連立一次方程式の解法
反復回数と計算量削減の工夫

— 共役勾配法について —

大阪大学工学部　都　田　艶　子

〔0〕は　じ　め　に

n次の連立一次方程式 $Ax=b$ の解法には、Gauss の消去法に代表される直接法（有限回の演算で丸め誤差がなければ厳密解に到達するもの）と、適当な近似解 x_k からはじめて、方程式より変形した同値な式 $x=\varphi(x)=Hx+c$ を用い、逐次代入 $x_{k+1}=\varphi(x_k)$ を行なうことによって解を求めていく反復法があり、その代表的なものに Gauss－Seidel 法，SOR法，SSOR法，Jacobi法がある。今、ここでとりあげる共役勾配法(Conjugate Gradient Method―略してCG法）は、反復解法であると同時に、有限回のステップで解に到達するという直接法の利点をあわせもち、この点で他の反復解法と異なる特徴をもっていることが知られる。しかしながら、その絶対値が減少数列である残差の列の線形結合により修正方向を決定していくアルゴリズムであることから、丸め誤差の影響に対し非常に敏感であるという欠点をもっていた。

しかし、係数行列Aが大型でかつ疎行列(sparse matrix)であるならば、その疎(sparse)である性質が大きくきいて、反復回数は理論上のn回よりずっと小さくて解に収束することが知られた。このことはきわめて注目してよいことであった。そしてまた、最初は一般化CG法(generalized CG method)として考えられた手法が、係数行列Aが大次元かつ sparse である場合に、非常な効果をもたらし、収束の速さを飛躍的に速めるということを発見したのは Reid[21]であり、それ以後、この generalized CG法の手法は、連立一次方程式がより解きやすい条件下におかれ、そして収束が速くなるようにあらかじめ条件を整えておくという Preconditioning の手法として非常な注目を集めている。

ここでは、このCG法に対する Preconditioning の手法について若干解説するとともに、さらに、そのアルゴリズムを考慮する中で、一反復における計算量を画期的に減少させる方法について述べる。

〔1〕古典的 Conjugate Gradient 法

係数行列Bが $n\times n$ の正則行列とすると、B^TB は正定値対称行列である。そこで、連立一次方程式 $Bx=b$ に対し、正値2次形式

$$\varphi(y)=\frac{1}{2}(\bar{x}-y)^T B^T B(\bar{x}-y)$$

を考えると、y が真の解 $\bar{x}$ に近いほど $\varphi(y)$ の値は小さくなり、$y=\bar{x}$ のとき $\varphi(y)=0$ となる。そこで、一つの近似解 x_k に対し、適当な修正をほどこして $\varphi(x_{k+1})<\varphi(x_k)$ となるような x_{k+1} を定め、真の解 $\bar{x}$、つまり $\varphi(y)$ の極小値0を与える点へ収束する列 $\{x_k\}$ をつくっていく方法が考えられる。修正方向ベクトルの定め方によりその方法は種々考えられるが、これらは総称して傾斜法と呼ばれる。共役勾配法(CG法)は、そのうちの一つである最大傾斜法(Steepest Descent Method)に修正を加えたものとして、最初に、1952年 Hestenes と Stiefel によって提唱されたものである。

ここで、改めて解くべき連立一次方程式を次のようにする。

$$Ax=b,\quad A;\ n\times n\text{ で、正定値対称} \qquad (1-1)$$

その最小値を求めることを目的とする誤差函数は

$$\varphi(x)=(\bar{x}-x\ ,\ A(\bar{x}-x))=(A^{-1}r\ ,\ r) \qquad (1-2)$$

ただし　$\bar{x}$; 真の解

$r=b-Ax$; 残差

とする。ある近似解 x_k から、修正ベクトル p_k を用いて新しい近似解 x_{k+1} は次のようにして求めるものとする。

$$x_{k+1}=x_k+\alpha_k p_k \qquad (1-3)$$

この修正方向において $\varphi(x_{k+1})$ を最小にする α_k は、$\dfrac{\partial\varphi(x_{k+1})}{\partial\alpha_k}=0$ より

$$\alpha_{k+1}=\frac{(p_k\ ,\ r_k)}{(p_k\ ,\ Ap_k)} \qquad (1-4)$$

であり、これから次の残差に対する式がでてくる。

$$r_{k+1}=b-Ax_k=r_k-\alpha_k Ap_k \qquad (1-5)$$

修正ベクトル p_{k+1} の選び方としては、p_1, p_2, …, p_k に対し、A−直交であるようにする。A−直交であるとは

$$\begin{aligned}&(p_i\ ,\ Ap_j)=0,\quad i\neq j\\&(p_i\ ,\ Ap_i)>0\end{aligned} \qquad (1-6)$$

なる関係を満たすことである。このようにして選ぶ p_1, p_2, …, p_n は一次独立なベクトル系であり、n 次元空間を張る基底をなしている。ゆえに、連立一次方程式の解 $\bar{x}=A^{-1}b$ はこのベクトル系で展開され、

$$\bar{x}=c_1p_1+c_2p_2+\cdots+c_np_n \qquad (1-7)$$

と書くことができる。各係数は $\{p_i\}$ の A−直交性より

$$c_k = \frac{(p_k\ ,\ b)}{(p_k\ ,\ Ap_k)} \quad , \quad k=1,2,\cdots,\ n \qquad (1-8)$$

である。反復の初期値 $x_1=0$ とすると、k 回目の反復での修正の最適係数 α_k は、$\{p_i\}$ の A−直交性より

$$\alpha_k = \frac{(p_k\ ,\ r_k)}{(p_k\ ,\ Ap_k)} = \frac{(p_k\ ,\ b)}{(p_k\ ,\ Ap_k)} = c_k \qquad (1-9)$$

であり、解 $\bar{x}$ の展開係数に等しい。共役勾配法の反復回数はたかだか n まで、というきわだった特徴はここからいえることである。

さて、この A−直交系のとり方にはまだ任意性が残っているが、共役勾配法では各近似解 x_1, x_2, … における残差ベクトルの列 r_1, r_2, …, r_k より、以下のようにしてつくる。この残差ベクトル系は直交系である。つまり、

$$(r_i\ ,\ r_j) = 0 \qquad (i \neq j) \qquad (1-10)$$

ゆえに、一次独立なベクトル系であるから、これにより Gram−Schmidt の直交化法を用いて A−直交系 p_1, p_2, … を求めていく。

$$p_1 = r_1$$
$$p_k = r_k - \sum_{j=1}^{k-1} r_{jk}\, p_j \ , \quad k=2,\ 3,\ \cdots,\ n \qquad (1-11)$$
$$\text{ただし} \qquad r_{jk} = \frac{(r_k\ ,\ Ap_j)}{(p_j\ ,\ Ap_j)}$$

$\{r_i\}$ の直交性と $\{p_i\}$ の A−直交性を用いて、上式は、以下のような簡単な形に書きなおすことができる。

$$p_{k+1} = r_{k+1} + \beta_k p_k \qquad (1-12)$$
$$\text{ただし} \qquad \beta_k = \frac{(r_{k+1}\ ,\ r_{k+1})}{(r_k\ ,\ r_k)}$$

以上より、共役勾配法のアルゴリズムは以下のようになる。

〔CG Algorithm 1〕

初期値 x_1

$r_1 = b - Ax_1$, $p_1 = r_1$

$k=1,\ 2,\ \cdots$ に対して

$$\alpha_k = \frac{(p_k\ ,\ r_k)}{(p_k\ ,\ Ap_k)} \ , \qquad \begin{aligned} x_{k+1} &= x_k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k Ap_k \end{aligned} \qquad (1-13)$$

$$\beta_k = \frac{(r_{k+1}\ ,\ r_{k+1})}{(r_k\ ,\ r_k)} \ , \qquad p_{k+1} = r_{k+1} + \beta_k p_k$$

さらに、$p_k = p_k/(r_k, r_k)$ とおきなおすと、簡単な計算により、上と同値な次のアルゴリズムがでてくる。

〔CG Algorithm 2〕

初期値；x_1

$r_1 = b - Ax_1$，　$p_1 = r_1$

$k = 1, 2, \cdots$ に対して

$$\alpha_k = \frac{1}{(p_k, Ap_k)}, \qquad x_{k+1} = x_k + \alpha_k p_k \qquad (1-14)$$

$$r_{k+1} = r_k - \alpha_k A p_k$$

$$\beta_k = \frac{1}{(r_{k+1}, r_{k+1})}, \qquad p_{k+1} = p_k + \beta_k r_{k+1}$$

以上が基本的なCG法のアルゴリズムである。解法自体に対する変形版も種々あるが、ここではふれない。なお、上の〔CG Algorithm 2〕は高橋版といわれる変形版である。以後、このアルゴリズムを基礎におく。これにもとづき、最初は一般化CG法(generalized Conjugate Gradient method)として考えられた Preconditioning について次節以降で述べる。

〔2〕 Preconditioned Conjugate Gradient Algorithm

連立一次方程式を解く際には、通常、その解きやすさというのが問題とされ、それは行列の条件数(condition number)の大きさで測られる。Aの条件数 $\kappa(A)$ は、ノルムを用いて、$\kappa(A) = \|A\| \cdot \|A^{-1}\|$ と定義される。またAの固有値 $\{\sigma_i\}$ が、大きさの順に $\sigma_n < \sigma_{n-1} < \cdots < \sigma_1$ ならば、これを用いて $\kappa(A) = \sigma_1/\sigma_n$ とも定義される。そしてこの時は spectral condition numberという。Preconditioning とは、もとの系を変換し、関連する条件数を減少させて、連立一次方程式が解きやすくなるように、あらかじめ条件を調整することである。

連立一次方程式に対する反復法への Preconditioning としては、次の2つのタイプが考えられる。適当な正則行列Cを用いて、

(a)　$AC^{-1}\tilde{x} = b$，　$x = C^{-1}\tilde{x}$

(b)　$C^{-1}Ax = C^{-1}b$

どちらの場合も、Aの条件数より $C^{-1}A$ 又は AC^{-1} の条件数がよくなるように、つまり、係数行列の固有値の多重度を増し、収束を速めることができるようにCは選ばれる。CG法では残差の情報を用いるので、(b)のタイプの Preconditioning を考慮する。

解くべき線形系

$$Ax = b, \qquad A\text{；正定値対称} \qquad (2-1)$$

に対し、適当な $n \times n$ の正則行列Mを選び、次のように一般化した系を考える。

$$M^{-1}Ax = M^{-1}b \qquad (2-2)$$

簡単にわかるように、M^{-1}はAの逆行列に近いものをとれば一番よい。従ってPreconditioningとは、Aの逆行列への非常によい近似行列をさがすことだともいえる。Aは正定値対称であるから、MもAと同様正定値対称であるとすると、(2－1)に同値な次式を解くべき線形系とすることができる。

$$M^{-\frac{1}{2}}AM^{-\frac{1}{2}}M^{\frac{1}{2}}x = M^{-\frac{1}{2}}b \qquad (2-3)$$

これを $\tilde{A}\tilde{x} = \tilde{b}$ と書き、この系に対してCG法のアルゴリズムを適用すると

〔 Algorithm 〕

$\tilde{x}_1$; (2－3)に対する初期値($M^{\frac{1}{2}}x$ の近似値)

$\tilde{r}_1 = \tilde{b} - \tilde{A}\tilde{x}_1$, $\tilde{p}_1 = \tilde{r}_1$

$k = 1, 2, \cdots$ に対して

$$\alpha_k = \frac{1}{(\tilde{p}_k, \tilde{A}\tilde{p}_k)}, \qquad \tilde{x}_{k+1} = \tilde{x}_k + \alpha_k \tilde{p}_k$$

$$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k \tilde{A}\tilde{p}_k$$

$$\beta_k = \frac{1}{(\tilde{r}_{k+1}, \tilde{r}_{k+1})}, \qquad \tilde{p}_{k+1} = \tilde{p}_k + \beta_k \tilde{r}_{k+1}$$

である。これを、もとの系(2－1)についての記法で書きなおすと、以下のようになる。

〔 Preconditioned CG Algorithm 〕

x_1 ; (2－1)に対する初期値

$r_1 = b - Ax_1$, $h_1 = M^{-1}r_1$, $p_1 = h_1$

$k = 1, 2, \cdots$ に対して $\qquad (2-4)$

$$\alpha_k = \frac{1}{(p_k, Ap_k)}, \qquad x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k Ap_k$$

$$h_k = M^{-1}r_{k+1}$$

$$\beta_k = \frac{1}{(r_{k+1}, h_k)}, \qquad p_{k+1} = p_k + \beta_k h_k$$

CG法のアルゴリズム(1－13)と、(2－4)を比較してわかるように、Preconditioning を考慮したとしても、アルゴリズムとしての違いは一ケ所 h_k の計算がはいってくるところである。新しい係数行列$\tilde{A}$を計算する必要はない。そこで、 h_k の計算のしやすさということも、Precon-ditioning matrix を作る時の一つの目安となる。それゆえ、Mとしては、三角行列の積ととることが行なわれている。CG法に対するよく知られた Preconditioning には、三角行列を用いる

SSOR型(Symmetric Successive Over-relaxation Methodに基づく)のもの(〔3〕で述べる)と、係数行列のsparse性を利用し、不完全三角行列分解したものからPreconditioning matrixをつくるもの(〔4〕で述べる)がある。

〔3〕 SSOR型のPreconditioning

〔2〕でみたことからもわかるように、一般に、よいPreconditioning matrixがもつ性質として、以下のことが考えられる。

(a) $M^{-1}A$ の spectral condition number が A より小さい。

(b) M をつくるのは簡単であること。

(c) M は A と同様の sparse 性をもつこと。

(d) M を係数行列とする連立一次方程式を解くことは、A を係数行列とするものより、ずっと簡単であること。

係数行列 A の分割に基づき、三角行列の積として Preconditioning matrix をつくるものを総称して、SSOR型[7]と呼ぶ。これは最初に O. Axelsson が SOR法の加速のための Preconditioning と、CG法のPreconditioningとの類似性を認めて、SSOR Preconditioningと名付けたことによる[5],[22]ようである。ここでは、この型に属するものの一つとして、高橋・野寺のPreconditioning[8],[9],[11]をあげておく。

係数行列 A を次のように分割する。

$$A=L+D+L^T \qquad (3-1)$$

ただし　D；対角行列

L；対角要素はすべて0の下半三角行列

そして、

$$A'=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}=I+L'+L'^T, \quad (I\text{；単位行列})$$

より、

$$C\equiv I+\omega L' , \qquad (3-2)$$

ただし　ω；SOR法加速パラメータ，　$0<\omega<2$

とし、

$$M^{-1}=(CC^T)^{-1} \qquad (3-3)$$

とおく。

A がsparseであるならば、C もまたsparseな三角行列である。ゆえに、Preconditioned CG Algorithm (2−4) における $h_k=M^{-1}r_{k+1}$ の計算は、次の2つの連立方程式を相ついで解くことに相当する。

1) $Cs = r_{k+1}$ （3－4）

2) $C^T h_k = s$

これらの連立一次方程式は、係数行列が三角行列であることから、簡単に解を求めることができる。

〔4〕 不完全分解によるPreconditioning

実の係数行列 A が

$$A = K - N \qquad (4-1)$$

ただし　K は正則かつ $K^{-1} \geq 0$

$N \geq 0$

であるよう分割できるとき、この分割を A の正則分割(regular splitting)といい、このK, N を用いて、一般に反復法は次のように定式化される。

$$Kx_{k+1} = Nx_k + b \qquad (4-2)$$

そしてこのとき、この反復法は収束するということが示されている。

たとえば

$A = L + D + U$

L；下半三角行列,　D；対角行列,　U；上半三角行列

と分割したとき、$K=D$, $N=-(L+D)$ とすると Jacobi 法であり、$K=D+L$, $N=-U$ ととると Gauss－Seidel 法, $K=\frac{1}{\omega}(D+\omega L)$, $N=\frac{1}{\omega}((1-\omega)D-\omega U)$ ととるのが SOR法である。

CG法はまたこの観点からみると、反復行列 $I-A$ とする Richardson法の変形版

$$x_{k+1} = (I-A)x_k + b \qquad (4-3)$$

に対する多項式加速のアルゴリズムの形となっている。$A=K-N$ と正則分割でき、このKを用いてあらたな係数行列を $K^{-1}A$ としたとき、これに対して

$$x_{k+1} = (I-K^{-1}A)x_k + K^{-1}b = K^{-1}Nx_k + K^{-1}b \qquad (4-4)$$

となる。これから、係数行列 Aに対し、正則分割したときの正則行列 K を Preconditioning matrix に用いると、反復法に対する議論がそのまま使え、収束性も保証される。Precondition－ed CG アルゴリズムとは、反復法 $x_{k+1} = K^{-1}Nx_k + K^{-1}b$ に対する多項式加速であるといえる。

さて、正定値である行列 A は、次のような三角行列の積に分解される。

$$A = LDU, \qquad (4-5)$$

$D = \mathrm{diag}(\rho_1, \rho_2, \cdots, \rho_n)$,　$\rho_i > 0$

L；対角要素１の下半三角行列,　U；対角要素１の上半三角行列

これに対し、不完全分解は、 $A=K-R$ と分解する方法である。Kは、以下のようにする。

$$K=\hat{L}\hat{D}\hat{U}, \qquad \hat{D}=\mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n), \quad \sigma_i>0 \tag{4-6}$$

$\hat{L}$; 対角要素１の下半三角行列

$\hat{U}$; 対角要素１の上半三角行列

Aが対称ならば $\hat{U}=\hat{L}^T$ となり不完全コレスキー分解である。$\hat{L}$, $\hat{D}$, $\hat{U}$ のとり方は次のようにする。

今、正定値対称行列 $A=(a_{ij})$ より、第一添字と第２添字の組を考え、次の２つの集合を考える。

$$\Pi=\{(i, j) \mid a_{ij} \neq 0\}$$
$$\Pi_0=\{(i, j) \mid a_{ij}=0\} \tag{4-7}$$

これらに基づき、 $\hat{L}$ と R（今、$\hat{U}=\hat{L}^T$ である）を、

$$\hat{L}=(\hat{l}_{ij}), \qquad \hat{l}_{ij}=0, \quad i<j$$
$$\hat{l}_{ij} \neq 0, \quad (i, j) \in \Pi \text{ かつ } i>j$$
$$\hat{l}_{ij}=0, \quad (i, j) \in \Pi_0 \text{ かつ } i>j \tag{4-8}$$
$$R=(r_{ij}), \qquad r_{ij}=0, \quad (i, j) \in \Pi$$
$$r_{ij} \neq 0, \quad (i, j) \in \Pi_0$$

この不完全コレスキー分解は、以下のようにして順次求めることができる。

$$\sigma_i=a_{ii}-\sum_{k=1}^{i-1} \hat{l}_{ik}^2 \sigma_k \tag{4-9}$$

$j=i+1, \cdots, n$ に対して

$$\hat{l}_{ji}=\begin{cases}(a_{ji}-\sum_{k=1}^{i-1} \hat{l}_{jk} \hat{l}_{ik} \sigma_k)/\sigma_i, & (j, i) \in \Pi \\ 0, & (j, i) \in \Pi_0\end{cases}$$

このように、係数行列 Aの非零要素の２つの添字の組の集合に基づいて、その上で行なう不完全分解は、Aが対称な M−matrix* であるなら、Aの正則分割であり、 $\sigma_i>0$, $i=1, 2, \cdots, n$ であるなら、作り方より一意である。このことを示したのは Meijerink と van der Vorst[2]

---

* 実行列 Aは、次の条件をみたすとき、M−matrix という。

$A=(a_{ij})$ としたとき、 $a_{ii}>0$, $a_{ij} \leq 0$, $i \neq j$, A ; 正則, $A^{-1} \geq 0$

であり、これらの結果を Manteuffel[10]が、H−matrix** に拡張できることを示している。

そこで、 Preconditioned CG Algorithm (2−4) の $h_k=M^{-1}r_{k+1}$ の計算についてSSOR型の場合と同様

$$C=\hat{L}\hat{D}^{\frac{1}{2}} \tag{4-10}$$

とおき、

$$M^{-1}=\left((\hat{L}\hat{D}^{\frac{1}{2}})(\hat{L}\hat{D}^{\frac{1}{2}})^{T}\right)^{-1} \tag{4-11}$$

とする。

〔5〕収束率について

CG法 (1−13) で x_k から x_{k+1} を計算したとき、誤差関数は

$$\varphi(x_k)-\varphi(x_{k+1})=\frac{(p_k,r_k)^2}{(p_k,Ap_k)}>0 \tag{5-1}$$

となり、必ず減少する。今、数列 $\{x_k\}$ 自身の収束をみるために、次のノルムを導入する。ベクトル u, 行列 B に対して

$$\|u\|_A=(u,Au)^{\frac{1}{2}},\qquad \|B\|_A=\sup\frac{\|Bv\|_A}{\|v\|_A},\quad \text{for } {}^{\forall}v. \tag{5-2}$$

真の解 $\bar{x}=A^{-1}b$ であるとして、誤差ベクトルを

$$\varepsilon^{(k)}=x_k-\bar{x} \tag{5-3}$$

とする。

CG法のアルゴリズムから、それは x_k に関する漸化式

$$x_{k+1}=\rho_{k+1}\{\gamma_{k+1}((I-A)x_k+b)+(1-\gamma_{k+1})x_k\}+(1-\rho_{k+1})x_{k-1} \tag{5-4}$$

ただし $\quad \gamma_{k+1}=\dfrac{(r_k,r_k)}{(r_k,Ar_k)}$

$$\rho_{k+1}=\left[1-\frac{1}{\rho_k}\cdot\frac{\gamma_{k+1}}{\gamma_k}\frac{(r_k,r_k)}{(r_{k-1},r_{k-1})}\right]^{-1},\qquad \rho_1=1$$

と書くことがわかる。$I-A$ は Richardson 法での反復行列である。さらにまた、任意の行列 G に対する行列多項式

** 行列 $A=(a_{ij})$ が、H−matrix であるとは、次のことがいえるときである。

行列 $B=(b_{ij})$ を $\quad b_{ii}=a_{ii}$

$$b_{ij}=-|a_{ij}|,\quad i\neq j$$

としてつくったとき、B はM−matrix である。

$Q_k(G) \equiv \alpha_{k,0} I + \alpha_{k,1} G + \cdots + \alpha_{k,k} G^k$, （実数 $\alpha_{n,i}$ は $\sum_{i=0}^{k} \alpha_{k,i} = 1$, $k = 0, 1, \cdots$ である）

とすると、上のことより $\varepsilon^{(k)}$ は次の形に書ける。

$$\varepsilon^{(k)} = Q_k(I - A)\varepsilon^{(0)} \qquad (5-5)$$

ゆえに、

$$\|\varepsilon^{(k)}\|_A = \|Q_k(I-A)\varepsilon^{(0)}\|_A \leq \|Q_k(I-A)\|_A \cdot \|\varepsilon^{(0)}\|_A \qquad (5-6)$$

A は正定値対称だから $Q_k(I-A)$ も対称、$AQ_k(I-A)A^{-1}$ も対称であるから、行列のノルムの性質より、A の固有値 λ_i, $i = 1, 2, \cdots, n$ とすると

$$\|Q_k(I-A)\|_A = \max_{1 \leq i \leq n} |Q_k(1-\lambda_i)| \qquad (5-7)$$

これから

$$\frac{\|\varepsilon^{(k)}\|_A}{\|\varepsilon^{(0)}\|_A} \leq \max_{1 \leq i \leq n} |Q_k(1-\lambda_i)| \qquad (5-8)$$

固有値を用いたもっと厳密な誤差の評価については、たとえば文献〔3〕,〔20〕等にその記述をゆずるとして、結果だけを述べると、$\kappa = \lambda_{\max} / \lambda_{\min}$ として

$$\frac{\|\varepsilon^{(k)}\|_A}{\|\varepsilon^{(0)}\|_A} \leq 2\left[\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k\right]^{-1} \qquad (5-9)$$

である。

〔6〕計算量削減の工夫

今まで述べてきたアルゴリズムをみればわかるように、CG法はそれぞれの反復において Ap_k という行列とベクトルの積を行なわねばならず、一反復での乗算回数は $O(2n^2)$ を下まわらない。我々は、この計算量を削減することも目的のひとつにして別の Preconditioning〔15〕を試みているが、最近、高橋・野寺の両氏が、係数行列の分割をうまく利用することにより、計算量を $O(n^2)$ におとすという画期的な方法〔14〕をだしたので、これについて若干ふれることにする。

係数行列 A の分割

$$A = L + D + L^T, \quad D \text{ ; 対角行列}, \quad L \text{ ; 対角要素 0 の下半三角行列} \qquad (6-1)$$

とし、A の不完全分解を

$$A=\hat{L}\hat{D}\hat{L}^{T}-R\,,\quad \hat{D}\ ;\ \text{対角行列} \qquad (6-2)$$

$\hat{L}$; 対角要素1の下半三角行列

R ; residual

とする。そしてこれらを用いて、SSOR型Preconditioning matrix を次のように書く。

$$M^{-1}=\left((\hat{D}+L)\hat{D}^{-1}(\hat{D}+L)^{T}\right)^{-1} \qquad (6-3)$$

もとの系に同値な系は、以下のものである。

$$(\hat{D}+L)^{-1}A(\hat{D}+L)^{-T}(\hat{D}+L)^{T}x=(\hat{D}+L)^{-1}b \qquad (6-4)$$

これを、$\tilde{A}\tilde{x}=\tilde{b}$ と書くと、任意の近似値 $\tilde{x}_i=(\hat{D}+L)^{T}x_i$, (x_i はもとの系での近似値) に対する残差 $\hat{r}_i$ は

$$\hat{r}_i=\tilde{b}-\tilde{A}\tilde{x}_i=(\hat{D}+L)^{-1}r_i \qquad (6-5)$$

$$r_i=b-Ax_i$$

ここで、

$$\hat{p}_i=(\hat{D}+L)^{T}p_i \qquad (6-6)$$

とおくと、Preconditioned CG Algorithm (2−4) は、次のように書きなおせる。

〔Modify 1〕

任意の初期値 ; x_1

$\hat{r}_1=b-\tilde{A}x_1$, $\hat{p}_1=\beta_1\hat{D}\hat{r}_1$

$k=1, 2, \cdots$ に対して

$$\alpha_k=\frac{1}{(\hat{p}_k\ ,\ \tilde{A}\hat{p}_k)}\,,\qquad x_{k+1}=x_k+\alpha_k(\hat{D}+L)^{-T}\hat{p}_k \qquad (6-7)$$

$$\hat{r}_{k+1}=\hat{r}_k-\alpha_k\tilde{A}\hat{p}_k$$

$$\hat{h}_k=\hat{D}\hat{r}_{k+1}$$

$$\beta_k=\frac{1}{(\hat{r}_{k+1}\ ,\ \hat{h}_k)}\,,\qquad \hat{p}_{k+1}=\hat{p}_k+\beta_k\hat{h}_k$$

ここで特に注目すべき点は、$(\hat{D}+L)^{-T}\hat{p}_k$ と $\tilde{A}\hat{p}_k$ の計算である。これについて、以下のように考える。

$$\tilde{A}\hat{p}_k=(\hat{D}+L)^{-1}A(\hat{D}+L)^{-T}\hat{p}_k=(\hat{D}+L)^{-1}(L+D+L^{T})(\hat{D}+L)^{-T}\hat{p}_k$$

$$=(\hat{D}+L)^{-1}\{(\hat{D}+L)+(\hat{D}+L^{T})-(2\hat{D}-D)\}(\hat{D}+L)^{-T}\hat{p}_k$$

$$=(\hat{D}+L)^{-T}\hat{p}_k+(\hat{D}+L)^{-1}\{\hat{p}_k-(2\hat{D}-D)(\hat{D}+L)^{-T}\hat{p}_k\} \qquad (6-8)$$

ここで、$t_k\equiv(\hat{D}+L)^{-T}\hat{p}_k$ とすると

$$q_k \equiv \tilde{A}\hat{p}_k = t_k + (\hat{D}+L)^{-1}\{\hat{p}_k - St_k\} \qquad (6-9)$$

ただし　　$S = 2\hat{D} - D$

ゆえに以上を整理すると、Preconditioned CG Algorithm は次のように修正される。

〔Modify 2〕

任意の初期値　x_1

$\hat{r}_1 = b - \tilde{A}x_1$,　$\hat{p}_1 = \beta_1 \hat{D}\hat{r}_1$

$k = 1, 2, \cdots$ に対して

$t_k = (\hat{D}+L)^{-T}\hat{p}_k$

$q_k = t_k + (\hat{D}+L)^{-1}(\hat{p}_k - St_k)$

$$\alpha_k = \frac{1}{(\hat{p}_k, q_k)}, \qquad x_{k+1} = x_k + \alpha_k t_k \qquad (6-10)$$

$$\hat{r}_{k+1} = \hat{r}_k - \alpha_k q_k$$

$$\hat{h}_k = \hat{D}\hat{r}_{k+1}$$

$$\beta_k = \frac{1}{(\hat{r}_{k+1}, \hat{h}_k)}, \qquad \hat{p}_{k+1} = \hat{p}_k + \beta_k \hat{h}_k$$

このようにすると、Aが $n\times n$ の要素を全部もつ行列(full matrix) であるとしたときに、t_k に対する計算において、三角行列を係数行列とする連立方程式を解くために乗算が $n(n-1)/2$, q_k の計算において St_k でn回、連立方程式を解くのに $n(n-1)/2$ 回、そして $\hat{D}$ は対角行列だから $\hat{D}\hat{r}_{k+1}$ の計算でn回、1回の反復における乗算の総計算量は $n^2 + 6n + 2$ となる。修正以前のアルゴリズム(2−4)(高橋・野寺の Preconditioning で)では、それは $2n^2 + 4n + 2$ であるから、演算量の減少は目をみはるものであるということがわかろう。

計算のために必要な記憶容量(working storage) の方面からみるとどうなるかをみてみよう。各反復において、SSOR型 Preconditioned CGアルゴリズムでは、(3−4)での s と h_{k+1} は重ねることが可能であるから、x, r, p, h, Ap の5本のベクトルを準備することを必要とする。修正したアルゴリズムでは、$\hat{h}_k$ は t_k 又は q_k に重ねることが可能であるから、用意すべきベクトルの数は、前と同様5本であるが、まえもって$\hat{D}$とSを記憶しておくための2本のベクトルが必要とされよう。しかし、演算量の減少による経済性の方が、多くの場合には有効であろう。

〔7〕数　値　例

ここで、Preconditioning の効果をみるために、簡単な数値例をあげておく。CG法のアルゴ

リズム（1−14）とPreconditioned CG法（2−4）を用い、高橋・野寺のPreconditioning で解いたときの反復回数の比較例である。このとき $\omega=1.0$ とした。

テストした係数行列は次の2つの場合である。

(a) $A=(a_{ij})$,

$$a_{ij}=n-|i-j|$$

(b) $A=$

$$\begin{pmatrix} 4 & & & & & \\ -1 & 4 & & & \text{対称} & \\ -1 & -1 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & -1 & -1 & 4 \end{pmatrix}$$

収束判定は $(r_k, r_k) \leq (10^{-6})^2$ とした。

I. (a)の場合

Aの次元数	反復回数	
	CG法（1−14）のアルゴリズム	Precond. CG法（2−4）のアルゴリズム
50	38	6
100	68	8
150	99	8
200	125	8
250	158	8

II. (b)の場合

Aの次元数	反復回数	
	CG法（1−14）のアルゴリズム	Precond. CG法（2−4）のアルゴリズム
50	34	20
100	62	32
150	90	45
200	118	57
250	146	68

例(a)の場合には高橋・野寺の Preconditioning はすばらしい効果を発揮する。(b)の場合については、CG法の約1/2の回数であるが、今1.0と固定したパラメータ ω は $0<\omega<2$ の範囲動かせるので、より少ない回数とすることは可能である。

〔8〕おわりに

以上が、連立一次方程式の反復解法（とくに共役勾配法）に対する Preconditioning と計算量削減の方法の概略である。大規模な構造解析等でみられる大次元疎行列を係数行列とする連立一次方程式は、行列が疎であるということと、Preconditioning によって、その収束を速めることができる。しかし、連立一次方程式の解法に対して行う Preconditioning というのは、注意深くほどこすと非常な優位性を発揮するものであるが、乱暴にとりあつかうと、かえって、問題の条件をより悪くする可能性もあるということは、注意しておかなければならない。その収束が反復行列の

固有値の分布に依存してくることは、収束率のところでみたとおりである。

ここでは係数行列 A は、正定値対称に限って述べてきた。行列のクラスをひろげた場合におけるCG法，Preconditioning の方法の種々等については、文献を参照されたい。

〔参 考 文 献〕

1) M. R. Hestenes, E. Stiefel : Methods of Conjugate Gradients for Solving Linear Systems, Nat. Bur. Standards J., Res 49, 1952, pp 409−436.

2) J. A. Meijerink, H. A. van der Vorst : An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-matrix, Math. of Comp., Vol. 31, 1977, pp 148−162.

3) A. Greenbaum : Comparison of Splitting used with the Conjugate Gradient Algorithm, Numer. Math., Vol. 33, 1979, pp 181−194.

4) A. Bjork : Solving Linear Least Squares Problems by Gram−Schmidt Orthogonalization, BIT, 1967.

5) O. Axelsson : Solution of Linear Systems of Equations, Lecture Note in Math., No. 572, Springer−Verlag, 1977, pp 1−51.

6) I. Gustafsson : A Class of First Order Factorization Methods, BIT, Vol. 18, 1978, pp 142−156.

7) O. Axelsson : On Preconditioned Conjugate Gradient Methods, AERE Report 9636, 1979, pp 24−35.

8) H. Takahashi, T. Nodera : New Variants of the Conjugate Gradient Algorithm, International Congress on Numerical Methods for Engineering in Paris, Dunod, 1980, pp 209−219.

9) H. Takahashi, T. Nodera : Preconditioned Conjugate Gradient Algorithms for Nonsymmetric Matrix , Keio Math. Sem. Rep., 1981, pp 51−70.

10) T. A. Manteuffel : An Incomplete Factorization Technique for Positive Definite Linear Systems, Math. of Comp., Vol. 34, No. 150, 1980, pp 473−497.

11) T. Nodera, H. Takahashi : Preconditioned Conjugate Gradient Algorithm for Solving Biharmonic Equation, 4−th IMACS International Symposium on Computer Methods for Partial Differential Equation, 1981.

12) P. Concus, G. H. Golub, D. P. O'Leary : A Generalized Conjugate

Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations, Sparse Matrix Computation (ed) J. R. Bunch, D. J. Rose, Academic-Press, 1976, pp 309-332.

13) J. Douglas Jr, T. Dupont ; Preconditioned Conjugate Gradient Iteration Applied to Galerkin Methods for a Mildly Nonlinear Dirichlet Problem, Sparse Matrix Computation (ed) J. R. Bunch, D. J. Rose, Academic Press, 1976, pp 333-348.

14) 高橋秀俊, 野寺隆 ; PCGアルゴリズムの効果的な実現の一考察, 情報処理学会第24回全国大会論文集, 1982, pp 901-902.

15) 都田艶子 ; CG法に対する一つの前処理, 情報処理学会第24回全国大会論文集, 1982, pp 899-900.

16) R. S. Varga ; Matrix Iterative Analysis, Prentice-Hall, 1962.

17) D. M. Young ; Iterative Solution of Large Linear Systems, Academic-Press, 1971.

18) 戸川隼人 ; 共役勾配法, 教育出版, 1977.

19) M. R. Hestenes ; Conjugate Direction Methods in Optimization, Springer-Verlag, 1980.

20) L. A. Hageman, D. M. Young ; Applied Iterative Methods, Academic-Press, 1981.

21) J. K. Reid ; On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations, Proc. Conference on Large Sparse Sets of Linear Equations, Academic-Press, 1971, pp 231-254.

22) O. Axelsson ; A Generalized SSOR Method, BIT, Vol.12, 1972, pp 443-467.