

## 非対称行列系の反復解法について

野寺 隆

(慶應義塾大学理工学部)

### 1. はじめに

近年、高次元動的非線形現象の数値解法で生じる大規模な連立1次方程式:

$$Ax = b, \quad (1.1)$$

を解くための PCG法 (Preconditioned Conjugate Gradient Method) は、正定値対称行列問題に対して、かなり効果的に働く良い方法であることが報告されてきている。しかし、非対称行列問題に対しては、一般に次のような正規方程式:

$$A^T Ax = A^T b, \quad (1.2)$$

または

$$\begin{aligned} AA^T u &= b, & (1.3a) \\ x &= A^T u, & (1.3b) \end{aligned}$$

のどちらか一方を構成して解かねばならないので、対称問題と比較して、その収束性に多少の難点がある。

大型の非対称行列系の連立1次方程式は、通常、非自己随伴楕円型偏微分方程式の境界値問題を離散近似 (例えば、FEM や FDM による) するとき生ずる場合が多い。以下、そのような非対称行列問題に対して CG法の系統を受け継ぐいくつかの反復解法 (特に、GCR法と Lanczos法に基づく算法) について最近の話題を追いつながりその地理を明らかにする。

GCR法 (Generalized Conjugate Residual Method) は、言うなれば降下法の系統の数値解法であり、チェビシェフ準反復法のようにパラメータの決定を行う必要もなく、その収束性も、係数行列  $A^T A$  のスペクトルではなく対称行列と同様に行列  $A$  のスペクトルに依存する。また、これらの方法と行列の preconditioning を併用すれば、その収束性を加速向上することもできる。

### 2. GCR法 (Generalized Conjugate Residual Method)

連立1次方程式  $Ax = b$  を解くことを考える。ただし、係数行列  $A$  は  $n \times n$  の正定値対称な部分を持つ非対称行列であるものとする。まず最初に、Eisenstat et al [5] による GCR法のバリエーションについて述べることにする。

#### 2.1 GCR法の基本算法

ここでは、4つの反復解法について述べることにするが、これらの方法はすべて次の基本形式で表される。

##### (基本算法)

(1) 初期値  $x_1$  を選び、残差ベクトル

$$r_1 = b - Ax_1,$$

を計算し、方向ベクトルを  $p_1 = r_1$  と置く。

(2) 次の手順を繰り返す。(  $k=1, 2, 3, \dots$  )

$$x_{k+1} = x_k + a_k p_k, \quad (2.1a)$$

$$r_{k+1} = r_k - a_k Ap_k, \quad (2.1b)$$

$$a_k = \frac{(r_k, Ap_k)}{(Ap_k, Ap_k)}, \quad (2.1c)$$

$$p_{k+1} \text{ を計算する。} \quad (2.1d)$$

この基本算法において、スカラー  $a_k$  は、 $\|r_{k+1}\|_2 = \|b - A(x_k + a_k p_k)\|_2$  を最小化するように選択することになり、残差の Euclid ノルムは、各反復で減少する。

方向ベクトル  $p_{k+1}$  の様々な選択により、次の4つの方法が考えられる。

#### [I] GCR法 (Generalized Conjugate Residual Method)

$$p_{k+1} = r_{k+1} + \sum_{j=1}^k b_{j,k} p_j, \quad (2.2a)$$

ただし、

$$b_{j,k} = - \frac{(Ar_{k+1}, Ap_j)}{(Ap_j, Ap_j)}, \quad (j \leq k) \quad (2.2b)$$

#### [II] ORTHOMIN (q) 法

$$p_{k+1} = r_{k+1} + \sum_{j=\max(1, k-q+1)}^k b_{j,k} p_j, \quad (2.3)$$

ただし、 $\{b_{j,k}\}$  は、GCR法の (2.2b) を利用する。

#### [III] GCR (q) 法

GCR法のアルゴリズムを  $q+1$  回ごとに再出発させる。総ての  $q+1$  回目の反復で  $x_{j+(q+1)}$  を新しい出発値として考えることになる。

#### [IV] MR法 (Minimum Residual Method)

$$p_{k+1} = r_{k+1}. \quad (2.4)$$

GCR法で生成された方向ベクトル  $\{p_k\}$  は、

$$(Ap_i, Ap_j) = 0, \quad \forall i \neq j \quad (2.5)$$

を満足する。つまり、 $\{p_k\}$  は  $A^T A$  直交していることになる。また、 $x_k$  は、affine部分空間  $\langle x_1 + \text{span}\{p_1, p_2, \dots, p_{k-1}\} \rangle$  上で、

$$E(x_k) = \|b - Ax_k\|_2, \quad (2.6)$$

を最小化する。ここで、 $r_k = b - Ax_k$  とおけば、(2.6)式は、GCR法の生成多項式を用いて、

$$\|r_k\|_2 = \min_{q_k \in Q_k} \|q_k(A)r_1\|_2, \quad (2.7)$$

と書き表すことができる。ただし、 $Q_k$  は、 $q_k(0) = 1$  を満足する実生成多項式の集合を示すものとする。

GCR法は、丸め誤差の影響を受けなければ、高々  $n$  回の反復で、(1.1) 式の厳密解を与えることになる。また、対称問題に対して、GCR法はCR法に類似しており、もし係数行列  $A$  が正定値対称ならば、(2.2) 式は次のような二項漸化式：

$$p_{k+1} = r_{k+1} + b_k p_k, \quad (2.8a)$$

$$b_k = - \frac{(Ar_{k+1}, Ap_k)}{(Ap_k, Ap_k)}. \quad (2.8b)$$

で表現され、CR法と同じ算法になる。しかし、GCR法の算法からもわかるように、大次元行列問題を取り扱った場合には、計算量と記憶領域は膨大なものとなる。そこで、Vinsome [11] は、GCR法の代替法として計算量と記憶領域を節約する ORTHOMIN ( $q$ ) 法を提案した。ORTHOMIN ( $q$ ) 法のベクトル  $p_{k+1}$  は、最も新しい  $q (\geq 0)$  個のベクトルのみ直交しているため、 $q$  本分のベクトルの記憶領域のみを必要とするだけである。この場合にも、 $x_k$  は affine 部分空間  $\langle x_1 + \text{span}\{p_{k-q+1}, \dots, p_{k-1}\} \rangle$  上で、 $E(x_k)$  を最小ならしめる。次に、GCR ( $q$ ) 法も GCR法の代替法として提案され、ORTHOMIN ( $q$ ) と同様に、 $q$  個の方向ベクトルのみを記憶するだけでよい。1 回の反復に於けるこれらの方法は、 $p_{k+1}$  の計算において、 $k$  個よりも少ない方向ベクトルを用いて計算することになるので計算量の減少になり、計算時間も短縮される。MR法は、ORTHOMIN ( $q$ ) 法と GCR ( $q$ ) 法で  $q=0$  とした特別な場合に対応しており、計算量や記憶領域の面では、もっとも妥当なものと言えよう。また、この方法は対称行列系を取り扱った場合には、最急降下法 (Steepest Descent Method) となる。表 2.1 に、ここで述べた各方法が必要とする計算量と記憶領域についてまとめた。GCR法の表現にはベクトル  $x_k, r_k, Ar_k, \{p_1, \dots, p_k\}$  と  $\{Ap_1, \dots, Ap_k\}$  のための記憶領域を含んでいる。 $Ap_k$  は、(2.2a) 式より再帰的に、

$$Ap_k = Ar_k + \sum_{j=\max(1, k-q)}^{k-1} b_{j, k-1} Ap_j, \quad (2.9)$$

で計算できるので、 $Ar_k$  (行列  $\times$  ベクトル) のみを計算するだけでよいことになる。

表 2.1 GCR法, GCR ( $q$ ) 法, ORTHOMIN ( $q$ ) 法, MR法が必要とする計算量 (1 回の反復) と記憶領域 ( $nz(A)$ : 行列  $\times$  ベクトルの計算量)

解法	計算量	記憶領域
GCR法	$(3(k+1)+4)n + nz(A)$	$(2k+3)n$
ORTHOMIN ( $q$ ) 法	$(3q+4)n + nz(A)$	$(2q+3)n$
GCR ( $q$ ) 法	$((3/2)q+4)n + nz(A)$	$(2q+3)n$
MR法	$4n + nz(A)$	$3n$

GCR ( $q$ ) 法の計算量は、 $q+1$  回の反復の平均であると言える。

前にも述べたが GCR法は、高々  $n$  回の反復で (1.1) の厳密解を与えるが、残りの 3 つの方法は、一般に有限回の反復で終了するとは言えない。しかし、現実ここで述べた総ての方法は、 $n$  回よりもかなり少ない反復回数で十分精度の良い近似解を計算することが可能である。

## 2.2 GCR法の誤差の上限

前説で述べた 4 つの方法について、誤差の評価式を考えることにする。今、係数行列  $A$  を

$$A = M - R, \quad (2.10)$$

ただし、

$$M = \frac{1}{2}(A + A^T) : \text{行列 } A \text{ の対称部}, \quad (2.11a)$$

$$R = -\frac{1}{2}(A - A^T) : \text{歪対称部}, \quad (2.11b)$$

ここで、行列  $M$  は正定値であるものと仮定する。今、行列  $A$  のジョルダン標準型を  $J (=T^{-1}AT)$  とする。ここで、 $\forall X \in \mathbb{R}^{n \times n}$  に対して、 $\Lambda(X)$  を行列  $X$  の固有値の集合とする。また、 $X$  の絶対値最小の固有値を  $\lambda_{\min}(X)$ 、 $X$  の絶対値最大の固有値を  $\lambda_{\max}(X)$ 、そしてスペクトル半径を  $\rho(X) (=|\lambda_{\max}(X)|)$  とする。行列  $X$  が正則であるならば、行列  $X$  の条件数を

$$\sigma(X) := \|X\|_2 \|X^{-1}\|_2, \quad (2.12)$$

と定義する。

GCR法と GCR ( $q$ ) 法の生成多項式を利用して、次のような誤差の上限を導出することができる。

[定理 2.1]  $\{r_k\}$  が GCR法で生成された残差ベクトル列であるならば、次式が成立する。

$$\|r_k\|_2 \leq \min_{q_k \in Q_k} \|q_k(A)\|_2 \cdot \|r_1\|_2, \quad (2.13)$$

ここで、行列  $A$  が固有ベクトルの完全集合を持つならば、次式が成立する。

$$\|r_k\|_2 \leq \sigma(T) M_k \|r_1\|_2, \quad (2.14)$$

ただし、

$$M_k := \min_{q_k \in Q_k} \max_{\lambda \in \Lambda(A)} |q_k(\lambda)|, \quad (2.15)$$

さらに、行列  $A$  が正規行列ならば、次式が成立する。

$$\|r_k\|_2 \leq M_k \|r_1\|_2. \quad (2.16)$$

[定理 2.2]  $\{r_k\}$  が GCR ( $q$ ) 法で生成された残差ベクトル列であるならば、次式が成立する。

$$\|r_{j+(q+1)}\|_2 \leq \left[ \min_{q_{q+1} \in Q_{q+1}} \|q_{q+1}(A)\|_2 \right]^j \times \|r_1\|_2. \quad (2.17)$$

ここで、行列Aが固有ベクトルの完全集合を持つならば、次式が成立する。

$$\|r_{j(q+1)}\|_2 \leq \left[ \sigma(T) M_{q+1} \right]^j \|r_1\|_2, \quad (2.18)$$

さらに、行列Aが正規行列ならば、次式が成立する。

$$\|r_{j(q+1)}\|_2 \leq \left[ M_{q+1} \right]^j \|r_1\|_2 \quad (2.19)$$

次に述べる定理は、ORTHOMIN(q)法が、収束することを保証し、GCR法およびGCR(q)法のもう1つの誤差の上限を与えることになる。

[定理2.3]  $\{r_k\}$ がGCR法で生成された残差ベクトル列であるならば、次式が成立する。

$$\|r_k\|_2 \leq \left[ 1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^T A)} \right]^{k/2} \times \|r_1\|_2. \quad (2.20)$$

および、

$$\|r_k\|_2 \leq \left[ 1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\min}(M)\lambda_{\max}(M) + \rho(R)^2} \right]^{k/2} \|r_1\|_2 \quad (2.21)$$

これらの証明は、S.C.Eisenstat et al. [5] 参照せよ。GCR法では、行列Mが正定値と言う条件があるが、もしMが正定値でないときには、次の例のように収束しないこともある。

(例1) (Young [13] より)

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

初期値を  $x_1 = [1, 2]$  から反復を始めると、GCR法は最初の反復で、 $a_1 = 0$  となり反復を先に進めることができなくなる。これは、 $\{r_1, Ar_1, A^2 r_1, \dots, A^k r_1\}$  ( $k < n$ ) が線形独立でも  $\{p_j\}_{j=1}$  は線形従属となることがあるからである。

### 2.3 AxelssonのGCR法

2.1節では、非対称行列を係数とする連立1次方程式を効果的に解くCR法の一般化として、方向ベクトル  $\{p_k\}$  を修正する解法について述べてきた。

Axelsson [2] は、前述の方法と基本的には同等であるが、より複雑な近似解ベクトル  $\{x_k\}$  を修正利用する次のような算法を提案した。

(AXELGCR法 (Axelsson's Generalized Conjugate Residual Method))

(1) 初期値  $x_1$  を選び、残差ベクトル

$$r_1 = b - Ax_1,$$

を計算し、方向ベクトルを  $p_1 = r_1$  と置く。

(2) 次の手順を繰り返す。 ( $k=1, 2, 3, \dots$ )

$$x_{k+1} = x_k + \sum_{j=1}^k a_{j,k} p_j, \quad (2.22a)$$

ただし、スカラー  $\{a_{j,k}\}_{j=1}^k$  は、 $\|r_{k+1}\|_2$  を最小にするように選ぶ。

$$r_{k+1} = r_k - \sum_{j=1}^k a_{j,k} A p_j, \quad (2.22b)$$

$$p_{k+1} = r_{k+1} + b_k p_k, \quad (2.22c)$$

$$b_k = - \frac{(A r_{k+1}, A p_k)}{(A p_k, A p_k)}. \quad (2.22d)$$

この算法では、スカラー  $\{a_{j,k}\}_{j=1}^k (= \alpha_k)$  は最小自乗問題：

$$\min \|B_k \alpha_k - r_k\|_2 \quad (2.23)$$

の解として決定することになる。ただし、 $B_k := [A p_1, A p_2, \dots, A p_k]$  とする。この場合、 $\{a_{j,k}\}_{j=1}^k$  は、

$$\|r_k\|_2 = \min_{q_k \in Q_k} \|q_k(A) r_1\|_2, \quad (2.24)$$

を満足するように決定することになるので、この方法もGCR法である。つまり、この方法は最小自乗GCR法とも言うのが適当であるかもしれない。

この方法を  $k+1$  回ごとに再出発させるならば、GCR法と同等な算法になる。また、ORTHOMIN(q)法と同様に、(2.22a) を適当に打ち切って、

$$x_{k+1} = x_k + \sum_{j=\max(1, k-q+1)}^k a_{j,k} p_j \quad (q \geq 1) \quad (2.25)$$

で置き換える方法も考えられる。ただし  $\{a_{j,k}\}_{j=k-q+1}^k$  は、 $\|r_{k+1}\|_2$  を最小にするように選ぶ必要があり、(2.23)と同様に次のような最小自乗問題の解として決定することになる。

$$\min \|B_k \alpha_k - r_k\|_2 \quad (2.26)$$

ただし、

$$\alpha_k := [a_{k-q+1}, \dots, a_k],$$

$$B_k := [A p_{k-q+1}, \dots, A p_k].$$

この算法をAXEL(q)法と命名することにする。  $q=1$  の

場合、この算法はORTHOMIN(1)法である。AXELGCR法とAXEL(q)法の計算量と記憶領域を表2.2に示した。

言うまでもないが、(2.23)式と(2.26)式の $\alpha_k$ は正規方程式：

$$B_k^T B_k \alpha_k = B_k^T r_k, \quad (2.27)$$

を利用して決定すればよい。

AXEL(q)の算法では、 $r_k \neq 0$ ならば $\{p_j\}_{j=1}^{k-q+1}$ は線形独立になるように $\alpha_k$ を決定しているため、途中で破綻をきたすことはまれである。算法が破綻をきたすたが1つの可能性は、 $r_{k+1}$ と $p_k$ が共線(collinear)になるときである。

表2.2 AXELGCR法、AXEL(q)法が必要とする計算量と記憶領域(nz(A):行列×ベクトルの計算量)

解法	計算量	記憶領域
AXELGCR法	$(3(k+1)+4)n + nz(A) + O(k^2)$	$2((k+2)+2)n + O(k^2)$
AXEL(q)法	$(3q+4)n + nz(A)$	$(2q+3)n$

#### 2.4 Young&JeaのORTHODIR法

Young&Jea [13]は、GCR法によく似たORTHODIR法を提案した。この方法は、行列Aの対称部である行列Mが正定値でなくても収束が保証される性質を持っている。

(ORTHODIR法)

(1) 初期値 $x_1$ を選び、残差ベクトル

$$r_1 = b - Ax_1,$$

を計算し、方向ベクトルを $p_1 = r_1$ と置く。

(2) 次の計算を繰り返す。(k=1,2,3,...)

$$x_{k+1} = x_k + a_k \bar{p}_k, \quad (2.27a)$$

$$r_{k+1} = r_k - a_k A \bar{p}_k, \quad (2.27b)$$

$$\bar{p}_{k+1} = A \bar{p}_k + \sum_{j=1}^k b_{j,k} \bar{p}_j, \quad (2.27c)$$

$$a_k = \frac{(r_k, A \bar{p}_k)}{(A \bar{p}_k, A \bar{p}_k)}, \quad (2.27d)$$

$$b_{j,k} = - \frac{(A^2 \bar{p}_k, A \bar{p}_j)}{(A \bar{p}_j, A \bar{p}_j)} \quad (j \leq k) \quad (2.27e)$$

ORTHODIR法の打ち切り版として、(2.27c)の代わりに

$$\bar{p}_{k+1} = A \bar{p}_k + \sum_{j=\min(1, k-q+1)}^k b_{j,k} \bar{p}_j,$$

を用いることもできる。これをORTHODIR(q)法とする。これらの方法の1回の反復に必要な計算量と記憶領域を表2.3に示した。この表から、これらの値はGCR法とORTHOMIN(q)法のものに等しいことになることは明らかである。

今、行列Aの対称部が正定値であるならば、GCR法で生成された方向ベクトル $\{p_k\}$ とORTHODIR法で生成された

方向ベクトル $\{\bar{p}_k\}$ の間に、 $\bar{p}_1 = p_1$ 、 $\bar{p}_k = r_k$ 、 $p_k$ が成立する。ただし、 $r_k$ は、あるスカラーである(詳細は、Young&Jea [13, 14]を参照せよ)。よってORTHODIR法は、GCR法に等しいことになる。さらに、行列Mが正定値でなくても、 $\text{span}\{\bar{p}_1, \dots, \bar{p}_k\} = \text{span}\{r_1, Ar_1, \dots, A^k r_1\}$ なので、ORTHODIR法は、一般の正則行列Aに対しても収束することになる。

[例2] 例1の問題でGCR法は破綻をきたしたが、ORTHODIR法を用いると、

1回目:  $a_1 = 0$ ,  $p_1 = [0, 1]^T$ ,  $x_2 = x_1$ ,

2回目:  $a_2 = 1$ ,  $x_3 = \bar{x}$ (厳密解)。

となり、2回の反復で収束する。

表2.3 ORTHODIR法、ORTHODIR(q)法が必要とする計算量と記憶領域(nz(A):行列×ベクトルの計算量)

解法	計算量	記憶領域
ORTHODIR法	$(3(k+1)+4)n + nz(A)$	$(2(k+2)+2)n$
ORTHODIR(q)法	$(3q+4)n + nz(A)$	$(2q+3)n$

#### 3. BCG法(Biconjugate Gradient Method)

この方法はFletcher [6]により提案され、おもに正定値でない対称な正則行列問題の解法として開発された。

[BCG法]

(1) 初期値 $x_1$ を選び、残差ベクトル

$$r_1 = b - Ax_1,$$

を計算し、 $\bar{p}_1 = \bar{r}_1 = p_1 = r_1$ と置く。

(2) 次の計算を繰り返す。(k=1,2,3,...)

$$x_{k+1} = x_k + a_k p_k, \quad (3.1a)$$

$$r_{k+1} = r_k - a_k A p_k, \quad (3.1b)$$

$$\bar{r}_{k+1} = \bar{r}_k - a_k A^T \bar{p}_k, \quad (3.1c)$$

$$p_{k+1} = r_{k+1} + b_k p_k, \quad (3.1d)$$

$$\bar{p}_{k+1} = \bar{r}_{k+1} + b_k \bar{p}_k, \quad (3.1e)$$

$$a_k = (\bar{r}_k, r_k) / (\bar{p}_k, A p_k), \quad (3.1f)$$

$$b_k = (\bar{r}_{k+1}, r_{k+1}) / (\bar{r}_k, r_k). \quad (3.1g)$$

この算法は非対称行列に対する固有値問題を Lanczos法で求めるときに算法と同値であり、次の条件を満足している。

$$\forall i \neq j \quad (r_i, \bar{r}_j) = (\bar{r}_i, r_j) = 0, \quad (3.2a)$$

$$(p_i, A p_j) = (\bar{p}_i, A p_j) = 0, \quad (3.2b)$$

一般の非対称行列Aに対して、この算法が破綻をきたすことなく安定であるという保証はどこにもない。

#### 4. Preconditioning

前節の方法は行列の前処理 (preconditioning) によって収束を加速することができる ([8])。今、preconditionerとしてある正則行列B (=B<sub>1</sub> B<sub>2</sub>) を考える。方程式(1.1)を解くことは、次の3つの方程式の1つを解くことに帰着できる。

$$B A x = B b, \quad (4.1)$$

$$A B (B^{-1} x) = b, \quad (4.2)$$

$$B_1 A B_2 (B_2^{-1} x) = B_1 b. \quad (4.3)$$

GCR法の各反復で生成される近似解は、ある部分空間上で残差のEuclidノルムを最小化するが、preconditioningを用いたときに最小化する値は、当然のことながらpreconditioningの技法の選択に依存する。例えば、(4.1)式にGCR法を適用する場合を考えてみると、各反復で残差のEuclidノルム:  $\|B(b - A x_k)\|_2$  を最小にする。また、(4.2)式にGCR法を適用した場合は、(1.1)式にGCR法を適用した場合と同じ残差のEuclidノルム:  $\|b - A x_k\|$  を最小にすることは明らかである。

(4.2)式にGCR法を適用した場合の算法は次のようになる。

(PGCRM (Preconditioned Generalized Conjugate Residual Method for Right preconditioning))

(1) 初期値  $x_1$  を選び、残差ベクトル

$$r_1 = b - A x_1,$$

を計算し、 $p_1 = r_1$  と置く。

(2) 次の計算を繰り返す。(k=1,2,3,...)

$$x_{k+1} = x_k + a_k p_k, \quad (4.4a)$$

$$r_{k+1} = r_k - a_k A p_k, \quad (4.4b)$$

$$p_{k+1} = B r_{k+1} + \sum_{j=1}^k b_{j,k} p_j, \quad (4.4c)$$

$$a_k = \frac{(r_k, A p_k)}{(A p_k, A p_k)}, \quad (4.4d)$$

$$b_{j,k} = \frac{-(A B r_{k+1}, A p_j)}{(A p_j, A p_j)}. \quad (4.4e)$$

この算法の計算量は、 $A B r_{k+1}$  の計算を除いて、GCR法の計算量に等しい。一般に、この算法では、Bを係数とする方程式( $B r_{k+1} = v$ )を解き、その結果に行列Aを掛けることになる。よって、 $B r_{k+1}$  を計算するためにGCR法よりも1本余分にベクトルを記憶する領域が必要になる。次に、(4.3)式にGCR法を適用した場合の算法について述べることにする。(4.3)式からもわかるように、(4.1)式と(4.2)式は、この式のpreconditioningの特別な場合であることは明らかである。

(PGCRMS (Preconditioned Generalized Conjugate Residual Method for Split preconditioning))

(1) 初期値  $x_1$  を選び、残差ベクトル

$$r_1 = b - A x_1,$$

及び、方向ベクトル

$$p_1 = B_1 r_1,$$

を計算する。

(2) 次の計算を繰り返す。(k=1,2,3,...)

$$x_{k+1} = x_k + a_k p_k, \quad (4.5a)$$

$$r_{k+1} = r_k - a_k B_1 A p_k, \quad (4.5b)$$

$$p_{k+1} = B_2 r_{k+1} + \sum_{j=1}^k b_{j,k} p_j, \quad (4.5c)$$

$$a_k = \frac{(r_k, B_1 A p_k)}{(B_1 A p_k, B_1 A p_k)}, \quad (4.5d)$$

$$b_{j,k} = - \frac{(B_1 A B_2 r_{k+1}, B_1 A p_j)}{(B_1 A p_j, B_1 A p_j)} \quad (4.5e)$$

この算法では、affine部分空間  $\langle x_1 + \text{span}\{p_1, B_1 A p_1, \dots, (B_1 A)^{k-1} p_1\} \rangle$  上で残差のEuclidノルム  $\|B_1(b - A x_k)\|$  を最小にする。必要な計算量と記憶領域は、PGCRM法に等しいことは明らかである。

これらの算法からpreconditioningとして、(4.2)及び(4.3)式を用いる場合には、 $B_2 r_{k+1}$  を計算するために1本余分に補助記憶ベクトルを必要とするが、(4.1)式を利用した場合には、上記算法で  $B_2 = I$  と置けばよいので、 $B_2 r_{k+1} = r_{k+1}$  となり、PGCR法は前処理をしないGCR法と同じ記憶領域で実現できることになる。では(4.1)~(4.3)式のどれを利用すれば良いかと言えば一般には、最小にする残差のEuclidノルムが前処理に無関係な(4.2)式を用いるのが妥当であるように思われる。しかし、ほぼ対称な行列問題を取り扱う時には、前処理した行列  $B_1 A B_2$  もほぼ対称になる可能性がある(4.3)式を利用した算法を用いるのが好ましい。

ORTHOMIN(q)法、GCR(q)法及びMR法の算法の実現においても、ほぼこの算法と同様である。

前処理行列(preconditioner)の構成方法を大別すれば行列分離と不完全行列分解の2つの方法に分けられるが詳

細は野寺 [8] を参照されたい。

### 5. 数値例

(問題) 偏微分方程式の境界値問題:

$$\Delta u + \beta \frac{\partial u}{\partial x} = f, \quad (x, y) \in \Gamma, \quad (5.1)$$

$$u = g, \quad (x, y) \in \partial \Gamma. \quad (5.2)$$

の近似解  $u_1, u_2, \dots, u_n$  を求めることを考える。ただし、 $\Gamma$  は正方形領域 (5.1) で、 $\beta$  は定数とする。この方程式を修正風上差分で離散化すれば、図5.2 に示すような分子構造を持つ。数値実験の結果を図5.3 ~ 5.7 及び表5.1 に示す。

### 6. おわりに

大型で疎な非対称行列を係数とする連立一次方程式の反復解法、特に variational method と呼ばれるものについて述べてきた。現在の所、これらの算法は、戦国時代の様相を示しており、どれをとっても一長一短があるように思われる。

最後に、今後、解決しなければならない問題点を列挙すると次のようになる。

- (1) 誤差の上限
- (2) 収束判定
- (3) 正定値でない行列系への応用
- (4) ベクトル計算機への実現

#### [参考文献]

[1] O.Axelsson, Solution of linear systems of equations, Lecture Note in Math.572, pp.1-51 (1976).

[2] O.Axelsson, Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations, Linear Algebra Appl. 29, pp.1-16 (1980).

[3] P.Concus and G.H.Golub, A generalized conjugate gradient method for nonsymmetric system of linear equations, in Lecture Note in Economics 134, Springer, pp.56-65 (1976).

[4] H.C.Elman, Preconditioned Conjugate Gradient methods for nonsymmetric systems of linear equations, in Advances in Computer methods for P.D.E.IV, pp.409-417 (1981).

[5] S.C.Eisenstat et al., Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. Vol.20, pp.345-357 (1983).

[6] R.Fletcher, Conjugate gradient methods for indefinite systems, in Lecture Note in Math. 506, Springer, pp.73-89 (1976).

[7] T.Nodera and H.Takahasi, PCG methods for non-symmetric matrix, Scientific Computing Vol.1

North Holland, pp.173-180 (1983).

[8] 野寺, 大型疎行列に対するPCG法, Seminar on Mathematical Sciences Vol.7, マセマテイカ (1983).

[9] T.Nodera, PCG method for solving singular perturbation problem (PART I), Keio MATH. SEM. Rep. No.7, pp.61-72 (1983).

[10] Y.Saad, The Lanczos bidiagonalization algorithm and other oblique projection methods for solving large unsymmetric systems, SIAM J. Numer.Math. Vol.49, pp.485-506 (1982).

[11] P.K.Vinsome, Orthomin, an iterative method for solving sparse sets of simultaneous linear equations, in Proc. Fourth Symposium on Reservoir Simulation Society of Petroleum Engineers of AIME, pp.149-159 (1976).

[12] O.Widlund, A Lanczos method for a class of nonsymmetric systems of linear equations, SIAM J. Numer. Anal. Vol.15, pp.801-812 (1978).

[13] D.M.Young and K.C.Jea, Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods, Linear algebra Appl., vol.34, pp.159-194 (1980).

[14] K.C.Jea and D.M.Young, On the simplification of generalized conjugate gradient methods for nonsymmetrizable linear systems, Linear Algebra Appl., pp.399-417 (1983).

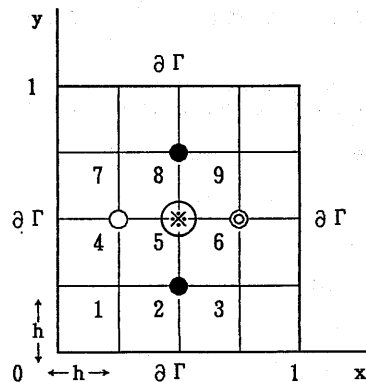


図5.1 正方形領域  $\Gamma$  (整合順序に番号付け)

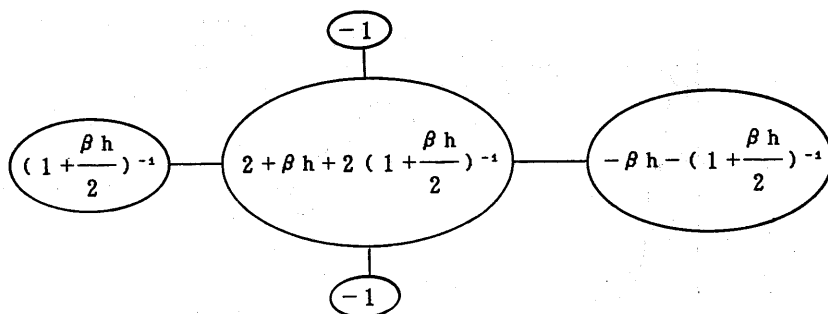


図5.2 修正風上差分の分子構造

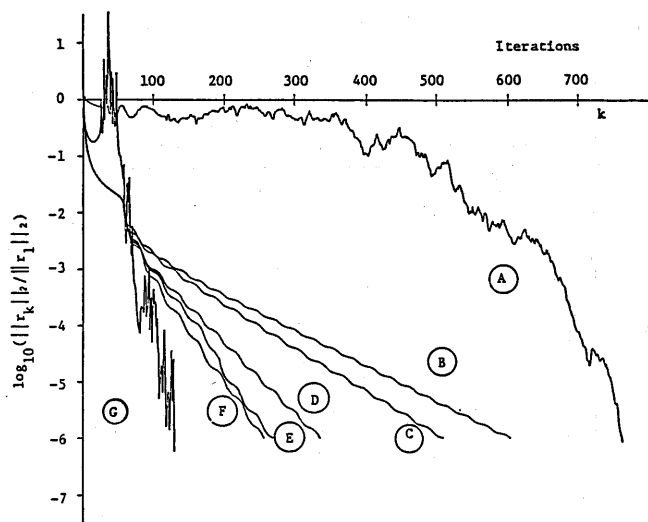


図5.3 残差ノルムと反復回数 ( $h = 1/40$ ), 前処理なし  
 A: 正規方程式 ( $AA^T u = b, x = A^T u$ )  
     に共役勾配法を適用  
 B: ORTHOMIN (1)      E: ORTHOMIN (4)  
 C: ORTHOMIN (2)      F: ORTHOMIN (5)  
 D: ORTHOMIN (3)      G: BCG

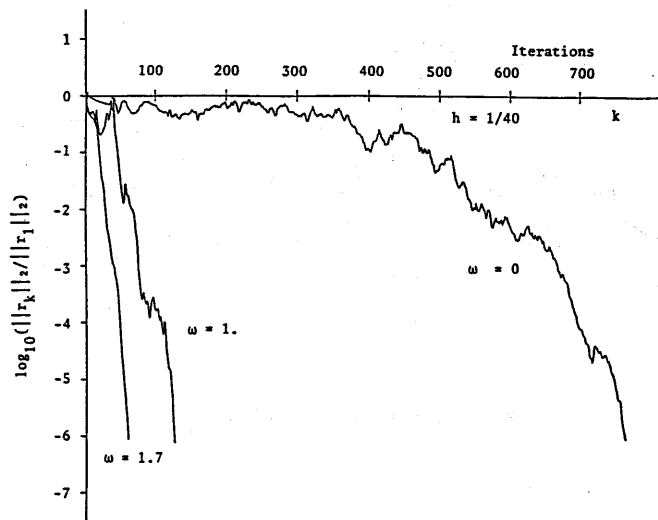


図5.4 残差ノルムと反復回数 ( $h = 1/40$ )  
 正規方程式 ( $AA^T u = b, x = A^T u$ )  
 に共役勾配法を適用

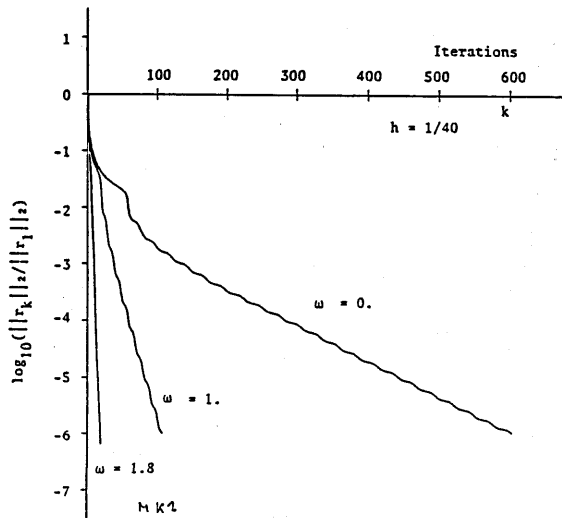


図5.5 残差ノルムと反復回数 ( $h = 1/40$ )  
 ORTHOMIN ( 1 )



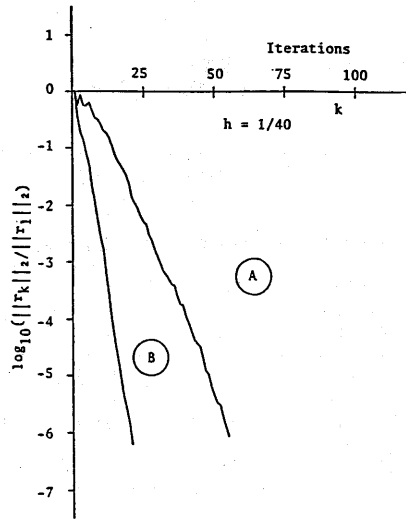


図5.6 残差ノルムと反復回数 ( $h = 1/40$ )  
 前処理 (行列分離:  $\omega = 1.8$ )  
 A: 正規方程式に共役勾配法を適用  
 B: ORTHOMIN (1)

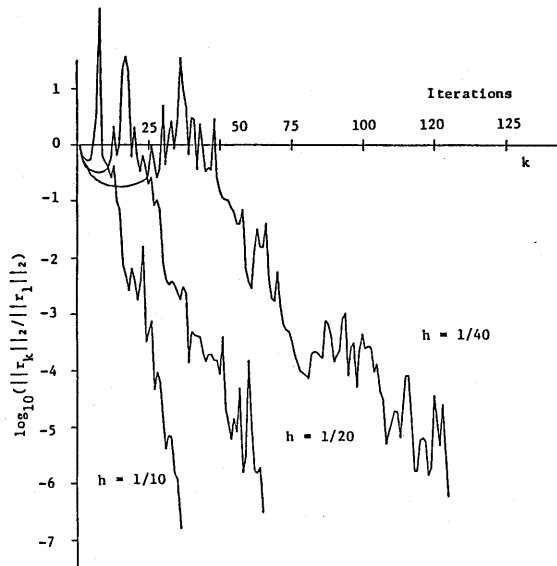


図5.7 BCG法の残差ノルムと反復回数, 前処理なし  
 ( $h = 1/10, 1/20, 1/40$ )

表5.1 ORTHOMIN ( q ) が収束判定条件 (  $\varepsilon = 10^{-6}$  )  
を満たすまでに必要な反復回数と乗算回数

q	前処理なし		前処理 ( 行列分離 ) ( $\omega = 1.8$ )	
	反復回数	乗算回数	反復回数	乗算回数
1	603	9994725	20	450060
2	510	10780380	20	541320
3	335	8609835	20	632580
4	270	8171280	18	651456
5	256	8915712	18	733590