

収束の三重の安全鍵を与える Crout 版 ILU 分解つき BiCGSafe 法

藤原 牧[†] 吉田 正 浩[†] 藤野 清 次^{††}

本論文では、Crout 版 ILU 分解を前処理に用いた BiCGStab 法、GPBi-CG 法そして BiCGSafe 法の安全収束性について評価する。また、積型 BiCG 法において初期条件として与える初期シャドウ残差ベクトル r_0^* の与え方を変えたときの反復法の収束性の違いを調べる。数値実験を通して、(i) 適切な前処理、(ii) 適切な反復法そして (iii) 適切な r_0^* の与え方を示し、収束性を大きく左右するこれら 3 つの鍵を適切に選ぶことにより安全な収束が実現することを明らかにする。

BiCGSafe Method with Crout Version of ILU Decomposition Giving Triple Safe Keys for Convergence

MAKI FUJIWARA,[†] MASAHIRO YOSHIDA[†] and SEIJI FUJINO^{††}

In this paper, we study the performance of BiCGStab, GPBi-CG and BiCGSafe methods preconditioned with ILU decomposition. The robustness of such methods depends on the following *three keys*: (1) stability of preconditioning, (2) robustness of iterative method, and (3) initial choice of shadow residual vector for iterative solutions, r_0^* . Good combination of these three keys typically leverages the performance. For the first key, we develop a Crout version of drop-threshold ILU (ILUC) with diagonal magnification parameter for robust and efficient preconditioning. The effectivenesses of these three keys are investigated through the experiments of a suite of sparse matrices from real applications including a modeling of electromagnetic-mechanical applications. Our preliminary results demonstrate that our ILUC-BiCGSafe scheme with random vector for r_0^* is very robust and efficient, indicating the best triple keys.

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³⁾ と略す）システムの反復法によってよく解かれる。また、問題の規模が大きくなればなるほど、前処理つき反復法に対してその高速性と収束の安定性が要求されるのは必然であるといえる。そのため、いままで数多くの前処理法と反復法が各々提案されてきた^{4),15),19)}。またよく知られているように、本論文で扱う一般化積型 BiCG 法（一般に GPBi-CG 法と呼ばれる²⁰⁾）は、通常のランチョス多項式（ $= R_{n+1}(\lambda)$ ）に加速多項式（ $= H_{n+1}(\lambda)$ ）を掛けて、積の形の多項式列を構成して反復法の収束を加速させる、という着想に基づく反復

法である。ここで、 λ は固有値を表す。また、従来の多くの積型反復法では、算法中に含まれる加速パラメータ ζ_n, η_n は、算法を構成する主要なベクトルの 1 つである真の残差ベクトル r_{n+1} （ $\equiv H_{n+1}(\lambda)R_{n+1}(\lambda)$ ）のノルムの最小化から決定されてきた。このアイデアの採用により、ランチョス多項式だけで算法を構成する場合に比べて、収束の安定性を大幅に増すことができた^{18),19)}。そのため、安定化版の最初の反復法である BiCGStab（下線は Stabilized の略）法は、その名前からも分かるように安定化双共役勾配法と呼ばれる。

一方、算法中に陽には現れないが数学的に真の残差ベクトル r_{n+1} に近い準残差ベクトルと呼ばれるベクトル $a_n r_n$ （ $\equiv H_{n+1}(\lambda)R_n(\lambda)$ ）のノルムの最小化から加速パラメータ ζ_n, η_n を決定するという新しい戦略が最近提案された⁶⁾。真の残差ベクトル r_{n+1} のノルムの最小化から得られる安定収束（stable convergence）と対比させて、新算法が持つ収束性は安全収束（safe convergence）と呼ばれた。そして数値実験を通して BiCGSafe 法が持つ安全収束性が明示された。

さらに、前処理の研究においては、最近 Crout 版

[†] 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical
Engineering, Kyushu University

^{††} 九州大学情報基盤センター
Computing and Communications Center, Kyushu Uni-
versity

の ILU 分解が提案された^{8),9)} .そして反復法 GMRES 法¹³⁾ の前処理として用いられ性能評価が行われた .さらに, 土木工学分野の密な係数行列を持つ問題に対しても固有値解析などにより反復法として GMRES 法を使った場合の性能評価が行われた¹⁰⁾ .この前処理は閾値 (tolerance value) に基づきフィルインを決定する分解法の 1 つであり, 従来の前処理が持つ問題点を解消したとされる .しかし, Crout 版 ILU 分解つき積型 BiCG 法の性能評価は未着手である⁷⁾ .

本論文では, 非常に解き難いとされる直流場渦電流解析のある問題に対して, Crout 版 ILU 分解を前処理に用いた BiCGStab 法¹⁸⁾ と GPBi-CG 法²⁰⁾ そして BiCGSafe 法⁶⁾ の安全収束性について評価する .また, 積型 BiCG 法において初期条件として与える初期シャドウ残差ベクトル r_0^* の与え方を変えて反復法の収束性の違いを調べる .本論文の目的は, (i) 適切な前処理, (ii) 適切な反復法そして, (iii) 適切な初期シャドウ残差ベクトル r_0^* の与え方を示し, 収束性を大きく左右するこれら三重の安全鍵を適切に選ぶことにより, 安全な収束が得られることを示すことである .

本論文の構成は次のとおりである .まず, 2 章において閾値を用いる前処理の問題点について述べる .次に, 3 章において加速係数 γ つき Crout 版 ILU 分解について記述する .4 章においては, 前処理つき BiCGSafe 法の算法および準残差による加速パラメータ ζ_n, η_n の決定法について述べる .5 章で数値実験の概要とその結果を記述する .取り上げた問題は電磁界解析における磁気軸受の渦電流問題である .数値実験を通して Crout 版 ILU 分解つき BiCGSafe 法の安全収束性について議論とその有用性の検証を行う .さらに, 疎行列データベースから選んだ 3 つの行列もテストし, 前処理部分の実装が難しく一般の不規則疎行列のとき前処理行列の作成に時間がかかり効率が悪いとされてきた ILUT(Lfil, tol) 分解の実装方法についても考察した .最後に 6 章でまとめを行う .

2. 従来の前処理とその問題点

よく知られているように, その処理の簡便さから係数行列 A の非零要素とその位置が一致するフィルインのみを採用する ILU(0) 分解がある .また, フィルインが発生するレベルを指定するタイプの ILU(level) 分解もよく使われる .さらに, 閾値に基づきフィルインを決定する分解もある .その代表的な方法に, ILUT(Lfil, tol) (the dual Threshold Incomplete LU factorization) 分解^{14),15)} がある .この分解法は行ごとに行列の上方から下方に向かって分解処理が行われる .ILU(0)

や ILU(level) 分解よりも多くのフィルインを考慮できるため, 収束性が大幅に向上することもあるが, 有限要素法などで現れる非零要素が不規則分布する行列の場合, 実装面においていくつかの問題点が報告されている¹⁵⁾ .そのため, 本研究の動機にもなった考案者らのグループによる論文⁸⁾ では, 分解において新たに発生したフィルインを収容する適切な配列位置の探索 (原論文⁸⁾ では “the cost of searching may make the factorization too expensive” と記述) の計算コストの高さ, また巻幅らの報告⁹⁾ による, 分解処理が下方に進行するに従って近似の「品質」が次第に低下することが報告されている .

3. 加速係数 γ つき Crout 版 ILU 分解

前述の ILUT 分解の改良版が Crout 版 ILU 分解^{8),9)} である .閾値 (= tol とする) と加速係数 (以下に示す分解中の下線を付けた式 (2) の係数 γ を指す) を用いた加速係数 γ つき Crout 版 ILU 分解の算法を以下に示す .解くべき線形方程式を $Ax = b$ とするとき, 行列 $A(= (a_{ij})) \in R^{N \times N}$, x, b は N 次元の解ベクトルと右辺ベクトルとする .また, $A \approx LU$ と不完全分解され, $L = (l_{ij})$, $U = (u_{ij})$ とする . z_j, d_k, w_j などは中間配列とする .本研究では解くべき行列の対角要素をすべて 1 としたので, 実質的に文献 11) による対角シフトと同一になる .また加速係数の有効性の理論的考察は文献 1) が詳しい .

[加速係数 γ つき Crout 版 ILU 分解]

```
do  $k = 1, \dots, N$ 
```

$$z_j = a_{kj} \quad (j = k, \dots, N) \quad (1)$$

$$\underline{z_k} = \gamma \times z_k \quad (2)$$

```
do  $i = 1, \dots, k - 1; l_{ki} \neq 0$ 
```

```
do  $j = k, \dots, N; u_{ij} \neq 0$ 
```

$$z_j = z_j - l_{ki} u_{ij} \quad (3)$$

```
end do
```

```
end do
```

$$w_j = a_{jk} \quad (j = k + 1, \dots, N) \quad (4)$$

```
do  $i = 1, \dots, k - 1; u_{ik} \neq 0$ 
```

```
do  $j = k + 1, \dots, N; l_{ji} \neq 0$ 
```

$$w_j = w_j - l_{ji} u_{ik} \quad (5)$$

```
end do
```

```
end do
```

$$d_k = z_k \quad (6)$$

```
do  $j = k + 1, \dots, N$ 
```

$$\text{if } (|z_j| > \text{tol}) \text{ then } u_{kj} = z_j \quad (7)$$

$$\text{if } (|w_j| > \text{tol}) \text{ then } l_{jk} = w_j/d_k \quad (8)$$

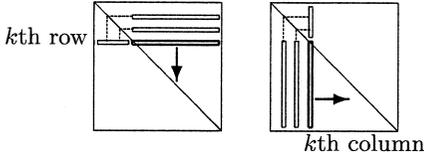


図1 Crout版ILU分解の概念図

Fig.1 Diagram of Crout version ILU decomposition.

end do

end do

図1にCrout版ILU分解の概念図を示す。図中の()印の方向に分解の処理が進行する。更新対象の第 k 行と第 k 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べ替えも必要なくなる。また、行列 U の要素は第 k 行よりも上方に位置する $1 \sim k-1$ 行の要素を使って更新される(Crout版ILU分解の式(3)を参照)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 $1 \sim k-1$ 列の要素を使って更新される(同分解中の式(5)を参照)。そのため、不完全分解における近似の度合いが行列 U と行列 L で偏らないことが期待される。

4. 前処理つきBiCGSafe法の算法

行列 A の前処理行列を K とし、次のように分解する。

$$A \approx K = K_1 K_2. \quad (9)$$

ここで、 $K_1^{-1} A K_2^{-1}$ を \tilde{A} 、 $K_2 x$ を \tilde{x} 、 $K_1^{-1} b$ を \tilde{b} としたとき、解くべき連立1次方程式 $Ax = b$ は次の方程式と数学的に同値である。

$$\tilde{A} \tilde{x} = \tilde{b}. \quad (10)$$

ここで各ベクトルの対応は以下ようになる。

$$\tilde{x}_n = K_2 x_n, \quad \tilde{p}_n = K_2 p_n, \quad \tilde{u}_n = K_2 u_n, \quad (11)$$

$$\tilde{z}_n = K_2 z_n, \quad \tilde{r}_n = K_1^{-1} r_n, \quad \tilde{y}_n = K_1^{-1} y_n, \quad (12)$$

$$\tilde{a}_r = K_1^{-1} a_r, \quad \tilde{r}_0^* = K_1^T r_0^*. \quad (13)$$

方程式(10)にBiCGSafe法を適用し、式(11)~式(13)を使って元方程式に戻すと、以下のような前処理つきBiCGSafe法の算法が導出される⁶⁾。 ε は収束判定に使う微小な値とする。

[前処理つきBiCGSafe法の算法]

x_0 is given, $r_0 = b - Ax_0$,

choose r_0^* such that $(r_0^*, r_0) \neq 0$,

e.g., $r_0^* = r_0$ or $r_0^* = \text{random number}$,

$\beta_{-1} = 0$,

for $n = 0, 1, \dots$ until $\|r_{n+1}\|_2 / \|r_0\|_2 \leq \varepsilon$ do :

begin

$$p_n = K^{-1} r_n + \beta_{n-1} (p_{n-1} - u_{n-1}),$$

$$Ap_n = AK^{-1} r_n + \beta_{n-1} (Ap_{n-1} - Au_{n-1}),$$

$$\alpha_n = (r_0^*, r_n) / (r_0^*, Ap_n),$$

compute ζ_n and η_n

$$u_n = K^{-1} (\zeta_n Ap_n + \eta_n y_n) + \eta_n \beta_{n-1} u_{n-1},$$

$$z_n = \zeta_n K^{-1} r_n + \eta_n z_{n-1} - \alpha_n u_n,$$

$$y_{n+1} = \zeta_n AK^{-1} r_n + \eta_n y_n - \alpha_n Au_n,$$

$$x_{n+1} = x_n + \alpha_n p_n + z_n,$$

$$r_{n+1} = r_n - \alpha_n Ap_n - y_{n+1}, \quad (14)$$

$$\beta_n = (\alpha_n / \zeta_n) \cdot (r_0^*, r_{n+1}) / (r_0^*, r_n),$$

end.

行列ベクトル積の計算回数は1反復あたり $A(K^{-1} r_n)$ 、 Au_n の2回、同様に、 K^{-1} の前進・後退代入の計算は $K^{-1}(\zeta_n Ap_n + \eta_n y_n)$ と $K^{-1} r_n$ の2回である。

4.1 2つの加速パラメータ ζ_n, η_n の決定

上のBiCGSafe法の算法中の残差ベクトル r_{n+1} ($\equiv H_{n+1}(\lambda) R_{n+1}(\lambda)$)の式(14)中には決定すべき加速パラメータ ζ_n, η_n が含まれていない。そのため、BiCGStab法をはじめとしてGPBi-CG法など多くの積型BiCG法で採用されてきた真の残差ベクトル r_{n+1} のノルムの最小化という手順でパラメータを決めることができない。そこで、算法中には陽に現れない補助ベクトル $a_r = r_n$ ($\equiv H_{n+1}(\lambda) R_n(\lambda)$)

$$a_r = r_n - \zeta_n AK^{-1} r_n - \eta_n y_n \quad (15)$$

の式の中に加速パラメータ ζ_n, η_n が含まれていることに着目する。また各々の定義式から分かるように、両者の違いは真の残差ベクトルでは R_{n+1} を使い、一方補助ベクトルでは R_n を使うという違いがある。しかし、加速多項式を積の形で使う収束列の中では2つのベクトルの違いは非常に小さいと予想される。そこで、 a_r を準残差ベクトルと呼ぶことにすると、準残差ベクトルの最小化、すなわち、

$$\|a_r\|_2 = \|r_n - \zeta_n AK^{-1} r_n - \eta_n y_n\|_2 \quad (16)$$

の最小化から加速パラメータを決める。ここで、ベクトル y_n, Ar_n によって張られる2次元部分空間への射影を表す大きさ $N \times 2$ の行列を $B \equiv (y_n Ar_n)$ と定義するとき、 ζ_n と η_n は次のように表せる。

$$\begin{pmatrix} \zeta_n \\ \eta_n \end{pmatrix} = -(B^T B)^{-1} B^T r_n.$$

これを解き ζ_n と η_n が求まる．ただし，中間ベクトル $v_n = AK^{-1}r_n$ とおく．

$$\zeta_n = \frac{(y_n, y_n)(v_n, r_n) - (y_n, r_n)(v_n, y_n)}{(v_n, v_n)(y_n, y_n) - (y_n, v_n)(v_n, y_n)}, \tag{17}$$

$$\eta_n = \frac{(v_n, v_n)(y_n, r_n) - (y_n, v_n)(v_n, r_n)}{(v_n, v_n)(y_n, y_n) - (y_n, v_n)(v_n, y_n)}. \tag{18}$$

また反復初回 $n = 0$ のときは， $\zeta_n = (v_n, r_n) / (v_n, v_n)$ ， $\eta_n = 0$ とする．

5. 数値実験

5.1 計算機環境と計算条件

数値実験には，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 7.3) を使用した．コンパイラは Intel Fortran compiler ver. 7.1 を用い，最適化オプションは -O3 を使用した．反復法の収束判定条件は $\|r_{n+1}\|_2 / \|r_0\|_2 \leq 10^{-7}$ とした．方程式の右辺項は基礎方程式の離散化から得られる値をそのまま使用し，初期近似解 x_0 はすべて 0，最大反復回数は 10,000 回とした．初期シャドウ残差ベクトル r_0^* には初期残差ベクトル r_0 または一樣乱数を用いた．行列はあらかじ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．不完全分解には，元の係数行列 A の非零要素とその位置が一致するフィルインだけを考慮する ILU(0) 分解および Crout 版 ILU 分解の 2 つを用いた．不完全分解中の加速係数 γ は 1.0 から 1.3 まで 0.002 刻みで合計 151 試行を行い収束性を調べた．反復法は BiCGStab 法，GPBi-CG 法，BiCGSafe 法を用いた．

5.2 収束安全性の検証テスト問題

ここでは，テスト問題として，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^{7),12),16)}．一般に，電気電子機器では，渦電流損を低減するため，非常に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，この部分に渦電流が流れることがある．ゆえに，発生する渦電流により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のある問題を箔要素を用いた有限要素法によりモデル化した場合，要素が扁平な形状になる．

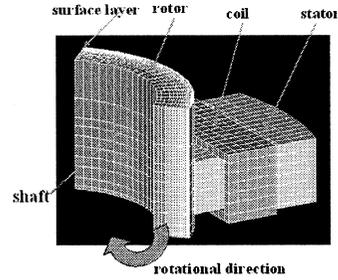


図 2 渦電流解析 1/8 モデル

Fig. 2 Eddy current analytic model (1/8 region).

その結果，係数行列の性質が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，ILU(0) 分解および加速係数 $\gamma = 1.0$ の Crout 版 ILU 分解つきの反復法では収束しなかった困難な問題である．ここで扱う行列は次元数 27,720，非零要素数 1,665,372，1 行あたりの平均非零要素数は約 60.08 の疎行列である．図 2 に渦電流解析モデル (1/8 モデル) とメッシュ分割図を示す．このモデルは，積層鉄心とコイルから構成される固定子と，軸の周りに渦電流を低減するため積層鉄心を設けた回転子で構成されている．本解析は次式を基礎方程式とする 3 次元非線形渦電流解析である¹²⁾．ここで， A ， ϕ および 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．

$$w = -\sigma \left(\frac{\partial A}{\partial t} + \text{grad } \phi \right), \tag{19}$$

$$\text{rot} (\nu \text{ rot } A) = J_0 + w, \tag{20}$$

$$\text{div}(w) = 0. \tag{21}$$

5.3 計算結果

表 1 に BiCGStab 法，GPBi-CG 法，BiCGSafe 法の収束性を比較した結果を示す．閾値は 0.005 とした．以下の表中の時間の単位はすべて秒とする．表中の初期シャドウ残差ベクトル r_0^* の第 3 欄で「 r_0 」の表記は， $r_0^* = r_0$ の場合を，同様に「rand」の表記は r_0^* に一樣乱数を代入した場合の結果を各々表す．また「収束試行度数」とは収束に成功した度数の合計である．表に載せた平均値は，加速係数 γ の探索区間 [1.018, 1.180] かつその区間に含まれる γ に対して収束した試行だけを集計した平均値で，未収束の場合は集計から除外した．探索区間 [1.018, 1.180] 中の試行度数は 82 である．表中で 82 試行すべての場合において収束したケースは太字で表した．表 1 に示した結果から，

- ILU(0) 分解の収束試行度数の割合がどの反復法

表 1 閾値=0.005 の場合の 3 つの反復法の収束性比較．ただし γ の探索区間 [1.018, 1.180] の 82 試行のみ集計

Table 1 Comparison of convergence of the iterative methods at tolerance of 0.005 for 82 cases with parameters γ included in the interval [1.018, 1.180].

前処理	反復法	r_0^*	収束試 行回数	平均反 復回数	平均 時間
ILU(0)	BiCGStab	r_0	54	3,864	117.9
		rand	54	5,730	174.3
	GPBi-CG	r_0	61	2,085	68.0
		rand	62	1,955	64.2
	BiCGSafe	r_0	60	1,769	56.0
		rand	62	2,074	65.6
ILUC	BiCGStab	r_0	33	2,684	104.0
		rand	82	2,055	84.7
	GPBi-CG	r_0	81	3,167	128.7
		rand	81	778	44.2
	BiCGSafe	r_0	82	2,318	95.4
		rand	82	765	43.0

表 2 閾値=0.01 および閾値=0.001 の場合の Crout 版 ILU 分解つき GPBi-CG 法および同 BiCGSafe 法の収束性比較．ただし γ の探索区間 [1.018, 1.180] の 82 試行のみ集計

Table 2 Comparison of convergence of the ILUC-GPBi-CG and ILUC-BiCGSafe methods at tolerances of 0.01 and 0.001 for 82 cases with parameters γ included in the interval [1.018, 1.18].

閾値 (fill-in)	反復法	r_0^*	収束試 行回数	平均反 復回数	平均 時間
0.01	GPBi-CG	r_0	68	4,484	148.4
		rand	77	985	43.8
	BiCGSafe	r_0	77	3,150	105.8
		rand	77	954	42.1
0.001	GPBi-CG	r_0	82	948	72.2
		rand	82	823	65.7
	BiCGSafe	r_0	82	895	68.4
		rand	82	705	58.8

でも低い，

- Crout 版 ILU 分解は，BiCGStab 法の $r_0^* = r_0$ の場合を除いて，調べたすべての反復法で収束性が良い，
- Crout 版 ILU 分解で r_0^* が一様乱数の場合，GPBi-CG 法，BiCGSafe 法いずれの反復法でも反復回数の平均値が 800 回以下，平均時間が約 43～44 秒と収束が安全かつ速いこと，

などが分かる．このように r_0^* に一様乱数を与えた場合に収束が良い理由は， r_0^* にすべての固有ベクトル成分が含まれている可能性が高く，そのため反復過程で増加した固有ベクトル成分を十分取り入れることができたためと考えられる^{5),17)}．

また，表 2 に閾値を 0.01 および 0.001 にした場合の GPBi-CG 法と BiCGSafe 法の数値実験結果を示す．収束性が悪い BiCGStab 法の結果は割愛した．表 2 の結果から，

- 閾値=0.01 の場合は閾値=0.005 の場合よりも収束性が若干悪くなったこと，
 - 閾値=0.001 に小さくした場合，GPBi-CG 法と BiCGSafe 法は初期シャドウ残差ベクトル r_0^* の与え方によらず安全収束性を実現したこと，
 - ただし，閾値=0.001 の場合の収束までの平均時間は閾値=0.005 の場合よりも増えたこと，
- などが分かる．

図 3 に加速係数 γ を変化させた場合の ILU(0) 分解つき反復法の収束性を示す．上図は BiCGStab 法，中図は GPBi-CG 法，下図は BiCGSafe 法の結果を各々示す．図中において，赤色の実線は $r_0^* = r_0$ のとき，青色の点線は $r_0^* =$ 乱数のときを各々表す．また，図 4 に閾値が 0.005 で加速係数 γ を変化させた場合の Crout 版 ILU 分解つき反復法の収束性を示す．図 3 と同様に，上図は BiCGStab 法，中図は GPBi-CG 法，下図は BiCGSafe 法の結果を各々示す．図の横軸は加速係数 γ の値，縦軸は収束までの反復回数を各々表す．最大反復回数は 10,000 回である．これらの図から，

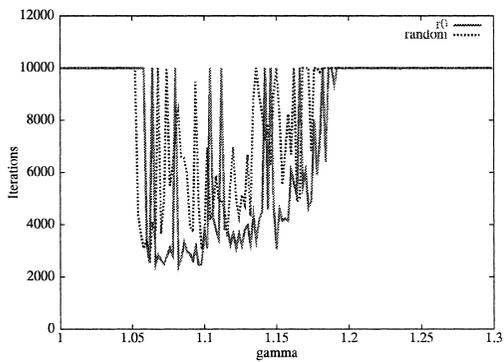
- BiCGStab 法は Crout 版 ILU 分解で初期シャドウ残差ベクトル r_0^* に一様乱数を代入すると収束性が改善されるが全般に収束性が良くないこと，
 - GPBi-CG 法は Crout 版 ILU 分解で r_0^* に一様乱数を代入しても収束しない場合があること，
 - BiCGSafe 法は全般に収束性が非常に良く，特に Crout 版 ILU 分解で r_0^* に一様乱数を代入したときは探索区間 [1.018, 1.180] のすべての場合で収束したこと，
- などが分かる．

さらに，図 5 に閾値を 0.01 と大きくした場合の Crout 版 ILU 分解つき GPBi-CG 法と同 BiCGSafe 法の収束性を示す．同様に，図 6 に閾値を 0.001 と小さくした場合の Crout 版 ILU 分解つき GPBi-CG 法と同 BiCGSafe 法の収束性を示す．これらの図から，

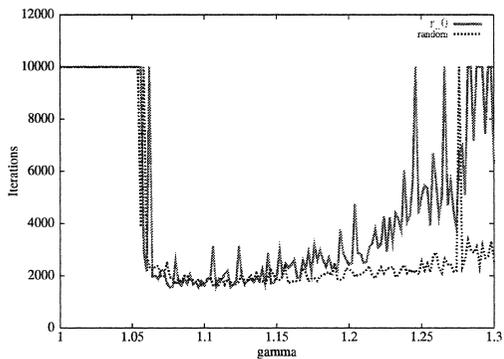
- $r_0^* = r_0$ のときの収束性は，指定した閾値の値によって大きく変動し非常に不安定であること，
- 一方， r_0^* が一様乱数の場合は閾値の違いによる影響の度合いが小さく，かつ収束が安全であること，

などが分かる．このような収束の「安全性」は非線形問題などの場合重要な鍵になることが多い^{12),16)}．

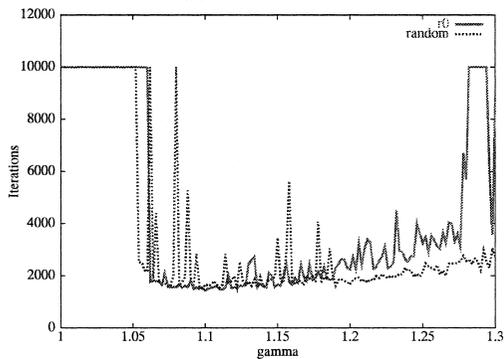
表 3 に ILUC 前処理つき BiCGSafe 法（加速係数は 1.05）における収束までの計算時間の合計に対する前処理時間の割合およびフィルインの個数などを示す．フィルインが多くなるに従って，前処理時間が増加し反復回数が減少するのが分かる．



(a) BiCGStab 法



(b) GPBi-CG 法



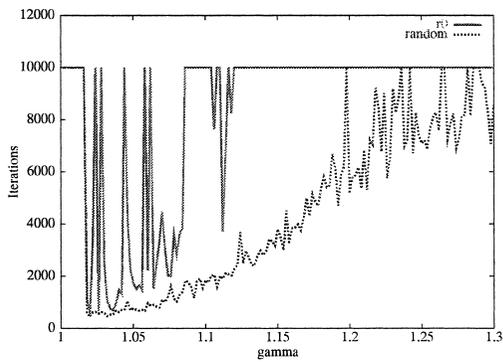
(c) BiCGSafe 法

図 3 加速係数 γ を変化させたときの ILU(0) 分解つき反復法の収束性 (上図: BiCGStab 法, 中図: GPBi-CG 法, 下図: BiCGSafe 法). 赤色の実線: $r_0^* = r_0$, 青色の点線: $r_0^* =$ 乱数のとき

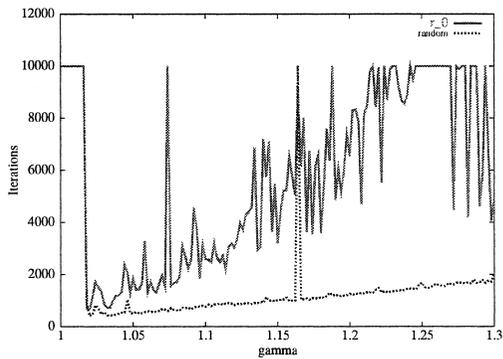
Fig. 3 Convergence of the iterative methods with ILU(0) decomposition when parameter γ was varied (a solid line in red color means a case with $r_0^* = r_0$ and a dotted line in blue color means a case with $r_0^* =$ random number).

5.4 前処理 ILUT(Lfil,tol) 法との比較

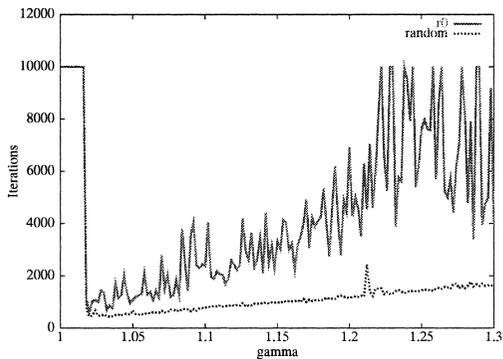
ここでは, フィルインの個数の上限を設けた前処理の性能評価を行う. n を行列の次元数, nnz を元の行列 A の総非零要素数, m を倍率とする. このとき, $Lfil = \frac{nnz}{2n} \times m$ で表される数を各行または各列で



(a) BiCGStab 法



(b) GPBi-CG 法

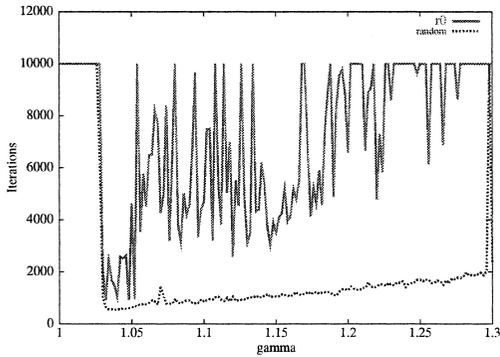


(c) BiCGSafe 法

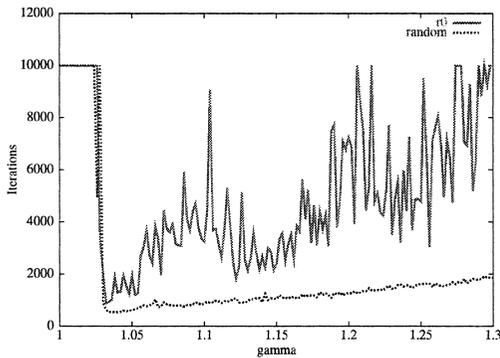
図 4 閾値=0.005 のとき加速係数 γ を変化させたときの Crout 版 ILU 分解つき反復法の収束性 (上図: BiCGStab 法, 中図: GPBi-CG 法, 下図: BiCGSafe 法). 赤色の実線: $r_0^* = r_0$, 青色の点線: $r_0^* =$ 乱数

Fig. 4 Convergence of the iterative methods with Crout version ILU decomposition at tolerance value of 0.005 when parameter γ was varied (a solid line in red color means a case with $r_0^* = r_0$ and a dotted line in blue color means a case with $r_0^* =$ random number).

のフィルインの個数の上限と設定する⁸⁾. すなわち, ILUC(Lfil, tol) 前処理は, まず ILUC 分解において閾値 tol より小さいフィルインを棄却し, その後さらに, 上三角行列 U の各行と下三角行列 L の各列のフィルインの個数が各々上限値: Lfil 個までの絶対値

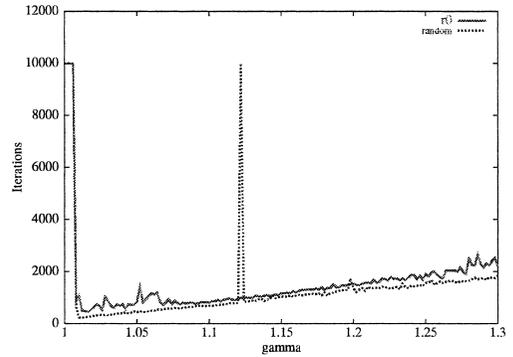


(a) GPBi-CG 法

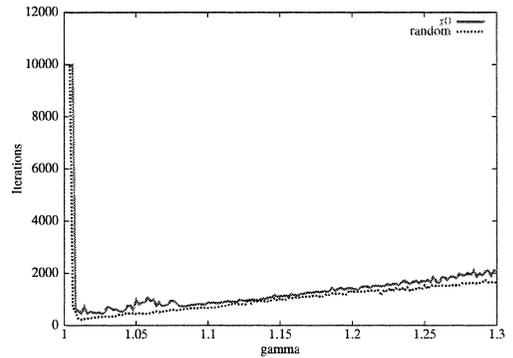


(b) BiCGSafe 法

図 5 閾値を 0.01 に大きくしたときの Crout 版 ILU 分解つき (上図) GPBi-CG 法と (下図) BiCGSafe 法の収束性
Fig. 5 Convergence of GPBi-CG and BiCGSafe methods with Crout version ILU decomposition at tolerance value of 0.01 when parameter γ was varied.



(a) GPBi-CG 法



(b) BiCGSafe 法

図 6 閾値を 0.001 に小さくした場合の Crout 版 ILU 分解つき (上図) GPBi-CG 法と (下図) BiCGSafe 法の収束性
Fig. 6 Convergence of GPBi-CG and BiCGSafe methods with Crout version ILU decomposition at tolerance value of 0.001 when parameter γ was varied.

表 3 ILUC 前処理つき BiCGSafe 法 (加速係数は 1.05) における前処理時間の割合およびフィルインの個数

Table 3 Ratio of preconditioning time and number of fill-in in ILUC-BiCGSafe method at $\gamma = 1.05$.

閾値	フィルインの個数	反復回数	前処理時間 (比)	反復時間	合計時間
0.01	1,347,836	562	4.53 (0.215)	16.58	21.11
0.005	1,895,662	557	5.33 (0.215)	19.43	24.76
0.001	3,872,549	442	11.77 (0.345)	22.35	34.12

の大きいフィルインを採用する前処理である。一方、ILUT(Lfil, tol) 前処理は、まず ILUT 分解において閾値 tol より小さいフィルインを棄却し、その後、上三角行列 U と下三角行列 L の各行のフィルインの個数が各々上限値: Lfil 個まで絶対値の大きいフィルインを採用する。表 4 にテスト行列の主な特徴を示す。行列 surface_033 以外は疎行列データベース²⁾ から選んだ。

ILUT(Lfil, tol) 前処理は 2 つの実装方法を考えた。可変長型 ILUT_v(Lfil, tol) 前処理と固定長型 ILUT_f(Lfil, tol) 前処理の実装方法を以下に示す。

表 4 テスト行列の主な特徴

Table 4 Characteristics of tested matrices.

行列	次元数	非零要素	非零要素/行	解析分野
surface_033	27,720	1,665,372	60.1	電磁界
Olafu	16,146	1,015,156	62.9	構造工学
Li	22,695	1,350,309	59.5	磁性流体
Bbmat	38,744	1,771,722	45.7	翼の構造計算

- 可変長型 ILUT_v(Lfil, tol) 前処理は、元の ILUT (Lfil, tol) 前処理において、前処理行列の第 k 行の処理のとき、最初は元の行列 A の非零要素だけを配列に収容し、フィルインが新たに発生するたびに配列の中の該当位置を探索してそこに挿入し、それより小さいフィルインを 1 個ずつ配列の後方に移動させる実装である。ILUT の節尾語_v は variable (配列の長さが可変長の意味) の略である。非零要素数が少ないときは効率が良いが、非零要素数が多くなると、該当位置の探索とフィルインの後方移動に手間を要する。
- 一方、固定長型 ILUT_f(Lfil, tol) 前処理は、元の

表 5 行列 surface_033 に対する ILUC(Lfil, tol), ILUT_v(Lfil, tol), ILUT_f(Lfil, tol) 前処理の性能

Table 5 Performance of preconditioners ILUC(Lfil, tol), ILUT_v(Lfil, tol) and ILUT_f(Lfil, tol) for matrix surface_033.

前処理 (Lfil=150.3)	閾値 tol	fill-in 個数	反復回数	前処理時間	反復時間	合計時間
ILUC	0.01	1,345,649	550	4.37	15.66	20.03
ILUT_v	0.01	1,691,557	544	8.59	16.73	25.32
ILUT_f	0.01	1,691,590	524	3.32	15.89	19.21
ILUC	0.005	1,900,296	481	5.54	16.18	21.72
ILUT_v	0.005	2,258,040	528	12.85	18.76	31.61
ILUT_f	0.005	2,258,090	518	3.87	18.48	22.35
ILUC	0.001	3,819,328	446	15.17	22.33	37.50
ILUT_v	0.001	3,839,422	407	32.37	20.82	53.19
ILUT_f	0.001	3,840,622	400	6.21	19.63	25.84

表 6 3つの行列に対する ILUC(Lfil, tol), ILUT_v(Lfil, tol), ILUT_f(Lfil, tol) 前処理の性能 (閾値 tol = 0.005)

Table 6 Performance of preconditioners ILUC(Lfil, tol), ILUT_v(Lfil, tol) and ILUT_f(Lfil, tol) for the three matrices at parameter tol = 0.005.

行列 (Lfil)	前処理	fill-in 個数	反復回数	前処理時間	反復時間	合計時間
Olafu (157.3)	ILUC	833,536	2,659	1.58	42.62	44.20
	ILUT_v	907,754	3,950	1.33	64.47	65.80
	ILUT_f	907,754	3,950	1.37	64.88	66.25
Li (148.8)	ILUC	1,531,510	47	3.69	1.26	4.95
	ILUT_v	2,298,359	34	12.62	1.09	13.71
	ILUT_f	2,298,359	34	3.41	1.07	4.48
Bb-mat (114.3)	ILUC	7,947,593	1,250	93.5	102.0	195.5
	ILUT_v	5,631,007	-	194.7	破綻	-
	ILUT_f	6,602,394	-	30.7	破綻	-

ILUT(Lfil, tol) 前処理において行列の第 k 行目の分解のとき、まず元の行列 A の 1 行分をそのまま値が零の要素も含めて配列にとり分解する。発生したフィルインをその配列の該当位置に代入するだけで済む。可変長型のようなフィルインの後方移動が不要になる。その後、新たに加わったフィルインも含めて非零の要素だけを抽出し別の小さな配列に収める。ILUT の節尾語 $_f$ は fixed (配列の長さは固定という意味) の略とする。

表 5 に行列 surface_033 に対する ILUC(Lfil, tol), ILUT_v(Lfil, tol), ILUT_f(Lfil, tol) 前処理の性能を示す。同様に、表 6 に 3 つの行列に対する ILUC(Lfil, tol), ILUT_v(Lfil, tol), ILUT_f(Lfil, tol) 前処理の性能 (倍率 $m = 5$ と閾値 $tol = 0.005$ は固定) を示す。いずれの場合も反復法は BiCGSafe 法とし、初期シャドウ残差 r_0^* は一様乱数を代入し、加速係数は 1.05 に固定した。表中の「破綻」は反復計算中に破綻したことを示す。表中の太い数字は 3 つの前処理で収束までの時間が最も短かったものを表す。表 5 と表 6

に示した結果から以下の観察ができる。

- 行列 surface_033 では、ILUT_f(Lfil, tol) 前処理の効率が良いが、ILUC(Lfil, tol) 前処理との差は小さい。
- 行列 Olafu では、ILUC(Lfil, tol) 前処理が最も効率が良い。また行列 Li では、ILUT_f(Lfil, tol) 前処理が効率が良いが、収束までの反復回数が他の行列のときと比べて非常に少ない。
- 行列 Bbmat では、ILUC(Lfil, tol) 前処理つき BiCGSafe 法だけが収束し、ILUT(Lfil, tol) 前処理はいずれも BiCGSafe 法の反復計算の途中で計算が破綻した。このことから ILUC(Lfil, tol) 前処理つき BiCGSafe 法の安全収束性が分かる。

6. まとめ

本研究では、Crout 版 ILU 分解に加速係数 γ を付加した前処理を用いた BiCGStab 法、GPBi-CG 法そして BiCGSafe 法の安全収束性について検証した。さらに、初期シャドウ残差ベクトル r_0^* の与え方を変えたときの反復法の収束性の違いについて検証した。その結果、以下の有用な知見が得られた。

- (1) 加速係数 γ を付加した Crout 版 ILU 分解は同じく加速係数つき ILU(0) 分解に比べて反復法の収束性を大幅に向上させる。
- (2) BiCGSafe 法の安全収束性は他の反復法に比べて優ることが多い。
- (3) さらに、初期シャドウ残差 r_0^* に一様乱数を代入すれば安全収束性の確かさが増える。

このように、加速係数 γ 付き Crout 版 ILU 分解前処理、BiCGSafe 法そして初期シャドウ残差 r_0^* への一様乱数の代入という三重の安全収束鍵を適切に選ぶことにより、安全な収束性が実現することを実際の問題および疎行列データベースの問題に対して明らかにした。

参考文献

- 1) Chow, E. and Saad, Y.: Experimental study of ILU preconditioner factorization for indefinite matrices, *J. Comput. Appl. Math.*, Vol.86, pp.387–414 (1997).
- 2) Tim Davis's sparse matrix collection. <http://www.cise.ufl.edu/research/sparse/matrices/>
- 3) Fletcher, R.: Conjugate Gradient preconditioning for indefinite systems, *Lecture Notes in Mathematics*, Vol.506, pp.73–89 (1976).
- 4) 藤野清次, 張 紹良: 反復法の数理, 朝倉書店

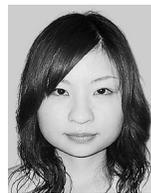
- (1996).
- 5) 藤野清次, 阿部邦美: 乱数を初期値に与えた時の積型反復解法の収束改善について, 日本応用数学会 2001 年度年会講演予稿集, pp.266–267 (2001).
 - 6) 藤野清次, 藤原 牧, 吉田正浩: 準残差の最小化に基づく積型 BiCG 法, 日本計算工学会論文集 (2005). <http://save.k.u-tokyo.ac.jp/jsces/>
 - 7) 藤原 牧, 藤野清次, 吉田正浩, 藤原耕二, 村松和弘: 二重の安全収束性を有する前処理つき BiCGSafe 法について, 第 8 回環瀬戸内シンポジウム予稿集, 松山, pp.78–83 (2005).
 - 8) Li, N., Saad, Y. and Chow, E.: Crout version of ILU for general sparse matrices, *SIAM J. Sci. Comput.*, Vol.25, pp.716–728 (2003).
 - 9) 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, Vol.7, pp.279–284, 土木学会 (2004).
 - 10) 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, 松山, pp.43–48 (2005).
 - 11) Manteuffel, T.A.: An incomplete factorization technique for positive definite linear systems, *Math. Comp.*, Vol.31, pp.473–497 (1980).
 - 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. Magnetics*, Vol.38, pp.597–600 (2002).
 - 13) Saad, Y. and Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.7, pp.856–869 (1986).
 - 14) Saad, Y.: ILUT: A dual threshold incomplete LU factorization, *Numer. Linear Algebra Appl.*, Vol.1, pp.387–402 (1994).
 - 15) Saad, Y.: *Iterative methods for sparse linear systems*, 2nd edition, SIAM Philadelphia (2003).
 - 16) 清水露規, 村松和弘ほか: 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, pp.105–110 (2005).
 - 17) 多田野寛人, 櫻井鉄也: Lanczos プロセスのリス

タートによる CGS 法の安定化, 日本応用数学会論文誌, Vol.15, pp.129–143 (2005).

- 18) van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.13, pp.631–644 (1992).
- 19) van der Vorst, H.A.: *Iterative Krylov preconditionings for large linear systems*, Cambridge University Press, Cambridge (2003).
- 20) Zhang, S.-L.: GPBi-CG: Generalized product-type preconditionings based on Bi-CG for solving nonsymmetric linear systems, *SIAM J. Sci. Comput.*, Vol.18, pp.537–551 (1997).

(平成 17 年 9 月 29 日受付)

(平成 18 年 1 月 3 日採録)



藤原 牧

2004 年 3 月九州大学工学部電気情報工学科卒業。2006 年 3 月九州大学大学院システム情報科学府修士課程修了。非対称行列用の前処理と反復法に興味を持つ。



吉田 正浩

2004 年 3 月九州大学工学部電気情報工学科卒業。2006 年 3 月九州大学大学院システム情報科学府修士課程修了。共役勾配法の実装と並列化に興味を持つ。



藤野 清次 (正会員)

1974 年京都大学理学部卒業。1993 年博士 (工学) 東京大学。2001 年九州大学情報基盤センター研究部教授。現在に至る。その間共役勾配法系統の反復法とその前処理の研究を行う。

日本応用数学会会員。