

ノルム前処理付き共役勾配法の性能評価

池田 優介[†] 藤野 清次^{††} 山田 知典^{†††}

共役勾配 (CG) 法は収束性を向上させるために前処理を行って使用されることが多い。現在最も多く用いられている前処理として不完全コレスキー (IC) 分解が挙げられる。しかし、この前処理には逐次的な演算が含まれるため並列計算機上で高速化を図るのが難しいことが知られている。一方、プログラミングの手間も少なく並列化も比較的容易な前処理として対角スケールリング法がある。さらに、共著者の1人によってノルムスケールリングという新しい前処理法が提案された。そこで、本発表では従来の対角スケールリング法とノルムスケールリング法の収束性を比較評価することにする。

Performance Estimation of a Norm-Scaled CG method

YUSUKE IKEDA,[†] SEIJI FUJINO^{††} and TOMONORI YAMADA ^{†††}

Conjugate Gradient method is often used with preconditioning for the purpose of gaining high convergence. Incomplete Cholesky decomposition is most effective among a variety of preconditioner in view of reduction of iterations. The sequential process, however, is involved in the IC decomposition, and is not suited to parallelism. Therefore, diagonal scaling is effective as a preconditioner of CG method. Similarly, a norm scaling technique newly proposed has the same property of diagonal scaling. In this article, performance of norm-scaled CG method will be verified.

1. はじめに

科学技術計算分野に現れる問題は、最終的に大規模連立1次方程式 $Ax = b$ を解くことに帰着されることが多い。特に、方程式の係数行列 A が対称正定値の場合、共役勾配 (Conjugate Gradient) 法⁵⁾ がよく用いられる。一般に CG 法は収束性を向上させるために行列に前処理を行ってから用られる。よく使用される前処理には、対角項のみを扱う対角スケールリング (Diagonal Scaling) 法¹⁾⁴⁾、非対角項も考慮する不完全コレスキー (Incomplete Cholesky) 分解法⁸⁾、重みつき ICCG 法⁶⁾、あるいは近似逆行列²⁾ を用いるものなど様々な前処理法がある。IC 分解法を用いると収束までの反復回数は大幅に減少するものの、演算の中に逐次的な計算が含まれているため並列化に余り向いていない。一方、対角スケールリング法は、IC 分解に比べて反復回数の減少の割合は少ないが、逐次的な計算がないので並列化に向いている。そのためいろいろな改良が加えられて現在もよく使用されている⁹⁾。

本研究で取り上げたノルムスケールリング (Norm

Scaling) 法の考え方は、対角スケールリング法の考え方をさらに発展させたものである¹⁰⁾。対角スケールリングでは行列の対角要素が1になるように係数の大きさを揃えるのに対して、新しいノルムスケールリング法では行及び列の2-ノルムが1になるように係数を正規化するところだけが異なる。対角項だけを前処理として使用するので、並列化に向けた前処理法であることは変わらない。本研究では、Matrix Market に収められている行列を評価用行列として取り上げ、ノルムスケールリング法と従来の対角スケールリング法を前処理として使用したときの CG 法の収束性の比較検討を行った。

2. 共役勾配法

連立一次方程式 $Ax = b$ を CG 法で解くことを考える。ここで、行列 A は $n \times n$ 対称正定値行列、 x, b は各々解ベクトルおよび右辺ベクトルを表す。CG 法の算法を以下に示す。収束判定用の ε は微小値である。

[CG 法の算法]

初期値 x_0 を与える

$$r_0 = b - Ax_0, p_0 = r_0$$

for $k = 0, 1, \dots$

$$\alpha_k = \frac{(r_k, r_k)}{(p_k, Ap_k)}$$

[†] 九州大学大学院システム情報科学府
Kyushu University

^{††} 九州大学情報基盤センター
Kyushu University

^{†††} 理化学研究所 ものつくり情報技術統合化研究プログラム
RIKEN

$x_{k+1} = x_k + \alpha_k p_k$
 $r_{k+1} = r_k - \alpha_k A p_k$
 if $\|r_{k+1}\|_2 / \|r_0\|_2 \leq \epsilon v$ stop
 $\beta_k = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)}$
 $p_{k+1} = r_{k+1} + \beta_k p_k$
 end.

3. 前処理と対角スケーリング CG 法

CG 法の収束性は解くべき方程式の係数行列の固有値の分布に大きく依存し、固有値が広く分布するより密集している方が CG 法は速く収束することが多い。そこで、CG 法が速く収束するように元の方程式の係数行列 $A = (a_{ij})$ を変換することを前処理と呼ぶ。

特に、対角スケーリング法では、係数行列の対角項がすべて 1 になるように正規化する。すなわち、前処理行列 $D = (d_{ij})$ を次のように定義する。

$$d_{ij} = \begin{cases} \sqrt{a_{ii}}, & (i=j \text{ のとき}) \\ 0 & (i \neq j \text{ のとき}) \end{cases} \quad (1)$$

このように定めた前処理行列 D を用いて、係数行列 A を次のように変換する。ここで、変換後の係数行列を $B = (b_{ij})$ とする。

$$B = DAD^T. \quad (2)$$

ただし、 T は転置を表す。変換された行列 B についても元の行列 A が持つ対称正定値性などの性質がそのまま保持されるので CG 法が適用できる。

3.1 対角スケーリング CG 法の算法

対角スケーリング CG 法の算法を以下に示す。

初期値 x_0 を与える
 $D^{-1} = \text{diag}[1/a_{11}, 1/a_{22}, \dots, 1/a_{nn}]$
 $r_0 = b - Ax_0$, $p_0 = D^{-1}r_0$
 for $k = 0, 1, \dots$
 $\alpha_k = \frac{(r_k, D^{-1}r_k)}{(p_k, Ap_k)}$
 $x_{k+1} = x_k + \alpha_k p_k$
 $r_{k+1} = r_k - \alpha_k A p_k$
 if $\|r_{k+1}\|_2 / \|r_0\|_2 \leq \epsilon$ stop
 $\beta_k = \frac{(r_{k+1}, D^{-1}r_{k+1})}{(r_k, D^{-1}r_k)}$
 $p_{k+1} = D^{-1}r_{k+1} + \beta_k p_k$
 end.

4. ノルムスケーリングによる前処理

ノルムスケーリングでは係数行列の各行のノルムが 1 になるように正規化する。ここで、あるベクトル $x = (x_1, x_2, \dots, x_n)^T$ のノルム (norm) とは以下の

ものをさす。

$$\text{norm}(x) = \left(\sum x_i \cdot x_i \right)^{1/2} \quad (3)$$

4.1 ノルムスケーリング前処理の計算手順

ここでは、対称行列を扱っているので、前処理も対称性を失わないような正規化が望ましい。具体的な計算手順は (2) 式による変換とよく似た以下のようなものになる。すなわち、(2) 式において変換行列 B の成分 b_{ij} は次のように表せる。

$$b_{ij} = \frac{a_{ij}}{d_{ii}d_{jj}} \quad (4)$$

ノルムスケーリング前処理では、変換後の行列のノルムが 1 になることから次のようにおける。

$$\begin{aligned} \text{norm}(B_i) &= \sqrt{\sum_j (b_{ij}b_{ij})} \\ &= \sqrt{\sum_j \left(\frac{a_{ij}}{d_{ii}d_{jj}} \right)^2} \\ &= 1 \end{aligned} \quad (5)$$

この式から変換行列 D の要素 d_{ii} を求められればよい。そのためには以下の非線形方程式を解けばよい。

$$d_{ii} = \sqrt{\sum_j \left(\frac{a_{ij}}{d_{jj}} \right)^2} \quad (6)$$

この方程式を解くために次の反復繰り返し計算を行う。

(1) 反復の初期値 $d_{ij}^{(0)}$ に対角スケーリングのときと同じ値を与える。

$$d_{ij}^{(0)} = \begin{cases} \sqrt{a_{ii}} & (i=j \text{ のとき}) \\ 0 & (i \neq j \text{ のとき}) \end{cases} \quad (7)$$

(2) その後、以下の式で変換行列 D の要素 d_{ii} を更新する。 ($k = 0, 1, 2, \dots$)

$$d_{ii}^{(k+1)} = \sqrt{\sum_j \left(\frac{a_{ij}}{d_{jj}^{(k)}} \right)^2} \quad (8)$$

書き下すと以下のように表せる。

$$\begin{aligned} d_{11} &\leftarrow \left(\left(\frac{a_{11}}{d_{11}} \right)^2 + \left(\frac{a_{12}}{d_{22}} \right)^2 + \dots + \left(\frac{a_{1n}}{d_{nn}} \right)^2 \right)^{1/2} \\ d_{22} &\leftarrow \left(\left(\frac{a_{21}}{d_{11}} \right)^2 + \left(\frac{a_{22}}{d_{22}} \right)^2 + \dots + \left(\frac{a_{2n}}{d_{nn}} \right)^2 \right)^{1/2} \\ &\vdots \\ d_{nn} &\leftarrow \left(\left(\frac{a_{n1}}{d_{11}} \right)^2 + \left(\frac{a_{n2}}{d_{22}} \right)^2 + \dots + \left(\frac{a_{nn}}{d_{nn}} \right)^2 \right)^{1/2} \end{aligned}$$

この繰り返し計算は非線形方程式の解を厳密に求めるものではないので、ある程度の収束条件を満たしたとき反復を止める。

[補足]

実際には、前処理にかかる計算時間を減らすために、 $p_{ii} = 1/d_{ii}^2$ とおき以下の繰返し計算を逐次的に行なっている。

$$\begin{aligned} p_{11} &\leftarrow 1/(a_{11}^2 p_{11} + a_{12}^2 p_{22} + \dots + a_{1n}^2 p_{nn}) \\ p_{22} &\leftarrow 1/(a_{21}^2 p_{11} + a_{22}^2 p_{22} + \dots + a_{2n}^2 p_{nn}) \\ &\vdots \\ p_{nn} &\leftarrow 1/(a_{n1}^2 p_{11} + a_{n2}^2 p_{22} + \dots + a_{nn}^2 p_{nn}) \end{aligned}$$

4.2 収束判定について

ノルムスケーリングの収束判定条件については以下の2通りの方法を検討した。ただし、 n は行列の次元数とする。 ϵ は微小な値とする。

[方法 1]

この方法は、変換された行列 B のノルムを計算し、以下のように誤差の平均値と 1 との差が微小値 ϵ 以下になったとき収束とみなすものである¹⁰⁾。

$$\left| \frac{\sum_i |B_i^{(k)}|}{n} - 1 \right| \leq \epsilon \quad (9)$$

ただし、Matrix Market のように次元数がいろいろ異なる行列を調べるとき、この評価では n の値によって ϵ の値のとり方が難しく適当でないことがわかった。

[方法 2]

この方法は、ノルムスケーリングの初期値として採用した対角スケーリングのときの各行のノルムの平均値を初期ノルムとする相対ノルムの値で収束判定を行なうものである。上付き添字 (k) は繰り返し回数を表す。

$$\left| \frac{\frac{\sum_i |B_i^{(k)}|}{n} - 1}{\frac{\sum_i |B_i^{(0)}|}{n} - 1} \right| \leq \epsilon \quad (10)$$

これは相対値に基づく判定であるので、次元数 n の大きさによらずに ϵ の値を定めることができる。実際には、定数 (3 ~ 5) 回で反復計算を打ち切った。

5. 数値実験

5.1 計算機環境と計算条件

数値実験は、プロセッサ Pentium 4、クロック数 2GHz、メモリ 512MB 上で行った。プログラム言語は Visual C++ 6.0、演算はすべて倍精度で行った。初期値 x_0 はすべて零、厳密解 x はすべて 1 とした。CG 法の収束判定条件は相対残差 $\frac{\|r_k\|_2}{\|r_0\|_2}$ が 10^{-9} 以下になったときとし、反復回数が次元数と同じ n 回になったとき計算を終了した。

5.2 数値実験 1

5.2.1 テスト行列について

テスト用の行列として Matrix Market⁶⁾ を採用した。Matrix Market とは、自然科学あるいは工学の分野で実際に現れた問題を 20 年近くに亘って蓄積されてきた行列群をさし、いまでは URL などを通じて簡単にダウンロードできる。それらは、例えば線形方程式の解法の評価などでよく利用される。この中から、構造解析 (以下、bcsstk 系の行列と呼ぶ) の分野から 4 つの行列を、さらに、偏微分方程式の分野から行列を 1 つ (行列 nos5)、合計 5 つの問題を選出した。bcsstk 系の行列は次元も大きく一行当たりの平均非零要素数も多い。一方、行列 nos5 は bcsstk 系の行列より次元も平均非零要素数も少ないのが特徴的である。これらの行列に対して、前処理として、(i) 対角スケールリ

ング (以下、対角 S 前処理と略す) 及び (ii) ノルムスケーリング (以下、ノルム S 前処理と略す) を施した行列に CG 法を適用し収束性を調べた。表 5.1 に行列の次元数と一行当たりの平均非零要素数を、表 5.2 に 2 つの分野の行列の解析内容の主な特徴 (Matrix Market から引用) を示す。また、図 5.1 から図 5.5 に各行列の非零要素分布を示す。

表 5.1: 行列の次元数と一行当たりの平均非零要素数

行列	次元数	非零要素数
bcsstk15	3948	29.84
bcsstk17	10974	39.06
bcsstk18	11948	12.48
bcsstk25	15439	16.34
nos5	468	11.05

表 5.2: 各行列の分野とその名称および主な解析内容

分野	行列	解析内容
構造解析	bcsstk15	海上プラットフォームのモジュール解析
	bcsstk17	高圧力容器解析
	bcsstk18	ギーナ原子力発電所
	bcsstk21	固有値解析
偏微分方程式	nos5	重調和方程式

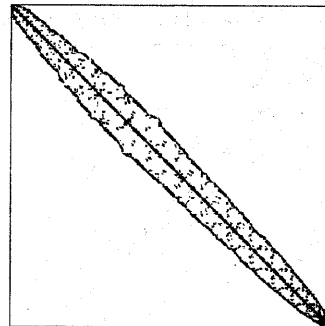


図 5.1: 行列 bcsstk15 の非零要素の分布

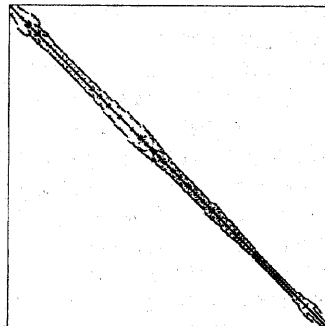


図 5.2: 行列 bcsstk17 の非零要素の分布

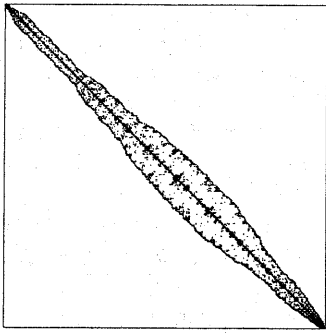


図 5.3: 行列 bcsstk18 の非零要素の分布

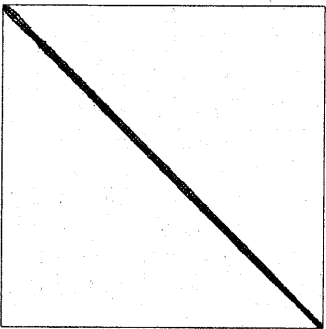


図 5.4: 行列 bcsstk25 の非零要素の分布

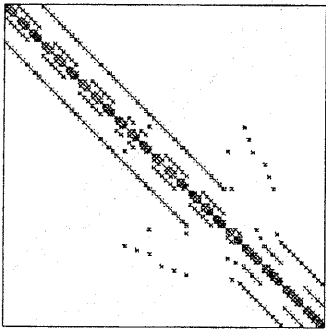


図 5.5: 行列 nos5 の非零要素の分布

5.2.2 実験結果と考察

表 5.3 に各前処理つき CG 法の収束までの反復回数と対角スケージングの反復回数を 1 としたときのノルム S 前処理の比率を示す。“ノルム S”の欄には前処理が 3 回の場合と同 5 回の場合の CG 法の反復回数を示す。構造解析の bcsstk 系の行列の比率がすべて 1.0 以下であったのに対し、行列 nos5 だけが 1.0 より僅かに大きな値になったことがわかる。ノルム S 前処理の反復計算 1 回に要する時間は、CG 法自体の反復計算 1 回に要する時間に比べて約 2.2~2.5 倍多くかかる。また、表 5.1 に示した 1 行当たりの平均非零要素数の大小とかなり相関関係があることがわかる。すなわち、ノルム S 前処理が非零要素数が多い行列

bcsstk15, bcsstk17, bcsstk25 に対して効果があったのに対して、それが比較的少なかった行列 bcsstk18 では同程度、行列 nos5 に対しては逆に収束性が僅かに悪くなった。

図 5.6 に行列 bcsstk25 に対する 3 つの前処理つき CG 法の収束過程の履歴を示す。図の縦軸には相対残差を常用対数目盛で表し、横軸には反復回数をとった。図では、対角 S の結果を実線で、ノルム S の結果のうち前処理 3 回の結果を破線で、そして前処理 5 回の結果を点線で表す。

表 5.3: 各前処理つき CG 法の反復回数

行列	対角 S	ノルム S	
		反復 3 回	反復 5 回
bcsstk15	561	532 (0.95)	528 (0.94)
bcsstk17	2815	2702 (0.96)	2684 (0.95)
bcsstk18	1205	1192 (0.99)	1161 (0.96)
bcsstk25	11740	10802 (0.92)	10721 (0.91)
nos5	235	243 (1.03)	244 (1.04)

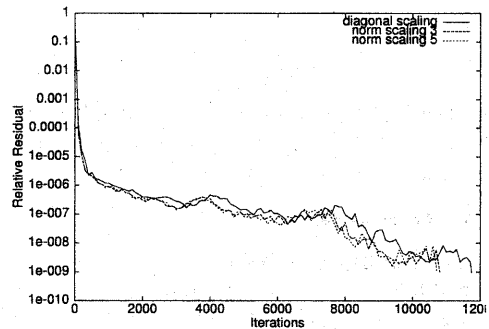


図 5.6: 行列 bcsstk25 に対する三つの解法の収束履歴

次に、ノルム S 前処理のとき、ノルムの平均値 $\frac{\sum_i |B_i^d|}{n}$ がどのように変わるかを表 5.4 に示す。なお、ノルム S 前処理では、対角スケージングのときに使った値を初期値としている。

表 5.4: ノルム S 前処理における反復回数が初期値および同 3, 5 回するときのノルムの値の変化

反復	ノルム S 前処理		
	初期値	3 回	5 回
bcsstk15	1.2207	0.9456	0.9747
bcsstk17	1.3650	0.9514	0.9820
bcsstk18	1.1720	0.9603	0.9835
bcsstk25	1.2414	0.9575	0.9855
nos5	1.1490	0.9570	0.9758

ノルム S 前処理により、CG 法の収束性の改善効果があまりなかった行列: bcsstk18 と nos5 のノルムの値は、それ以外の行列のノルムの値に比べて、初期値の段階でノルムの値が比較的 1.0 に近い値になっている。また、4.1 節で述べたノルムの計算方法では 2 乗

和をとっているため、1行当たりの非零要素数が多くなればなる程ノルムの値は大きく評価される傾向がある。また、対角S前処理をノルムS前処理の初期値として使っているの、そのときのノルムの値は1よりも大きな値になる。ノルムS前処理により、CG法の収束までの反復回数が減少した bcsstk 系の行列のうち、行列：bcsstk15, bcsstk17, bcsstk25 は次元数も大きく、1行当たりの非零要素数も多い。したがって、一行当たりの非零要素数が多い行列ほど、ノルムS前処理によりCG法の収束性向上の度合いが大きいと思われる。

5.3 数値実験 2

前小節の数値実験1の結果から、1行当たりの非零要素数が多い場合にノルムS前処理がCG法の収束性にとって有効なのは？という予想が成り立つ。そこで、その予想を確かめるために、以下のように元の行列の要素の値の一部だけを修正した新しい行列を考え、それをテスト行列として使用した。ただ、元の行列、すなわち、bcsstk 系の行列および行列 nos5 の非対角項の要素の値を修正するのは難しく現実的でない。ここでは、以下のように対角項のみに1.0よりも小さな値(ここでは、0.9とした)を掛けた行列に対して、ノルムS前処理つきCG法の収束性を調べた。元の行列の要素を a_{ij} 、重みを掛けた行列の要素を \hat{a}_{ij} で表す。

$$\hat{a}_{ij} = \begin{cases} 0.9 \times a_{ii} & (i = j \text{ のとき}) \\ a_{ij} & (i \neq j \text{ のとき}) \end{cases} \quad (11)$$

この操作によって、ノルムS前処理におけるノルムの値は次のように評価でき、元の行列のときのノルムの値より大きな値が得られる。このようにすることにより、平均非零要素数が多い行列の場合と同じような状況を作り出したことになる。

$$\begin{aligned} \text{norm}(\hat{B}_i) &= \sqrt{\sum_j (b_{ij} b_{ij})} \\ &= \sqrt{\sum_j \left(\frac{a_{ij}}{d_{ii} d_{jj}} \right)^2} \\ &= \sqrt{\sum_j \left(\frac{a_{ij}}{\hat{a}_{ii} \hat{a}_{jj}} \right)^2} \\ &= \sqrt{\sum_j \left(\frac{a_{ij}}{0.9^2 a_{ii} a_{jj}} \right)^2} \\ &> \sqrt{\sum_j \left(\frac{a_{ij}}{a_{ii} a_{jj}} \right)^2} \end{aligned}$$

元の行列：bcsstk18 と nos5 に対して、上記の係数操作を施した行列を各々 **bcsstk18-改** と **nos5-改** と呼ぶことにする。元の行列および行列 **bcsstk18-改** と **nos5-改** に対する実験結果を表5.5と表5.6に示す。対角S前処理のときと同じ値をノルムS前処理の初期値とし、そのときのノルムの値を"*"印つきで対角S前処理の欄に記した。行列：bcsstk18から**bcsstk18-改**

の場合には、ノルムの値が、1.1720から1.2075に、同じく行列：nos5から**nos5-改**の場合には、1.1490から1.1801に、各々大きくなったことがわかる。さらに、行列 **bcsstk18-改** では、収束性の改善の度合いが大きくなった。一方、行列：**nos5-改** ではノルムS前処理の効果が現れるようになった。

表 5.5: 行列 **bcsstk18** と **bcsstk18-改** に対するノルムの値、CG法の反復回数そして比率の変化の様子

	対角 S	ノルム S 前処理	
		3 回	5 回
bcsstk18			
ノルム	*1.1720	0.9603	0.9835
反復回数	1205	1192	1161
比率	1.00	0.99	0.96
bcsstk18-改			
ノルム	*1.2075	0.9612	0.9853
反復回数	10757	9971	9687
比率	1.00	0.93	0.90

表 5.6: 行列 nos5 と **nos5-改** に対するノルムの値、CG法の反復回数そして比率の変化の様子

	対角 S	ノルム S 前処理	
		3 回	5 回
nos5			
ノルム	*1.1490	0.9570	0.9758
反復回数	235	243	244
比率	1.00	1.03	1.04
nos5-改			
ノルム	*1.1801	0.9554	0.9750
反復回数	231	216	217
比率	1.00	0.94	0.94

行列 **bcsstk18-改** と **nos5-改** に対する対角S前処理およびノルムS前処理つきCG法の収束の履歴を図5.7と図5.8に示す。対角S前処理つきCG法の収束の履歴を実線で、ノルムS前処理つきCG法の反復3回のときの同履歴を破線で、そして反復5回のときの同履歴を点線で各々表す。

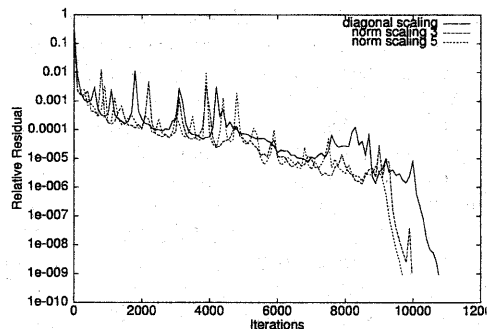


図 5.6: 行列 **bcsstk18-改** に対する対角S前処理およびノルムS前処理つきCG法の収束の履歴

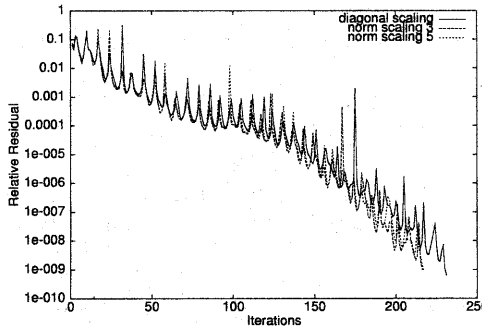


図 5.6: 行列 nos5-改に対する対角 S 前処理およびノルム S 前処理つき CG 法の収束の履歴

6. おわりに

対角 S 前処理およびノルム S 前処理を施した行列に対する CG 法の収束性を調べた。その結果、ノルム S 前処理は構造解析分野の問題に対して有効であることが再確認された。特に、構造解析分野の問題のように、1 行当たりの非零要素数が多い行列ほどノルム S 前処理は有効に働いた。また、対角 S 前処理のときと同じ値を初期値とするノルム S 前処理では、反復の初期値のノルムの平均値が 1 よりも大きい程、ノルム S 前処理による利得が大きいこともわかった。これを元の行列の対角成分だけを小さくした行列で確かめた。

謝辞

本研究を進めるにあたり、東京大学 矢川元基教授、国立情報学研究所 速水謙教授、図書館情報大学長谷川秀彦助教授にご協力と多くのご助言を戴きました。心より感謝の念を表します。

参 考 文 献

- 1) Bauer, F., *Optically Scaled Matrices*, Numer. Math., Vol.5, pp.73-87, 1963.
- 2) Benzi, M., Meyer, C., Tuma, M., *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., Vol.17, No.5, pp.1135-1149, 1996.
- 3) Davis, T., *University of Florida Sparse Matrix Collection*, NA Digest, Vol.94, Issue 42, Oct. 1994. <http://www.cise.ufl.edu/~davis/sparse/>
- 4) 速水謙, 原田紀夫: 対角項スケールリングを施した共役勾配法のベクトル計算機における有効性について, 情報処理学会論文誌, Vol.30, No.11, pp.1364-1375, 1989.
- 5) Hestenes, M., Stiefel, E., *Method of Conjugate Gradient for Solving Linear Systems*, J. Res. Nat. Stand., No.49, pp.409-436, 1952.
- 6) 池田優介, *Matrix Market* における共役勾配法の収束性評価, 九州大学工学部電気情報工学科卒業研究, 2002.3.

- 7) *Matrix Market Collection*: <http://math.nist.gov/MatrixMarket/>
- 8) Meijerink, J., van der Vorst, H.A., *Iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., Vol.31, pp.148-162, 1977.
- 9) 塩谷隆二, 金山寛, 田上大助, 荻野正雄: バランシング領域分割法による 3 次元大規模構造解析, Trans. of JSCES, No.20000017, 2000.
- 10) 山田知典, 矢川元基: ノルム前処理付き共役勾配法による有限要素解析, Trans. of JSCES, No.20010033, 2001.