

対角要素を修正するCrout版ILU前処理の収束性評価

藤原, 牧

九州大学大学院システム情報科学府情報工学専攻 : 修士課程

吉田, 正浩

九州大学大学院システム情報科学府情報工学専攻 : 修士課程

藤野, 清次

情報基盤センター

<https://doi.org/10.15017/1516230>

出版情報 : 九州大学大学院システム情報科学紀要. 11 (1), pp. 45-50, 2006-03-24. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

対角要素を修正する Crout 版 ILU 前処理の収束性評価

藤原牧*・吉田正浩*・藤野清次**

Convergence Estimation of Crout Version of ILU Preconditioner by Means of Modification for Diagonal Entries

Maki FUJIWARA, Masahiro YOSHIDA and Seiji FUJINO

(Received December 9, 2005)

Abstract: This paper presents variants of the Crout version of incomplete LU (ILU) decomposition with modification for diagonal entries. As variants of ILUC decomposition, we considered single and double compensated modifications for diagonal entries. Furthermore we also examined two type of modifications with and without normalization for each fillin. As a result, numerical experiments for realistic electromagnetic problem indicates that GPBi-CG and BiCGSafe methods with ILUC decomposition with single compensated modification without normalization for diagonal entries perform well.

Keywords: Crout version of ILU decomposition, Preconditioning, BiCGSafe method, BiCGStab method, GPBi-CG method, Modification for diagonal entries

1. はじめに

大型で疎な非対称行列を係数行列 A として持つ連立一次方程式は前処理つき双共役勾配法(以下, BiCG法と略す)系統の反復法によって解かれることが多く, 今までに様々な反復法が提案され, 効率的で収束が安定な反復法が研究されてきた¹⁷⁾. その中でも一般化積型BiCG法(以下, GPBi-CG法と呼ぶ)¹⁹⁾は, 通常ランチョス多項式に加速多項式を掛けて反復法の収束を加速させるという着想に基づき考案された反復法であり, 多くの実際問題の求解に適用されてきた. また, 著者達もGPBi-CG法の収束安全性をさらに増したBiCGSafe法を新しく提案した⁵⁾.

一方, 前処理(preconditioning)は反復法の収束性を大幅に向上させる有力かつ必須なものとしていまや確立されている. 多くの前処理が提案されているが, 本研究では最近提案されかつ非対称行列用の不完全分解の中のCrout版のILU分解(以下, ILUC分解と呼ぶ)に注目した. この分解は前処理行列の近似精度を調節する閾値というパラメータを1個使用する. 対称正定値行列に対しては元の係数行列 A の対角要素を修正するロバストなRIC分解がよく知られている¹⁾. またパラメータを自動調整する機能を持つRIC分解も提案され, 実際問題に対してその有効性が実証されてきた⁹⁾. 一方, 非対称行列に対しても, より対角優位な行列にするために, 元の係数行列 A の対角要素を修正する分解法が研究されている²⁾. また, 加速係数を対角要素に掛ける前処理もよく知られているが調節すべきパラ

メータの数が1個から2個に増加するという新たな問題も生まれてきた. そこで, 本研究では, ILUC分解の収束性をさらに向上させ, かつ調節すべきパラメータの数を増やさないという方針に基づき, 棄却したフィルインに対応して対角要素を修正する方法の有効性について検証する.

本論文の構成は次のとおりである. 第2節でCrout版ILU分解の概要について記述する. 次に, 第3節で, 前処理の中で閾値により棄却するフィルインが生じた場合に対応する対角要素を1個だけ修正する前処理を提案する. また, 対称正定値行列用のRIC分解を拡張し, 棄却するフィルインに対応した対角要素2個を修正する分解についても検討する. 第4節では, パラメータの数が1個だけの前処理付きの反復法の収束性を調べて, 対角要素を修正するタイプの不完全分解法の有用性を比較実証する. 第5節ではまとめと今後の課題を述べる.

2. Crout版ILU分解

Fig. 1にCrout版ILU分解(以下, ILUC分解と略す)の概念図を示す. Fig. 1(a)に更新対象の第 k 行そして Fig. 1(b)に同じく更新対象の第 k 列の場合を示す. 更新対象の行および列を太い実線で囲った. また図中の点線はフィルインに対応する更新の計算で使用される行または列を各々表す. このとき, 上三角行列 U は行方向にフィルインが格納されるのに対して, 下三角行列 L は列方向にフィルインが格納される. このようにフィルインを格納することによりフィルインの並び替えも必要がなくなる. また, 行列 U の要素は第 k 行よりも上方に位置する第1行 ~ 第 $k-1$ 行の要素を使って更新されるのに対して, 行列 L の要素は第 k 列よりも左側に位置する第1列 ~ 第 $k-1$ 列の要素

平成17年12月9日受付

* 情報工学専攻修士課程

** 情報基盤センター

素を使って更新される。また、下向きの矢印“↓”および右向きの矢印“⇒”は、ILUC分解の処理が進行する方向を各々表す。すなわち、ILUC分解の場合、上三角行列 U では分解は上から下に進み、一方下三角行列 L では分解は左から右に進むことになる。

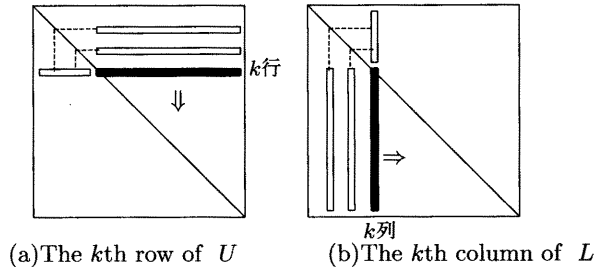


Fig.1 Schematic procedure of Crout version of ILU decomposition.

3. 単一補償および二重補償ILUC分解

ここでは、前処理中に棄却されたフィルインに基づいて対角要素を修正する手法について考える。対称正定値行列の場合、IC分解の途中で分解の破綻が起きないように改良された対称正定値行列用の前処理であるロバストIC分解¹⁾(Robust IC分解、以下RIC分解と呼ぶ)がある。RIC分解では、係数行列 A を

$$A \approx U^T U - R - R^T - D \quad (1)$$

と不完全分解する。ここで、行列 U は分解後の上三角行列を表し、行列 R は誤差行列を表す。 T は行列の転置を表す。さらに、対角行列 D は、行列 U のフィルインが棄却されたとき行列 U の対角要素の符号が負にならないように対角要素を修正する働きをする。

3.1 単一補償ILUC分解

非対称行列の場合、前処理中に棄却されたフィルインに基づいて対角要素を修正するILUC前処理について考える。ここでは、係数行列 A を

$$A \approx LU \quad (2)$$

のように、上三角行列 U と下三角行列 L を使って不完全分解する。sc_ILUC分解のアルゴリズムを以下に示す。アルゴリズム中の式(9)と式(14)がフィルインに対応する対角要素を各々修正する部分に該当する。またフィルイン z_j に対して正規化(normalization)する場合を式(7)に、正規化しない場合を式(8)に示す。フィルイン w_j に対する場合も同様で、各々式(12)と式(13)に示す。また、変数 z_k は式(4)と式(15)からわかるように上三角行列 U の対角要素 u_{kk} の作業用の一時的変数を表す。

[sc_ILUC分解のアルゴリズム]

```

 $u_{kk} = a_{kk} (= 1.0), (k = 1, \dots, n)$ 
 $l_{kk} = 1.0 (k = 1, \dots, n)$ 
do  $k = 1, \dots, n$ 
   $z_k = u_{kk}$ 
   $z_j = a_{kj} (j = k + 1, \dots, n)$ 
  do  $i = 1, \dots, k - 1; l_{ki} \neq 0$ 
     $z_j = z_j - l_{ki} u_{ij} (j = k, \dots, n; u_{ij} \neq 0)$ 
  end do
  do  $j = k + 1, \dots, n$ 
     $\zeta = |z_j| / \sqrt{z_k}$  : 正規化するとき
    ( $\zeta = |z_j|$  : 正規化しないとき)
    if  $\zeta < \text{tol}$  then
       $z_j = 0, z_k = (1 + \zeta) z_k,$ 
    end if
  end do
   $w_j = a_{jk} (j = k + 1, \dots, n)$ 
  do  $i = 1, \dots, k - 1; u_{ik} \neq 0$ 
     $w_j = w_j - l_{ji} u_{ik} (j = k + 1, \dots, n; l_{ji} \neq 0)$ 
  end do
  do  $j = k + 1, \dots, n$ 
     $\zeta = |w_j| / \sqrt{l_{kk}}$  : 正規化するとき
    ( $\zeta = |w_j|$  : 正規化しないとき)
    if  $\zeta < \text{tol}$  then
       $w_j = 0, l_{kk} = (1 + \zeta) l_{kk},$ 
    end if
  end do
   $u_{kk} = z_k$ 
  do  $j = k + 1, \dots, n$ 
     $u_{kj} = z_j, l_{jk} = w_j / u_{kk}$ 
  end do
end do

```

Fig. 2に、sc_ILUC分解の場合の、前処理中で上三角行列 U のフィルイン u_{kj} (“o”印)が棄却されたとき、 U の対角項である u_{kk} (“●”印)が修正され、同様に、前処理中に下三角行列 L のフィルイン l_{hk} (“o”印)が棄却されたとき、下三角行列 L の対角要素である l_{kk} (“●”印)が修正されるときに対応関係を表す。このように一つのフィルインにつき対角要素1個を修正する分解を単一補償(single-compensated) ILUC分解 (以下、sc_ILUC分解と略す)と呼ぶ。sc_ILUC分解は、棄却されたフィルインに基づいて対角要素を修正することによって、前処理行列をより対角優位な行列に近づける働きがある。

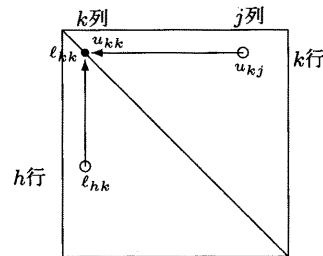


Fig.2 Correspondence between rejected fill-ins (u_{kj} and l_{hk}) and modified diagonal (u_{kk} and l_{kk}) entries for upper (lower) triangular matrix U (L) in case of single-compensated ILUC decomposition.

3.2 二重補償ILUC分解

対称行列に対するRIC分解における対角要素の補償アルゴリズムを単純に拡張すると以下に示すようなアルゴリズムが得られる。この場合、一つのフィルインにつき対角要素が2個修正されることになる。そこで、このアルゴリズムを二重補償 (double-compensated) ILUC分解(以下, dc_ILUC分解と略す)と呼ぶ。

dc_補償ILUC分解の場合, sc_補償ILUC分解のアルゴリズムにおいて, 上三角行列 U において生じたフィルイン z_j に対する対角要素 z_k と u_{jj} (“●”印)の修正手順は以下ようになる。すなわち, 正規化の場合はsc_補償ILUC分解のアルゴリズム中の式(7)が式(17)に, 正規化しない場合は式(8)が式(18)に修正される。いずれの場合も式(9)の後に式(19)が追加される。

$$\zeta = |z_j|/\sqrt{z_k u_{jj}} \quad \text{: 正規化するとき} \quad (17)$$

$$(\zeta = |z_j| \quad \text{: 正規化しないとき}) \quad (18)$$

if $\zeta < \text{tol}$ then

$$z_j = 0, \quad z_k = (1 + \zeta)z_k, \quad (19)$$

$$u_{jj} = (1 + \zeta)u_{jj}$$

end if

同様に, dc_補償ILUC分解の場合, 下三角行列 L において生じたフィルイン w_j に対する対角要素 l_{kk} と l_{jj} (“●”印)の修正手順は以下ようになる。すなわち, 正規化の場合はsc_補償ILUC分解のアルゴリズム中の式(12)が式(20)に, 正規化しない場合は式(13)が式(21)に修正される。いずれの場合も式(14)の後に式(22)が追加される。

$$\zeta = |w_j|/\sqrt{l_{kk}l_{jj}} \quad \text{: 正規化するとき} \quad (20)$$

$$(\zeta = |w_j| \quad \text{: 正規化しないとき}) \quad (21)$$

if $\zeta < \text{tol}$ then

$$w_j = 0, \quad l_{kk} = (1 + \zeta)l_{kk}, \quad (22)$$

$$l_{jj} = (1 + \zeta)l_{jj}$$

end if

Fig. 3に, dc_ILUC分解の場合の, 前処理中で上三角行列 U のフィルイン u_{ij} (“○”印)が棄却されたとき, U の対角項である u_{kk} と u_{jj} の2箇所 (“●”印)が修正され, 同様に, 前処理中に下三角行列 L のフィルイン l_{hk} (“○”印)が棄却されたとき, 下三角行列 L の対角要素である l_{kk} と l_{jj} の2箇所 (“●”印)が修正される時の対応関係を表す。

4. 数値実験

4.1 計算機環境と計算条件

数値実験はCPUにPentium4を搭載したPC(クロック周波数 3.2 GHz, メモリ: 2 GB, OS: Red Hat Linux version 7.1)を使用し, 計算はすべて倍精度演算で行った。コンパイラはIntel Fortran compiler ver. 8.0を用い, 最適化オプションは-O3を使用した。反復法の収束判定は l_2

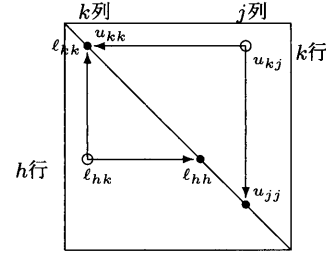


Fig.3 Correspondence between rejected fill-ins (u_{kj} and l_{hk}) and modified diagonal entries (u_{kk}, u_{jj}, l_{kk} and l_{hh}) for upper (lower) triangular matrix U (L) in case of double-compensated ILUC decomposition.

ノルムの残差の比が $\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 \leq 10^{-7}$ を満たしたときとした。方程式の右辺項は基礎方程式から得られる値を使用し, 初期近似解 \mathbf{x}_0 はすべて0, 最大反復回数は10000回とした。初期シャドウ残差ベクトル \mathbf{r}_0^* には一様乱数を代入した。行列は予め対角スケーリングによって対角項をすべて1に正規化した。反復法はBiCGStab法, GPBi-CG法とBiCGSafe法の三つの反復法を用いて, 前処理には加速係数(= γ とする)つきILU前処理, sc_ILUC前処理とdc_ILUC前処理の3種類の前処理に対して数値実験を行った。ここで, ILU前処理とは元の係数行列 A の非零要素と同じ位置に生じたフィルインのみ考慮する前処理を意味し, これは加速係数 $\gamma = 1$ のときに対応する。

4.2 テスト問題

箔要素を用いて積層鉄心の表面層を考慮した磁気軸受の渦電流解析から生じた行列を取り扱った^{12) 14)}。一般に, 電気電子機器では, 渦電流損を低減するため, 薄い鋼板を積層した積層鉄心がよく用いられる。そして, 高い設計精度を確保するため, その表面を研磨する必要がある。このとき, 場合によっては, 層間の絶縁が破れ, この部分に流れる渦電流のために被る損失を極力抑えることが重要になる。この非常に薄い表面層を有限要素法によりモデル化した場合, 要素形状が歪で扁平になることが多い。そのとき係数行列の性質も悪くなり反復法がなかなか収束しない事態が生じる。事実, 今回の行列の場合も, 通常の不完了LU分解つき反復法と対角要素を修正しない元のILUC分解つき反復法では両法とも収束しなかった。本研究で扱う直流場渦電流問題の行列は次元数 27720, 非零要素数 1665372, 1行当たりの平均非零要素数は約60.08の行列である。また解くべき3次元非線形渦電流解析の基礎方程式は以下のように表される。

$$\mathbf{w} = \sigma \left(\frac{\partial \mathbf{A}}{\partial t} + \text{grad } \phi \right) \text{ とおくとき} \quad (23)$$

$$\text{rot}(\nu \text{rot } \mathbf{A}) = \mathbf{J}_0 - \mathbf{w} \quad (23)$$

$$\text{div}(-\mathbf{w}) = 0 \quad (24)$$

ここで、 \mathbf{A} , ϕ および \mathbf{J}_0 は磁気ベクトルポテンシャル, 電気スカラーポテンシャル, 強制電流密度を各々表す. また, ν と σ は磁気抵抗率, 導電率を各々表す.

4.3 計算結果

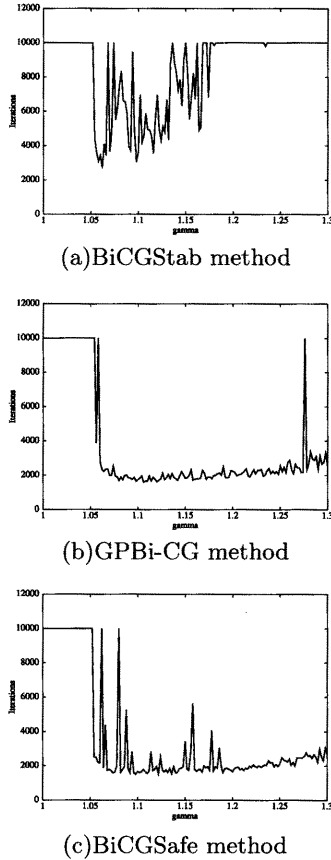
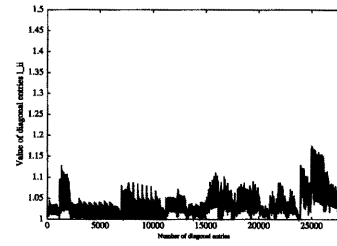


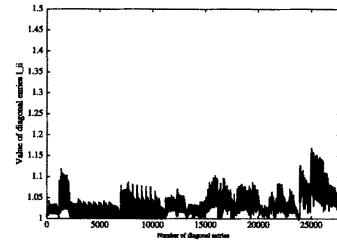
Fig.4 Convergence of the iterative methods with usual ILU decomposition when parameter γ was varied (a solid line means a case with τ_0^* = random number).

Fig. 4に加速係数 γ を対角項に掛けて不完全分解の破綻を回避したILU分解つき反復法の数値実験結果を示す. 図の横軸は加速係数の値, 縦軸は収束までの反復回数を表す. 最大反復回数は10000回である. 加速係数は1.0から1.3まで0.002刻みで合計151通り変化させ三つの反復法の収束性を調べ, 加速係数 γ ごとの反復回数を実線でプロットした. これら三つの図から以下のことが観察される.

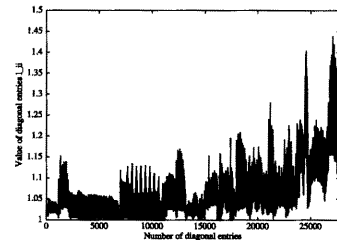
- 他の反復法に比べてBiCGStab法の収束性が悪い.
- 加速係数 γ がおよそ1.06よりも大きな値のとき反復法が収束するが, 過度に大きいと収束しない.
- いずれのILU分解つき反復法でも収束回数が加速係数によって大きく変動する.
- 加速係数 γ がおよそ1.06から1.15の区間の値のとき,



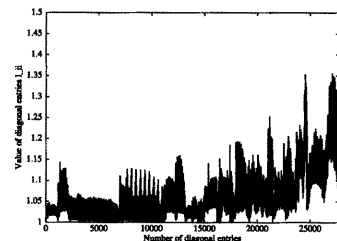
(a)sc_ILUC with normalization



(b)sc_ILUC without normalization



(c)dc_ILUC with normalization



(d)dc_ILUC without normalization

Fig.5 Plots of diagonal entries l_{ii} of lower triangular matrix L in the single-compensated and double-compensated ILUC decomposition.

反復回数は約1700~2400回程度で少ない.

Fig. 5に, sc_ILUC分解およびdc_ILUC分解のときの下三角行列 L の対角要素 l_{ii} をプロットした.図の横軸は下三角行列 L の対角要素 l_{ii} の番号, 同じく図の縦軸は対角要素 l_{ii} の値を各々表す. sc_ILUC分解のときの方がdc_ILUC分解のときよりも全般に値が小さいことがわかる. また正規化の有無はほとんど影響していないように思われる.

次に, Table 1に加速係数 γ が区間[1.052, 1.180]のときのILU前処理付き反復法の収束性を比較した. この場合試行回数は全部で65回である. この区間内の加速係数 γ だけについて統計数字をまとめたのは, Fig. 4(a)に示した結果からわかるように, 加速係数 γ が区間[1.052, 1.180]の外側ではBiCGStab法がまったく収束しなかったためであ

る。表の中で、“% of Conv.”は全試行回数に対する収束した試行回数の割合を表す。また“Ave. Itr.”は収束した場合のみの平均反復回数、同様に“Ave. Time”は収束した場合のみの平均計算時間[単位：秒]を表す。また、括弧内の数字はBiCGStab法の結果を1.0としたときのGPBi-CG法とBiCGSafe法の平均反復回数と平均計算時間の比率を各々表す。

Table 1 Comparison of convergence of the iterative methods with usual ILU decomposition for parameters γ included in the interval of [1.052, 1.180].

Method	% of Conv.	Ave. Itr.	Ave. Time
BiCGStab	54/65 (83.1%)	5723 (1.0)	174.3 (1.0)
GPBi-CG	62/65 (95.4%)	1955 (0.341)	64.2 (0.368)
BiCGSafe	62/65 (95.4%)	2074 (0.362)	65.6 (0.376)

Table 2(a)にフィルインに正規化した場合のsc_ILUC分解付き反復法の結果を、同様に**Table 2(b)**に正規化しない場合のsc_ILUC分解付き反復法の結果を各々示す。また**Table 3(a)**にフィルインに正規化した場合のdc_ILUC分解付き反復法の結果を、同様に**Table 3(b)**に正規化しない場合のdc_ILUC分解付き反復法の結果を各々示す。表中の“Pre-t”は前処理にかかった時間[単位：秒]，“Itr-t”は反復計算にかかった時間[同秒]，“Tot-t”は前処理時間と反復時間の合計の時間[同秒]を示す。“Ave.”は反復1回当たりの平均時間[同ミリ秒]を示す。閾値 (= Tolとする)は0.005, 0.003, 0.001, 0.0008の4通りについて数値実験を行った。また各反復法で合計時間が最も短かったときの結果を太字で表した。

これらの表から以下のことがわかる。

- sc_ILUC分解の方がdc_ILUC分解よりもいずれの反復法でもまた閾値の大小にかかわらず収束までの合計時間が短い。反復回数もおおよそ305~320回程度で収束した。
- sc_ILUC分解とdc_ILUC分解いずれの場合も、閾値が0.001程度のときが収束までの合計時間が最も短いことが多い。
- sc_ILUC分解とdc_ILUC分解いずれの場合も、閾値が大きくなってもGPBi-CG法とBiCGSafe法では反復回数の増加の比率はそれほど大きくないが、BiCGStab法では反復回数が急に増加する。特に、dc_ILUC分解の場合のBiCGStab法の反復回数の増加が著しい。
- sc_ILUC分解の場合、閾値を変化させてもGPBi-CG

Table 2 Comparison of convergence of the iterative methods with sc_ILUC decomposition.

Tol	Method	Itr.	Pre-t	Itr-t	Tot-t	Ave.
.005	BiCGStab	1523	5.8	50.7	56.5	33.3
	GPBi-CG	765	5.7	27.1	32.8	35.4
	BiCGSafe	796	5.8	27.3	33.1	34.3
.003	BiCGStab	850	6.9	31.8	38.7	37.5
	GPBi-CG	772	6.9	30.8	37.7	39.9
	BiCGSafe	753	6.9	29.3	36.2	38.9
.001	BiCGStab	318	12.8	16.0	28.8	50.3
	GPBi-CG	305	12.0	15.9	27.9	52.2
	BiCGSafe	320	12.1	16.4	28.5	51.2
.0008	BiCGStab	299	13.8	15.8	29.5	52.7
	GPBi-CG	300	13.7	16.4	30.1	54.5
	BiCGSafe	315	14.1	17.0	31.1	53.9

(a)with normalization

Tol	Method	Itr.	Pre-t	Itr-t	Tot-t	Ave.
.005	BiCGStab	1669	5.3	55.8	61.1	33.4
	GPBi-CG	787	5.3	27.9	33.2	35.4
	BiCGSafe	764	5.3	26.3	31.7	34.5
.003	BiCGStab	1065	6.4	40.9	46.8	37.9
	GPBi-CG	621	6.4	24.7	31.1	39.8
	BiCGSafe	586	6.4	22.8	29.2	38.9
.001	BiCGStab	372	11.1	18.5	29.7	49.8
	GPBi-CG	304	11.2	15.8	27.0	52.0
	BiCGSafe	287	11.5	14.6	26.1	50.9
.0008	BiCGStab	285	12.8	15.0	27.8	52.7
	GPBi-CG	282	12.8	15.4	28.1	54.5
	BiCGSafe	281	12.8	15.1	27.9	53.7

(b)without normalization

法とBiCGSafe法の収束までの合計時間は30秒台と非常に安定している。

- 反復1回当たりの反復時間の差は三つの反復法ではあまり見られない。
- 閾値が小さいとき、前処理時間と収束までの反復時間はほぼ同程度の大きさになる。
- 閾値が小さいとき、すなわち前処理行列の近似精度が高いとき、三つの反復法の収束性の違いはほとんど見られない。

5. ま と め

Crout版ILU分解において、フィルインに対応する対角要素を1箇所だけ修正するsc_ILUC分解を新しく提案し、数値実験にてその有効性を検証した。比較した前処理は、調節すべきパラメータ(加速係数)が1個の従来の加速係数つきILU分解そして対称正定値行列用のRIC分解を拡

Table 3 Comparison of convergence of the iterative methods with dc_ILUC decomposition.

Tol	Method	Itr.	Pre-t	Itr-t	Tot-t	Ave.
.005	BiCGStab	8068	5.9	267.3	273.1	33.1
	GPBi-CG	1755	5.5	62.0	67.5	35.3
	BiCGSafe	1508	5.7	51.3	57.0	34.0
.003	BiCGStab	3207	6.4	120.5	127.1	37.6
	GPBi-CG	1158	6.7	45.7	52.4	39.5
	BiCGSafe	1209	6.6	46.6	53.2	38.5
.001	BiCGStab	674	12.6	35.5	48.0	52.6
	GPBi-CG	583	12.2	30.5	42.8	52.4
	BiCGSafe	576	12.3	29.6	41.9	51.4
.0008	BiCGStab	588	13.3	31.1	44.4	52.8
	GPBi-CG	546	13.4	30.0	43.4	54.9
	BiCGSafe	549	13.3	29.7	42.9	54.1

(a)with normalization

Tol	Method	Itr.	Pre-t	Itr-t	Tot-t	Ave.
.005	BiCGStab	6356	5.6	215.3	221.0	33.9
	GPBi-CG	1470	5.6	52.7	58.3	35.9
	BiCGSafe	1491	5.5	51.8	57.3	34.8
.003	BiCGStab	3124	6.7	120.0	126.7	38.4
	GPBi-CG	1105	6.7	44.7	51.4	40.5
	BiCGSafe	1096	6.7	43.2	49.9	39.4
.001	BiCGStab	642	11.6	32.5	44.2	50.7
	GPBi-CG	576	11.9	30.3	42.2	52.7
	BiCGSafe	562	11.7	29.0	40.7	51.6
.0008	BiCGStab	670	13.1	35.8	48.9	53.4
	GPBi-CG	539	13.3	29.9	43.2	55.4
	BiCGSafe	530	13.1	28.9	42.1	54.6

(b)without normalization

張して、フィルインに対応する対角要素を2箇所修正する dc_ILUC前処理の2種類である。その結果、収束までの計算時間および閾値の変化に対する安定性の面で、比較した二つの前処理に比べてsc_ILUC分解の優位性が明らかになった。今後は、sc_ILUC分解の理論の確立およびより多くの事例についてsc_ILUC分解の有効性示していきたい。

参 考 文 献

- 1) Ajiz, M. A., Jennings, A., A robust incomplete Choleski-conjugate gradient algorithm, *Int. J. Numer. Methods Engrg.*, **20**(1984), 949-966.
- 2) Chow, E., Saad, Y., Experimental Study of ILU Preconditioners for Indefinite Matrices, *J. Comput. Appl. Math.*, **86**(1997), 387-414.

- 3) Fletcher, R.: Conjugate Gradient preconditioning for indefinite systems, *Lecture Notes in Mathematics* **506**(1976), 73-89.
- 4) Fujino, S.: Complexity on the order of polynomial sequence used in GPBi-CG preconditioning and its effect to convergence, *Trans. of INFORMATION*, **6**(2003), 13-23.
- 5) 藤野清次, 藤原 牧, 吉田正浩: 準残差の最小化に基づく積型 BiCG法, *日本計算工学会論文集*, 2005. <http://save.k.u-tokyo.ac.jp/jsces/trans/trans2005/No.20050028.pdf>
- 6) 藤原牧, 藤野清次, 吉田正浩, 藤原耕二, 村松和弘: 二重の安全収束性を有する前処理つきBiCGSafe法について, 第8回環瀬戸内シンポジウム予稿集, (2005), 78-83.
- 7) 藤原牧, 吉田正浩, 藤野清次: 棄却した要素に基づいて対角項を修正するCrout版ILU-BiCGSafe法の収束性評価, 第9回環瀬戸内シンポジウム予稿集, pp.58-63, 金沢大学, 2005.11.12-14.
- 8) Li, N., Saad, Y., and Chow, E., Crout version of ILU for general sparse matrices, *SIAM J. Sci. Comput.*, **25**(2003), 716-728.
- 9) 柿原正伸, 藤野清次: 緩和係数 ω を自動決定する対角緩和準ロバストICCG法の収束性, *情報処理学会論文集 コンピュータシステム*, Vol.46 No.SIG4(ACS9) (2005), 45-55.
- 10) 巻幡憲俊, 宇都宮智昭, 渡邊英一: 波浪解析問題のための境界要素法へのILUCの適用, *応用力学論文集 土木学会*, **7**(2004), 279-284.
- 11) 巻幡憲俊, 宇都宮智昭, 渡邊英一: OSP-ILUCの非エルミート密行列への適用性に関する固有値分布からの検討, 第8回環瀬戸内シンポジウム予稿集, 43-48, 松山, 2005.
- 12) Muramatsu, K., Takahashi, N., et al., Three-dimensional steady-state eddy-current analysis of moving conductor using edge elements and moving-coordinate system, *IEEE Trans. on Magnetics*, **38**(2002), 597-600.
- 13) Saad, Y., *Iterative Methods for Sparse Linear Systems*, SIAM Philadelphia, 2003.
- 14) 清水露規, 村松和弘, 亀有昭久 他: 箔要素を用いて積層鉄心の表面層を考慮した磁気軸受の渦電流解析, *電気学会研究会資料 SA-05-18*, (2005), 105-110.
- 15) 後保範, ベクトル計算機向きICCG法, *京都大学数理解析研究所講究録 No.514*, 1984, 110-134.
- 16) van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **13**(1992), 631-644.
- 17) van der Vorst, H.A.: *Iterative Krylov preconditionings for large linear systems*, Cambridge University Press, Cambridge, 2003.
- 18) 山口遼一, 野寺隆: 大規模行列系に対する頑強な2階ILU分解前処理, *情報処理学会研究会報告 2005-HPC-103*, 7-12, 2005.
- 19) Zhang, S.-L.: GPBi-CG: Generalized product-type preconditionings based on Bi-CG for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **18**(1997), 537-551.

