

前処理つき反復法における自動残差修正法の有効性の検証

村上 啓一¹ 藤野 清次²

概要：IDR(s)法の偽収束を防ぐ手法として、櫻井らにより自動残差修正法が提案された。さらに、自動残差修正法は拡張され、塚田らによりGBiCGSTAB(s, L)法に適用された。一般に、前処理を適用すると丸め誤差の影響が減るため、反復法は前処理と共に用いられることが多い。しかし、前処理つきの場合には、自動残差修正法の効果について報告がほとんどされていない。そこで、本論文では、自動残差修正法を前処理つき反復法に適用し、その有効性を明らかにする。

キーワード：自動残差修正法，IDR(s)法，IDRstab(s, L)法，GBiCGSTAB(s, L)法，前処理

Verification of effectiveness of Auto-Correction technique applied to preconditioned iterative methods

KEIICHI MURAKAMI¹ SEIJI FUJINO²

Abstract: Auto-Correction technique has been proposed by Sakurai *et al.* in order to avoid spurious convergence. Tukada *et al.* expanded the AC technique, and applied it to their GBiCGSTAB(s, L) method. In general, iterative methods are used together with preconditioning. Therefore, in this paper, we apply the AC technique to preconditioned iterative methods, and verify effectiveness of the AC technique.

Keywords: Auto-Correction technique, IDR(s) method, IDRstab(s, L) method, GBiCGSTAB(s, L) method, preconditioning

1. はじめに

2008年，Sonneveldらは非対称行列を係数に持つ連立一次方程式の解法として，IDR(s) (Induced Dimension Reduction)法を提案した [15]。Sleijpenらは，IDR(s)法にBiCGstab(ℓ)法 [12]の戦略を導入することで，IDRstab(s, L)法 [13][14]を導出した。また，谷尾らはIDRstab(s, L)法と数学的に等価な解法であるGBiCGSTAB(s, L)法 [16]を提案した。これらは共に，IDR(s)法に高次の安定化多項式を付加した解法であり，Bi-CGSTAB法等の従来の解法よりも優れた収束性を示すことで注目を集めている。

しかしながら，IDR(s)法システムの解法には，パラメータを

大きく設定するとき偽収束が起こるという問題がある。偽収束とは，反復計算中に得られる残差と近似解の満たすべき関係 $r_k = b - Ax_k$ が丸め誤差の影響で崩れ，要求する精度の近似解が求まっていない状態で計算が終了する現象である。櫻井らは，文献 [11]において偽収束の発生と関連の高い指標を実験的に見出し，IDR(s)法の偽収束を防ぐ手法として自動残差修正法を提案した。塚田らは櫻井らの自動残差修正法をGBiCGSTAB(s, L)法に適用し有効性を示した [17]。

一方，偽収束を防ぐ別の方法として，前処理の利用が考えられる。前処理とは係数行列を近似した行列を用いて，連立一次方程式を，同じ解を持つ，より解きやすい方程式に変換する手法であり，代表的な前処理としてILU(0)，SSOR等が挙げられる [2][3][4]。多くの数値例において，前処理の適用により反復法の収束性が劇的に改善される事が示されている。丸め誤差は反復が増えるに従って蓄積されるため，前処理により反復回数を減らすことでIDR(s)系

¹ 九州大学大学院システム情報科学府情報学専攻
Graduate School of Information Science and Electrical Engineering, Kyushu University

² 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

統の解法における偽収束を防ぐ効果が期待できる。

本研究では, IDRstab(s, L) 法, 及び GBiCGSTAB(s, L) 法の偽収束を防ぎ, かつ収束性を向上させる事を目的とする。自動残差修正法を前処理つき反復法に適用し, 偽収束発生という問題に対して自動残差修正法がどの程度有効であるかを明らかにする。

2. 前処理つき AC-GBiCGSTAB(s, L) 法

解くべき連立一次方程式を

$$Ax = b \quad (1)$$

とする。ここで, $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ は各々係数行列, 解ベクトル, 右辺項を意味する。

2.1 自動残差修正法

IDR 系統の解法では, パラメータ s や L を大きくした場合に偽収束と呼ばれる現象が起こることがある。偽収束とは, 丸め誤差の影響で反復終了時の残差 r_{k+1} と近似解の関係 $r_{k+1} = b - Ax_{k+1}$ が崩れ, 解が収束していない状態で反復計算が終了する現象である。

IDR(s) 法における残差の更新式は次式で表される。

$$r_{k+1} = (I - \omega_k A)(r_k - dR_k c). \quad (2)$$

ここで, ω_k は 0 でないスカラー, c はサイズ s のベクトルである。 dR_k は残差の差分ベクトル $dr_k = r_{k+1} - r_k$ を用いて $dR_k := (dr_{k-1}, \dots, dr_{k-s})$ と定義される。 dr_k の更新方法には, 漸化式による計算と残差の定義から $dr_k = -Adx_k$ とする直接計算の二通りの方法がある。直接計算は行列ベクトル積が必要となるため漸化式計算よりも必要な計算量が多い。しかし, 漸化式計算は直接計算よりも丸め誤差の影響を受けやすいため, 偽収束となりやすい。

偽収束を防ぐ手法として, 次式で定義される IDR 系統の解法における誤差の指標次式で定義される指標

$$I_k = \frac{\|dr_k\|}{\|b\|} \times \text{Range}(c). \quad (3)$$

を用いる自動残差修正法が提案されている [11]。式 (3) 中に用いられている $\text{Range}(c)$ は次式で定義される。

$$\text{Range}(c) = \frac{\max_{1 \leq i \leq s} \|c(i)\|}{\min_{1 \leq j \leq s} \|c(j)\|}. \quad (4)$$

自動残差修正法では, I_k が閾値 I_{th} より大きいと dr_k を直接計算により更新し, そうでないときは漸化式による更新を行うことで IDR(s) 法の偽収束を防ぐ手法である。

以下に, 自動残差修正法の算法を示す。

自動残差修正法

Set I_k

if $I_k < I_{th}$ then

 Compute dr_k recursively

else

 Compute $dr_k = -Adx_k$

end if

$r_{k+1} = r_k + dr_k$

一方, GBiCGSTAB(s, L) 法には指標 I_k を拡張する事により自動残差修正法を適用する事ができる [17]。GBiCGSTAB(s, L) 法において, 残差の更新は次式で行われる。

$$r_{k+1} = r_k - \sum_{k=1}^L U_{k,1}^{(j)} \alpha_k^j - [r_{k,1}^L, \dots, r_{k,L}^L] \gamma_{k+1} \quad (5)$$

ここで, γ_{k+1} , $U_{k,1}^j$ は,

$$\begin{aligned} \gamma_{k+1} &= \operatorname{argmin}_{\gamma} \|r_0 - [r_1, \dots, r_L] \gamma\|, \\ U_{k,1}^j &= AU_k^j \end{aligned} \quad (6)$$

で得られる。GBiCGSTAB(s, L) 法に対する誤差の指標 I_k は次式で表される。

$$I_k = \frac{\|r_k\|}{\|r_0\|} \times \max_j (\text{Range}(\alpha_j)) \times \text{Range}(\gamma) \quad (7)$$

2.2 Eisenstat trick を用いた SSOR 前処理つき反復法

反復法の代表的な前処理として, ILU 前処理や SSOR 前処理 [2] が挙げられる。SSOR 前処理は Eisenstat trick と呼ばれる計算量削減手法 [6] と共に用いることで, 優れた収束性を示す事が報告されている [3][4][8][9]。

SSOR 前処理では, 係数行列を $A = L_A + D + U_A$ と分離して得られる行列を用いて, 前処理行列 K を

$$K = (L_A + D/\omega)(D/\omega)^{-1}(U_A + D/\omega) \quad (8)$$

とする。ここで, L_A, D, U_A は各々狭義下三角行列, 対角行列, 狭義上三角行列を, ω は緩和係数を意味する。

SSOR 前処理は Eisenstat trick と呼ばれる手法を用いて実装することで計算量を削減することができる [6]。両側前処理後の行列を

$$\begin{aligned} \tilde{A} &= ((U_A + D/\omega)^{-1} + (L_A + D/\omega)^{-1} \\ &\quad \times (I + (1 - 2/\omega)D(U_A + D/\omega)^{-1}))(D/\omega) \end{aligned} \quad (9)$$

と式変形し, 行列ベクトル積 $\tilde{A}v$ を以下の手順で計算する [3][4]。

1. $y = (U_A + D/\omega)^{-1}(D/\omega)v$
2. $z = (D/\omega)v + (1 - 2/\omega)Dy$
3. $w = (L_A + D/\omega)^{-1}z$
4. $\tilde{A}v = y + w$

以下, Eisenstat trick を用いて実装した SSOR 前処理を

E-SSOR 前処理と略す .

以下に , E-SSOR 前処理つき AC-GBiCGSTAB(s, L) 法の
の算法を示す .

E-SSOR 前処理つき AC-GBiCGSTAB(s, L) 法の算法

1. Let \mathbf{x}_0 be an initial guess, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, set $\tilde{R}_0 \in \mathbb{R}^{n \times s}$
2. Compute $\tilde{\mathbf{x}}_0 = (D/\omega)^{-1}(U_A + D/\omega)\mathbf{x}_0$, $\tilde{\mathbf{r}}_0 = (L_A + D/\omega)^{-1}\mathbf{r}_0$
3. Set U_0 , Compute U_p , ($p = 1$)
4. Solve $M\boldsymbol{\alpha} = \mathbf{m}$
5. $\tilde{\mathbf{r}}_0 = \tilde{\mathbf{r}}_0 - U_1\boldsymbol{\alpha}$, $\tilde{\mathbf{x}}_0 = \tilde{\mathbf{x}}_0 + U_0\boldsymbol{\alpha}$
6. while $\|\mathbf{r}_k\|/\|\mathbf{r}_0\| \geq \epsilon$ do
7. {BiCG - PART}
8. for $j = 1 \dots L$ do
9. if $(k = 0) \cap (j = 1)$ then
10. $M_0 = \tilde{R}_0^T U_1$, $\mathbf{m}_0 = \tilde{R}_0^T \mathbf{r}_0$
11. Go to line 8
12. end if
13. for $i = 1 \dots s$ do
14. if $i = 1$ then
15. Solve $M\boldsymbol{\beta} = \mathbf{m}$
16. $U_p \mathbf{e}_1 = \tilde{\mathbf{r}}_p - U_p \boldsymbol{\beta}$, ($p = 0, 1, \dots, j - 1$)
17. else
18. Solve $[\mathbf{m}, M[1 : i - 2], M[i : s]] \boldsymbol{\beta} = M\mathbf{e}_{i-1}$
19. $U_p \mathbf{e}_i = U_{p+1} \mathbf{e}_{i-1} - [\mathbf{r}_p, U_{p+1}[1 : i - 2], U_p[i : s]] \boldsymbol{\beta}$
($p = 0, 1, \dots, j - 1$)
20. end if
21. $\mathbf{y} = (U_A + D/\omega)^{-1}(D/\omega)\mathbf{e}_i$
22. $\mathbf{z} = (D/\omega)U\mathbf{e}_i + (1 - \frac{2}{\omega})D\mathbf{y}$
23. $\mathbf{w} = (L_A + D/\omega)^{-1}\mathbf{z}$
24. $U\mathbf{e}_1 = \mathbf{y} + \mathbf{w}$
25. $U_p \mathbf{e}_i = \frac{U_p \mathbf{e}_i}{\|U_p \mathbf{e}_i\|}$, ($p = 0, 1, \dots, j$)
26. $M\mathbf{e}_i = \tilde{R}_0^T U\mathbf{e}_i$
27. end do
28. Solve $M\boldsymbol{\alpha} = \mathbf{m}$
29. $\tilde{\mathbf{r}}_p = \tilde{\mathbf{r}}_p - U_{p+1}\boldsymbol{\alpha}$, ($p = 0, 1, \dots, j - 1$)
30. $\mathbf{y} = (U_A + D/\omega)^{-1}(D/\omega)\tilde{\mathbf{r}}_{j-1}$
31. $\mathbf{z} = (D/\omega)\tilde{\mathbf{r}}_{j-1} + (1 - \frac{2}{\omega})D\mathbf{y}$
32. $\mathbf{w} = (L_A + D/\omega)^{-1}\mathbf{z}$
33. $\tilde{\mathbf{r}}_j = \mathbf{y} + \mathbf{w}$
34. end do
35. {MR - PART}
36. $\gamma = \operatorname{argmin}_j \|\tilde{\mathbf{r}}_0 - [\tilde{\mathbf{r}}_1, \dots, \tilde{\mathbf{r}}_L]\gamma\|$
37. $I_k = \frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_0\|} \times \max_j (\operatorname{Range}(\boldsymbol{\alpha}_j)) \times \operatorname{Range}(\gamma)$
38. $d\mathbf{x}_k = \sum_{j=1}^L U_k \boldsymbol{\alpha}_j + [\tilde{\mathbf{r}}_0, \dots, \tilde{\mathbf{r}}_{L-1}]\gamma$, $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + d\mathbf{x}_k$
39. if $I_k > \theta$ then
40. $\mathbf{y} = (U_A + D/\omega)^{-1}(D/\omega)d\mathbf{x}_k$
41. $\mathbf{z} = (D/\omega)d\mathbf{x}_k + (1 - \frac{2}{\omega})D\mathbf{y}$

42. $\mathbf{w} = (L_A + D/\omega)^{-1}\mathbf{z}$
43. $\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - (\mathbf{y} + \mathbf{w})$
44. else
45. $\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k - [\tilde{\mathbf{r}}_1, \dots, \tilde{\mathbf{r}}_L]\gamma$
46. end if
47. $U_{k+1} = U_k - [U_1, \dots, U_L]\gamma$
48. $M_{k+1} = -\gamma M_k$, $\mathbf{m}_k = \tilde{R}_0^T \tilde{\mathbf{r}}_k$, $k = k + 1$
49. Compute $\mathbf{r}_k = (L_A + D/\omega)\tilde{\mathbf{r}}_k$
50. end while
51. Compute $\mathbf{x}_k = (U_A + D/\omega)^{-1}(D/\omega)\tilde{\mathbf{x}}_k$

3. 数値実験

4. 計算機環境と計算条件

計算機環境と計算条件は以下の通りである .

計算機環境

- (1) 計算は倍精度浮動小数点演算で行った .
- (2) 計算機は Dell PowerEdge R210II (CPU: Intel Xeon E3-1220 , クロック周波数: 3.1GHz , メモリ: 8Gbytes , OS: Scientific Linux 6.0) を使用した .
- (3) コンパイラは Intel Fortran Compiler version 11.0 を用い , 最適化オプションは “-fast” を使用した .
- (4) 時間の計測には時間計測関数 cputime を使用した .

計算条件

- (1) 収束判定条件は相対残差の 2 ノルム:
 $\|\mathbf{r}_{k+1}\|_2/\|\mathbf{r}_0\|_2 \leq 10^{-12}$ とした .
- (2) 初期近似解 \mathbf{x}_0 はすべて 0 とした .
- (3) 行列は予め対角スケーリングによって対角項をすべて 1.0 に正規化した .
- (4) 最大反復回数は行列の次元数が 10000 以下のとき 10000 回 , それ以上のときは次元数と等しい回数とした .
- (5) パラメータ s , 及び L は 14 個の行列に対して各々 1 , 2 , 4 , 6 , 8 の 5 通り , 合計 350 通り調べた .
- (6) SSOR 前処理及び ILU(0) 前処理のパラメータ ω は 1.0 に固定した .

5. テスト行列

表 1 に非対称行列 14 個の特徴を示す . 表 1 中の行列はフロリダ大学の疎行列データベース [5] から選出した .

6. 実験結果

表 2 に行列 dc3 に対する各前処理つき反復法の収束性を示す .

表中の “ TRR ” は反復終了時の真の相対残差 (True Relative Residual) $\|\mathbf{b} - A\mathbf{x}_{k+1}\|/\|\mathbf{b}\|$ の常用対数の値を , “ NaN ”

表 1 非対称行列の特徴

Table 1 Characteristics of test matrices.

行列	次元数	非零要素数	平均非零要素数	解析分野
air-cfl5	1,536,000	19,435,428	12.65	流体力学
airfoil_2d	14,214	259,688	18.27	
poisson3Db	85,623	2,374,949	27.74	
sherman5	3,312	20,793	70.22	
raefsky2	3,242	293,551	90.55	
watt_2	1,856	11,550	6.22	
2007OK01	54,903	3,990,483	72.68	構造解析
comsol	1,500	97,645	65.10	
sme3Dc	42,930	3,148,656	73.30	
add20	2,395	17,319	7.23	回路
dc3	116,835	766,396	6.56	
memplus	17,758	126,150	7.10	
epb3	84,617	463,625	5.48	熱
k3plates	11,107	378,927	34.12	音響

は計算が破綻したことを意味する。また、計算時間が最も短い値は太字とした。ただし、TRR が-8 以上のケースは除外した。

表 2 行列 dc3 に対する前処理つき反復法の収束性:

(a) 前処理なし IDRstab(s, L) 法

Table 2 Convergence rate of preconditioned iterative methods for matrix dc3:

(a) non-preconditioned IDRstab(s, L) method.

s	L	AC なし			AC 付き		
		反復回数	反復時間	TRR	反復回数	反復時間	TRR
1	1	3,667	9.34	-8.59	6,116	14.72	-10.24
1	2	3,129	8.72	-8.66	3,827	9.87	-10.42
1	4	3,265	10.78	-9.27	3,498	10.81	-10.41
1	6	2,653	10.28	-7.38	3,405	12.48	-11.40
1	8	3,057	13.11	-6.73	2,499	10.58	-10.86
2	1	1,319	4.44	-9.08	1,911	5.33	-10.78
2	2	1,274	4.72	-8.54	1,727	5.41	-10.68
2	4	1,322	5.98	-8.06	1,430	5.59	-10.66
2	6	1,442	7.76	-6.33	1,464	6.90	-10.85
2	8	1,586	9.70	-4.05	1,653	9.12	-11.34
4	1	824	3.78	-7.67	932	3.39	-9.74
4	2	854	4.60	-7.38	957	4.20	-9.76
4	4	904	5.79	-7.24	956	5.17	-9.71
4	6	1,024	7.83	-5.38	1,002	6.57	-9.75
4	8	1,084	9.58	-5.50	1,010	7.83	-9.71
6	1	783	4.53	-7.17	848	3.82	-9.40
6	2	790	5.16	-6.30	930	4.89	-9.39
6	4	790	6.40	-6.82	881	5.95	-9.34
6	6	888	8.57	-4.62	847	7.01	-9.40
6	8	1,182	13.29	-1.46	1,017	9.95	-9.39
8	1	674	4.60	-6.19	793	4.27	-9.60
8	2	728	5.65	-6.26	861	5.45	-9.69
8	4	800	7.65	-6.75	766	6.20	-9.65
8	6	764	8.76	-5.41	817	8.08	-9.68
8	8	1,160	15.51	-1.54	868	10.21	-7.55

表 3 に前処理つき反復法の精度比較を示す。表中の数字

(b) ILU(0) 前処理つき IDRstab(s, L) 法
(b) ILU(0) preconditioned IDRstab(s, L) method.

s	L	AC なし			AC 付き		
		反復回数	反復時間	TRR	反復回数	反復時間	TRR
1	1	889	4.16	-7.91	1,084	4.90	-10.67
1	2	877	4.31	-8.06	1,035	4.86	-10.96
1	4	881	4.79	-8.36	899	4.69	-11.05
1	6	865	5.20	-7.16	878	5.08	-10.72
1	8	913	5.86	-7.12	875	5.60	-10.35
2	1	497	2.75	-8.22	628	3.11	-10.65
2	2	524	3.11	-7.08	628	3.33	-10.44
2	4	542	3.65	-3.74	605	3.68	-10.50
2	6	560	4.24	-4.13	588	4.06	-11.15
2	8	578	4.80	-4.34	585	4.51	-10.11
4	1	369	2.52	-7.13	404	2.38	-10.91
4	2	414	3.08	-6.15	435	2.80	-10.63
4	4	444	3.84	-6.04	451	3.43	-10.87
4	6	424	4.19	-2.63	457	4.01	-11.31
4	8	484	5.35	-0.69	444	4.43	-8.88
6	1	377	3.01	-5.71	417	2.81	-10.54
6	2	398	3.47	-8.18	415	3.12	-10.87
6	4	398	4.11	-5.87	398	3.59	-11.52
6	6	426	5.04	-4.68	424	4.45	-9.63
6	8	454	6.09	-1.58	396	4.76	-11.08
8	1	368	3.34	-6.28	370	2.86	-9.89
8	2	386	3.86	-6.23	404	3.47	-9.95
8	4	368	4.34	-5.21	401	4.16	-9.92
8	6	494	6.77	-3.35	437	5.33	-9.91
8	8	512	7.95	-2.79	363	5.09	-10.09

は TRR が各範囲内の値となったケースの数を表し、括弧内の数字は全体に占める割合の百分率の値を意味する。

表 3 より、IDRstab(s, L) 法について以下の観察が得られる。

- (1) 前処理なしするとき、AC なしときは TRR が“-8 以上”が 160 ケースから AC ありでは 7 ケースに減少した。
 - (2) 前処理の有無に関わらず、AC ありの場合に“max”の方がケースが増加した。
 - (3) SSOR 前処理するとき、AC ありのときは TRR が“-8 以上”は 1 ケースのみであった。
 - (4) AC なし、前処理なしのときは TRR が“-8 以上”は 160 ケースから、ILU(0) 前処理つきの場合 136 ケース、SSOR 前処理つきの場合 118 ケースに減少した。
- GBCGSTAB(s, L) 法について以下の観察が得られる。
- (1) 前処理なしのとき、AC なしで TRR が“-8 以上”が 113 ケースから AC ありでは 31 ケースに減少した。
 - (2) 前処理に関わらず、“max”が AC なしときより AC ありの方がケースが同じ、または減少した。
 - (3) SSOR 前処理のとき、AC ありのときは TRR が“-8 以上”は 0 ケースであった。
 - (4) AC なし、前処理なしのとき TRR が“-8 以上”は 113 ケースから、ILU(0) 前処理つきの場合 86 ケース、SSOR 前処理つきの場合 88 ケースに減少した。

(c) E-SSOR 前処理つき IDRstab(s, L) 法
 (c) E-SSOR preconditioned IDRstab(s, L) method.

s	L	AC なし			AC つき		
		反復回数	反復時間	TRR	反復回数	反復時間	TRR
1	1	1,175	4.48	-10.58	1,016	3.74	-10.71
1	2	729	2.80	-10.54	910	3.49	-10.71
1	4	601	2.54	-10.10	664	2.70	-10.73
1	6	661	3.17	-8.64	713	3.27	-10.71
1	8	689	3.57	-8.92	612	3.17	-8.52
2	1	509	2.30	-10.59	678	2.66	-10.56
2	2	536	2.53	-10.60	607	2.51	-10.71
2	4	494	2.70	-10.63	502	2.45	-10.73
2	6	506	3.16	-8.93	492	2.77	-10.73
2	8	530	3.72	-8.84	511	3.29	-10.76
4	1	374	2.11	-10.47	374	1.78	-10.78
4	2	384	2.34	-9.77	397	2.06	-10.71
4	4	384	2.79	-9.21	385	2.45	-10.73
4	6	394	3.36	-6.87	394	2.95	-10.68
4	8	364	3.51	-5.16	404	3.49	-10.63
6	1	328	2.22	-10.20	352	1.94	-10.62
6	2	328	2.44	-8.56	341	2.11	-10.71
6	4	314	2.81	-8.16	343	2.61	-10.72
6	6	384	4.05	-7.24	341	3.11	-10.72
6	8	454	5.48	-4.34	396	4.21	-9.81
8	1	314	2.44	-9.66	322	2.07	-10.71
8	2	314	2.71	-9.14	311	2.25	-10.61
8	4	368	3.85	-7.04	365	3.28	-10.71
8	6	386	4.75	-4.52	327	3.55	-10.72
8	8	440	6.24	-2.64	363	4.61	-9.43

(d) 前処理なし GBiCGSTAB(s, L) 法
 (d) non-preconditioned GBiCGSTAB(s, L) method.

s	L	AC なし			AC つき		
		反復回数	時間 [s]	TRR	反復回数	時間 [s]	TRR
1	1	5,563	11.22	-10.30	9,047	19.49	-9.80
1	2	4,463	9.54	-9.52	5,740	22.34	-10.34
1	4	3,047	2.71	-9.82	3,612	9.08	-10.57
1	6	2,771	7.74	-9.70	2,117	6.00	-10.78
1	8	2,223	6.95	-8.27	2,575	8.15	-10.29
2	1	1,478	3.21	-10.02	1,550	3.50	-10.38
2	2	1,301	3.03	-10.40	1,552	3.69	-10.91
2	4	1,367	3.67	-9.92	1,401	3.84	-10.53
2	6	1,349	3.57	-9.62	1,462	1.83	-10.54
2	8	1,559	4.98	-7.88	1,558	5.39	-10.41
4	1	854	2.17	-9.55	912	2.41	-9.69
4	2	849	2.37	-9.68	875	2.51	-9.60
4	4	879	2.93	-9.25	917	3.12	-9.66
4	6	959	3.71	-8.68	942	3.72	-9.70
4	8	999	4.40	-7.74	1,011	4.52	-9.64
6	1	769	2.26	-9.27	780	2.38	-9.30
6	2	797	2.63	-9.17	745	2.53	-9.32
6	4	811	3.28	-9.31	767	3.17	-9.30
6	6	839	4.01	-7.47	851	4.13	-9.28
6	8	951	5.26	-6.10	906	5.06	-9.28
8	1	692	2.32	-9.82	751	2.59	-9.73
8	2	701	2.69	-9.43	718	2.81	-9.73
8	4	719	3.45	-8.56	731	3.56	-9.67
8	6	809	4.66	-6.90	818	4.78	-9.74
8	8	935	6.26	-4.25	942	6.40	-9.74

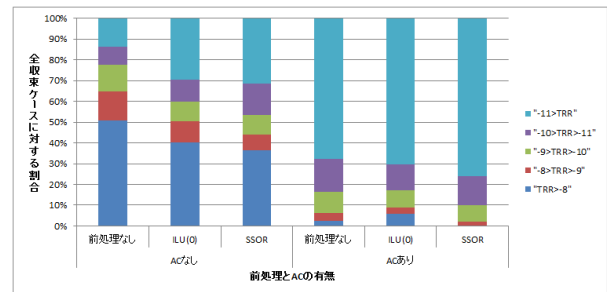
IDRstab(s, L) 法と GBiCGSTAB(s, L) 法の比較として、以下の観察が得られる。

- (1) AC なし、前処理なしのとき IDRstab(s, L) 法の TRR が “-11 以下” が 43 ケースであったのに対して、GBiCGSTAB(s, L) 法は 103 ケースであった。
- (2) AC なし、前処理なしのとき IDRstab(s, L) 法の “max” が 35 ケースであったのに対して、GBiCGSTAB(s, L) 法は 63 ケースであった。
- (3) AC あり、前処理なしのとき IDRstab(s, L) 法の TRR が “-11 以下” が 212 ケースであったのに対して、GBiCGSTAB(s, L) 法は 188 ケースであった。
- (4) AC あり、SSOR 前処理のとき IDRstab(s, L) 法の TRR が “-11 以下” が 233 ケースであったのに対して、GBiCGSTAB(s, L) 法は 259 ケースであった。

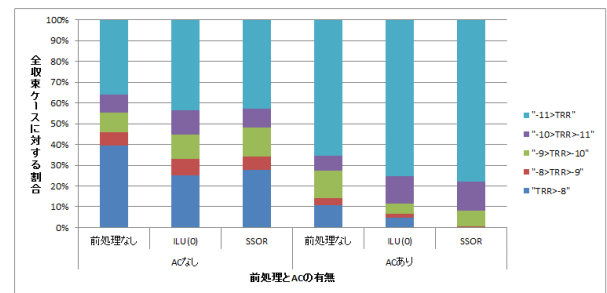
図 1 に、前処理つき反復法の TRR 分布を示す。図 1 より、全体的な傾向として、GBiCGSTAB(s, L) 法が IDRstab(s, L) 法より近似解の精度が良い事がわかる。しかし、前処理なしの場合の AC つき GBiCGSTAB(s, L) 法は同 IDRstab(s, L) 法よりも TRR が “-8” 以上となったケースの割合が多い。

7. まとめ

自動残差修正法の前処理つき反復法に対する効果を調べた。前処理つき IDRstab(s, L) 法、及び GBiCGSTAB(s, L)



(a) IDRstab(s, L)



(b) GBiCGSTAB(s, L)

図 1 前処理つき反復法の TRR 分布

Fig. 1 TRR distribution of preconditioned iterative method.

法に自動残差修正法を適用した場合、近似解の精度が改善される結果が得られた。

謝辞 本研究を進めるに当たり多大な協力を頂いた、平成 24 年 3 月 九州大学工学部電気情報工学科卒業 平田裕貴

(e) ILU(0) 前処理つき GBiCGSTAB(s, L) 法
(e) ILU(0) preconditioned GBiCGSTAB(s, L) method.

s	L	AC なし			AC つき		
		反復回数	反復時間	TRR	反復回数	反復時間	TRR
1	1	954	3.75	-10.24	983	4.22	-10.35
1	2	960	3.85	-9.68	891	3.88	-10.64
1	4	848	3.58	-9.41	968	4.50	-10.53
1	6	864	3.83	-7.21	913	4.53	-10.97
1	8	896	4.19	-8.26	897	4.77	-10.49
2	1	543	2.25	-8.95	607	2.68	-10.57
2	2	522	2.23	-9.81	562	2.56	-10.47
2	4	552	2.51	-9.29	555	2.74	-10.45
2	6	540	2.61	-8.60	552	2.95	-10.81
2	8	552	2.83	-6.88	563	3.23	-11.03
4	1	400	1.83	-0.41	454	2.20	-10.34
4	2	430	2.04	0.61	423	2.16	-10.50
4	4	440	2.27	0.32	429	2.45	-10.75
4	6	420	2.37	0.06	429	2.71	-10.61
4	8	480	2.91	0.09	445	3.10	-11.81
6	1	420	2.07	-0.01	403	2.11	-10.85
6	2	420	2.20	0.13	376	2.11	-11.23
6	4	448	2.63	-0.69	398	2.57	-10.89
6	6	420	2.73	0.43	425	3.10	-10.50
6	8	448	3.20	-0.73	396	3.22	-10.70
8	1	423	2.25	1.76	364	2.06	-9.91
8	2	396	2.27	0.48	387	2.37	-9.87
8	4	432	2.82	2.33	401	2.88	-9.90
8	6	486	3.56	3.21	437	3.58	-9.86
8	8	504	4.13	3.25	435	4.04	-9.92

(f) E-SSOR 前処理つき GBiCGSTAB(s, L) 法
(f) E-SSOR preconditioned GBiCGSTAB(s, L) method.

s	L	AC なし			AC つき		
		反復回数	反復時間	TRR	反復回数	反復時間	TRR
1	1	1,158	3.49	-10.72	952	3.20	-10.72
1	2	916	2.67	-10.71	750	2.43	-10.69
1	4	616	1.82	-10.62	741	2.55	-10.69
1	6	684	2.19	-10.75	712	2.69	-10.70
1	8	704	2.40	-10.66	647	2.62	-10.70
2	1	540	1.63	-10.71	596	1.98	-10.70
2	2	510	1.54	-10.71	521	1.74	-10.71
2	4	468	1.51	-10.72	498	1.82	-10.70
2	6	504	1.76	-10.65	529	2.14	-10.66
2	8	504	1.92	-10.11	487	2.15	-10.69
4	1	360	1.20	-10.71	414	1.50	-10.72
4	2	350	1.21	-10.71	378	1.43	-10.71
4	4	360	1.38	-10.50	366	1.60	-10.71
4	6	390	1.67	-8.48	366	1.82	-10.70
4	8	400	1.89	-7.93	407	2.26	-10.71
6	1	336	1.24	-10.69	328	1.31	-10.70
6	2	336	1.32	-10.68	316	1.35	-10.70
6	4	336	1.52	-10.65	314	1.60	-10.70
6	6	336	1.74	-6.10	340	2.01	-10.72
6	8	392	2.27	-4.35	340	2.29	-10.72
8	1	297	1.21	-10.60	295	1.28	-10.71
8	2	306	1.35	-10.01	291	1.40	-10.65
8	4	324	1.68	-9.41	330	1.90	-10.72
8	6	378	2.29	-7.32	382	2.61	-10.71
8	8	360	2.47	-4.31	364	2.86	-10.71

氏に心より感謝する。

参考文献

- [1] 相原研輔, 阿部邦美, 石渡恵美子: 近似解の精度を改善する IDRstab 法 (科学技術計算における理論と応用の新展開), 数理解析研究所講究録, 京都大学, No.1791, pp.11-20 (2012).
- [2] Axelsson, O.: A generalized SSOR method, BIT, Vol.12, pp.443-467 (1972).
- [3] Chan, T. F., van der Vorst, H. A.: Approximate and Incomplete Factorizations, Parallel Numerical Algorithms, ICASE/LaRC Interdisciplinary Series in Sci. and Eng., Kluwer Academic, Vol.4, pp.167-202 (1997).
- [4] Chen, X., Toh, K. C., Phoon, K. K.: A modified SSOR preconditioner for sparse symmetric indefinite linear systems of equations, Int. J. Numer. Meth. Engng, Vol.65, pp.785-807 (2006).
- [5] Davis, T.: University of Florida Spares Matrix Collection: <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- [6] Eisenstat, S. C.: Efficient implementation of a class of preconditioned conjugate gradient methods, SIAM J. Sci. Stat. Compute., Vol.2, pp.1-4 (1981).
- [7] 藤野清次, Sonneveld, P., 尾上勇介, van Gijzen, M.B.: IDR(s)-SOR 法の提案, 日本応用数学会論文誌, Vol.20, No.4, pp.289-308 (2010).
- [8] 東慶幸, 藤野清次, 尾上勇介: Eisenstat 版 GS 型前処理付き MRTR 法の収束性について, 日本計算工学会論文集, No.2011, 20110006 (2011).
- [9] 村上 啓一, 藤野 清次, 尾上 勇介, 平良 賢剛: メモリアクセスの視点からの Eisenstat 版前処理の考察, 日本シミュ

表 3 前処理つき反復法の精度比較

Table 3 Comparison of the accuracy between the preconditioned iterative methods.

		(a) IDRstab(s, L)						合計
AC	前処理	TRR					max	
		≥ -8	$-8 \sim -9$	$-9 \sim -10$	$-10 \sim -11$	$-11 \geq$		
なし	なし	160 (46%)	44 (13)	41 (12)	27 (8)	43 (12)	35 (10)	350
	ILU(0)	136 (46)	34 (13)	32 (12)	36 (8)	99 (12)	13 (10)	
	SSOR	118 (34)	25 (7)	30 (9)	49 (14)	102 (29)	25 (7)	
あり	なし	7 (2)	12 (3)	32 (9)	50 (14)	212 (61)	37 (11)	
	ILU(0)	19 (5)	10 (3)	28 (8)	42 (12)	233 (67)	18 (5)	
	SSOR	1 (0)	6 (2)	25 (7)	45 (13)	243 (69)	30 (9)	

- [10] レーション学会論文誌, Vol.3, No.2, pp.36-47 (2011).
- [10] Saad, Y., van der Vorst, H.A.: Iterative solution of linear systems in the 20th century, J. of Compute. Appl. Math., Vol.123, pp.1-33 (2000).
- [11] 櫻井隆雄, 直野健, 恵木正史, 猪貝光洋, 木立啓之, 小路将徳: 高速度性と信頼性を両立する AC-IDR(s) 法の提案と実装, 情報処理学会 HPCS2009, pp.81-88, (2009).
- [12] Sleijpen, G.L.G., Fokkema, D.R.: BiCGstab(ℓ) for equations involving unsymmetric matrices with complex spectrum, Electronic Transactions on Numerical Analysis,

(b) GBiCGSTAB(s, L)

AC	前処理	TRR					max	合計
		≥ -8	$-8 \sim -9$	$-9 \sim -10$	$-10 \sim -11$	$-11 \geq$		
なし	なし	113 (32%)	19 (5)	27 (8)	25 (7)	103 (29)	63 (18)	350
	ILU(0)	86 (25)	27 (8)	39 (13)	40 (8)	149 (39)	9 (9)	
	SSOR	88 (25)	21 (6)	45 (13)	28 (8)	137 (39)	31 (9)	
あり	なし	31 (9)	10 (3)	38 (11)	20 (6)	188 (54)	63 (18)	
	ILU(0)	16 (5)	7 (2)	16 (5)	46 (13)	259 (74)	6 (2)	
	SSOR	0 (0)	2 (1)	24 (7)	46 (13)	253 (72)	25 (7)	

Vol.1, pp.11-32 (1993).

- [13] Sleijpen, G.L.G., Sonneveld, P., van Gijzen, M.B.: Bi-CGSTAB as an induced dimension reduction method, Appl. Numer. Math., Vol.60, pp.1100-1114 (2010).
- [14] Sleijpen, G.L.G., van Gijzen, M.B.: Exploiting BiCGstab(ℓ) strategies to induce dimension reduction, SIAM J. Sci. Comput., Vol.35, No.5, pp.2687-2709 (2010).
- [15] Sonneveld, P., van Gijzen, M.B.: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., Vol.31, No.2, pp.1035-1062 (2008).
- [16] Tanio, M., Sugihara, M.: 一般化 Bi-CGSTAB(s, L) (= 一般化 IDR(s, L)), 数理解析研究所講究録, 京都大学, No.1638, pp.95-111 (2009).
- [17] 塚田健, 深堀康紀, 谷尾真明, 杉原正顯: 自動残差修正機能付き GBiCGSTAB(s, L) 法 (科学技術計算アルゴリズムの数理的基盤と展開), 数理解析研究所講究録, 京都大学, No.1733, pp.149-159 (2011).
- [18] 塚田健: 一般化 IDR 定理に基づく反復解法に関する研究, 東京大学大学院情報理工学系研究科修士論文 (2010).
- [19] van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing, Vol.12, pp.631-644 (1992).
- [20] Wesseling, P., Sonneveld, P.: Numerical Experiments with a Multiple Grid-and a Preconditioned Lanczos Type Methods, Lecture Notes in Math., Springer, No.771, pp.543-562 (1980).