ISSN 1349-1113
JAXA-RR-07-008

宇宙航空研究開発機構研究開発報告
JAXA Research and Development Report

一般化最小残差（GMRES）法の安定性検証

坂下　雅秀，松尾　裕一，村山　光宏

2008 年 2 月

宇宙航空研究開発機構
Japan Aerospace Exploration Agency

一般化最小残差（GMRES）法の安定性検証*

坂下　雅秀*1,　　松尾　裕一*1,　　村山　光宏*2

The Stability Experiment of the Generalized Minimal Residual(GMRES) Method

Masahide Sakashita*1, Yuichi Matsuo*1 and Mitsuhiro Murayama*2

Abstract

In order to apply the GMRES method, faster and more stable than the LU-SGS method, to the three-dimensional hybrid-unstructured-grid finite-volume method Euler／Navier-Stokes solver JTAS (JAXA Tohoku-university Aerodynamic Simulation code), we developed a GMRES library code. Additionally, we solve some linear systems with simple coefficient matrices to examine stability of the GMRES method. As a result, we confirm the algorithm is more stable than Symmetric Gauss-Seidel method, and the preconditioning works well. We also find that under very ill-condition cases, the GMRES method shows unstable behavior.

Key Word: Krylov subspace method, GMRES, preconditioning, LU-SGS, Symmetric Gauss-Seidel, Navier-Stokes, CFD, approximate factorization

概　要

3 次元ハイブリッド非構造格子有限体積法 Euler／Navier-Stokes ソルバ JTAS（JAXA Tohoku university Aerodynamic Simulation code）に対して，LU-SGS 法より高速で安定な解法である GMRES 法を適用するため, その準備として GMRES 法のライブラリを作成した. そして，簡単な行列を係数行列に持つ連立方程式を解くことにより，その安定性の検証を行った．その結果，Symmetric Gauss-Seidel 法と比較して，より安定な計算が行えることが確認できた. また, 前処理の有効性も確認できた. 一方で, 係数行列の条件が悪い場合には, GMRES 法も不安定になる場合のあることがわかった.

* 平成 19 年 12 月 25 日受付
*1 情報・計算工学センター　計算機運用・利用技術チーム
（JAXA's Engineering Digital Innovetion Center, Computing Resource Management Team）
*2 航空プログラム　国産旅客機チーム
（Aviation Program Group, Civil Transport Team）

1. はじめに

本報告書は，3 次元非構造格子有限体積法Euler／Navier-StokesソルバJTAS（JAXA Tohoku university Aerodynamic Simulation code）で使用されているLU-SGS法に代えて，より高速で安定な連立方程式の解法であるGMRES（Generalized Minimal Residual）法[*1*]を導入し，ソルバの安定化及び高速化をはかる目的で行った作業について報告するものである．今回の作業では，その前段階としてGMRES法のライブラリを作成し，その安定性評価を行った．

現在，宇宙航空研究開発機構(JAXA)では，次世代超音速機技術の基礎研究として小型超音速実験機（NEXST－1）に関するプロジェクトが進められている[*2*][*3*]．このプロジェクトにおいては，複雑な形状の回りにおける剥離や再付着を伴う複雑な流れ場に対する数値流体力学（CFD；Computational Fluid Dynamics）による解析技術が求められている．このような解析には，非構造格子法（Unstructured Grid Method）が良く用いられる．非構造格子法は，

(1) 構造格子に比べて格子生成が比較的容易
(2) 流れ場の重要な場所に格子を細分化して局所的重点的に配置し精度向上を図ることが可能
(3) 最適設計時における形状変化に伴う計算格子の修正が容易

といった特徴を持つ．JAXAにおいては，非構造格子ソルバとして，主にJTASが用いられている[*4*][*5*][*6*]．JTASは，東北大学で開発されたTAS(Tohoku university Aerodynamic Simulation code)[*6*]をもとにJAXAに導入されているCeNSS(Central Numerical Simulation System)と呼ばれる大規模SMP（Symmetric Multiple Processor）クラスタシステム（富士通製PRIMEPOWER HPC2500）に適合するよう若干の変更が加えられたコードであり，オリジナルのTASと区別する意味でJTASと呼ばれている．

現在，JTAS では Euler／Navier－Stokes 方程式を離散化して得られる連立方程式の解法として，LU-SGS 法が用いられている．しかし，近年では LU-SGS 法より優れた連立方程式の解法が考案されており，その成果を反映することにより，より高速で安定に解を得ることが期待できる．そこで，より優れた解法である GMRES 法を JTAS に組み込むこととした．本報告では，GMRES 法の概要を示すとともに，実際に組み込む前に GMRES ライブラリの単体試験として行った安定性の検証結果について報告する．

2. 数値解法

ここでは，GMRES 法が，どのような解法であるかについて説明する．

2.1 Krylov 部分空間法

連立一次方程式の反復解法として最も簡単な方法は，近似解として適当なベクトル $\mathbf{x}_0$ を与え，得られた残差によりこれを修正し新たな近似解 $\mathbf{x}_1$ を計算することを以下同様に繰り返すことであろう．即ち，N 次元の未知ベクトル $\mathbf{x}$，定数ベクトル $\mathbf{b}$ 及び N 行 N 列の係数行列 $\mathbf{A}$ によって与えられる N 元一次連立方程式

$$\mathbf{Ax} = \mathbf{b} \tag{2.1}$$

に対して，反復 k 番目（k=1,2,⋯）の近似解 $\mathbf{x}_k$ 及び残差ベクトル $\mathbf{r}_k$ を，

$$\begin{aligned} \mathbf{x}_k &= \mathbf{x}_{k-1} + \mathbf{r}_{k-1} \\ \mathbf{r}_k &= \mathbf{b} - \mathbf{Ax}_k \end{aligned} \qquad \textit{for } k=1,2,.... \tag{2.2}$$

として与え，適当なベクトル $\mathbf{x}_0$ の元に，残差 $\mathbf{r}_k$ が十分小さくなるまで繰り返す方法である．このような反復解法を Richardson の反復法という．

ところで，このとき得られる近似解 $\mathbf{x}_1$，$\boldsymbol{x}_2$，$\boldsymbol{x}_3$⋯は，それぞれ

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{r}_0 \\ \mathbf{x}_2 &= \mathbf{x}_0 + 2\mathbf{r}_0 + \mathbf{Ar}_0 \\ \mathbf{x}_3 &= \mathbf{x}_0 + 3\mathbf{r}_0 - 3\mathbf{Ar}_0 + \mathbf{A}^2\mathbf{r}_0 \\ &\vdots \end{aligned}$$

であることから，ベクトル $\mathbf{z}_k$ が係数行列 $\mathbf{A}$ のべき乗と残差 $\mathbf{r}_0$ の積（$\mathbf{r}_0$，$\mathbf{Ar}_0$，$\mathbf{A}^2\mathbf{r}_0$，⋯$\mathbf{A}^{k-1}\mathbf{r}_0$）の適当な線形結合によって作られるものすれば，一般に近似解 $\mathbf{x}_k$ を

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{z}_k \tag{2.3}$$

と表すことができる．このとき，ベクトル列 $\mathbf{r}_0$，$\mathbf{Ar}_0$，$\mathbf{A}^2\mathbf{r}_0$，⋯$\mathbf{A}^{k-1}\mathbf{r}_0$ が互いに一次独立であれば，これを k 次元ベクトル空間の基底と考えることができる．この k 次元ベクトル空間は，真の解 $\mathbf{x}$ が存在すると考えられる N 次元ベクトル空間の部分空間となることから，このベクトル空間を Krylov 部分空間（Krylov subspace）という．

ところで，残差 $\mathbf{r}_0$ と係数行列 $\mathbf{A}$ のべき乗が作る基底は，互いに直行しているとは限らないので，Richardson の反復法は仮に収束条件を満足していたとしても，有限回の反復で収束する保障はない．一方で，この基底を基に Gram-Schmidt の直交化等の方法を用いて（正規）直交基底を生成することにより，原理上 N 回の反復で厳密解に収束する一連の「反復」解法を構築することができる．このような Krylov 部分空間上の直交基底を基にした反復解法を Krylov 部分空間法という．

Krylov 部分空間法は，生成された直交基底の基に $\mathbf{z}_k$

を一意に決定する方法により大きく二つに分類することができる．そのひとつは，残差ベクトル $\mathbf{r}_k$ がそれまでに得られた残差ベクトル列 $\mathbf{r}_0$，$\mathbf{r}_1$，…，$\mathbf{r}_{k\text{-}1}$ と直交するように近似解 $\mathbf{x}_k$ を選ぶ方法であり，共役勾配法（Conjugate Gradient Method, CGM）の系列がそれに相当する．残りのもうひとつは，残差ベクトル $\mathbf{r}_k$ が最小になるように近似解 $\mathbf{x}_k$ を選択する方法であり，共役残差法（Conjugate Residual Method, CRM）の系列がそれに相当する．

2.2 Arnoldi 法

Krylov 部分空間において直交基底を生成するためには，Arnoldi 法と呼ばれるアルゴリズムが使用される．Arnoldi 法により m（$m \leqq N$）次元部分空間の正規直交基底を生成する具体的な方法をアルゴリズム 2.1 に示す．

アルゴリズム 2.1 Arnoldi 法

1. *Choose a vector* $\mathbf{v}_1$ *of norm* 1
2. *For* $j = 1,2,....,m$ *Do:*
3. $\mathbf{w}_j \quad = \mathbf{A}\mathbf{v}_j$
4. $h_{i,j} \quad = \left(\mathbf{w}_j, \mathbf{v}_i\right) \quad for\ i = 1,2,..., j$
5. $\mathbf{w}_j \quad = \mathbf{w}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
6. $h_{j+1,j} = \left\| w_j \right\|$
7. $\mathbf{v}_{j+1} \quad = \mathbf{w}_j / h_{j+1,j}$
8. *End Do*

上のアルゴリズムでは．適当な単位ベクトル $\mathbf{v}_1$ から始めて，ステップ 3 で（直交するとは限らない）基底を生成している．ベクトル列 $\mathbf{r}_0$，$\mathbf{A}\mathbf{r}_0$，$\mathbf{A}^2\mathbf{r}_0$，…$\mathbf{A}^{k\text{-}1}\mathbf{r}_0$ は，行列 $\mathbf{A}$ の固有値に収束する可能性があるので，Arnoldi 法では $\mathbf{v}_1$，$\mathbf{A}\mathbf{v}_1$，$\mathbf{A}\mathbf{v}_2$，…$\mathbf{A}\mathbf{v}_{k\text{-}1}$ 等として基底ベクトルを生成している．ステップ 4 及び 5 では，Gram-Schmidt の直交化による直交基底の生成が行われ，ステップ 7 で生成された直交基底の正規化が行われている．このとき，$h_{i,j}$ は一般の行列 $\mathbf{A}$ に対して相似変換して得られる Hessenberg 行列の要素となっている（APPENDIX A 参照）．

ここでは，直交基底の生成に Gram-Schmidt の直交化を用いたが，この方法は計算機による有限桁の計算において誤差が累積し易いことが知られている．そこで，Gram-Schmidt の直交化の計算順序を変更し，すでに一部直交化されたベクトルを用いて順次直交化を繰り返す修正 Gram-Schmidt の直交化†や，Householder 変換を応用した直交化を利用して，直交基底を生成する場合もある．但し，Gram-Schmidt の直交化と比較して，修正 Gram-Schmidt の直交化では並列化に際して各プロセスごとの同期を取る回数が増加し，Householder 変換による直交化では演算量が増加する．

もし，係数行列 $\mathbf{A}$ が対称行列であった場合には，相似変換によって得られる Hessenberg 行列も対称行列でなければならない．対称な Hessenberg 行列は三重対角行列であるから，

$$
\begin{aligned}
h_{i,j} \quad &= 0, \qquad for\ \ 1 \le i < j-1 \\
h_{j,j+1} &= h_{j+1,j} \quad\ j = 1,2,...,m
\end{aligned}
\tag{2.4}
$$

となる．このことにより Arnoldi 法はより簡略化され，行列 $\mathbf{A}$ が対称行列である場合に適用可能な Lanczos（ランチョス）法が得られる．$\alpha_j = h_{j,j}$，$\beta_j = h_{j+1,j} = h_{j,j+1}$ と置いた場合の具体例をアルゴリズム 2.2 に示す．

アルゴリズム 2.2 Lanczos 法

1. *Choose a vector* $\mathbf{v}_1$ *of norm* 1. *Set* $\beta_1 = 0, \mathbf{v}_0 = 0$
2. *For* $j = 1,2,....,m$ *Do:*
3. $\mathbf{w}_j \quad = \mathbf{A}\mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$
4. $\alpha_j \quad = \left(\mathbf{w}_j, \mathbf{v}_j\right)$
5. $\mathbf{w}_j \quad = \mathbf{w}_j - \alpha_j \mathbf{v}_j$
6. $\beta_{j+1} = \left\| w_j \right\|$
7. $\mathbf{v}_{j+1} = \mathbf{w}_j / \beta_{j+1}$
8. *End Do*

2.3 GMRES 法

係数行列 $\mathbf{A}$ が対称でない場合に適用可能な Krylov 部分空間法のひとつに GMRES（Generalized Minimal RESidual）法がある．この解法は，その名前が示す通り残差ベクトル $\mathbf{r}_k$ が最小になるように近似解 $\mathbf{x}_k$ を選択する方法である．いま，Arnoldi 法により得られた正規直交基底を $\mathbf{v}_1$，…，$\mathbf{v}_m$ とし，線形結合の定数を y_1，…，y_m とすれば，（2.3）式は，

$$
\begin{aligned}
\mathbf{x}_m &= \mathbf{x}_0 + \sum_{i=1}^{m} \mathbf{v}_i y_i \\
&= \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m
\end{aligned}
\tag{2.5}
$$

と書くことが出来る．ここで，$\mathbf{V}_m$ は N 行 m 列の行列 $\mathbf{V}_m = [\mathbf{v}_0 \quad \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_m]$（Arnoldi 法によって生成された正規直交基底 $\mathbf{v}_i$ を列ベクトルとして持つ行列）であり，$\mathbf{y}_m$ は m 次元ベクトル $\mathbf{y}_m = [y_0 \quad y_1 \quad \cdots \quad y_m]^{\mathrm{T}}$ である．GMRES 法とは，この線形結合定数ベクトル $\mathbf{y}_m$ を求めるアルゴリズムのことである．即ち，GMRES 法は，m 次元ベクトル $\mathbf{y}_m$ の関数 $J(\mathbf{y}_m)$ を

$$
\begin{aligned}
J(\mathbf{y}_m) &= \left\| \mathbf{b} - \mathbf{A}\mathbf{x}_m \right\| \\
&= \left\| \mathbf{b} - \mathbf{A}\left(\mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m\right) \right\|
\end{aligned}
\tag{2.6}
$$

としたときに，$J(\mathbf{y}_m)$ を最小にするベクトル $\mathbf{y}_m$ を求める

† Gram-Schmidt では，最初に直交化される全てのベクトルに対して非直交成分を計算し，最後に直交ベクトルを得る．一方，修正 Gram-Schmidt では，順次直交化を繰り返す．

方法を与える．

GMRES 法により近似解 $\mathbf{x}_m$ を求める方法の概略は以下の通りである．まず，最初のベクトル $\mathbf{v}_1$ として初期残差ベクトル $\mathbf{r}_0$ を正規化したもの（$\mathbf{v}_1$=$\mathbf{r}_0$／‖$\mathbf{r}_0$‖）を選び Arnordi 法（アルゴリズム 2.1）により正規直交化基底及び Hesssenberg 行列の要素を生成する．次に，関数 $J(\mathbf{y}_m)$を最小にするベクトル $\mathbf{y}_m$ を求める．(2.6)式は，

$$J(\mathbf{y}_m) = \left\| \beta\, \mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}_m \right\| \tag{2.7}$$

と変形できる（APPENDIX B 参照）．ただし，β は初期残差ベクトル $\mathbf{r}_0$ の大きさ（β=‖$\mathbf{r}_0$‖）であり，$\mathbf{e}_1$ は最初の成分のみ 1 である m+1 次元の単位ベクトル（$\mathbf{e}_1=[1\ 0\ \cdots\ 0]^{\mathrm{T}}$）である．また，$\widetilde{\mathbf{H}}_m$ は APPENDIX A の(A.5)式に示す Hessenberg 行列の最後に一行を加えた m+1 行 m 列の行列である．この(2.7)式を最小化する問題は，Givens 回転を用いた QR 分解法により行列$\widetilde{\mathbf{H}}_m$ を上三角行列に変換することにより解くことができ（APPENDIX C 参照），ベクトル $\mathbf{y}_m$ が求まる．このとき同時に，近似解 $\mathbf{x}_m$ を求めることなく収束判定をすることが可能である．近似解 $\mathbf{x}_m$ は，収束した後に(2.5)式より一回のみ求めればよい．

以上の手順をまとめてアルゴリズム 2.3 として示す．これが基本となる GMRES 法のアルゴリズムである．

アルゴリズム 2.3　GMRES 法

1. *Choose* $\mathbf{x}_0$ *and*
   compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|$ *and* $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$
2. *Set* $g_1 = \beta$
3. *For* $j = 1,2,....,n$ *Do:* (GMRES iteration loop)
4. $\mathbf{w}_j = \mathbf{A}\mathbf{v}_j$
5. $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$ *for* $i = 1,2,...,j$
6. $\mathbf{w}_j = \mathbf{w}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
7. $h_{j+1,j} = \| w_j \|$
8. *if* $h_{j+1,j} = 0$ *then set* $m = j$ *and Go To step* 26
9. $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
10. *Set* $r_{1,j}^{(0)} = h_{1,j}$
11. *For* $i = 1,2,....,j-1$ *Do:*
12. $tmp1 = c_i r_{i,j}^{(i-1)} + s_i h_{i+1,j}$
13. $tmp2 = -s_i r_{i,j}^{(i-1)} + c_i h_{i+1,j}$
14. $r_{i,j}^{(i)} = tmp1$
15. $r_{i+1,j}^{(i)} = tmp2$
16. *End Do*
17. $c_j = r_{j,j}^{(j-1)} \Big/ \sqrt{\left(r_{j,j}^{(j-1)}\right)^2 + h_{j+1,j}^2}$
18. $s_j = h_{j+1,j} \Big/ \sqrt{\left(r_{j,j}^{(j-1)}\right)^2 + h_{j+1,j}^2}$
19. $g_{j+1} = -s_j g_j$
20. $g_j = c_j g_j$
21. $r_{j,j}^{(j)} = c_j r_{j,j}^{(j-1)} + s_j h_{j+1,j}$
22. $\left(r_{j+1,j}^{(j)} = 0 \right)$
23. *if* $|s_j g_j| < \varepsilon$ *then set* $m = j$ *and Go To step* 26
24. *End Do*
25. *Set* $m = n$
26. *For* $j = m, m-1,....,1$ *Do:*
27. $y_j = g_j$
28. *For* $i = j+1, j+2,....,m$ *Do:*
29. $y_j = y_j - r_{j,i}^{(j)} y_i$
30. *End Do*
31. $y_j = y_j / r_{j,j}^{(j)}$
32. *End Do*
33. *Compute* $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m$

2.4　GMRES 法の大規模連立方程式への適用

「2.3 GMRES 法」で示したアルゴリズムを大規模な連立一次方程式の解法として実際に適用しようとすると，直交基底 $\mathbf{v}_i$ 及び Hessenberg 行列を保存する必要があるためメモリ不足により破綻してしまう．これを回避する方法の主なものとして，リスタート GMRES（Restarted GMRES）法と不完全 GMRES（Truncated GMRES）法がある．

2.4.1　リスタート GMRES 法

リスタート GMRES 法では，アルゴリズム 2.3 ステップ 3 のループを連立方程式の元数 N より小さい値 m（例えば 10，20 等）で実行し，得られた近似解 $\mathbf{x}_m$ をあらたな初期ベクトル $\mathbf{x}_0$ として再び同じ計算を繰り返す．このようなリスタート GMRES 法は，一般に GMRES(m)法と表記される．このとき，いくつかの m 次元のベクトル空間が生成される．もし，連立方程式の真の解 $\mathbf{x}$ が存在する N 次元ベクトル空間（の部分空間）がこれらのベクトル空間の直積で表されるならば，GMRES(m)法が GMRES 法に一致することは明らかである．ただし，直積で表される保障はない．

2.4.2　不完全 GMRES 法

Arnoldi 法は，既に得られた全ての基底ベクトルに直交する基底を順次生成する．この過程を，高々直前 k-1 本の基底ベクトルにのみ直交する基底を生成する，不完全直交化過程（Incomplete Orthogonalization process）で置き換えることにより，必要なメモリ量を節約すると同時に演算量を削減することが可能である．アルゴリズ

ム 2.4 に，不完全直交化過程を示す．

アルゴリズム 2.4　不完全直交化過程

1. $For\ j = 1,2,\cdots,m\ Do:$
2. $\mathbf{w}_j = \mathbf{A}\mathbf{v}_j$
3. $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i) \quad for\ i = \max\{1, j-k+1\},\cdots,j$
4. $\mathbf{w}_j = \mathbf{w}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
5. $h_{j+1,j} = \| w_j \|$
6. $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
7. $End\ Do$

この不完全直交化過程により，アルゴリズム 2.3 のステップ 3 から 8 を単純に置き換えた GMRES 法を QGMRES（Quasi-GMRES）法という．

不完全直交過程によって得られる Hessenberg 行列 $\mathbf{H}_m$ は帯行列となる．このことを利用すると，近似解 $\mathbf{x}_m$ を行列の積を含む(2.5)式ではなく，行列の積を含まない漸化式

$$\mathbf{x}_m = \mathbf{x}_{m-1} + g_m^{(m)}\mathbf{p}_m \tag{2.8}$$

によって計算することが可能である（APPENDIX D 参照）．ただし，$\mathbf{p}_m$ は m 行 m 列の行列 $\mathbf{P}_m=[\mathbf{p}_1,\cdots,\mathbf{p}_m]$を

$$\mathbf{P}_m = \mathbf{V}_m \tilde{\mathbf{R}}_m^{-1}$$

で定義したときの，最後（m 番目）の列ベクトルである．また，スカラ $g_m^{(m)}$及び行列$\tilde{\mathbf{R}}_m$については，APPENDIX C の(C.4)式及び(C.16)式を参照されたい．QGMRES 法に対して，(2.8)式を適用することにより，アルゴリズム 2.5 に示す DQGMRES（Direct Quasi-GMRES）法が得られる．以下では，DQGMRES 法のことを不完全（Truncated）GMRES 法という．

アルゴリズム 2.5　DQGMRES 法

1. $compute\ \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0,\ \beta = \|\mathbf{r}_0\|\ and\ \mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$
2. $For\ j = 1,2,....,until\ convergence\ Do:$
3. $\mathbf{w}_j = \mathbf{A}\mathbf{v}_j$
4. $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i) \quad for\ i = \max\{1, j-k+1\},...,j$
5. $Compute\ c_i, s_i, g_j^{(j)}, r_{i,j}^{(i)}\ and\ so\ on\ (i = j-k,\cdots)$
6. $\mathbf{p}_j = \frac{1}{r_{j,j}^{(j)}}\left(\mathbf{v}_j - \sum_{i=j-k}^{j-1} r_{i,j}^{(i)}\mathbf{p}_i\right)$
7. $\mathbf{x}_j = \mathbf{x}_{j-1} + g_j^{(j)}\mathbf{p}_j$
8. $if\ \left|g_j^{(j)}\right| < \varepsilon\ then\ Stop$
9. $End\ Do$

ところで，不完全 GMRES 法では，全ての基底ベクトル $\mathbf{v}_i$ が直交する訳ではないので，この基底で展開された近似解は，もはや残差 ‖ $\mathbf{b}-\mathbf{A}\mathbf{x}_m$ ‖ を最小にする訳ではない．しかし，近似的に最小化することは期待できる．

また，残差を評価する(C.21)式は，その導出過程で基底ベクトル $\mathbf{v}_i$ の直交性を使用しているので成立しない．代わりに，

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_m\|_2 \leqq \sqrt{m-k+1}\left|g_{m+1}^{(m)}\right| \tag{2.9}$$

なる不等式が成立することが知られている．ここで，$m \leqq k$ のとき，k は m で置き換えられる．この不等式は一般に過大評価であり，実際は残差の良い近似になっていると言われてはいるものの，｜$g_{m+1}^{(m)}$｜によって近似解の残差を評価すると，収束していないにもかかわらず収束したものとみなしてしまう可能性がある．

より厳密に残差を評価するには，漸化式

$$\mathbf{z}_{m+1} = -s_m\mathbf{z}_m + c_m\mathbf{v}_{m+1} \tag{2.10}$$

によって与えられるベクトル $\mathbf{z}_{m+1}$ のノルムを評価すればよい．ただし，$\mathbf{z}_1=\mathbf{v}_1$ である．この場合，ベクトル $\mathbf{z}_i$ を保存するための余分なメモリと，1 ステップ当たり $3N$ 回の演算を必要とする，また，ノルムを計算するには $2N$ 回の演算が必要である．

そこで，スカラ量に関する漸化式

$$\varsigma_{m+1} = |s_m|\varsigma_m + |c_m| \tag{2.11}$$

を使って，｜ζ_{m+1}｜で残差を評価する方法もある．ただし，ζ_1=‖ $\mathbf{v}_1$ ‖ である．この方法は，厳密な評価を行うものではないが，(2.9)式よりは精確な評価が行える．これら，不完全 GMRES 法における残差の評価方法の詳細については，Saad(2000)[7]を見よ．

2.5　前処理付き GMRES 法

いま，適当な行列 $\mathbf{M}$ が与えられたとして，その逆行列 $\mathbf{M}^{-1}$ を(2.1)式の両辺に左からかけて得られる方程式

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \tag{2.12}$$

を考える．(2.12)式を解いて得られる解 $\mathbf{x}$ は，(2.1)式を解いて得られるそれと同一であるから，(2.1)式を解くかわりに(2.12)式を解いてもよい．もし，$\mathbf{M}$=$\mathbf{A}$ であり $\mathbf{M}^{-1}$ が具体的に求められるならば，(2.12)式は解けたことになる．一般に $\mathbf{M}^{-1}$ を厳密に求めることは非常に困難であるが，行列 $\mathbf{M}$ として係数行列 $\mathbf{A}$ の疎性を考慮して fill in を無視した不完全 LU 分解を行ったものを選んだり，近似的に係数行列 $\mathbf{A}$ の逆行列を求めて $\mathbf{M}^{-1}$ とするなどの様々な方法により，(2.12)式に対して Krylov 部分空間法を適用した方が，(2.1)式に対して適用する場合に比べて，より安定で早い収束を期待できる場合がある．(2.1)式を変形して(2.12)式を得る処理を前処理（preconditioning）と言い，行列 $\mathbf{M}$ を前処理行列（preconditioner）と言う．

(2.12)式による前処理の場合，前処理行列を左から作用させているので，特に左前処理（left preconditioning）とも言う．左前処理付き GMRES（left preconditioned GMRES）法のアルゴリズムを得るには，(2.12)式に素直に GMRES 法を適用すればよい．これにより，左前処理付き GMRES 法としてアルゴリズム 2.6 を得る．

アルゴリズム 2.6　左前処理付き GMRES 法

1. *Choose* $\mathbf{x}_0$ *and compute* $\mathbf{r}_0 = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$, $\beta = \|\mathbf{r}_0\|$ *and* $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$
2. *For* $j = 1,2,....,m$ *Do:* (GMRES iteration loop)
3. $\mathbf{w}_j = \mathbf{M}^{-1}\mathbf{A}\mathbf{v}_j$
4. $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$ *for* $i = 1,2,..., j$
5. $\mathbf{w}_j = \mathbf{w}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
6. $h_{j+1,j} = \| w_j \|$
7. $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
8. *End Do*
9. *Compute* $\mathbf{y}_m$ *the minimizer of* $\|\beta \mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}\|$ *and* $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m$
10. *If converged stop, else set* $\mathbf{x}_0 = \mathbf{x}_m$ *and Go To* 1

アルゴリズム 2.6 において，ステップ 3 の計算は前処理行列としてどのようなものを選ぶかによって，連立方程式

$$\mathbf{M}\mathbf{w}_j = \widetilde{\mathbf{v}}_j, \quad \widetilde{\mathbf{v}}_j = \mathbf{A}\mathbf{v}_j \tag{2.13}$$

を解くことを意味する場合がある．このとき，初期残差 $\mathbf{r}_0$ を計算するには，連立方程式

$$\mathbf{M}\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \tag{2.14}$$

を解くことになる．

前処理の方法としては，(2.12)式のように前処理行列を左から作用させる左前処理の他に，前処理行列 $\mathbf{M}$ を不完全 LU 分解して係数行列 $\mathbf{A}$ の両側から作用させる方法と係数行列の右側から作用させる方法がある．これらの方法はそれぞれ，分離前処理（split preconditioning）及び右前処理（right preconditioning）と呼ばれる．分離前処理は，前処理行列を $\mathbf{M}$=$\mathbf{LU}$ と分解して，

$$\begin{aligned} \mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}\mathbf{u} &= \mathbf{L}^{-1}\mathbf{b} \\ \mathbf{x} &= \mathbf{U}^{-1}\mathbf{u} \end{aligned} \tag{2.15}$$

を解く．ただし，$\mathbf{L}$ は下三角行列であり，$\mathbf{U}$ は上三角行列である．また，右前処理は，

$$\begin{aligned} \mathbf{A}\mathbf{M}^{-1}\mathbf{u} &= \mathbf{b} \\ \mathbf{u} &= \mathbf{M}\mathbf{x} \end{aligned} \tag{2.16}$$

を解く．ただし，以下に示されるされるように，右前処理付き GMRES 法は, 新しいベクトル $\mathbf{u}$ 及びその初期値 $\mathbf{u}_0$ 使わずに構成することが可能である．まず，前処理系における初期残差 $\mathbf{r}_0$=$\mathbf{b}-\mathbf{A}\mathbf{M}^{-1}\mathbf{u}_0$ は，それに等しい $\mathbf{b}-\mathbf{A}\mathbf{x}_0$ によって計算可能である．その後全ての Krylov 部分空間上のベクトルを求める際に，変数 $\mathbf{u}$ が参照されることはない．最後に，(2.15)式の近似解は,

$$\mathbf{u}_m = \mathbf{u}_0 + \sum_{i=1}^{m} \mathbf{v}_i y_i \tag{2.17}$$

によって与えられる．ここに，$\mathbf{u}_0$=$\mathbf{M}\mathbf{x}_0$ である．これに $\mathbf{M}^{-1}$ を乗ずれば，求めたい $\mathbf{x}$ についての近似解を得ることができる．

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{M}^{-1}\left[\sum_{i=1}^{m} \mathbf{v}_i y_i\right] \tag{2.18}$$

即ち，右前処理では，前処理行列（の逆行列）の乗算は最後に解を求めるために必要となる．このことは，左前処理の場合において初期残差を求めるために最初に必要だったのとは対照的である．右前処理付き GMRES 法はアルゴリズム 2.7 のようになる．

アルゴリズム 2.7　右前処理付き GMRES 法

1. *Choose* $\mathbf{x}_0$ *and compute* $\mathbf{r}_0 = (\mathbf{b} - \mathbf{A}\mathbf{x}_0)$, $\beta = \|\mathbf{r}_0\|$ *and* $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$
2. *For* $j = 1,2,....,m$ *Do:* (GMRES iteration loop)
3. $\mathbf{w}_j = \mathbf{A}\mathbf{M}^{-1}\mathbf{v}_j$
4. $h_{i,j} = (\mathbf{w}_j, \mathbf{v}_i)$ *for* $i = 1,2,..., j$
5. $\mathbf{w}_j = \mathbf{w}_j - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
6. $h_{j+1,j} = \| w_j \|$
7. $\mathbf{v}_{j+1} = \mathbf{w}_j / h_{j+1,j}$
8. *End Do*
9. *Compute* $\mathbf{y}_m$ *the minimizer of* $\|\beta \mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}\|$ *and* $\mathbf{x}_m = \mathbf{x}_0 + \mathbf{M}^{-1}\mathbf{V}_m \mathbf{y}_m$
10. *If converged stop, else set* $\mathbf{x}_0 = \mathbf{x}_m$ *and Go To* 1

アルゴリズム 2.6 の場合と同様，アルゴリズム 2.7 においても，ステップ 3 の計算は前処理行列としてどのようなものを選ぶかによって，連立方程式

$$\mathbf{M}\widetilde{\mathbf{w}}_j = \mathbf{v}_j, \quad \mathbf{w}_j = \mathbf{A}\widetilde{\mathbf{w}}_j \tag{2.19}$$

を解くことを意味することがある．この場合，近似解 $\mathbf{x}_m$ を求めるには，$\mathbf{s}$=$\mathbf{M}^{-1}\mathbf{V}_m\mathbf{y}_m$ とおき，連立方程式

$$\mathbf{M}\mathbf{s} = \mathbf{V}_m\mathbf{y}_m \tag{2.20}$$

を解いた後，$\mathbf{x}_m$=$\mathbf{x}_0$+$\mathbf{s}$ を計算することになる．

行列 $\mathbf{A}$ が対称の場合は，前処理付き共役勾配法が適用可能である．このとき，内積の定義に前処理行列 $\mathbf{M}$ を介在させることにより，前処理行列 $\mathbf{M}$ の逆行列と係数行列 $\mathbf{A}$ の積 $\mathbf{M}^{-1}\mathbf{A}$ について，対称性を保存することができる．そして，Cholesky 分解された前処理行列 $\mathbf{M}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$ の場合には，左前処理付き共役勾配法と分離前処理付き共役勾配法は理論上一致する．同様にして，係数行列 $\mathbf{A}$ が対称行列に近い場合には，その対称性に近い性質を保持できる分離前処理付き GMRES 法を構成することが可能であり，因数分解された前処理行列を使用するのが最も適当であると考えられる．

$\mathbf{A}$ が一般の行列の場合，前処理行列 $\mathbf{M}$ が悪条件を持つような特別な場合を除けば，三つの前処理の間に一般的な相違は，ほとんど見られない．左前処理付き GMRES 法において生成される基底が張るベクトル空間と，右前処理のそれは一致している．ただし，右前処理の場合には，元々の方程式における残差ノルム $\|\mathbf{b}-\mathbf{A}\mathbf{x}_m\|$ そのものを最小にするのに対して，左前処理の場合には，前処理された残差ノルム $\|\mathbf{M}^{-1}(\mathbf{b}-\mathbf{A}\mathbf{x}_m)\|$ を最小にするという違いがある．この違いは，左前処理（あるいは分離前処理）において収束判定を行う場合に注意が必要であることを意味する．

2.6　LU−SGS 法

解析対象となる流れ場は近似的に定常であるという仮定の下に，Navier-Stokes 方程式の定常解を求めることを考える．三次元非定常圧縮性 Navier－Stokes 方程式の積分形表示は，

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV+\int_{\partial\Omega}\left[\mathbf{F}(\mathbf{Q})-\mathbf{G}(\mathbf{Q})\right]d\mathbf{S}=0 \qquad (2.21)$$

で与えられる．ここに，$\mathbf{Q}=[\rho,\rho u,\rho v,\rho w,e]^{\mathrm{T}}$は保存量（Conservative variables）ベクトルであり，ρは密度，$\mathbf{u}=[u,v,w]^{\mathrm{T}}$ は流速，eはエネルギーである．また，$\mathbf{F}(\mathbf{Q})$及び$\mathbf{G}(\mathbf{Q})$はそれぞれ非粘性流束ベクトル及び粘性流束ベクトルである．定常状態では，

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV=0 \qquad (2.22)$$

であり，したがって，

$$\int_{\partial\Omega}\left[\mathbf{F}(\mathbf{Q})-\mathbf{G}(\mathbf{Q})\right]d\mathbf{S}=0 \qquad (2.23)$$

を満たす保存量ベクトル $\mathbf{Q}$ を求めればよい．しかし，(2.23)式を解くことは困難であることが多い．そのため，(2.21)式を時間発展させて行き，十分な時間経過の後に漸近的に(2.22)式が成立した時の解を定常解とみなすこととする．

(2.21)式は，各物理量の無次元化を行った後，これを数値的に計算するために離散化される．

まず，空間方向について，ここではセル節点有限体積法によって非構造格子系に離散化する．有限体積法における検査体積（Control Volume）は，図 2.1 に示したように，各格子節点周りに要素の中心A，要素表面の中心B，D及び要素を構成する辺の中点Cを結んで出来る面を境界とする多面体として定義される．このような検査体積の定義方法は，非重合二重格子（non-overlapping dual cell）と呼ばれる．

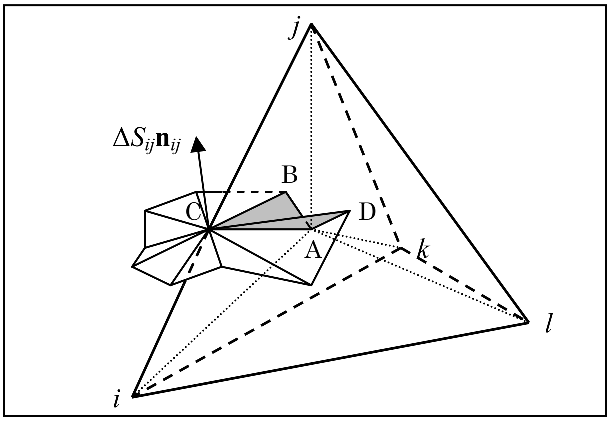

図 2.1　三角錐要素における検査体積境界面

この空間離散化によって(2.24)式が得られる．

$$V_i\frac{\partial\mathbf{Q}_i}{\partial t}=-\sum_{j(i)}\left[\mathbf{F}(\mathbf{Q})_{ij}^{n+1}-\frac{1}{\mathrm{Re}}\mathbf{G}(\mathbf{Q})_{ij}^{n+1}\right]\Delta S_{ij}\mathbf{n}_{ij} \qquad (2.24)$$

ここに，V_iは節点iのまわりの検査体積であり，ΔS_{ij}と$\mathbf{n}_{ij}$はそれぞれ節点iとそれに隣接する節点jとの間の検査体積表面の面積及びその単位法線ベクトル（点iから見て外向きが正）である．また，$\mathbf{\Sigma}_{j(i)}$は節点iの周りの多面体検査体積において，それを構成する全ての面についての総和を取ることを意味する．Reはレイノルズ数である．

次に，(2.24)式について時間方向の離散化を行う．ここでは，Euler陰解法を適用することにより，

$$\begin{aligned}V_i\frac{\Delta\mathbf{Q}_i^n}{\Delta t}&=\mathbf{R}_i^{n+1}\\ \mathbf{R}_i^{n+1}&=-\sum_{j(i)}\left[\mathbf{f}(\mathbf{Q},\mathbf{n})_{ij}^{n+1}-\mathbf{g}(\mathbf{Q},\mathbf{n})_{ij}^{n+1}\right]\Delta S_{ij}\end{aligned} \qquad (2.25)$$

を得る．ただし，$\Delta\mathbf{Q}_i^{\mathrm{n}}\equiv\mathbf{Q}_i^{\mathrm{n+1}}-\mathbf{Q}_i^{\mathrm{n}}$，$\mathbf{f}(\mathbf{Q},\mathbf{n})\equiv\mathbf{F}(\mathbf{Q})\mathbf{n}$及び$\mathbf{g}(\mathbf{Q},\mathbf{n})\equiv(1/\mathrm{Re})\mathbf{G}(\mathbf{Q})\mathbf{n}$とおいた．また，$\Delta t$は時間刻み幅を，上付き添え字$n$は時間ステップを表す．(2.25)式は

$$\mathbf{R}_i^{n+1}=\mathbf{R}_i^n+\frac{\partial\mathbf{R}_i^n}{\partial\mathbf{Q}_i}\Delta\mathbf{Q}_i \qquad (2.26)$$

とおくことにより線形化することができる．全ての節点についての方程式をまとめて書けば，

$$\mathbf{A}\Delta\mathbf{Q}^n=\mathbf{R}^n \qquad (2.27)$$

を得る．ただし，

$$\mathbf{A}=\frac{\Delta \mathbf{V}}{\Delta t}\mathbf{I}-\frac{\partial \mathbf{R}^n}{\partial \mathbf{Q}} \tag{2.28}$$

$$\mathbf{R}=-\sum_{j(i)}\left[\mathbf{f}(\mathbf{Q},\mathbf{n})_{ij}^n-\mathbf{g}(\mathbf{Q},\mathbf{n})_{ij}^n\right]\Delta S_{ij} \tag{2.29}$$

であり，$\partial\mathbf{R}／\partial\mathbf{Q}$はヤコビ行列である．このとき，(2.27)式は，$\Delta t\rightarrow\infty$の極限においてNewton法†による求解に一致し，良く知られているように2次の収束性を示す．

ところで，(2.29)式の$\mathbf{R}_i^n$が(2.23)式を差分化して得られる式そのものであることに注意されたい．したがって，(2.27)式の時間ステップを進め$\Delta\mathbf{Q}$=[$\Delta\mathbf{Q}_i$]が0に収束すれば，$\mathbf{R}_i^n$=0となり求める定常解が得られたことになる．このことから，$\mathbf{R}_i^n$は残差と呼ばれる．また，(2.27)式の右辺（RHS)の評価は十分厳密に行う必要があるものの，左辺（LHS）の評価は収束を疎外しない程度の粗いもので良いことがわかる．JTASにおいては，数値流束の評価をHLLEW法‡により行うと同時に高次精度差分を用いることにより右辺の評価の高精度化が計られている．一方で，左辺に含まれるヤコビ行列の評価は，簡便な方法によって行うことで，行列$\mathbf{A}$の評価を効率化することが一般的である．このとき，$\partial\mathbf{R}／\partial\mathbf{Q}$は残差$\mathbf{R}_i^n$の厳密なヤコビ行列とはならないので，2次の収束性は失われるが，その代わりに，行列$\mathbf{A}$の評価に要する時間を短縮することができる．

いま，ヤコビ行列を評価するために使用する数値流束ベクトルを

$$\mathbf{f}(\mathbf{Q},\mathbf{n})_{ij}=\frac{1}{2}\Big[\mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})+\mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij}) \\ -\left(\left|\lambda_{ji}\right|\mathbf{Q}_j-\left|\lambda_{ij}\right|\mathbf{Q}_i\right)\Big] \tag{2.30}$$

によって定義する．これは，Roeの流束関数（Roe's flux function）においてRoe行列をそのスペクトル半径で置き換えたものである．このとき，両辺を保存量ベクトルによって偏微分することにより，数値流束ベクトルのヤコビ行列の要素

$$\frac{\partial\mathbf{f}(\mathbf{Q},\mathbf{n})_{ij}}{\partial\mathbf{Q}_i}=\frac{\partial\mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial\mathbf{Q}_i}+\left|\lambda_{ij}\right|\mathbf{I} \tag{2.31}$$

及び

$$\frac{\partial\mathbf{f}(\mathbf{Q},\mathbf{n})_{ij}}{\partial\mathbf{Q}_j}=\frac{\partial\mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial\mathbf{Q}_j}-\left|\lambda_{ji}\right|\mathbf{I} \tag{2.32}$$

が得られる．(2.30)及び(2.31)式において，｜λ_{ij}｜はヤコビ行列$\mathbf{J}\equiv\partial\mathbf{f}/\partial\mathbf{Q}$のスペクトル半径$|U|+a$に，粘性ベクトルのヤコビ行列を無視する代わりの効果を付け加えたもの，即ち，

$$\left|\lambda_{ij}\right|=\chi\left(\left|U_i\right|+a\right)+2\frac{\mu_i}{\mathrm{Re}\,\rho_i\left|\mathbf{x}_j-\mathbf{x}_i\right|} \tag{2.33}$$

である．ただし，Uは検査体積表面法線方向速度であり，aは音速，μは粘性係数，χは経験定数1.01である．そして，$\mathbf{x}_i$及び$\mathbf{x}_j$はそれぞれ節点i及びjの位置ベクトルである．ここで，(2.31)及び(2.32)式より，この流束ベクトルを用いてヤコビ行列を評価する方法が，JamesonとTurkelによって提案されたヤコビ行列$\mathbf{J}$の分割方法[9]

$$\mathbf{J}^{\pm}=\mathbf{J}\pm|\lambda|\mathbf{I} \tag{2.34}$$

に従っていることが判る．粘性項については，流束ベクトルのスペクトル半径で補正したので残差$\mathbf{R}$のヤコビ行列では無視する．以上によりヤコビ行列の要素，

$$\frac{\partial\mathbf{R}(\mathbf{Q}_i)}{\partial\mathbf{Q}_i}=\frac{1}{2}\sum_{j(i)}\left[\frac{\partial\mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial\mathbf{Q}_i}+\left|\lambda_{ij}\right|\mathbf{I}\right]\Delta S_{ij} \tag{2.35}$$

及び

$$\frac{\partial\mathbf{R}(\mathbf{Q}_i)}{\partial\mathbf{Q}_j}=\frac{1}{2}\left[\frac{\partial\mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial\mathbf{Q}_j}-\left|\lambda_{ji}\right|\mathbf{I}\right]\Delta S_{ij} \tag{2.36}$$

が得られる．このとき，(2.36)式はヤコビ行列$\partial\mathbf{R}／\partial\mathbf{Q}$の$i$行$j$列目のブロック要素であると同時に，流束ベクトルの対称性から，j行j列目のブロック対角項の一部である．同様に，(2.35)式は，$\partial\mathbf{R}／\partial\mathbf{Q}$の$i$行$i$列目のブロック対角項であると同時に，$\Sigma_{j(i)}$[・]内の各項は，符号を変えて$j$行$i$列目のブロック要素となる．ところで，面$ij$に接するふたつの検査体積$i$及び$j$において，いずれか大きい節点番号を持つ検査体積を$j$とし，他方を$i$とすることが必ずできる．即ち，$j>i$とすることができる．すると$i$行$j$列目のブロック要素は，必ずヤコビ行列$\partial\mathbf{R}／\partial\mathbf{Q}$の上三角要素となり，$j$行$i$列目のブロック要素は同じく下三角要素となる（図2.2参照）．また，面ijはそれが唯一含む辺ijによって指定できる．したがって，係数行列$\mathbf{A}$を上三角行列$\mathbf{U}$，下三角行列$\mathbf{L}$及び対角行列$\mathbf{D}$の和

$$\mathbf{A}=\mathbf{D}+\mathbf{L}+\mathbf{U} \tag{2.37}$$

で表した場合に，そのゼロでない各ブロック要素は，例えばアルゴリズム2.8のように全ての辺ijについて計算することにより求められる．

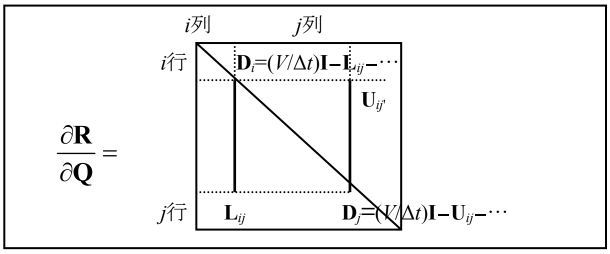

図2.2　ヤコビ行列の対角項と非対角項の関係

† xの関数$f(x)$において$f(x)$=0を満たすxを求める反復解法．反復n回目の近似解をx_nとしたときに，次の近似解x_{n+1}は，$x_{n+1}=x_n-f(x_n)／f'(x_n)$．ただし，$f'(x)$は$f(x)$の導関数．

‡ Harten-Lax-Van Lee-Einfeldt-Wada法[8]

アルゴリズム 2.8　ヤコビ行列のブロック要素の計算

1. $set\ \mathbf{D}_i = \frac{V_i}{\Delta t}\mathbf{I}$
2. $For\ ij = 1,2,....,all\ of\ edges\ Do:$
3. $Compute\ \mathbf{U}_{ij} = \left[\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial \mathbf{Q}_j} - |\lambda_{ji}|\mathbf{I}\right]\Delta S_{ij}$
4. $Compute\ \mathbf{L}_{ij} = \left[-\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} - |\lambda_{ij}|\mathbf{I}\right]\Delta S_{ij}$
5. $\mathbf{D}_i = \mathbf{D}_i - (-\mathbf{L}_{ij})$
6. $\mathbf{D}_j = \mathbf{D}_j - (-\mathbf{U}_{ij})$
7. $End\ Do$

アルゴリズム 2.8 よりゼロでない上三角要素及び下三角要素を保存するためには，それぞれ辺の数だけのブロック要素を保存する領域があればよいことが判る．また，これを書き下せば(2.38)式を得る．

$$
\begin{aligned}
\mathbf{U}_{ji} &= \left[\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial \mathbf{Q}_j} - |\lambda_{ji}|\mathbf{I}\right]\Delta S_{ij} \\
&= -\left[\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ji})}{\partial \mathbf{Q}_j} + |\lambda_{ji}|\mathbf{I}\right]\Delta S_{ij} \\
\mathbf{L}_{ij} &= \left[\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ji})}{\partial \mathbf{Q}_i} - |\lambda_{ij}|\mathbf{I}\right]\Delta S_{ij} \\
&= -\left[\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} + |\lambda_{ij}|\mathbf{I}\right]\Delta S_{ij} \\
\mathbf{D}_i &= \frac{V_i}{\Delta t}\mathbf{I} - \sum_{j(i)}\left[\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} + |\lambda_{ij}|\mathbf{I}\right]\Delta S_{ij}
\end{aligned}
\tag{2.38}
$$

ただし，$\mathbf{n}_{ji}$は検査体積jの面ijにおける外向き法線ベクトルであり，$\mathbf{n}_{ji}=-\mathbf{n}_{ij}$である．

以上のことから，(2.27)式は

$$(\mathbf{D}+\mathbf{L}+\mathbf{U})\Delta\mathbf{Q}^n = \mathbf{R}^n \tag{2.39}$$

と書くことができた．(2.39)式は，近似因数分解（AF；Approximate Factorization）法によって解ける．いま，

$$(\mathbf{D}+\mathbf{L}+\mathbf{U}) = (\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U}) + (\mathbf{L}\mathbf{D}^{-1}\mathbf{U}) \tag{2.40}$$

が成り立つ．Δt が十分小さければ $1+\Delta t \sim O(1)$であるから，右辺第一項が $O(1)$であるのに対して，右辺第二項は $O(\Delta t^2)$となる†．したがって，この項を無視することにすれば，(2.39) 式は，

$$(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})\Delta\mathbf{Q}^n = \mathbf{R}^n \tag{2.41}$$

と左辺を（近似的に）因数分解することができる．(2.41)式は $\Delta\mathbf{Q}^{n-1/2} \equiv \mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})\Delta\mathbf{Q}^n$ とおくことにより，

$$
\begin{aligned}
&Forward\ \ sweep: \\
&\quad (\mathbf{D}+\mathbf{L})\Delta\mathbf{Q}^{n-1/2} = \mathbf{R}^n \\
&Backward\ sweep: \\
&\quad (\mathbf{D}+\mathbf{U})\Delta\mathbf{Q}^{n} = \mathbf{D}\Delta\mathbf{Q}^{n-1/2}
\end{aligned}
\tag{2.42}
$$

と二段階に分けることができる．$(\mathbf{D}+\mathbf{L})$及び$(\mathbf{D}+\mathbf{U})$がそれぞれ下及び上三角行列であることを考慮すれば，(2.40)式は以下のようにして解くことができる．

$$
\begin{aligned}
&step\ 1: \Delta\mathbf{Q}^{n-1/2} = \mathbf{D}^{-1}\left[\mathbf{R}^n - \mathbf{L}\Delta\mathbf{Q}^{n-1/2}\right] \\
&step\ 2: \Delta\mathbf{Q}^{n} \quad\ = \Delta\mathbf{Q}^{n-1/2} - \mathbf{D}^{-1}\mathbf{U}\Delta\mathbf{Q}^n
\end{aligned}
\tag{2.43}
$$

(2.43)式を具体的に書き下せば，

$step\ 1:$

$$\Delta\mathbf{Q}_i^{n-1/2} = \mathbf{D}_i^{-1}\left[\mathbf{R}_i^n - \frac{1}{2}\sum_{j>i}\left\{\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} - |\lambda_{ij}|\right\}\Delta S_{ij}\Delta\mathbf{Q}_j^n\right]$$

$step\ 2:$

$$\Delta\mathbf{Q}_i^{n} = \Delta\mathbf{Q}_i^{n-1/2} - \frac{1}{2}\mathbf{D}_i^{-1}\sum_{j<i}\left\{\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} - |\lambda_{ji}|\right\}\Delta S_{ij}\Delta\mathbf{Q}_j^{n-1/2} \tag{2.44}$$

となる．ただし，

$$\mathbf{D}_i = \frac{V_i}{\Delta t}\mathbf{I} - \sum_{j(i)}\left[\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i} + |\lambda_{ij}|\mathbf{I}\right]\Delta S_{ij} \tag{2.45}$$

である．ここで，ヤコビ行列の性質，

$$\sum_{j(i)}\frac{\partial \mathbf{f}(\mathbf{Q}_i,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i}\Delta S_{ij} = 0 \tag{2.46}$$

を考慮し，加えて，ヤコビ行列$\partial\mathbf{f}/\partial\mathbf{Q}$の計算を省くため，

$$\frac{\partial \mathbf{f}(\mathbf{Q}_j,\mathbf{n}_{ij})}{\partial \mathbf{Q}_i}\Delta\mathbf{Q}_j \sim \Delta\mathbf{f}_{ij} \equiv \mathbf{f}(\mathbf{Q}+\Delta\mathbf{Q},\mathbf{n}) - \mathbf{f}(\mathbf{Q},\mathbf{n}) \tag{2.47}$$

なる近似を行うことにする．また，節点iと節点jの節点番号の大小関係は，例えばベクトル化のために超平面（hyper plane）を導入したような場合には，その節点の属する超平面番号の大小関係に置き換えられるため，下三角形要素に属する節点の集合$j\in L(i)$と上三角形要素に属する節点の集合$j\in U(i)$でより一般的に表すことにする．これにより，最終的に行列要素を保存する必要なし（matrix-free form）に計算可能な(2.48)式を得る．

$step\ 1:$

$$\Delta\mathbf{Q}_i^{n-1/2} = \mathbf{D}_i^{-1}\left[\mathbf{R}_i^n - \frac{1}{2}\sum_{j\in L(i)}\left\{\Delta\mathbf{f}(\mathbf{Q}_j^{n-1/2},\mathbf{n}_{ij}) - |\lambda_{ij}|\Delta\mathbf{Q}_j^n\right\}\Delta S_{ij}\right]$$

$step\ 2:$

$$\Delta\mathbf{Q}_i^{n} = \Delta\mathbf{Q}_i^{n-1/2} - \frac{1}{2}\mathbf{D}_i^{-1}\sum_{j\in U(i)}\left\{\Delta\mathbf{f}(\mathbf{Q}_j^{n},\mathbf{n}_{ij}) - |\lambda_{ji}|\right\}\Delta S_{ij}\Delta\mathbf{Q}_j^{n-1/2} \tag{2.48}$$

ただし，

† Δt が十分大きいと右辺第二項は $O(\Delta t)$となり，線形化による誤差と共に２次の収束性を得られなくなる要因のひとつとなる．

$$\mathbf{D}_i = \left(\frac{V_i}{\Delta t} - \sum_{j(i)} \left| \lambda_{ij} \right| \Delta S_{ij} \right) \mathbf{I} \tag{2.49}$$

である．以上の解法は，ヤコビ行列のブロック対角項$\mathbf{D}_i$が対角化されたことにより，逆行列の計算を必要としない（近似的なLU分解による）直接法となっている．一方で，(2.48)式は，反復解法の一種である二段階の対称Gauss−Seidel（SGS）法（の特別な場合）となっている（APPENDIX E参照）．そこで，このような解法はLU−SGS法と呼ばれている[*10*]，[*11*]，[*12*].

2.7 LU−SGS 前処理付き GMRES 法

LU−SGS 法は, 前処理として利用することが可能である（APPENDIX E）．したがって，LU−SGS 法が採用されているEuler/Navier-Stokesソルバに GMRES法を適用しようとした場合，その前処理として LU−SGS 法を用いるのは極めて自然なことである．例えば，前処理に不完全 LU 分解（Incomplete LU decomposition）を用いた場合には，不完全 LU 分解を行うためのコストがかかると共に，得られた行列を保存するための領域が別途必要となる．LU−SGS 法では，既に保存されている係数行列 **A** をそのまま利用することが可能であり，LU 分解にかかるコストも前処理行列のための余分な保存領域も必要としない．LU−SGS 法を前処理とした GMRES 法の構築は極めて簡単である．前処理行列 **M** を

$$\mathbf{M} = (\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U}) \tag{2.50}$$

とおけばよい．そして，左前処理の場合にはアルゴリズム 2.6 に従うと共に，連立方程式(2.13)式及び(2.14)式を LU−SGS 法を用いて解けばよい. 一方，右前処理の場合にはアルゴリズム 2.7 に従い，(2.19)及び(2.20)式を解くことになる. LU−SGS 法により連立方程式を解くためのプログラムは，既に当該ソルバに含まれている．これにより，GMRES 法導入に伴うプログラミングの負担も軽減される.

3. 計算結果

ここでは，GMRES 法により簡単な係数行列の基に連立方程式を解いた結果について示す．以下では，全てのケースにおいて, 修正 Gram-Schmidt の直交化を用いて直交基底を生成法した.

<u>ケース１</u>　係数行列として Toeplitz 行列の一種である

$$\mathbf{A} = \begin{pmatrix} \alpha & \beta & 0 & 0 & \cdots \\ 0 & \alpha & \beta & 0 & \cdots \\ \gamma & 0 & \alpha & \beta & \cdots \\ 0 & \gamma & 0 & \alpha & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.1}$$

において，α=2，β=1 とし，γ をパラメータとして連立方程式 **Ax=b** を Symmetric Gauss-Seidel（SGS）法と GMRES 法によって解いた結果について，反復回数 n に対する残差‖$\mathbf{b}\text{-}\mathbf{A}\mathbf{x}_n$‖の値を図 3.1 に示す．定数ベクトルは $\mathbf{b}$=**1**（全ての要素を１）とし，初期ベクトルは $\mathbf{x}_0$=**0** とした．また，連立方程式の次元 N は 100 である．

図 3.1 より明らかなように係数行列が対角優位†な場合（γ=1.0）においては，SGS 法が GMRES 法の約半分程度の反復回数で収束するものの，非対角項が優位となる γ=1.5 では一時的な残差の増大が見られ，γ=2.0 では更なる残差の増大と 10^{-8} 程度で収束が頭打ちとなり解法が不安定となっていることがわかる．実際，γ=2.5 では発散する．一方，GMRES 法は一貫して安定した収束を見せ，その安定性において SGS 法に比べ優位であることがわかる.

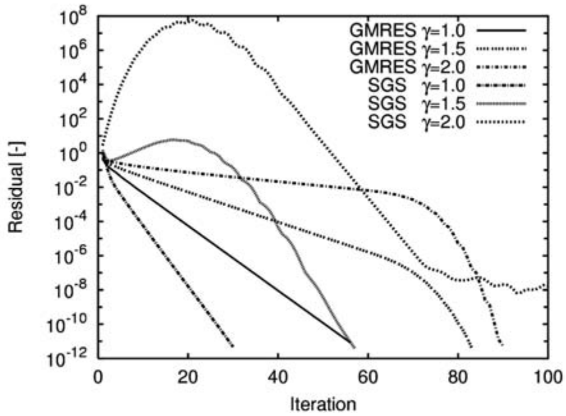

図 3.1　SGS 法と GMRES 法の比較

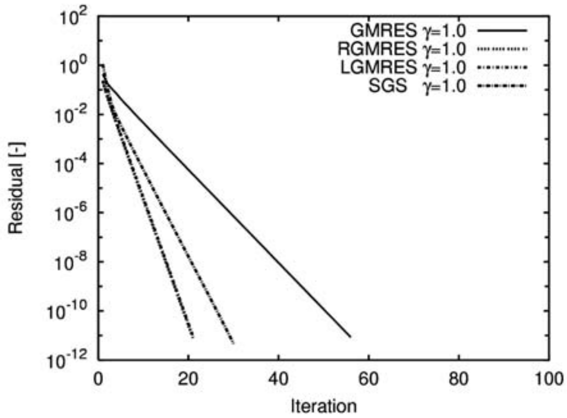

図 3.2　SGS 法と前処理付 GMRES 法の比較（γ=1.0）

<u>ケース２</u>　ケース１の γ=1.0 の場合について，SGS 法，GMRES 法，SGS 右前処理付 GMRES（RGMRES）法

† ここでは，行列 $\mathbf{A}=[a_{i,j}]$において全ての行 i において，対角項が非対角項の和より大きい場合，すなわち$|a_{i,i}|\geqq\Sigma\,|a_{i,j}|$のとき，対角優位行列であるものとする．逆に$|a_{i,i}|<\Sigma\,|a_{i,j}|$のとき非対角優位行列であるとする.

および SGS 左前処理付 GMRES（LGMRES）法について収束性を比較した結果を図 3.2 に示す．SGS 前処理法については，APPENDIX H を参照されたい．

この図において，右前処理付 GMRES（RGMRES）法と左前処理付 GMRES（LGMRES）法は，ほぼ一致しており，SGS 法より早い収束を見せている．これにより，前処理を行うことで収束性が改善されることが確認された．また，右前処理と左前処理に有意な差のないことが確認された．

ケース３　ケース１の γ=2.0 の場合について，SGS 法，GMRES 法，SGS 右前処理付 GMRES（RGMRES）法および SGS 左前処理付 GMRES（LGMRES）法についてその収束性を比較した結果を図 3.3 に示す．

この図においてもケース２の場合と同様，RGMRES 法と LGMRES 法はほぼ一致しており，SGS 法より安定でかつ早い収束性を見せていることがわかる．これにより，非対角優位な場合においても，前処理として SGS 法が適用可能であることがわかった．

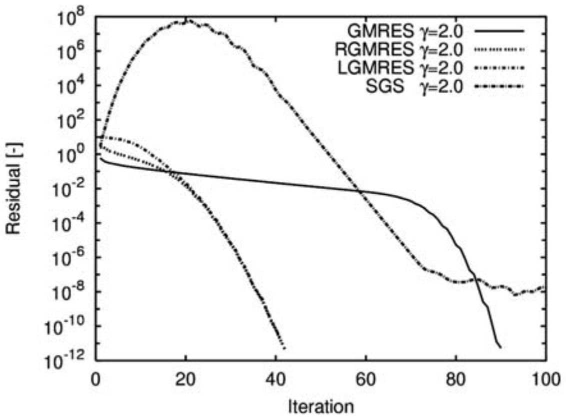

図 3.3　SGS 法と前処理付 GMRES 法の比較（γ=2.0）

ケース４　ケース１において，γ=2.0, N=1000 として，GMRES 法とリスタート GMRES（RGMRES）法の比較を行った結果を図3.4に，GMRES法と不完全GMRES（TGMRES）法の比較を行った結果を図 3.4 にそれぞれ示す．（ ）内には，基底ベクトル数を示した．

基底ベクトルの数を 10 とした場合（RGMRES(10)および TGMRES(10)）において明らかな収束性の低下が見られるが，30 では GMRES 法と同様の収束性を示しており，リスタート GMRES 法および不完全 GMRES 法の有効性が確認された．一方で，リスタート GMRES 法と不完全 GMRES 法では，同じ基底ベクトル 10 において明らかに収束性に差がみられる．リスタート GMRES 法に比べ，不完全 GMRES 法の実装が極めて困難であることを考え合わせれば，実用性においてはリスタート GMRES 法の方が優れているといえる．

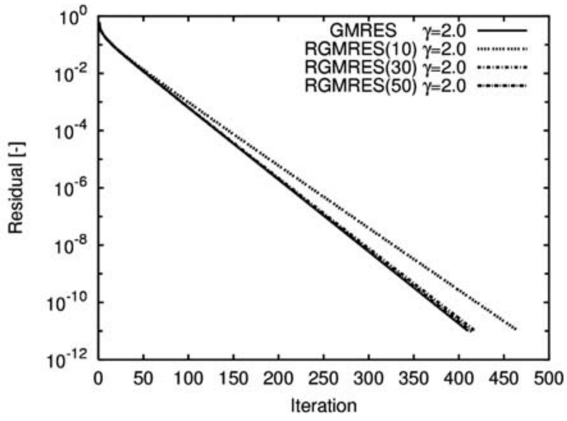

図 3.4　GMRES 法とリスタート GMRES 法の比較

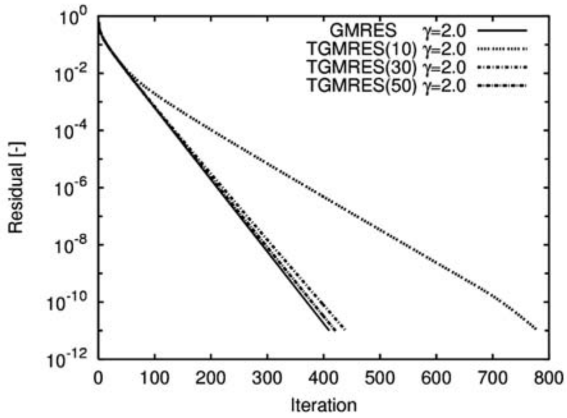

図 3.5　GMRES 法と不完全 GMRES 法の比較

ケース５　最後に GMRES 法に限らず，全ての Krylov 部分空間法に共通する収束性に関して問題となる典型的な例を図 3.6 に示す．この例では，係数行列 $\mathbf{A}$ として，三重対角行列

$$\mathbf{A} = \begin{pmatrix} 2.0 & 1.0 & 0 & 0 & \cdots \\ 1.2 & 2.0 & 1.0 & 0 & \cdots \\ 0 & 1.2 & 2.0 & 1.0 & \cdots \\ 0 & 0 & 1.2 & 2.0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.2}$$

を用い，N=100 とした．定数ベクトルおよび初期ベクトルは，ケース１同様 $\mathbf{b}=\mathbf{1}$，$\mathbf{x}_0=\mathbf{0}$ である．反復回数 n に対する残差 $\|\mathbf{b}-\mathbf{A}\mathbf{x}_n\|$ を図示した．

このケースでは，100 元連立方程式に対して反復 99 回までは，ほとんど残差の減少がみられず，10^{-2} のオーダーに留まっているにもかかわらず，反復 100 回目ではほぼ厳密解に収束している．これは，GMRES 法を含む Krylov 部分空間法が N 回の反復で必ず厳密解に収束することの一例となっている．一方で，数百万元，数千万元あるいは数億元の大規模連立方程式の解法として適用するには，N 回より十分早く必要な精度の解が得られることが求められており，Krylov 部分空間法の収束特性の欠点としてよく知られる典型例ともなっている．

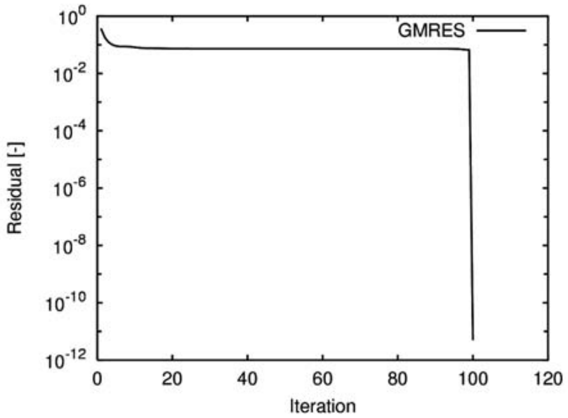

図 3.6　Krylov 部分空間法の収束性

4.　考察

「3. 計算結果」のケース 1 に示した計算結果のうち，GMRES 法についてのみ，改めて図 4.1 に示す．ここでは，γ=2.5，γ=3.0 および γ=3.5 の場合の計算結果も合わせて示している．

GMRES 法に限らず Krylov 部分空間法に属するアルゴリズムは，N 元連立方程式 $\mathbf{Ax}=\mathbf{b}$ に対して，その係数行列 $\mathbf{A}$ が正則である限り，N 回の反復で必ず厳密解に収束することが数学的に保障されており，行列 $\mathbf{A}$ の（正則性以外の）性質には依存しないと考えられている．ところが，図 4. 1 に示された計算結果では，非対角項 γ が大きくなるに従い収束性が悪くなり，γ=3.5 では計算が不安定になって，十分な収束が得られなくなっている．このことから，明らかに非対角項が優位になるに従い，計算が不安定になるという相関が存在することがわかる．

ところで，同じく厳密解を与えるアルゴリズムに，直接法の一種である Gauss の消去法がある．この解法も，数学的には係数行列 $\mathbf{A}$ が正則であれば，必ず厳密解が求まるアルゴリズムである．しかし，この解法を数値解法として適用すると，対角項に比べて非対角項が大きい場合に不安定になるため，行の入れ替えにより最も大きな行列要素を対角項にするピボッティング（pivoting）と呼ばれる操作を併用する必要があることが知られている．例えば，係数行列 $\mathbf{A}=[a_{i,j}]$において，非対角項と対角項の比の総乗 $\Pi(a_{i,j}/a_{i,i})$，i=1,N は，連立方程式の次元 N および各項 $a_{i,j}/a_{i,i}$ が有限である限り，数学的に発散することはない．しかし，例えば，1.1^{900}～$1.8\text{x}10^{37}$ であることを考えれば，大規模連立方程式における数値計算の場合には，発散は容易に起こりうる．Krylov 部分空間法の基となる Arnoldi 法においては，正規直交基底の生成過程に誤差が含まれる可能性のあることは既に知られている．しかし，それとは別に，Arnoldi 法には，係数行列 $\mathbf{A}$ のべき乗および Hessenberg 行列の要素 $h_{j+1,j}$ による除算が含まれている．図 4.1 に示された結果からは，係数行列の非対角項が優位になることにより，これらの値の比が発散傾向を示し，Gauss の消去法と同じような数値計算上の不安定性をもたらしていると考えることができる．

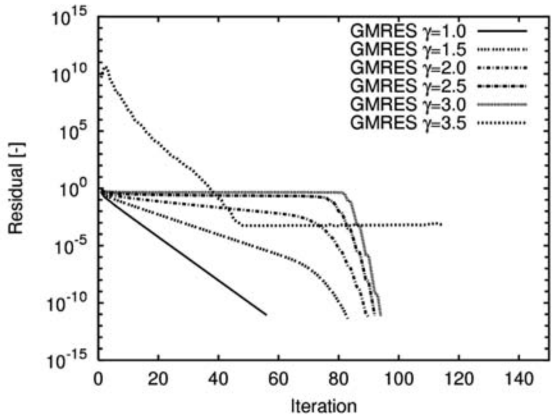

図 4.1　GMRES 法の非対角項依存性

図 4.2 に，「3. 計算結果」ケース 1 に示した条件の下，γ=3.0 とした場合の GMRES 法及び SGS 左前処理付 GMRES（LGMRES)法の収束状況を示す．GMRES 法では，およそ反復 80 回程度まで明確な収束が見られないものの，その後急速に残差が減少し，ほぼ厳密解に近い値が得られている．一方で，前処理を行ったケースでは，反復初期に発散傾向が見られ，その後収束に向かうものの，残差がおよそ 10^{-7} 程度に減少した段階で頭打ちとなっており，計算が不安定となっていることがわかる．この原因は，SGS 法が，連立方程式(2.13)式

$$\mathbf{M}\mathbf{w}_j = \tilde{\mathbf{v}}_j\,,\quad \tilde{\mathbf{v}}_j = \mathbf{A}\mathbf{v}_j \qquad (2.13)$$

を正しく解けないことにある．前処理行列 $\mathbf{M}$ は，係数行列 $\mathbf{A}$ の十分な近似になっていることが望ましい．このことは，行列 $\mathbf{M}$ が行列 $\mathbf{A}$ の性質もよく受け継いでいることを意味する．したがって，行列 $\mathbf{A}$ が非対角優位であるならば，行列 $\mathbf{M}$ も非対角優位である可能性がある．この場合，SGS 法が(2.13)式を正しく解けなくなる可能性がある．前処理の性格上(2.13)式は，厳密に解かれる必要はないが，必要な精度が得られないと前処理付 GMRES 法が不安定になることを図 4.2 の例は示しており，前処理にも限界のあることがわかる．

一方で，我々の目的は CFD の分野において表れる連立方程式を短時間のうちに，安定して，精度よく解くことにある．したがって，一般の悪条件（ill-conditioned）な係数行列を持つ連立方程式が解けなければならない必然性はない．実際，連立方程式の係数行列の条件を改善するための試みとして，Navier-Stokes 方程式の定常解を時間発展の下に解くこと等が行われている．

結局，より良い Euler／Navier-Stokes ソルバを開発するためには，古典的反復解法よりは安定な GMRES 法

のような解法を適用することにとどまらず，このような係数行列改善の試みを組み合わせていくことも重要であると言える．

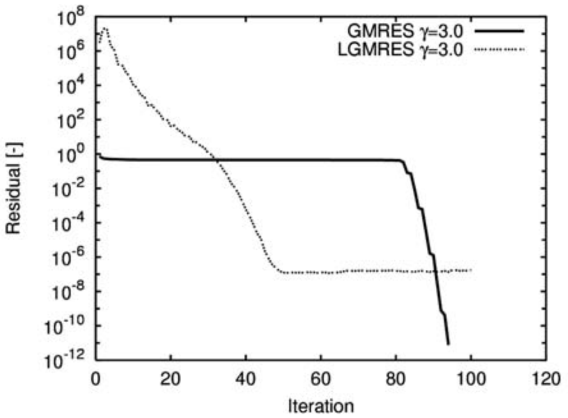

図 4.2　悪条件の場合の前処理付 GMRES 法

5.　まとめ

簡単なToeplitz行列を係数行列とした連立方程式に対して，GMRES 法の安定性と前処理の有効性が確認できた．一方で，非対角項が優位になるに従い，GMRES 法も不安定となり，その安定性に限界のあることも確認された．また，前処理が上手く働かない場合のあることもわかった．これらのことから，Euler／Navier－Stokes ソルバの開発には，より良い連立方程式の解法を選択するにとどまらず，Euler／Navier－Stokes 方程式を離散化する際に，最終的に得られる連立方程式の係数行列の条件が良くなるような工夫も重要であるといえる．

今後，実際に JTAS コードに組み込み，より現実的な計算条件の下に，GMRES 法の安定性に加え，計算時間の評価を行う必要がある．

なお，反復法の収束性に関する解析は，係数行列の固有値を基に行われるのが一般的である．しかし，大規模連立方程式では，その固有値を求めることが困難であることから，それがどのような分布を持っているのかについて，全く明らかになっていない．一方で，対角項と非対角項の大小比較は，より簡単に行うことができる．この観点から実問題に対する安定性を検討することにより，現象論的な知見が得られると期待される．

APPENDIX A　Arnoldi 法の行列表現

ここでは，Arnoldi 法の行列表現について考える．アルゴリズ 2.1 のステップ 5 及び 8 より，

$$\mathbf{A}\mathbf{v}_j = \sum_{i=1}^{j+1} h_{i,j}\mathbf{v}_i \quad for\ j = 1,2,\ldots,m \tag{A.1}$$

である．この m 本 (j=1,2,⋯,m) の式をまとめて書けば，

$$\mathbf{A}\mathbf{V}_m = \left[\sum_{i=1}^{2} h_{i,1}\mathbf{v}_i \ \ \sum_{i=1}^{3} h_{i,2}\mathbf{v}_i \ \cdots \ \sum_{i=1}^{m+1} h_{i,m}\mathbf{v}_i\right] \tag{A.2}$$

ただし，$\mathbf{V}_m$ は N 行 m 列の行列 $\mathbf{V}_m=[\mathbf{v}_0 \ \ \mathbf{v}_1 \ \ \cdots \ \ \mathbf{v}_m]$（正規直交基底 $\mathbf{v}_i$ を列ベクトルとして持つ行列）であり，右辺は $\Sigma h_{i,j}\mathbf{v}_i$（$j$=1,2,⋯,$m$）を列ベクトルとして持つ N 行 m 列の行列である．ところで，(A.2)式右辺の第 1 列ベクトルは，

$$\begin{aligned}\sum_{i=1}^{2} h_{i,1}\mathbf{v}_i &= h_{1,1}\mathbf{v}_1 + h_{2,1}\mathbf{v}_2 \\ &= \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2\end{bmatrix}\begin{bmatrix}h_{1,1}\\ h_{2,1}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \cdots & \mathbf{v}_m & \mathbf{v}_{m+1}\end{bmatrix}\begin{bmatrix}h_{1,1}\\ h_{2,1}\\ \vdots \\ h_{m,1}\\ h_{m+1,1}\end{bmatrix} \\ &= \mathbf{V}_{m+1}\mathbf{h}_1^{(m+1)}\end{aligned}$$

と書ける．但し，$i > j+1$ のとき $h_{i,j}=0$ であるとした．また，ベクトル $\mathbf{h}$ の上付き添え字はベクトルの次元を表すものとし，$\mathbf{h}_1^{(m+1)}=[h_{1,1}\ h_{2,1}\ \cdots\ h_{m+1,1}]^{\mathrm{T}}$ と置いた．同様にして，$\mathbf{h}_k^{(m+1)}=[h_{1,k}\ h_{2,k}\ \cdots\ h_{m+1,k}]^{\mathrm{T}}$（$k$=1,2,⋯,$m$）と置けば，一般に右辺第 k 列ベクトルは，

$$\sum_{i=1}^{k+1} h_{i,k}\mathbf{v}_i = \mathbf{V}_{m+1}\mathbf{h}_k^{(m+1)} \tag{A.3}$$

となる．よって，N 行 m+1 列の行列 $\tilde{\mathbf{H}}_m$ を $\tilde{\mathbf{H}}_m=[\mathbf{h}_1^{(m+1)}\ \mathbf{h}_2^{(m+1)}\ \cdots\ \mathbf{h}_{m+1}^{(m+1)}]^{\mathrm{T}}$ とおけば，

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\tilde{\mathbf{H}}_m$$

$$= \begin{bmatrix}\mathbf{v}_1 \cdots \mathbf{v}_{m+1}\end{bmatrix} \begin{array}{c}\begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ 0 & h_{3,2} & \cdots & h_{3.m-1} & h_{3.m} \\ 0 & 0 & \cdots & h_{4,m-1} & h_{4,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \cdots & 0 & h_{m+1,m} \end{bmatrix} \\ \begin{array}{ccccc}\mathbf{h}_1^{(m+1)} & \mathbf{h}_2^{(m+1)} & \cdots & \mathbf{h}_{m-1}^{(m+1)} & \mathbf{h}_m^{(m+1)}\end{array}\end{array}$$

を得る．この式の両辺に左から $\mathbf{V}_m^{\mathrm{T}}$ を掛ければ，

$$
\begin{aligned}
\mathbf{V}_m^{\mathrm{T}}\mathbf{A}\mathbf{V}_m &= \mathbf{V}_m^{\mathrm{T}}\mathbf{V}_{m+1}\widetilde{\mathbf{H}}_m \\
&= \begin{bmatrix} (\mathbf{v}_1,\mathbf{v}_1)(\mathbf{v}_1,\mathbf{v}_2)\cdots(\mathbf{v}_1,\mathbf{v}_{m+1}) \\ \vdots \qquad \vdots \qquad \ddots \qquad \vdots \\ (\mathbf{v}_m,\mathbf{v}_1)(\mathbf{v}_m,\mathbf{v}_2)\cdots(\mathbf{v}_m,\mathbf{v}_{m+1}) \end{bmatrix}\widetilde{\mathbf{H}}_m \\
&= m\text{行}\left\{ \overbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}}^{m+1\text{列}} \right.\widetilde{\mathbf{H}}_m \\
&= \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}\begin{bmatrix} \mathbf{h}_1^{(m)} & \cdots & \mathbf{h}_m^{(m)} \\ 0 & \cdots & h_{m+1,m} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{h}_1^{(m)} & \mathbf{h}_2^{(m)} & \cdots & \mathbf{h}_m^{(m)} \end{bmatrix}
\end{aligned}
$$

となる．但し，$(\mathbf{v}_i,\mathbf{v}_j)=\delta_{i,j}$ $(i,j=1,\cdots,m)$であることを用いた．ここで，$\mathbf{H}_m=[\,\mathbf{h}_1^{(m)}\ \mathbf{h}_2^{(m)}\ \cdots\ \mathbf{h}_m^{(m)}\,]^{\mathrm{T}}$とおけば，

$$\mathbf{V}_m^{\mathrm{T}}\mathbf{A}\mathbf{V}_m = \mathbf{H}_m$$

と書ける．以上により，Arnoldi 法の行列表現

$$
\begin{aligned}
\mathbf{A}\mathbf{V}_m &= \mathbf{V}_{m+1}\widetilde{\mathbf{H}}_m \\
\mathbf{V}_m^{\mathrm{T}}\mathbf{A}\mathbf{V}_m &= \mathbf{H}_m
\end{aligned} \tag{A.4}
$$

が得られた．但し，

$$
\begin{aligned}
\mathbf{V}_m &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{m-1} & \mathbf{v}_m \end{bmatrix} \\
\mathbf{H}_m &= \begin{bmatrix} \mathbf{h}_1^{(m)} & \mathbf{h}_2^{(m)} & \cdots & \mathbf{h}_{m-1}^{(m)} & \mathbf{h}_m^{(m)} \end{bmatrix} \\
&= \begin{bmatrix} h_{i,j} \end{bmatrix} \\
\widetilde{\mathbf{H}}_m &= \begin{bmatrix} \mathbf{h}_1^{(m)} & \mathbf{h}_2^{(m)} & \cdots\mathbf{h}_{m-1}^{(m)} & \mathbf{h}_m^{(m)} \\ 0 & 0 & \cdots 0 & h_{m+1,m} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{H}_m \\ 0\,0\cdots\ 0\ h_{m+1,m} \end{bmatrix} \\
\mathbf{h}_i^{(m)} &= \begin{bmatrix} h_{1,i} & h_{2,i} & \cdots & h_{m,i} \end{bmatrix},\ i=1,\cdots m \\
h_{i,j} &= 0,\quad if\ i > j+1
\end{aligned} \tag{A.5}
$$

である．ここで，$i>j+1$ のとき $h_{i,j}=0$ であるから，行列 $\mathbf{H}_m$ は Hessenberg 行列である．このことから，Arnoldi 法とは，係数行列 $\mathbf{A}$ を Hessenberg 行列 $\mathbf{H}_m$ に相似変換する直交行列 $\mathbf{V}_m$ を生成するアルゴリズムであることがわかる．

APPENDIX B 最小化すべき関数 $J(\mathbf{y}_m)$

ここでは(2.6)式

$$J(\mathbf{y}_m) = \| \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m) \| \tag{B.1}$$

より，(2.7)式として示した最小化すべき関数 $J(\mathbf{y}_m)$

$$J(\mathbf{y}_m) = \left\| \beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}_m\mathbf{y}_m \right\| \tag{B.2}$$

の導出を行う．

いま，ベクトル $\mathbf{y}_m$ のベクトル関数 $\mathbf{J}(\mathbf{y}_m)$を

$$\mathbf{J}(\mathbf{y}_m) = \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m) \tag{B.3}$$

で定義する．右辺をまとめ直せば，

$$
\begin{aligned}
\mathbf{J}(\mathbf{y}_m) &= (\,\mathbf{b} - \mathbf{A}\mathbf{x}_0\,) - \mathbf{A}\mathbf{V}_m\mathbf{y}_m \\
&= \mathbf{r}_0 - \mathbf{A}\mathbf{V}_m\mathbf{y}_m
\end{aligned}
$$

ここで，GMRES 法では，直交基底の最初のベクトル $\mathbf{v}_1$ として $\mathbf{v}_1=\mathbf{r}_0/\|\mathbf{r}_0\|$ を選ぶので，$\beta=\|\mathbf{r}_0\|$ とおけば，$\mathbf{r}_0=\beta\,\mathbf{v}_1$ であるから

$$\mathbf{J}(\mathbf{y}_m) = \beta\,\mathbf{v}_1 - \mathbf{A}\mathbf{V}_m\mathbf{y}_m$$

また，$\mathbf{e}_1$ を最初の成分のみ 1 である $m+1$ 次元の単位ベクトルとすれば，

$$
\begin{aligned}
\mathbf{V}_{m+1}\mathbf{e}_1 &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m & \mathbf{v}_{m+1} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&= \mathbf{v}_1
\end{aligned}
$$

であり，APPENDIX A (A.4)式より

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\widetilde{\mathbf{H}}_m$$

であるから，

$$
\begin{aligned}
\mathbf{J}(\mathbf{y}_m) &= \beta\,\mathbf{V}_{m+1}\mathbf{e}_1 - \mathbf{V}_{m+1}\widetilde{\mathbf{H}}\,\mathbf{y}_m \\
&= \mathbf{V}_{m+1}\left(\beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}\,\mathbf{y}_m\right)
\end{aligned}
$$

ところで，$(\mathbf{v}_i,\mathbf{v}_j)=\delta_{i,j}$ $(i,j=1,\cdots,m)$であるから，$\mathbf{I}$ を $m+1$ 行 $m+1$ 列の単位行列としたときに，

$$\mathbf{V}_{m+1}^{\mathrm{T}}\mathbf{V}_{m+1} = \mathbf{I}$$

であり，任意の $m+1$ 次元ベクトルを $\mathbf{q}$ としたときに，

$$
\begin{aligned}
\|\mathbf{V}_{m+1}\,\mathbf{q}\| &= (\mathbf{V}_{m+1}\,\mathbf{q}, \mathbf{V}_{m+1}\,\mathbf{q})^{1/2} \\
&= \left(\mathbf{q}, \mathbf{V}_{m+1}^{\mathrm{T}}\mathbf{V}_{m+1}\,\mathbf{q}\right)^{1/2} \\
&= (\mathbf{q}, \mathbf{q})^{1/2} \\
&= \|\mathbf{q}\|
\end{aligned}
$$

であるから，

$$\mathbf{J}(\mathbf{y}_m) = \left(\beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}\,\mathbf{y}_m\right)$$

即ち，

$$
\begin{aligned}
\mathbf{J}(\mathbf{y}_m) &= \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y}_m) \\
&= \left(\beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}\,\mathbf{y}_m\right)
\end{aligned} \tag{B.4}
$$

よって，(B.1)式より(B.2)式が導かれた．

APPENDIX C 関数 $J(\mathbf{y}_m)$を最小にする $\mathbf{y}_m$

ここでは，関数 $J(\mathbf{y}_m)$

$$J(\mathbf{y}_m) = \left\| \beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}_m \right\| \qquad \text{(C.1)}$$

を最小にするベクトル $\mathbf{y}_m$ を求める方法を示す．この問題は，ベクトルの直交行列による変換が，ベクトルの向きを変えるものの大きさ（ノルム）は変えないことを利用し，行列 $\widetilde{\mathbf{H}}_m$ を上三角行列に変換することにより解くことが出来る．上三角行列に変換するための直交変換としては，Givens 回転による QR 分解を用いる．このとき，行列 $\widetilde{\mathbf{H}}_m$ は m+1 行 m 列の行列であるから，最後の行（m+1 行目）の要素は，すべて 0 となる．

いま，回転行列 $\mathbf{\Omega}_i$ (i=1,⋯,m)を

$$\mathbf{\Omega}_i = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_i & s_i & & \\ & & & -s_i & c_i & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow\ i\ \text{行} \\ \leftarrow\ i+1\ \text{行} \\ \\ \\ \\ \end{array} \qquad \text{(C.2)}$$

で定義する．GMRES 法の反復 m 回目においてこの行列は，m+1 行 m+1 列の正方行列となる．そして，これらの回転行列の積を $\mathbf{Q}_m$ とおく．即ち，

$$\mathbf{Q}_m = \prod_{i=1}^{m} \mathbf{\Omega}_i \qquad \text{(C.3)}$$

このとき，$\mathbf{Q}_m\mathbf{Q}^{\mathrm{T}}{}_m=\mathbf{Q}^{\mathrm{T}}{}_m\mathbf{Q}_m=\mathbf{I}$ であれば

$$\begin{aligned} J(\mathbf{y}_m) &= \left\| \beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}_m \right\| \\ &= \left\| \mathbf{Q}_m \left(\beta\,\mathbf{e}_1 - \widetilde{\mathbf{H}}_m \mathbf{y}_m \right) \right\| \\ &= \left\| \mathbf{Q}_m \beta\,\mathbf{e}_1 - \mathbf{Q}_m \widetilde{\mathbf{H}}_m \mathbf{y}_m \right\| \\ &= \left\| \widetilde{\mathbf{g}}_m - \widetilde{\mathbf{R}}_m \mathbf{y}_m \right\| \end{aligned}$$

となる．ただし，

$$\begin{aligned} \widetilde{\mathbf{g}}_m &\equiv \mathbf{Q}_m \beta\,\mathbf{e}_1 \\ \widetilde{\mathbf{R}}_m &\equiv \mathbf{Q}_m \widetilde{\mathbf{H}}_m \end{aligned} \qquad \text{(C.4)}$$

と置いた．ここで，(C.4)第二式が QR 分解に相当する†．これにより

$$J(\mathbf{y}_m) = \left\| \widetilde{\mathbf{g}}_m - \widetilde{\mathbf{R}}_m \mathbf{y}_m \right\| \qquad \text{(C.5)}$$

を最小にするベクトル $\mathbf{y}_m$ を求めればよいこととなる．

† 通常行列 $\mathbf{H}$ を QR 分解する場合には，$\mathbf{H}=\mathbf{QR}$ と分解する．$\mathbf{QQ}^{\mathrm{T}}=\mathbf{Q}^{\mathrm{T}}\mathbf{Q}=\mathbf{I}$ に選ぶので，このときには $\mathbf{Q}^{\mathrm{T}}\mathbf{H}=\mathbf{R}$ となる．

まず，回転行列 $\mathbf{\Omega}_i$ を左からかけた時に行列 $\widetilde{\mathbf{H}}_m$ が上三角行列になるように c_i 及び s_i (i=1,⋯,m)を定める．m 行 m+1 列の行列 $\mathbf{R}_m{}^{(1)}$を

$$\mathbf{R}_m^{(1)} = \mathbf{\Omega}_1 \widetilde{\mathbf{H}}_m$$

とすれば，その要素表示は，

$$\begin{bmatrix} r_{1,1}^{(1)} & r_{1,2}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\ r_{2,1}^{(1)} & r_{2,2}^{(1)} & \cdots & r_{2,m-1}^{(1)} & r_{2,m}^{(1)} \\ & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & 0 & h_{m+1,m} \end{bmatrix}$$

$$= \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & \ddots \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m-1} & h_{2,m} \\ & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\ & & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & 0 & h_{m+1,m} \end{bmatrix}$$

と書ける．行列 $\widetilde{\mathbf{H}}_m$ の三行目以降は，回転行列 $\mathbf{\Omega}_i$ によって変更を受けないことに注意されたい．ここで，行列 $\widetilde{\mathbf{H}}_m$ を上三角行列に変換するために，$r^{(1)}{}_{2,1}=0$ としたい．したがって，

$$r_{2,1}^{(1)} = -s_1 h_{1,1} + c_1 h_{2,1} = 0 \qquad \text{(C.6)}$$

また，行列 $\mathbf{Q}_m$ が $\mathbf{Q}_m\mathbf{Q}^{\mathrm{T}}{}_m=\mathbf{Q}^{\mathrm{T}}{}_m\mathbf{Q}_m=\mathbf{I}$ であるためには，$\mathbf{\Omega}_i\mathbf{\Omega}^{\mathrm{T}}{}_i = \mathbf{\Omega}^{\mathrm{T}}{}_i\mathbf{\Omega}_i = \mathbf{I}$ であればよい‡ので，

$$\begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix} = \begin{bmatrix} c_i^2 + s_i^2 & 0 \\ 0 & c_i^2 + s_i^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

より，回転行列 $\mathbf{\Omega}_i$ の要素 c_i 及び s_i は関係式

$$c_i^2 + s_i^2 = 1 \qquad \text{(C.7)}$$

を満たせばよい．ここに，$i = 1,\cdots,m$ である．(C.6)及び(C.7)式より，

$$\begin{aligned} c_1 &= \frac{h_{1,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}} \\ s_1 &= \frac{h_{2,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}} \end{aligned} \qquad \text{(C.8)}$$

及び

‡ 正方行列 $\mathbf{A}$ 及び $\mathbf{B}$ に対して，$(\mathbf{AB})^{\mathrm{T}}=\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$ であるから，$\mathbf{Q}_m\mathbf{Q}^{\mathrm{T}}{}_m=(\mathbf{\Omega}_1\ldots\mathbf{\Omega}_m)(\mathbf{\Omega}_1\ldots\mathbf{\Omega}_m)^{\mathrm{T}}=\mathbf{\Omega}_1\ldots\mathbf{\Omega}_m\mathbf{\Omega}^{\mathrm{T}}{}_m\ldots\mathbf{\Omega}^{\mathrm{T}}{}_1$．よって，$\mathbf{\Omega}_i\mathbf{\Omega}^{\mathrm{T}}{}_i=\mathbf{I}$ ならば $\mathbf{Q}_m\mathbf{Q}^{\mathrm{T}}{}_m=\mathbf{I}$.

$$
\begin{aligned}
r_{1,j}^{(1)} &= c_1 h_{1,j} + s_1 h_{2,j} \\
&= \frac{h_{1,1} h_{1,j} + h_{2,1} h_{2,j}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}} \\
r_{2,j}^{(1)} &= -s_1 h_{1,j} + c_1 h_{2,j} \\
&= \frac{-h_{2,1} h_{1,j} + h_{1,1} h_{2,j}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}}
\end{aligned} \tag{C.9}
$$

を得る．ただし，$j=1,\cdots,m$ である．次に行列 $\mathbf{R}_m^{(2)}$を，

$$
\mathbf{R}_m^{(2)} = \mathbf{\Omega}_2 \mathbf{R}_m^{(1)} \left(= \mathbf{\Omega}_2 \mathbf{\Omega}_1 \widetilde{\mathbf{H}}_m \right)
$$

とすれば，その要素表示は，

$$
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
0 & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\
 & r_{3,2}^{(2)} & r_{3,3}^{(2)} & \cdots & r_{3,m-1}^{(2)} & r_{3,m}^{(2)} \\
 & & h_{4,3} & \cdots & h_{4,m-1} & h_{4,m} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & h_{m,m-1} & h_{m,m} \\
 & & & & 0 & h_{m+1,m}
\end{bmatrix}
=
\begin{bmatrix}
1 & & & & & \\
 & c_2 & s_2 & & & \\
 & -s_2 & c_2 & & & \\
 & & & 1 & & \\
 & & & & \ddots & \\
 & & & & & 1
\end{bmatrix}
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
r_{2,1}^{(1)} & r_{2,2}^{(1)} & \cdots & r_{2,m-1}^{(1)} & r_{2,m}^{(1)} \\
 & h_{3,2} & \cdots & h_{3,m-1} & h_{3,m} \\
 & & \ddots & \vdots & \vdots \\
 & & & h_{m,m-1} & h_{m,m} \\
 & & & 0 & h_{m+1,m}
\end{bmatrix}
$$

と書けるので，

$$
\begin{aligned}
c_2 &= \frac{r_{2,2}^{(1)}}{\sqrt{\left(r_{2,2}^{(1)} \right)^2 + h_{3,2}^2}} \\
s_2 &= \frac{h_{3,2}}{\sqrt{\left(r_{2,2}^{(1)} \right)^2 + h_{3,2}^2}}
\end{aligned} \tag{C.10}
$$

及び

$$
\begin{aligned}
r_{2,j}^{(2)} &= c_2 r_{2,j}^{(1)} + s_2 h_{3,j} \\
&= \frac{r_{2,2}^{(1)} r_{2,j}^{(1)} + h_{3,2} h_{3,j}}{\sqrt{\left(r_{2,2}^{(1)} \right)^2 + h_{3,2}^2}} \\
r_{3,j}^{(2)} &= -s_2 r_{2,j}^{(1)} + c_2 h_{3,j} \\
&= \frac{-h_{3,2} r_{2,j}^{(1)} + r_{2,2}^{(1)} h_{3,j}}{\sqrt{\left(r_{2,2}^{(1)} \right)^2 + h_{3,2}^2}}
\end{aligned} \tag{C.11}
$$

が得られる．ただし，$j = 2,\cdots,m$ である．以下順次行列 $\mathbf{\Omega}_i$ を左からかけて行けば $i = m$-1 のとき，

$$
\mathbf{R}_m^{(m-1)} = \mathbf{\Omega}_{m-1} \mathbf{R}_m^{(m-2)} \left(= \left(\prod_{i=1}^{m-1} \mathbf{\Omega}_i \right) \widetilde{\mathbf{H}}_m \right)
$$

より，

$$
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
0 & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\
 & 0 & r_{3,3}^{(3)} & \cdots & r_{3,m-1}^{(3)} & r_{3,m}^{(3)} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & r_{m-1,m-1}^{(m-1)} & r_{m-1,m}^{(m-1)} \\
 & & & & 0 & r_{m,m}^{(m-1)} \\
 & & & & 0 & r_{m+1,m}
\end{bmatrix}
=
\begin{bmatrix}
1 & & & & \\
 & \ddots & & & \\
 & & 1 & & \\
 & & & c_{m-1} & s_{m-1} & \\
 & & & -s_{m-1} & c_{m-1} & \\
 & & & & & 1
\end{bmatrix}
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
0 & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\
 & 0 & r_{3,3}^{(3)} & \cdots & r_{3,m-1}^{(3)} & r_{3,m}^{(3)} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & r_{m-1,m-1}^{(m-2)} & r_{m-1,m}^{(m-2)} \\
 & & & & 0 & h_{m,m} \\
 & & & & 0 & h_{m+1,m}
\end{bmatrix}
$$

とおけるので，

$$
\begin{aligned}
c_{m-1} &= \frac{r_{m-1,m-1}^{(m-2)}}{\sqrt{\left(r_{m-1,m-1}^{(m-2)} \right)^2 + h_{m,m-1}^2}} \\
s_{m-1} &= \frac{h_{m,m-1}}{\sqrt{\left(r_{m-1,m-1}^{(m-2)} \right)^2 + h_{m,m-1}^2}}
\end{aligned} \tag{C.12}
$$

及び

$$
\begin{aligned}
r_{m-1,j}^{(m-1)} &= c_{m-1} r_{m-1,j}^{(m-2)} + s_{m-1} h_{m,j} \\
&= \frac{r_{m-1,m-1}^{(m-2)} r_{m-1,j}^{(m-2)} + h_{m,m-1} h_{m,j}}{\sqrt{\left(r_{m-1,m-1}^{(m-2)} \right)^2 + h_{m,m-1}^2}} \\
r_{m,j}^{(m-1)} &= -s_{m-1} h_{m-1,j} + c_{m-1} h_{m,j} \\
&= \frac{-h_{m,m-1} r_{m-1,j}^{(m-2)} + r_{m-1,m-1}^{(m-2)} h_{m,j}}{\sqrt{\left(r_{m-1,m-1}^{(m-2)} \right)^2 + h_{m,m-1}^2}}
\end{aligned} \tag{C.13}
$$

を得る．ただし，$j=m$-1$,m$ である．最終的に，

$$
\widetilde{\mathbf{R}}_m = \mathbf{R}_m^{(m)} = \mathbf{\Omega}_m \mathbf{R}_m^{(m-1)} \left(= \left(\prod_{i=1}^{m} \mathbf{\Omega}_i \right) \widetilde{\mathbf{H}}_m \right)
$$

即ち，

$$
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
0 & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\
 & 0 & r_{3,3}^{(3)} & \cdots & r_{3,m-1}^{(3)} & r_{3,m}^{(3)} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & r_{m-1,m-1}^{(m-1)} & r_{m-1,m}^{(m-1)} \\
 & & & & 0 & r_{m,m}^{(m)} \\
 & & & & 0 & 0
\end{bmatrix}
=
\begin{bmatrix}
1 & & & & \\
 & \ddots & & & \\
 & & 1 & & \\
 & & & c_m & s_m \\
 & & & -s_m & c_m
\end{bmatrix}
\begin{bmatrix}
r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\
0 & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\
 & 0 & r_{3,3}^{(3)} & \cdots & r_{3,m-1}^{(3)} & r_{3,m}^{(3)} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & r_{m-1,m-1}^{(m-1)} & r_{m-1,m}^{(m-1)} \\
 & & & & 0 & r_{m,m}^{(m-1)} \\
 & & & & 0 & r_{m+1,m}
\end{bmatrix}
$$

において，

$$
\begin{aligned}
r_{m,m}^{(m)} &= c_m r_{m,m}^{(m-1)} + s_m h_{m+1,m} \\
0 &= -s_m r_{m,m}^{(m-1)} + c_m h_{m+1,m}
\end{aligned}
$$

また，

$$c_m^2 + s_m^2 = 1$$

であるから，$i = m$ のときにも同様に

$$
\begin{aligned}
c_m &= \frac{r_{m,m}^{(m-1)}}{\sqrt{\left(r_{m,m}^{(m)}\right)^2 + h_{m+1,m}^2}} \\
s_m &= \frac{h_{m+1,m}}{\sqrt{\left(r_{m,m}^{(m)}\right)^2 + h_{m+1,m}^2}}
\end{aligned}
\tag{C.14}
$$

及び

$$
\begin{aligned}
r_{m,m}^{(m)} &= c_m h_{m,m} + s_m h_{m+1,m} \\
&= \frac{r_{m,m}^{(m-1)} r_{m,m}^{(m-1)} + h_{m+1,m}\, h_{m,m}}{\sqrt{\left(r_{m,m}^{(m-1)}\right)^2 + h_{m+1,m}^2}} \\
r_{m+1,m}^{(m)} &= -s_m h_{m,m} + c_m h_{m+1,m} \\
&= \frac{-h_{m+1,m} r_{m,m}^{(m-1)} + r_{m,m}^{(m-1)} h_{m+1,m}}{\sqrt{\left(r_{m,m}^{(m-1)}\right)^2 + h_{m+1,m}^2}} \\
&= 0
\end{aligned}
\tag{C.15}
$$

が得られる．以上によって，回転行列 $\mathbf{\Omega}_i$ (i=1,⋯,m)の要素 c_i 及び s_i と，この回転行列により行列$\tilde{\mathbf{H}}_m$ を上三角行列に変換して得られる行列 $\mathbf{R}_m^{(m)}$の要素 $r^{(k)}_{k,j}$(j=1,⋯, m ;k=1⋯,j)が得られた．

ところで，m は一般に 1 から N まで変化する．しかし, m=l-1 のときに得られたこれらの値は m=l のときにもそのまま利用可能である．即ち，m=l のときには，回転行列 $\mathbf{\Omega}_l$ の要素 c_l 及び s_l と，行列 $\mathbf{R}_l^{(l)}$の l 列目の要素 $h^{(j)}_{,j,l}$(j,1,⋯,l)のみを計算すればよい．したがって，これらの要素を計算するアルゴリズムは, 以下のようになる.

アルゴリズム C.1　c_l,　s_l と $r^{(j)}_{,j,l+1}$ の計算

1. $Set\ r_{1,l}^{(0)} = h_{1,l}$
2. $For\ j = 1,2,....,l-1\ Do:$
3. $\quad tmp1 = c_j r_{j,l}^{(j-1)} + s_j h_{j+1,l}$
4. $\quad tmp2 = -s_j r_{j,l}^{(j-1)} + c_j h_{j+1,l}$
5. $\quad r_{j,l}^{(j)} = tmp1$
6. $\quad r_{j+1,l}^{(j)} = tmp2$
7. $End\ \ Do$
8. $c_l = r_{l,l}^{(l-1)} \Big/ \sqrt{\left(r_{l,l}^{(l-1)}\right)^2 + h_{l+1,l}^2}$
9. $s_l = h_{l+1,l} \Big/ \sqrt{\left(r_{l,l}^{(l-1)}\right)^2 + h_{l+1,l}^2}$
10. $r_{l,l}^{(l)} = c_l r_{l,l}^{(l-1)} + s_l h_{l+1,l}$
11. $\left(r_{l+1,l}^{(l)} = 0 \right)$

ここで，ステップ 3～6 に示したように，一旦計算結果をスカラ変数に保存することで, 行列$\tilde{\mathbf{H}}_m$ の要素 $h_{i,j}$ を記憶する領域と行列 $\mathbf{R}_l^{(l)}$の要素 $h^{(i)}_{,i,j}$ を記憶する領域を同じ領域とすることが可能となる.

次に，$\tilde{\mathbf{g}}_m$=$\mathbf{Q}_m \beta\, \mathbf{e}_1$ を計算する．

$$\tilde{\mathbf{g}}_m = \left[g_1^{(m)}\ g_2^{(m)} \cdots g_m^{(m)}\ g_{m+1}^{(m)}\right]^{\mathrm{T}} \tag{C.16}$$

とおけば,

$$
\tilde{\mathbf{g}}_m = \mathbf{\Omega}_m \begin{bmatrix} \tilde{\mathbf{g}}_{m-1} \\ 0 \end{bmatrix}
=
\begin{bmatrix}
1 & & & & \\
 & \ddots & & & \\
 & & 1 & & \\
 & & & c_m & s_m \\
 & & & -s_m & c_m
\end{bmatrix}
\begin{bmatrix} g_1^{(m-1)} \\ g_2^{(m-1)} \\ \vdots \\ g_m^{(m-1)} \\ 0 \end{bmatrix}
=
\begin{bmatrix} g_1^{(m-1)} \\ g_2^{(m-1)} \\ \vdots \\ c_m g_m^{(m-1)} \\ -s_m g_m^{(m-1)} \end{bmatrix}
$$

であるから，

$$
\begin{aligned}
g_j^{(m)} &= g_j^{(m-1)} \qquad for\ j = 1,\cdots,m-1 \\
g_m^{(m)} &= c_m g_m^{(m-1)} \\
g_{m+1}^{(m)} &= -s_m g_m^{(m-1)}
\end{aligned}
\tag{C.17}
$$

となる．ここで，m=1 のとき，

$$g_1^{(1)} = c_1\beta$$
$$g_2^{(1)} = -s_1\beta \tag{C.18}$$

であるから，$\tilde{\mathbf{g}}_m$ を求めるアルゴリズムは，以下のようになる．

アルゴリズム C.2 $\tilde{\mathbf{g}}_m$ の計算

1. *Set* $g_1 = \beta$
2. *For* $i = 1,2,\ldots,m$ *Do:* (GMRES iteration loop)
3. $g_{i+1} = -s_i g_i$
4. $g_i = c_i g_i$
5. *End Do*

ただし，ステップ 2 は GMRES 法の反復計算ループであり，反復計算一回ごとにステップ 3 及び 4 の計算を行えばよい．これにより，$\tilde{\mathbf{g}}_m = \mathbf{Q}_m \beta \mathbf{e}_1$ を求めることができた．ちなみに，

$$\tilde{\mathbf{g}}_1 \equiv \mathbf{\Omega}_1(\beta_1\mathbf{e}_1) = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \beta \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} c_1\beta \\ -s_1\beta \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\tilde{\mathbf{g}}_2 \equiv \mathbf{\Omega}_2\mathbf{\Omega}_1(\beta_1\mathbf{e}_1) = \begin{bmatrix} c_1\beta \\ -c_2s_1\beta \\ s_2s_1\beta \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\tilde{\mathbf{g}}_3 \equiv \mathbf{\Omega}_3\mathbf{\Omega}_2\mathbf{\Omega}_1(\beta_1\mathbf{e}_1) = \begin{bmatrix} c_1\beta \\ -c_2s_1\beta \\ c_3s_2s_1\beta \\ -s_3s_2s_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\vdots$$

であるから，$g_i^{(m)}$は以下の式で与えられる．

$$g_i^{(m)} = (-1)^{i-1}\beta\, c_i \prod_{j=1}^{i-1} s_i \qquad for\ i = 1,\cdots,m$$

$$g_{m+1}^{(m)} = (-1)^{m-1}\beta \prod_{j=1}^{i-1} s_i$$

以上のことから，関数 $J(\mathbf{y}_m)$を最小にするベクトル $\mathbf{y}_m$ は，次のようにして求めることができる．いま，m 行の列ベクトル $\mathbf{g}_m$ 及び m 行 m 列の行列 $\mathbf{R}_m$ をそれぞれ

$$\tilde{\mathbf{g}}_m = \begin{bmatrix} \mathbf{g}_m \\ g_{m+1}^{(m)} \end{bmatrix}$$

及び

$$\tilde{\mathbf{R}}_m = \begin{bmatrix} \mathbf{R}_m \\ 0 \end{bmatrix}$$

で定義する．このとき，

$$\begin{aligned} J(\mathbf{y}_m) &= \left\| \tilde{\mathbf{g}}_m - \tilde{\mathbf{R}}_m\mathbf{y}_m \right\| \\ &= \left\| \begin{bmatrix} \mathbf{g}_m \\ g_{m+1}^{(m)} \end{bmatrix} - \begin{bmatrix} \mathbf{R}_m \\ 0 \end{bmatrix} \mathbf{y}_m \right\| \\ &= \sqrt{(\mathbf{g}_m - \mathbf{R}_m\mathbf{y}_m)^2 + \left(g_{m+1}^{(m)}\right)^2} \end{aligned} \tag{C.19}$$

となる†．ここで，$(g_{m+1}{}^{(m)})^2 \geqq 0$ かつ定数であるから関数 $J(\mathbf{y}_m)$は，

$$\mathbf{g}_m - \mathbf{R}_m\mathbf{y}_m = 0 \tag{C.20}$$

のとき最小となる．よって，この連立方程式を解いてベクトル $\mathbf{y}_m$を求めればよい．行列 $\mathbf{R}_m$は上三角行列なので，この連立方程式は簡単に解ける．(C.19)式を要素表示すれば，

$$\mathbf{g}_m - \mathbf{R}_m\mathbf{y}_m = \begin{bmatrix} g_1^{(m)} \\ g_2^{(m)} \\ \vdots \\ g_m^{(m)} \end{bmatrix} - \begin{bmatrix} r_{1,1}^{(1)} & r_{1,2}^{(1)} & r_{1,3}^{(1)} & \cdots & r_{1,m-1}^{(1)} & r_{1,m}^{(1)} \\ & r_{2,2}^{(2)} & r_{2,3}^{(2)} & \cdots & r_{2,m-1}^{(2)} & r_{2,m}^{(2)} \\ & & r_{3,3}^{(3)} & \cdots & r_{3,m-1}^{(3)} & r_{3,m}^{(3)} \\ & & & \ddots & \vdots & \vdots \\ & & & & r_{m-1,m-1}^{(m-1)} & r_{m-1,m}^{(m-1)} \\ & & & & & r_{m,m}^{(m)} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

であるから，m 行目から順に計算できて，

$$y_m = g_m^{(m)} \big/ r_{m,m}^{(m)}$$
$$y_{m-1} = \left(g_{m-1}^{(m)} - r_{m-1,m}^{(m-1)} y_m\right) \Big/ r_{m-1,m-1}^{(m-1)}$$
$$y_{m-2} = \left(g_{m-2}^{(m)} - r_{m-2,m-1}^{(m-2)} y_{m-1} - r_{m-2,m}^{(m-2)} y_m\right) \Big/ r_{m-2,m-2}^{(m-2)}$$
$$\vdots$$
$$y_j = \left(g_j^{(m)} - \sum_{i=j+1}^{m} r_{j,i}^{(j)} y_i\right) \Big/ r_{j,j}^{(j)}$$
$$\vdots$$

となる．これにより，関数 $J(\mathbf{y}_m)$を最小にするベクトル $\mathbf{y}_m$ が得られた．

ところで，得られたベクトル $\mathbf{y}_m$ に対して(C.20)式より $\mathbf{g}_m$-$\mathbf{R}_m\mathbf{y}_m$=0 であるから，(C.19)式より

† ベクトル $\mathbf{x}=[x_1 \ \cdots \ x_k\, x_{k+1} \ \cdots \ x_m]^{\mathrm{T}}$，$\mathbf{x}_1=[x_1 \ \cdots \ x_k]^{\mathrm{T}}$ 及び $\mathbf{x}_2=[x_{k+1} \ \cdots \ x_m]^{\mathrm{T}}$ において，$\|\mathbf{x}\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2$ ．

$$J(\mathbf{y}_m)=\sqrt{(\mathbf{g}_m-\mathbf{R}_m\mathbf{y}_m)^2+\left(g^{(m)}_{m+1}\right)^2} \\ =\left|g^{(m)}_{m+1}\right| \tag{C.21}$$

となる．一方，(C.17)式において $\mathbf{g}^{(m)}{}_{m+1}=-s_m\mathbf{g}_m{}^{(m)}$であるから，$s_m=0$ のとき(C.21)式より $J(\mathbf{y}_m)=0$ である．このことから，残差ノルム $\|\mathbf{b}-\mathbf{A}\mathbf{x}_\mathrm{m}\|$ を評価するために，近似解 $\mathbf{x}_\mathrm{m}$ を計算する必要がないことが判る．$|s_m\mathbf{g}_m{}^{(m)}|$ が十分小さく(ε)なった場合に収束したと見做せば良い．そして，近似解 $\mathbf{x}_m$ は収束後に一回のみ計算すればよい．

これにより，関数 $J(\mathbf{y}_m)$を最小にするベクトル $\mathbf{y}_m$ を求めるアルゴリズムは，例えばアルゴリズム C.3 のようになる．

アルゴリズム C. 3　関数 $J(\mathbf{y}_m)$を最小にするベクトル $\mathbf{y}_m$

1. *Set* $g_1=\beta$
2. *For* $j=1,2,\ldots,n$ *Do:* (GMRES iteration loop)
   (*Perform Arnoldi Process*)
3. *Set* $r^{(0)}_{1,j}=h_{1,j}$
4. *For* $i=1,2,\ldots,j-1$ *Do:*
5. $tmp1=c_ir^{(i-1)}_{i,j}+s_ih_{i+1,j}$
6. $tmp2=-s_ir^{(i-1)}_{i,j}+c_ih_{i+1,j}$
7. $r^{(i)}_{i,j}=tmp1$
8. $r^{(i)}_{i+1,j}=-tmp2$
9. *End Do*
10. $c_j=r^{(j-1)}_{j,j}\Big/\sqrt{\left(r^{(j-1)}_{j,j}\right)^2+h^2_{j+1,j}}$
11. $s_j=h_{j+1,j}\Big/\sqrt{\left(r^{(j-1)}_{j,j}\right)^2+h^2_{j+1,j}}$
12. $g_{j+1}=-s_jg_j$
13. $g_j=c_jg_j$
14. $r^{(j)}_{j,j}=c_jr^{(j-1)}_{j,j}+s_jh_{j+1,ji}$
15. $\left(r^{(j)}_{j+1,j}=0\right)$
16. *if* $\left|s_jg_j\right|<\varepsilon$ *then set* $m=j$ *and Go To step* 19
17. *End Do*
18. *Set* $m=n$
19. *For* $j=m,m-1,\ldots,1$ *Do:*
20. $y_j=g^{(j)}_j$
21. *For* $i=j+1,j+2,\ldots,m$ *Do:*
22. $y_j=y_j-r^{(j)}_{j,i}y_i$
23. *End Do*
24. $y_j=y_j\big/r^{(j)}_{j,j}$
25. *End Do*

APPENDIX D DQGMRES 法における近似解

ここでは，DQGMRES 法において近似解 $\mathbf{x}_m$ を求める漸化式(2.8)式

$$\mathbf{x}_m=\mathbf{x}_{m-1}+g^{(m)}_m\mathbf{p}_m \tag{D.1}$$

の導出を行う．

GMRES（DQGMRES）法の近似解 $\mathbf{x}_m$ は，(2.5)式

$$\mathbf{x}_m=\mathbf{x}_0+\mathbf{V}_m\mathbf{y}_m \tag{D.2}$$

によって与えられる．ベクトル $\mathbf{y}_m$ は，(C.20)式より連立方程式

$$\mathbf{R}_m\mathbf{y}_m=\mathbf{g}_m$$

を解いて得るのであるから，

$$\mathbf{y}_m=\mathbf{R}^{-1}_m\mathbf{g}_m \tag{D.3}$$

(D.3)式を(D.2)式に代入して，

$$\mathbf{x}_m=\mathbf{x}_0+\mathbf{V}_m\mathbf{R}^{-1}_m\mathbf{g}_m$$

$\mathbf{P}_m\equiv\mathbf{V}_m\mathbf{R}_m{}^{-1}$ とおけば，

$$\mathbf{x}_m=\mathbf{x}_0+\mathbf{P}_m\mathbf{g}_m \tag{D.4}$$

ここで，(C.17)式より

$$\mathbf{g}_m=\begin{bmatrix}\mathbf{g}_{m-1}\\ g^{(m)}_m\end{bmatrix}$$

であるから，これを(D.4)式に代入して，

$$\mathbf{x}_m=\mathbf{x}_0+\mathbf{P}_{m-1}\mathbf{g}_{m-1}+g^{(m)}_m\mathbf{p}_m \\ =\mathbf{x}_{m-1}+g^{(m)}_m\mathbf{p}_m \tag{D.5}$$

これにより，(D.1)式，即ち(2.8)式が得られた．

ところで，行列 $\mathbf{P}_m$ の定義より $\mathbf{P}_m\mathbf{R}_m=\mathbf{V}_m$ であるから DQGMRES 法の場合，

$$[\mathbf{p}_1\cdots\mathbf{p}_m]\begin{pmatrix} r^{(1)}_{1,1} & \cdots & r^{(1)}_{1,k} & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & r^{(2)}_{2,k+1} & \cdots & \vdots \\ & & r^{(k)}_{k,k} & \vdots & \cdots & 0 \\ & & 0 & \ddots & \cdots & r^{(m-k+1)}_{m-k+1,m} \\ & & & & \ddots & \vdots \\ & & & & & r^{(m)}_{m,m} \end{pmatrix} \\ =[\mathbf{v}_1\cdots\mathbf{v}_m]$$

両辺の一番最後の列（m 列目）を取り出せば,

$$[\mathbf{p}_1 \cdots \mathbf{p}_m] \begin{pmatrix} 0 \\ \vdots \\ 0 \\ r_{m-k+1,m}^{(m-k+1)} \\ \vdots \\ r_{m,m}^{(m)} \end{pmatrix} = \mathbf{v}_m$$

より,

$$\sum_{i=m-k+1}^{m} r_{i,m}^{(i)} \mathbf{p}_i = \mathbf{v}_m$$

よって，ベクトル $\mathbf{p}_m$ は,

$$\mathbf{p}_m = \frac{1}{r_{m,m}^{(m)}} \left(\mathbf{v}_m - \sum_{i=m-k+1}^{m-1} r_{i,m}^{(i)} \mathbf{p}_i \right) \tag{D.6}$$

となる．このとき，$\mathbf{p}_1$ は以下のようにして計算すればよい．まず，m=1 の場合の近似解 $\mathbf{x}_1$ は,

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v}_1 \mathbf{y}_1 \tag{D.7}$$

によって与えられる．ベクトル $\mathbf{y}_1$ は,

$$\mathbf{y}_1 = \mathbf{R}_1^{-1} \mathbf{g}_1 \tag{D.8}$$

によって計算され，$\mathbf{R}_1$ 及び $\mathbf{g}_1$ はそれぞれ

$$\begin{aligned} \tilde{\mathbf{R}}_1 &= \mathbf{\Omega}_1 \tilde{\mathbf{H}}_1 \\ &= \begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} \\ &= \begin{pmatrix} r_{11}^{(1)} \\ 0 \end{pmatrix} \\ \tilde{\mathbf{g}}_1 &= \begin{pmatrix} g_1^{(1)} \\ g_2^{(1)} \end{pmatrix} \end{aligned}$$

と定義より,

$$\begin{aligned} \mathbf{R}_1 &= r_{11}^{(1)} \\ \mathbf{g}_1 &= g_{11}^{(1)} \end{aligned} \tag{D.9}$$

である．結局，(D.7)，(D.8)及び(D.9)式より m=1 の場合の近似解 $\mathbf{x}_1$ は,

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v}_1 g_{11}^{(1)} / r_{11}^{(1)} \tag{D.10}$$

となる．これと(D.1)式の比較より $\mathbf{p}_1 = \mathbf{v}_1 / r_{11}{}^{(1)}$ とすればよいことが判る．

これらのことから，(D.5)式を書き直せば,

$$\begin{aligned} \mathbf{x}_m &= \mathbf{x}_{m-1} + \frac{1}{r_{m,m}^{(m)}} g_m^{(m)} \left(\mathbf{v}_m - \sum_{i=m-k+1}^{m-1} r_{i,m}^{(i)} \mathbf{p}_i \right) \\ \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{v}_1 g_{11}^{(1)} / r_{11}^{(1)} \end{aligned} \tag{D.11}$$

となり，(D.11)式は不完全直交過程によって計算された直前 $k-1$ 本の基底ベクトル $\mathbf{p}_i$ ，$i=m-k+1,\cdots,m-1$ を保存しておくことにより計算できる．

左前処理を行った方程式に DQGMRES 法を適用する場合は，GMRES 法の場合同様，通常の残差に代えて，前処理系での残差 $\mathbf{r}_0=\mathbf{M}^{-1}(\mathbf{b}-\mathbf{A}\mathbf{x}_0)$を求め，（直交するとは限らない）基底ベクトル $\mathbf{w}_j$ を $\mathbf{w}_j=\mathbf{M}^{-1}\mathbf{A}\mathbf{v}_j$ として求めればよい．また，右前処理の場合には，基底ベクトル $\mathbf{w}_j$ を $\mathbf{w}_j=\mathbf{A}\mathbf{M}^{-1}\mathbf{v}_j$ によって求め，近似解 $\mathbf{x}_m$ を漸化式(D.11)式に代わって,

$$\begin{aligned} \mathbf{x}_m &= \mathbf{x}_{m-1} + \frac{1}{r_{m,m}^{(m)}} g_m^{(m)} \left(\mathbf{M}^{-1} \mathbf{v}_m - \sum_{i=m-k+1}^{m-1} r_{i,m}^{(i)} \mathbf{p}_i \right) \\ \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{M}^{-1} \mathbf{v}_1 g_{11}^{(1)} / r_{11}^{(1)} \end{aligned} \tag{D.12}$$

によって求めればよい．右前処理付き GMRES 法における近似解 $\mathbf{x}_m$ は，(2.5)式の代わりに(2.18)式により求められるのであるから，(2.18)式において $\mathbf{Z}_m=\mathbf{M}^{-1}\mathbf{V}_m$，即ち，$\mathbf{Z}_m=[\mathbf{z}_1,\ldots\mathbf{z}_m]^{\mathrm{T}}=[\mathbf{M}^{-1}\mathbf{v}_1,\ldots,\mathbf{M}^{-1}\mathbf{v}_m]^{\mathrm{T}}$ 等とおくことにより，前処理なしの DQGMRES 法の場合と全く同様に(D.12)式を導くことが可能である．前処理行列の逆行列と基底ベクトルの積 $\mathbf{M}^{-1}\mathbf{v}_j$ は，基底ベクトル $\mathbf{w}_j$ を計算するときに保存しておけば，一回の反復計算の中で二度計算する必要はない．

APPENDIX E 前処理としての反復法

ここでは，GMRES 法の前処理として，連立方程式の反復解法を適用する方法について示す．

良く知られているように，連立方程式の数値解法には，大きく分けて直接法（direct method）と反復法（iterative method）がある．反復法は，さらに定常反復法（stationary iterative method）と非定常反復法（non-stationary iterative method）に分けることができる．定常反復法は，与えられた連立一次方程式を不動点（fixed-point）方程式†

$$\mathbf{x} = f(\mathbf{x}) \tag{E.1}$$

に変形し，適当な初期値 $\mathbf{x}_0$ のもとに漸化式

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) \tag{E.2}$$

によって，近似解 $\mathbf{x}_{k+1}$ を逐次的に求めていく方法である．一方，非定常反復法は反復ごとに異なる修正を行うものである．その代表的な方法である Krylov 部分空間法では，反復ごとに異なる修正方向ベクトル $\mathbf{p}$ 及び修正量 α を求めて，近似解 $\mathbf{x}_{k+1}$ を

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{E.3}$$

として得る．ここでは，定常反復法に限って，それを前処理として利用する方法について示す．

† 関数 f によって $\mathbf{x}$ の値が変化しないことからこの名前がある．

いま，与えられた連立方程式

$$\mathbf{Ax} = \mathbf{b}$$

において，係数行列 $\mathbf{A}$ を適当な行列 $\mathbf{M}$ 及び $\mathbf{N}$ の差

$$\mathbf{A} = \mathbf{M} - \mathbf{N} \tag{E.4}$$

で表すことにすれば，定常反復法の一般形

$$\mathbf{x}_{k+1} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x}_k + \mathbf{M}^{-1}\mathbf{b} \tag{E.5}$$

が得られる．ところで，(E.5)式は，

$$\mathbf{x}_{k+1} = \mathbf{G}\mathbf{x}_k + \mathbf{f} \tag{E.6}$$

と書き直すことができる．ただし，

$$\mathbf{f} = \mathbf{M}^{-1}\mathbf{b} \tag{E.7}$$

であり，

$$\begin{aligned}\mathbf{G} &= \mathbf{M}^{-1}\mathbf{N} \\ &= \mathbf{M}^{-1}(\mathbf{M} - \mathbf{A}) \\ &= \mathbf{I} - \mathbf{M}^{-1}\mathbf{A}\end{aligned} \tag{E.8}$$

である．反復計算が収束すれば $\mathbf{x}_{k+1}$=$\mathbf{x}_k$ であるから，(E.6)式の反復計算は，連立方程式

$$(\mathbf{I} - \mathbf{G})\mathbf{x} = \mathbf{f} \tag{E.9}$$

を解いていることになる．これに，(E.7)式と(E.8)式を代入すれば，

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \tag{E.10}$$

となるので，一般に反復法は，その収束を前提にした時，前処理された連立方程式(2.12)を解いていることと等価となり，行列 $\mathbf{M}$ が前処理行列であることが示された．

以下に，主な反復法における行列 $\mathbf{M}$ と $\mathbf{N}$ の具体的な形を示す．いま，係数行列 $\mathbf{A}$ の対角行列，下三角行列及び上三角行列を $\mathbf{D}$=$[d_i]$,$\mathbf{L}$=$[l_{i,j}]$，$i<j$ 及び $\mathbf{U}$= $[u_{i,j}]$，$i>j$ とし

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U} \tag{E.13}$$

であるとする．さらに，$\mathbf{x}_k$=$[x_1^{(k)}, x_2^{(k)}, \cdots, x_N^{(k)}]^{\mathrm{T}}$ とする．

(1) Jacobi 法

Jacobi 法は，

$$x_i^{(k+1)} = \left(b_i - \sum_{j \neq i}^{i-1} l_{i,j} x_j^{(k)} - \sum_{j=i+1}^{N} u_{i,j} x_j^{(k)}\right) \Big/ d_i \tag{E11}$$

として与えられる．これを，行列を使って表せば，

$$\mathbf{D}\mathbf{x}_{k+1} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}_k \tag{E.12}$$

であるから，反復法の一般形として

$$\begin{aligned}\mathbf{M}_{JA}\mathbf{x}_{k+1} &= \mathbf{N}_{JA}\mathbf{x}_k + \mathbf{b} \\ \mathbf{M}_{JA} &= \mathbf{D} \\ \mathbf{N}_{JA} &= -\mathbf{L} - \mathbf{U}\end{aligned} \tag{E.13}$$

を得る．

(2) Gauss-Seidel 法

Gauss-Seidel 法は，

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} l_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^{N} u_{i,j} x_j^{(k)}\right) \Big/ d_i \tag{E.14}$$

として与えられる．このとき，i は 1 から N まで正順に計算される．これを，行列を使って表せば，

$$\mathbf{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{L}\mathbf{x}_{k+1} - \mathbf{U}\mathbf{x}_k) \tag{E.15}$$

であるから，反復法の一般形として

$$\begin{aligned}\mathbf{M}_{GSf}\mathbf{x}_{k+1} &= \mathbf{N}_{GSf}\mathbf{x}_k + \mathbf{b} \\ \mathbf{M}_{GSf} &= \mathbf{D} + \mathbf{L} \\ \mathbf{N}_{GSf} &= -\mathbf{U}\end{aligned} \tag{E.16}$$

を得る．

Gauss-Seidel 法においては，計算順序を逆にすることも可能である．即ち，一般項を

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} l_{i,j} x_j^{(k)} - \sum_{j=i+1}^{N} u_{i,j} x_j^{(k+1)}\right) \Big/ d_i \tag{E.17}$$

として，i を N から 1 まで逆順に計算する．これを，行列を使って表せば，

$$\mathbf{x}_{k+1} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{L}\mathbf{x}_k - \mathbf{U}\mathbf{x}_{k+1}) \tag{E.18}$$

であるから，反復法の一般形として

$$\begin{aligned}\mathbf{M}_{GSb}\mathbf{x}_{k+1} &= \mathbf{N}_{GSb}\mathbf{x}_k + \mathbf{b} \\ \mathbf{M}_{GSb} &= \mathbf{D} + \mathbf{U} \\ \mathbf{N}_{GSb} &- \mathbf{L}\end{aligned} \tag{E.19}$$

が得られる．

(3) Symmetric Gauss-Seidel（SGS）法

Symmetric Gauss-Seidel 法は，正順（forward）と逆順（backward）の Gauss-Seidel 法を組み合わせることにより得られる．即ち，

$$\begin{cases}\mathbf{M}_{GSf}\mathbf{x}_{k+1/2} = \mathbf{N}_{GSf}\mathbf{x}_k + \mathbf{b} \\ \mathbf{M}_{GSb}\mathbf{x}_{k+1} = \mathbf{N}_{GSb}\mathbf{x}_{k+1/2} + \mathbf{b}\end{cases} \tag{E.20}$$

である．(E.20)第一式より

$$\mathbf{x}_{k+1/2} = \mathbf{M}_{GSf}^{-1}\mathbf{N}_{GSf}\mathbf{x}_k + \mathbf{M}_{GSf}^{-1}\mathbf{b} \tag{E.21}$$

これを(E.20)第二式に代入すれば，

$$\begin{aligned}&\mathbf{M}_{GSb}\mathbf{x}_{k+1} \\ &\quad = \mathbf{N}_{GSb}\left(\mathbf{M}_{GSf}^{-1}\mathbf{N}_{GSf}\mathbf{x}_k + \mathbf{M}_{GSf}^{-1}\mathbf{b}\right) + \mathbf{b} \\ &\quad = \mathbf{M}_{GSb}^{-1}\mathbf{N}_{GSb}\mathbf{M}_{GSf}^{-1}\mathbf{N}_{GSf}\mathbf{x}_k \\ &\qquad\qquad + \mathbf{M}_{GSb}^{-1}\left(\mathbf{N}_{GSb}\mathbf{M}_{GSf}^{-1} + \mathbf{I}\right)\mathbf{b}\end{aligned} \tag{E.22}$$

ここで，

$$\begin{aligned}\mathbf{N}_{GSb}\mathbf{M}_{GSf}^{-1}+\mathbf{I}&=-\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}+\mathbf{I}\\&=-\mathbf{D}(\mathbf{D}+\mathbf{L})^{-1}-\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}\\&\qquad+\mathbf{I}+\mathbf{D}(\mathbf{D}+\mathbf{L})^{-1}\\&=\mathbf{D}(\mathbf{D}+\mathbf{L})^{-1}\end{aligned}\tag{E.23}$$

であるから，(E.23)式は，

$$\begin{aligned}\mathbf{x}_{k+1}&=(\mathbf{D}+\mathbf{U})^{-1}\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\mathbf{x}_k\\&\qquad+(\mathbf{D}+\mathbf{U})^{-1}\mathbf{D}(\mathbf{D}+\mathbf{L})^{-1}\mathbf{b}\end{aligned}\tag{E.24}$$

となる．この両辺に右辺第二項ベクトル $\mathbf{b}$ の係数行列の逆行列をかければ，

$$\begin{aligned}&(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})\mathbf{x}_{k+1}\\&\quad=(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})(\mathbf{D}+\mathbf{U})^{-1}\\&\qquad\times\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\mathbf{x}_k+\mathbf{b}\\&\quad=(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\mathbf{x}_k+\mathbf{b}\end{aligned}\tag{E.25}$$

を得る．ここで，右辺第一項 $\mathbf{x}_k$ の係数行列を整理すれば，

$$\begin{aligned}&(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}\mathbf{L}(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\\&\quad=(\mathbf{L}+\mathbf{L}\mathbf{D}^{-1}\mathbf{L})(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\\&\quad=\mathbf{L}\mathbf{D}^{-1}(\mathbf{D}+\mathbf{L})(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\\&\quad=\mathbf{L}\mathbf{D}^{-1}\mathbf{U}\end{aligned}\tag{E.26}$$

であるから，結局，Symmetric Gauss-Seidel 法の一般形は，

$$\begin{aligned}&\mathbf{M}_{SGS}\mathbf{x}_{k+1}=\mathbf{N}_{SGS}\mathbf{x}_k+\mathbf{b}\\&\quad\mathbf{M}_{SGS}=(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})\\&\quad\mathbf{N}_{SGS}=\mathbf{L}\mathbf{D}^{-1}\mathbf{U}\end{aligned}\tag{E.27}$$

となる．

(4) Successive Over-Relaxation（SOR）法

SOR 法における近似解 $\mathbf{x}_{k+1}$ は，加速パラメータ ω を使って，

$$\begin{aligned}x_i^{(k+1)}&=\frac{\omega}{d_i}\left(b_i-\sum_{j=1}^{i-1}l_{i,j}x_i^{(k+1)}-\sum_{j=i+1}^{n}u_{i,j}x_i^{(k)}\right)\\&\qquad+(1-\omega)x_i^{(k)}\end{aligned}\tag{E.28}$$

と与えられる．これは，Gauss-Seidel 法による近似解 $\mathbf{x}_{k+1}$ を与える右辺第一項と反復一回前の近似解 $\mathbf{x}_k$ である右辺第二項の ω を重みとした線形結合となっている．加速パラメータ ω が $0<\omega<2$ のとき，この反復法は収束し，$1<\omega<2$ の適当な値をとるとき，収束が加速されることが知られている†．このとき，i は 1 から N まで正順に計算される．これを，行列を使って表せば，

$$\mathbf{x}_{k+1}=\omega\mathbf{D}^{-1}(\mathbf{b}-\mathbf{L}\mathbf{x}_{k+1}-\mathbf{U}\mathbf{x}_k)+(1-\omega)\mathbf{x}_k\tag{E.29}$$

であるから，$\mathbf{x}_{k+1}$ を含む項を左辺に移項し整理すれば，

$$\begin{aligned}\frac{1}{\omega}(\mathbf{D}+\omega\mathbf{L})\mathbf{x}_{k+1}&=\mathbf{b}-\mathbf{U}\mathbf{x}_k+\frac{1}{\omega}(1-\omega)\mathbf{D}\mathbf{x}_k\\&=\mathbf{b}+\frac{1}{\omega}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]\mathbf{x}_k\end{aligned}\tag{E.30}$$

したがって，反復法の一般形として

$$\begin{aligned}&\mathbf{M}_{SORf}\mathbf{x}_{k+1}=\mathbf{N}_{SORf}\mathbf{x}_k+\mathbf{b}\\&\quad\mathbf{M}_{SORf}=\frac{1}{\omega}(\mathbf{D}+\omega\mathbf{L})\\&\quad\mathbf{N}_{SORf}=\frac{1}{\omega}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]\end{aligned}\tag{E.31}$$

を得る．

Gauss-Seidel 法の場合同様，SOR 法においても，計算順序を逆にすることが可能である．即ち，一般項を

$$\begin{aligned}x_i^{(k+1)}&=\frac{\omega}{d_i}\left(b_i-\sum_{j=1}^{i-1}l_{i,j}x_i^{(k)}-\sum_{j=i+1}^{n}u_{i,j}x_i^{(k+1)}\right)\\&\qquad+(1-\omega)x_i^{(k)}\end{aligned}\tag{E.32}$$

として，i を N から 1 まで逆順に計算する．この場合，反復法の一般形は，(E.31)式の行列 $\mathbf{L}$ と $\mathbf{U}$ を入れ替えることによって容易に求まり，

$$\begin{aligned}&\mathbf{M}_{SORb}\mathbf{x}_{k+1}=\mathbf{N}_{SORb}\mathbf{x}_k+\mathbf{b}\\&\quad\mathbf{M}_{SORb}=\frac{1}{\omega}(\mathbf{D}+\omega\mathbf{U})\\&\quad\mathbf{N}_{SORb}=\frac{1}{\omega}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]\end{aligned}\tag{E.33}$$

となる．

(5) Symmetric Successive Over-Relaxation（SSOR）法

SSOR 法は，正順と逆順の SOR 法を組み合わせることにより得られる．即ち，

$$\begin{aligned}&\mathbf{M}_{SORf}\mathbf{x}_{k+1/2}=\mathbf{N}_{SORf}\mathbf{x}_k+\mathbf{b}\\&\mathbf{M}_{SORb}\mathbf{x}_{k+1}=\mathbf{N}_{SORb}\mathbf{x}_{k+1/2}+\mathbf{b}\end{aligned}\tag{E.34}$$

である．(E.34)第一式より

$$\mathbf{x}_{k+1/2}=\mathbf{M}_{SORf}^{-1}\mathbf{N}_{SORf}\mathbf{x}_k+\mathbf{M}_{SORf}^{-1}\mathbf{b}\qquad\text{E.35)}$$

これを(E.34)第二式に代入すれば，

$$\begin{aligned}&\mathbf{M}_{SORb}\mathbf{x}_{k+1}\\&\quad=\mathbf{N}_{SORb}\left(\mathbf{M}_{SORf}^{-1}\mathbf{N}_{SORf}\mathbf{x}_k+\mathbf{M}_{SORf}^{-1}\mathbf{b}\right)+\mathbf{b}\end{aligned}\tag{E.36}$$

両辺に $\mathbf{M}^{-1}{}_{SORb}$ をかけて整理すれば，

$$\begin{aligned}&\mathbf{x}_{k+1}\\&\quad=\mathbf{M}_{SORb}^{-1}\mathbf{N}_{SORb}\mathbf{M}_{SORf}^{-1}\mathbf{N}_{SORf}\mathbf{x}_k\\&\qquad+\mathbf{M}_{SORb}^{-1}\left(\mathbf{N}_{SORb}\mathbf{M}_{SORf}^{-1}+\mathbf{I}\right)\mathbf{b}\end{aligned}\tag{E.37}$$

ここで，右辺 $\mathbf{x}_k$ の係数行列は，

† 実際には，$0<\omega<1$ のとき，収束が加速される場合も存在する．

$$\mathbf{M}_{SORb}^{-1}\mathbf{N}_{SORb}\mathbf{M}_{SORf}^{-1}\mathbf{N}_{SORf}$$
$$=\omega(\mathbf{D}+\omega\mathbf{U})^{-1}\frac{1}{\omega}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]$$
$$\times\omega(\mathbf{D}+\omega\mathbf{L})^{-1}\frac{1}{\omega}[(1-\omega)\mathbf{D}-\omega\mathbf{U}] \quad \text{(E.38)}$$
$$=(\mathbf{D}+\omega\mathbf{U})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]$$
$$\times(\mathbf{D}+\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$

であり，同様に定数ベクトル $\mathbf{b}$ の係数行列は，

$$\mathbf{M}_{SORb}^{-1}\left(\mathbf{N}_{SORb}\mathbf{M}_{SORf}^{-1}+\mathbf{I}\right)$$
$$=\omega(\mathbf{D}+\omega\mathbf{U})^{-1}$$
$$\times\left\{[(1-\omega)\mathbf{D}-\omega\mathbf{L}](\mathbf{D}+\omega\mathbf{L})^{-1}+\mathbf{I}\right\} \quad \text{(E.39)}$$
$$=\omega(\mathbf{D}+\omega\mathbf{U})^{-1}$$
$$\times\left\{[(2-\omega)\mathbf{D}-(\mathbf{D}+\omega\mathbf{L})](\mathbf{D}+\omega\mathbf{L})^{-1}+\mathbf{I}\right\}$$
$$=\omega(2-\omega)(\mathbf{D}+\omega\mathbf{U})^{-1}\mathbf{D}(\mathbf{D}+\omega\mathbf{L})^{-1}$$

であるから，

$$\mathbf{x}_{k+1}=(\mathbf{D}+\omega\mathbf{U})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]$$
$$\times(\mathbf{D}+\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]\mathbf{x}_k \quad \text{(E.40)}$$
$$+\omega(2-\omega)(\mathbf{D}+\omega\mathbf{U})^{-1}\mathbf{D}(\mathbf{D}+\omega\mathbf{L})^{-1}\mathbf{b}$$

を得る．この両辺に右辺第二項，ベクトル $\mathbf{b}$ の係数行列の逆行列をかければ，

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U})\mathbf{x}_{k+1}$$
$$=\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U}) \quad \text{(E.41)}$$
$$\times(\mathbf{D}+\omega\mathbf{U})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]$$
$$\times(\mathbf{D}+\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]\mathbf{x}_k+\mathbf{b}$$

となる．右辺第一項 $\mathbf{x}_k$ の係数行列を整理すれば，

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U})(\mathbf{D}+\omega\mathbf{U})^{-1}$$
$$\times[(1-\omega)\mathbf{D}-\omega\mathbf{L}](\mathbf{D}+\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$
$$=\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}$$
$$\times[(2-\omega)\mathbf{D}-\mathbf{D}-\omega\mathbf{L}](\mathbf{D}+\omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}] \quad \text{(E.42)}$$
$$=\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}$$
$$\times\left[(2-\omega)\mathbf{D}(\mathbf{D}+\omega\mathbf{L})^{-1}-\mathbf{I}\right][(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$
$$=\frac{1}{\omega(2-\omega)}\left[(2-\omega)\mathbf{I}-\mathbf{I}-\omega\mathbf{L}\mathbf{D}^{-1}\right][(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$
$$=\frac{1}{\omega(2-\omega)}[(1-\omega)\mathbf{D}-\omega\mathbf{L}]\mathbf{D}^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$

であるから，

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U})\mathbf{x}_{k+1}$$
$$=\frac{1}{\omega(2-\omega)}[(1-\omega)\mathbf{D}-\omega\mathbf{L}] \quad \text{(E.43)}$$
$$\times\mathbf{D}^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]\mathbf{x}_k+\mathbf{b}$$

が得られる．よって，SSOR 法の一般形は，

$$\mathbf{M}_{SSOR}\mathbf{x}_{k+1}=\mathbf{N}_{SSOR}\mathbf{x}_k+\mathbf{b}$$
$$\mathbf{M}_{SSOR}=\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U})$$
$$\mathbf{N}_{SSOR}=\frac{1}{\omega(2-\omega)}[(1-\omega)\mathbf{D}-\omega\mathbf{L}] \quad \text{(E.44)}$$
$$\times\mathbf{D}^{-1}[(1-\omega)\mathbf{D}-\omega\mathbf{U}]$$

となる．ちなみに，前処理行列として $\mathbf{M}_{SSOR}$ を使用した場合，例えば，前処理系の連立方程式(2.13)式 $\mathbf{M}\mathbf{w}_j=\tilde{\mathbf{v}}_j$ を解くには，まず，

$$\tilde{\mathbf{w}}=\mathbf{D}^{-1}(\mathbf{D}+\omega\mathbf{U})\mathbf{w} \quad \text{(E.45)}$$

とおく．すると与えられた連立方程式は，

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}+\omega\mathbf{L})\tilde{\mathbf{w}}=\tilde{\mathbf{v}} \quad \text{(E.46)}$$

となるので，LU-SGS 法の場合同様，

$$step\ 1:\tilde{\mathbf{w}}=\omega(2-\omega)\mathbf{D}^{-1}[\tilde{\mathbf{v}}-\omega\mathbf{L}\tilde{\mathbf{w}}]$$
$$step\ 2:\mathbf{w}=\mathbf{D}^{-1}(\mathbf{D}\tilde{\mathbf{w}}-\omega\mathbf{U}\mathbf{w}) \quad \text{(E.47)}$$
$$=\tilde{\mathbf{w}}-\omega\mathbf{D}^{-1}\mathbf{U}\mathbf{w}$$

と二段階の計算をすればよい．これは，勿論，$\mathbf{N}_{SSOR}=\mathbf{0}$ とした時の SSOR 法そのものである．(E.47)式を要素で表せば，

$$step\ 1:\tilde{w}_i=\omega(2-\omega)\frac{1}{d_i}\left[\tilde{v}_i-\omega\sum_{j=1}^{i-1}l_{i,j}\tilde{w}_j\right]$$
$$i=1,\cdots,N \quad \text{(E.52)}$$
$$step\ 2:w=\tilde{w}_i-\omega\frac{1}{d_i}\sum_{j=i+1}^{N}u_{i,j}w_j$$
$$i=N,\cdots,1$$

となる．

以上に示した反復法のうち，Gauss-Seidel 法と SOR 法では，仮に係数行列 $\mathbf{A}$ が対称であった場合でも，その対称性が保存されないが，SGS 法と SSOR 法では保存されることに注意されたい．このことは，それぞれの解法の一般形を見れば明らかである．

ところで，SGS 法の一般形(E.27)式と近似因数分解法における因数分解の式(2.40)式の比較より，LU-SGS 法は SGS 法において行列 $\mathbf{N}=\mathbf{0}$ とみなせる特別な場合であることがわかる．これは，既に見たように時間発展の Navier-Stokes 方程式において, 時間ステップ Δt が十分に小さい時に, 行列 $\mathbf{N}$ が二次のオーダーとなり無視できることによるものであり，一般には無視できるものではない．

また，(E.10)式は，行列 $\mathbf{N}$ が無視できるか否かに関わらず導くことができる．したがって，SGS 法の特別な場合である LU-SGS 法も前処理として利用可能であり，こ

の時，前処理行列 $\mathbf{M}_{LU\text{-}SGS}$ は，当然，$\mathbf{M}_{LU\text{-}SGS}=\mathbf{M}_{SGS}$ である．

参考文献

[1] Saad, Y. and Schultz M. H., "Generalized Minimal Residual Algorithm for Solving NonsymmetricLinear Systems", SIAM J. SCI. STAT. COMUT., Vol. 7, No.3, pp 856-869, 1986.

[2] Iwamiya, T., "NAL SST Project and Aerodynamic Design of Experimental Aircraft", Proceedings of the 4th ECCOMAS Computational Fluid Dynamics Conference, Wiley, Chichester, England, U.K., pp. 580-585, 1998.

[3] 高木正平, 坂田公夫 他., "［特集］超音速実験機計画について", 日本流体力学会誌ながれ 18-5, pp. 275-307, 1999.

[4] 藤田健, 松島紀左, 中橋和博., "非構造格子 CFD を用いた逆問題設計システムの高度化", 第 15 回数値流体力学シンポジウム予稿集 D05-3, 2001.

[5] 高橋克倫, 藤田健 他., "NAL 小型超音速実験機 NEXST-1 の結合分離金具形状修正の CFD 解析", 第 17 回数値流体力学シンポジウム予稿集 F2-3, 2003.

[6] "小型超音速実験機（NEXST-1）の舵角変化時における空力特性変化の数値解析", http://www.ista.jaxa.jp/res/c02/a06_01.html

[7] Saad, Y., "Iterative Methods for Space Linear Systems", 2nd edition, SIAM, 2000.

[8] Obayashi, S. and Guruswamy, G. P., "Convergence Acceleration of a Navier-Stokes Solver for Efficient Static Aeroelastic Computation", AIAA Journal, Vol. 33, No. 6, pp. 1134-1141, 1995.

[9] Jameson, A. and Turkel, E., "Implicit Scheme and LU Decompositions", Math. Comp., Vol. 37, No. 156, pp. 385-397, 1981.

[10] Men'shov, I. and Nakamura, Y., "Implementation of LU-SGS Method for an Arbitrary Finite Volume Descretization", Proc. of Japanese 9th CFD Symposium, pp. 123-124, 1995.

[11] Sharov, D. and Nakahashi, K., "Reordering of 3-D Hybrid Unstructured Grids for Vectorized LU-SGS Navier-Stokes Computations", AIAA 97-2102, 1997.

[12] "並列型の非構造格子ソルバープログラムユーザーズマニュアル" 平成 15 年 2 月 28 日財団法人青葉工業振興会.

宇宙航空研究開発機構研究開発報告　JAXA-RR-07-008

発　　行　平成 20 年 2 月 29 日
編集・発行　宇宙航空研究開発機構
〒182-8522 東京都調布市深大寺東町 7-44-1
URL：http://www.jaxa.jp/
印刷・製本　（株）東京プレス

本書及び内容についてのお問い合わせは、下記にお願いいたします。
宇宙航空研究開発機構 情報システム部 研究開発情報センター
〒305-8505 茨城県つくば市千現 2-1-1
TEL：029-868-2079 FAX：029-868-2956

宇宙航空研究開発機構
Japan Aerospace Exploration Agency