからと考える。そこで、粗いメッシュに対してはより大きな ε を用いる方法なども試みたが、一定の効果をおよぼすことはできたものの、劇的な効果をおよぼすまでには至らなかつた。一方、GMRES 法はかなり小さい ε に対しても収束させることができる。実際、収束に時間はかかるものの $\varepsilon = 1.0 \times 10^{-8}h^2$ 程度までは発散せずに収束させることができる。すなわち、GMRES 法は小さな ε に対しても安定して残差を減らすことができる。また、GMRES 法は反復回数が増えるごとに収束性が増す傾向がある (superlinear な性質と呼ぶ)。そのため、2 つ解法をあわせることにより、前半部分は MG 法の圧倒的な速さ、後半部分は GMRES 法の安定した速さと superlinear の性質により、全体として安定した速さを保っていると考えられる。

4.9.3 階層の検討

本小節においては Multigrid 法においてどの程度の階層を用いればよいかを検証していく。表 4.9 および 4.12 の実験結果から $2^n \times 2^n$ メッシュの階層は (n-1) 段 ~ (n-2) 段程度が良いことが分かった。これは単純に、段数が増えるほど粗いグリッドの行列が小さくなり、粗いグリッドで解く際の計算量が減るからである。実際階層数が少ないほど、反復回数は少ないことが多いが、計算時間の観点で見ると階層数ができるだけ多いほうが良く、従って (n-1) 段 ~ (n-2) 段程度が最も良いことが多い。n 段に関してはメ

メモリ不足などで体系的なデータを取ることはできなかったが、採取できたデータを見ると (n-1) 段~(n-2) 段と同程度の収束性を有していた。なお n+1 段とすると、最も粗いメッシュが 1×1 になるため、行列が境界条件のみとなり解くことができない。

例えば 128×128 なら 5~7 段の階層で解けば良い。

表 4.7: 各階層に対する EBEMPPSMG-GMRES 法の収束性の比較 (メッシュサイズは 128×128)

手法名	ε	ε_{MG}	階層	反復回数	計算時間 (秒)
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	3	38	31
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	4	46	20
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	5	37	11
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	6	43	13

表 4.8: 各階層に対する EBEMPPSMG-GMRES 法の収束性の比較 (メッシュサイズは 256×256)

手法名	ε	ε_{MG}	階層	反復回数	計算時間 (秒)
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	3	52	175
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	4	101	190
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	5	45	63
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	6	47	61

4.10 ε_{MG} の定め方について

4.10.1 ε_{MG} の定め方についての検証 (1)

本節では各 ε に対する ε_{MG} の定め方について検証していく。実験的には ε が $5.0 \times 10^{-1}h^2$ 以下程度なら ε_{MG} は $5.0 \times 10^{-1}h^2$ 程度が良い。 ε が $1.0 \times 10^{-1}h^2$ であっても

$1.0 \times 10^{-3}h^2$ であっても ε_{MG} としては同じ $5.0 \times 10^{-1}h^2$ を用いることが重要である。この理由を以下に説明する。

EBEMPMG 法は本来、表 4.9 で見たように単体では $\varepsilon = 1.0 \times 10^{-1}h^2$ で発散してしまった。すなわち $\varepsilon = 5.0 \times 10^{-1}h^2$ 程度が下限に近いと考えられる。従って、これ以下の ε_{MG} を用いると、MG 法は機能しなくなり、そのため smoother として用いた際にも機能しなくなると考えられるからである。 ε が $\varepsilon = 5.0 \times 10^{-1}h^2$ より大きな場合は、 $\alpha = 1.1 \sim 2$ 程度として $\varepsilon_{MG} = \alpha\varepsilon$ で定義すればよい。この程度の大きさの ε であれば、EBEMPPSMG 法を用いずとも EBEMPMG 法でも収束するレベルであるため、 α として大きな値を取ると逆に収束が妨げられる。

表 4.9: 各 ε_{MG} に対する EBEMPPSMG-GMRES 法の収束性の比較

手法名	ε	階層	ε_{MG}	反復回数	計算時間 (秒)
EBEMPPSMG-GMRES	$1.0 \times 10^{-2}h^2$	6	$3.0h^2$	109	37
EBEMPPSMG-GMRES	$1.0 \times 10^{-2}h^2$	6	h^2	94	31
EBEMPPSMG-GMRES	$1.0 \times 10^{-2}h^2$	6	$5.0 \times 10^{-1}h^2$	98	33
EBEMPPSMG-GMRES	$1.0 \times 10^{-2}h^2$	6	$3.0 \times 10^{-1}h^2$	131	44
EBEMPPSMG-GMRES	$1.0 \times 10^{-2}h^2$	6	$1.0 \times 10^{-1}h^2$	926	344

ε が非常に小さな値で解きづらい問題でも、Multigrid 部分の ε_{MG} を大きめにとることにより収束させることが可能になる。これは、Multigrid 部分では ε_{MG} に関する問題を解いていることになるため、修正量 Δx は容易に求まるためと言えよう。 ε_{MG} を大きくするほど Multigrid 部分は解きやすくなるが、求まった修正量 Δx は、元々 ε で与えられている問題からは大きくかけ離れたもの（乖離したもの）になり、修正を行ってもほとんど効果が無くなる。すなわち、修正量 Δx は容易に求まり、かつ修正にある程度効果がある最適な ε_{MG} を定める必要があると言えよう。

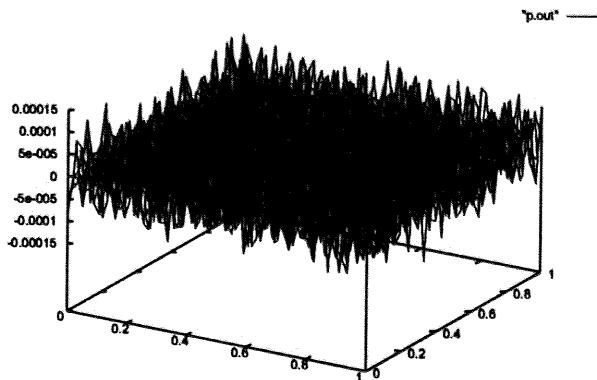


図 4.13: 減衰せずに残る誤差

4.10.2 ε_{MG} の定め方についての検証 (2)

ここまでの考察を実験的に検証していく。初期値を乱数で発散させて、EBEMPPSMG-GMRES 法を用いて

$$A(\varepsilon)\mathbf{x} = \mathbf{0}$$

を解き、ある地点で計算を打ち切り、その時点で残っている誤差を調べたところ、図 4.13 のようになった。 $\varepsilon = 1.0 \times 10^{-3}h^2$, $\varepsilon_{MG} = 5.0 \times 10^{-1}h^2$ とした。また、計算は 100 ステップ目で打ち切った。このときの残差は 10^{-5} 程度である。サイズは 64×64 で要素数は 4096、節点数は 8450、自由度は 12289、要素は $4/4c$ を用いた。この図から、圧力振動等の特別なモードが収束を妨げているわけではないことがわかる。EBEMPPSMG-GMRES 法では $\varepsilon = 1.0 \times 10^{-3}h^2$ より小さな ε に対しては、これ以上の収束性の向上は望めないと考えられる。その原因は以下のように考えられる。

Multigrid 法における pre-smoother 及び post-smoother の回数を変化させて、100 回の時点での最も粗いグリッドにおける

$$\frac{\|\mathbf{r} - \text{MG}(\varepsilon_{MG})\delta\mathbf{x}^k\|}{\|\mathbf{r}\|}$$

の出力する。この値が 1 より大幅に小さくなっているということは MG 法による修正量はきちんと求まっているということを意味する。結果は表 4.10 のとおりである。この結果から、course-grid における MG 法による修正量 $\delta\mathbf{x}^k$ はきちんと求まってい

表 4.10: 100 回反復時の $\frac{\|r - \text{MG}(\varepsilon_{MG})\Delta x^k\|}{\|r\|}$ の値

各レベルにおける smoother の回数	$\frac{\ r - \text{MG}(\varepsilon_{MG})\Delta x^k\ }{\ r\ }$ の値
1 回	0.974
5 回	3.14×10^{-2}
10 回	9.59×10^{-4}

が、 ε と ε_{MG} の間の乖離が大きいため、 ε_{MG} に対して求めた δx^k が ε に対する x^k あっていないといえる。すなわち修正

$$x^{k+1} = x^k + \delta x^k$$

の効果が薄いということに他ならない。

4.10.3 ε_{MG} の定め方についての結言

ε は問題そのものの解き易さを定める。この値が大きいほど解きやすい問題となる。この値が小さいとある地点で残差が停滞し、そこからなかなか落ちなくなる。

ε_{MG} は収束の安定性を定める。この値が大きいほど安定度が増し、収束しやすくなる。但し、 ε に対して大きすぎる値を取ると、 $A(\varepsilon)$ と $A(\varepsilon_{MG})$ の乖離が大きくなり、修正の効果がなくなる。その結果、安定度はあるものの収束は非常に遅くなる。

$\varepsilon_{MG} - \varepsilon$ は収束の速さを定める。この値が小さいと $A(\varepsilon)$ と $A(\varepsilon_{MG})$ の乖離が小さく、修正が非常に良く効くため収束が速い。但し、 ε_{MG} を ε に近づけすぎると、すなわち $\varepsilon_{MG} - \varepsilon$ の値を小さくしすぎると安定性がなくなり発散する。

結論としては、与えられた ε に対して発散しない範囲でできる限り小さい ε_{MG} を探せば良い。ここまでの考察から、現在は以下の各 ε に対して以下のような ε_{MG} が最適と考えられる。

表 4.11: 各 ε に対する最適な ε_{MG}

ε	ε_{MG}
$1.0 \times 10^1 h^2$	$1.1 \times 10^2 h^2$
$1.0 h^2$	$5.0 \times h^2$
$2.5 \times 10^{-1} h^2$	$5.0 \times 10^{-1} h^2$
$1.0 \times 10^{-1} h^2$	$5.0 \times 10^{-1} h^2$
$1.0 \times 10^{-2} h^2$	$5.0 \times 10^{-1} h^2$
$1.0 \times 10^{-3} h^2$	$5.0 \times 10^{-1} h^2$

4.11 EBEMPPSMG-GMRES法における自由度と反復回数の相関関係

本節ではEBEMPPSMG-GMRES法における自由度と反復回数の相関関係について考えていく。一般にCG法のような反復ソルバーを用いると、反復回数は各方向のメッシュの分割数 n に比例して増えていってしまう。一方MG法が完全に成功すれば、反復回数は表3.1などで見たように n によらず一定、すなわち自由度がどれだけ増えても現在の収束判定条件 1.0×10^{-8} を用いている限り10回程度での収束が見込める。EBEMPPSMG-GMRES法を用いた場合、表4.12のように反復数が約1.3倍ずつ増えていっている。よって現時点では自由度に依存しないソルバーとは言えないが、ILU-GMRES法などに比べて反復数の増加を格段に抑えることができているといえる。

4.12 本章の結言

本章において提案したMG系統の解法であるEBEMPPSMG法は、今後Navier-Stokes問題に拡張した場合に必要とされる安定化パラメータ $\varepsilon = 1.0 \times 10^{-1} h^2$ に対して十分な収束性を示した。またEBEMPPSMG法を前処理として用いたEBEMPPSMG-GMRES法は更に高い収束性を示し、しかもILU-GMRES法などに比べて自由度に依存しづらい。従って、本章の目標であった「定常Stokes方程式において反復数が自由度に大きく依存しない理想的な収束性を達成できるソルバーを開発すること」はEBEMPPSMG-GMRES法の開発により達成できたといえるだろう。

表 4.12: メッシュ分割数 n に対する EBEMPPSMG-GMRES 法の収束性の比較

手法名	ε	ε_{MG}	階層	n	反復回数	計算時間 (秒)
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	6	64	26	2
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	6	128	43	9
EBEMPPSMG-GMRES	$2.5 \times 10^{-1}h^2$	$5.0 \times 10^{-1}h^2$	6	256	47	44
手法名	ε	ε_{MG}	階層	n	反復回数	計算時間 (秒)
ILU-GMRES	$2.5 \times 10^{-1}h^2$	×	×	64	126	1
ILU-GMRES	$2.5 \times 10^{-1}h^2$	×	×	128	290	31
ILU-GMRES	$2.5 \times 10^{-1}h^2$	×	×	256	1135	652

次章では定常 Stokes 問題に移流項が加わった定常 Navier-Stokes 問題に対して同様のソルバーを開発していく。この際、移流項の非線形性および非対称性に十分注意を払う必要があると考えられる。