降順に展開した漸化式を使わない積型反復法

関本, 幹

藤野, 清次
九州大学情報基盤研究開発センター

https://doi.org/10.15017/20056

出版情報：九州大学大学院システム情報科学紀要. 16 (2), pp.70-74, 2011-09-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第 16 巻　第 2 号　平成 23 年 9 月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.16, No.2, September 2011

降順に展開した漸化式を使わない積型反復法

関本　　幹* ・ 藤野　清次**

Product-type Iterative Method without Reverse-ordered Recurrence

Takashi SEKIMOTO* and Seiji FUJINO**

(Received July 22, 2011)

Abstract: We consider to solve a linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$ by iterative method. Product-type of iterative methods such as GPBiCG, BiCGSafe and GPBiCG_AR method constitute a sequence of polynomial by multiplying Lanczos polynomials by acceleration polynomial. In this paper, we focus on the order of the development of recurrence for GPBiCG method, and propose a GPBiCGSafe method by improving GPBiCG_AR method. Through numerical experiments, we make clear that the GPBiCGSafe method has excellent convergence rate and robustness.

Keywords: Iterative method, GPBiCG_AR method, GPBiCGSafe method, Preconditioning

1. はじめに

自然科学や工学などの様々な解析分野において，数値シミュレーションは重要な役割を果たしてきた．それらの様々な解析分野から生じる問題は，偏微分方程式で記述されることが多い．偏微分方程式は有限要素法などにより離散化することで，連立一次方程式に帰着される．この連立一次方程式の解法として反復法が提案されている．

GPBiCG(Generalized Product-type Bi-Conjugate Gradient) 法[1]や BiCGSafe 法[2]，GPBiCG_AR(Associate Residual) 法[3]などの積型反復法は，通常のランチョス多項式に加速多項式を掛けて，積の形の多項式列を構成することで反復法の収束を加速させるという発想に基づく反復法である．最近，阿部らにより GPBiCG 法の変形版が提案された[4]．そこでは，残差ベクトルの更新で用いられる加速多項式に交代漸化式の代わりに三項漸化式が適用された．以下では、それらを AS(Abe-Sleijpen)_GPBiCG(_v1-v4) 法と呼ぶ．

本論文では，GPBiCG 法において，交代漸化式から三項漸化式に変形するときに用いられる降順な漸化式の存在に着目し，それらを消しさることによって GPBiCG_AR 法の収束性の改良を図ったので，その結果を検証する．

本論文の構成は以下の通りである．第 2 節で，GPBiCG 法の概要と漸化式の展開について述べ，GPBiCG_AR 法を改良した GPBiCGSafe 法を導く．第 3 節で，数値実験により，GPBiCGSafe 法の収束性を検証する．そして，第 5 節で，まとめと今後の課題を述べる．

平成 23 年 7 月 22 日受付
*情報学専攻修士課程
**情報基盤研究開発センター

2. 積型反復法

2.1 GPBiCG 法の算法の中の降順展開した漸化式

解くべき連立一次方程式を

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

とする．ただし，係数行列 A は大きさ $n \times n$ の実非対称行列, $\boldsymbol{x}$ と $\boldsymbol{b}$ は各々次数 n の解ベクトルと右辺ベクトルとする．初期近似解を $\boldsymbol{x}_0$ とし，初期残差ベクトルを $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ とする．このとき，BiCG 法において，反復 n 回目の残差ベクトル $\boldsymbol{r}_n^{\mathrm{BiCG}}$ と補助ベクトル $\boldsymbol{p}_n^{\mathrm{BiCG}}$ は以下のように表される．

$$\boldsymbol{r}_n^{\mathrm{BiCG}} = R_n(A)\boldsymbol{r}_0, \quad \boldsymbol{p}_n^{\mathrm{BiCG}} = P_n(A)\boldsymbol{r}_0. \tag{2}$$

多項式 $R_n(A)$ と $P_n(A)$ は次の交代漸化式を満たす．

$$R_0(\lambda) = 1, P_0(\lambda) = 1, \tag{3}$$
$$R_n(\lambda) = R_{n-1}(\lambda) - \alpha_{n-1}\lambda P_{n-1}(\lambda), \tag{4}$$
$$P_n(\lambda) = R_n(\lambda) + \beta_{n-1}P_{n-1}(\lambda), \quad n = 1, 2, \ldots. \tag{5}$$

BiCG 法の残差列を $\{\boldsymbol{r}_0^{\mathrm{BiCG}}, \boldsymbol{r}_1^{\mathrm{BiCG}}, \ldots, \boldsymbol{r}_n^{\mathrm{BiCG}}\}$ とする．このとき，適当な手続きにより多項式列 $\{H_0, H_1, \ldots, H_n\}$ を生成し, $\{H_0(A)\boldsymbol{r}_0^{\mathrm{BiCG}}, H_1(A)\boldsymbol{r}_1^{\mathrm{BiCG}}, \ldots, H_n(A)\boldsymbol{r}_n^{\mathrm{BiCG}}\}$ を構成することで，BiCG 法の残差列の収束の加速を図る．このように，残差が BiCG 法の残差と加速多項式との積で定義された解法は**積型反復法**と呼ばれる．すなわち，積型反復法の残差ベクトル $\boldsymbol{r}_n$ は

$$\boldsymbol{r}_n \;=\; H_n(A)\boldsymbol{r}_n^{\mathrm{BiCG}} = H_n(A)R_n(A)\boldsymbol{r}_0 \tag{6}$$

と表される．次に，固有値 λ を使って多項式列 $\{G_n(\lambda)\}$ と $\{H_n(\lambda)\}$ を次の (交代) 漸化式で構成する．

$$H_0(\lambda) \;=\; 1,\; G_0(\lambda) = \zeta_0, \tag{7}$$

$$H_n(\lambda) \;=\; H_{n-1}(\lambda) - \lambda G_{n-1}(\lambda), \tag{8}$$

$$G_n(\lambda) \;=\; \zeta_n H_n(\lambda) + \eta_n G_{n-1}(\lambda),\; n = 1, 2, \ldots. \tag{9}$$

通常，式 (8) と式 (9) の漸化式 H_n と G_n は，$H_1 \to G_1 \to H_2 \to G_2 \to H_3 \to G_3 \to \cdots$ の順番で更新される．すなわち，式 (8) と式 (9) は**昇順に展開**される．一方，GPBiCG 法では，多項式列 $\{G_n(\lambda)\}$ は昇順に展開するのではなく，次のように**降順に展開**して使用された．すなわち，n ステップ目の $G_n(\lambda)$ の多項式を次の $(n+1)$ ステップ目の $H_{n+1}(\lambda)$ を使って降順に展開した漸化式を使用した．

$$G_n(\lambda) \;=\; -(H_{n+1}(\lambda) - H_n(\lambda))/\lambda \tag{10}$$

このように**降順に展開した**漸化式を使うと，多項式列 $\{H_n(A)\}$ は次の 3 項から成る漸化式が得られるという利点がある．一方，数学的には同値でも，降順展開された漸化式は，まるめ誤差の影響を受け易くなる可能性がある．

$$H_0(\lambda) \;=\; 1,\; H_1(\lambda) = (1 - \zeta_0\lambda)H_0(\lambda), \tag{11}$$

$$H_{n+1}(\lambda) \;=\; (1 + \eta_n - \zeta_n\lambda)H_n(\lambda) - \eta_n H_{n-1}(\lambda), n = 1, 2, \ldots \tag{12}$$

ここで，ζ_n, η_n は新規のパラメータである．生成された多項式列 $\{H_n(\lambda)\}$ は，任意の n に対して $H_n(0) = 1$ を満たすので，$H_{n+1}(0) - H_n(0) = 0$ である．ここで，漸化式 (12) は式 (9) を式 (8) に代入し，それを式 (10) を使って書き直すと容易に得られる．以下では，多項式 $H_n(\lambda), G_n(\lambda)$ を H_n, G_n と略記す．得られた GPBiCG 法の算法を示す．

GPBiCG 法の算法

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,

Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0,\; \beta_{-1} = 0$,

set $\boldsymbol{p}_{-1} = \boldsymbol{t}_{-1} = \boldsymbol{u}_{-1} = \boldsymbol{w}_{-1} = \boldsymbol{z}_{-1} = 0$,

for $n = 0, 1, \cdots$ until $||\boldsymbol{r}_n|| \leq \varepsilon\, ||\boldsymbol{r}_0||$ do :

begin

$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}(\boldsymbol{p}_{n-1} - \boldsymbol{u}_{n-1}), \tag{13}$$

$$\alpha_n = \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_n)}, \tag{14}$$

$$\boldsymbol{y}_n = \boldsymbol{t}_{n-1} - \boldsymbol{r}_n - \alpha_n\boldsymbol{w}_{n-1} + \alpha_n A\boldsymbol{p}_n, \tag{15}$$

$$\boldsymbol{t}_n = \boldsymbol{r}_n - \alpha_n A\boldsymbol{p}_n, \tag{16}$$

$$\boldsymbol{a}_n = \boldsymbol{t}_n,\quad \boldsymbol{b}_n = \boldsymbol{y}_n,\quad \boldsymbol{c}_n = A\boldsymbol{t}_n, \tag{17}$$

$$\zeta_n = \frac{(\boldsymbol{b}_n, \boldsymbol{b}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{a}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{18}$$

$$\eta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{19}$$

$$(\text{if } n = 0,\ \text{then } \zeta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)},\ \eta_n = 0) \tag{20}$$

$$\boldsymbol{u}_n = \zeta_n A\boldsymbol{p}_n + \eta_n(\boldsymbol{t}_{n-1} - \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{u}_{n-1}), \tag{21}$$

$$\boldsymbol{z}_n = \zeta_n\boldsymbol{r}_n + \eta_n\boldsymbol{z}_{n-1} - \alpha_n\boldsymbol{u}_n, \tag{22}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n\boldsymbol{p}_n + \boldsymbol{z}_n, \tag{23}$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{t}_n - \eta_n\boldsymbol{y}_n - \zeta_n A\boldsymbol{t}_n, \tag{24}$$

$$\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{n+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}, \tag{25}$$

$$\boldsymbol{w}_n = A\boldsymbol{t}_n + \beta_n A\boldsymbol{p}_n, \tag{26}$$

end

ここでは，算法を構成する過程で降順な漸化式 (10) が使われる様子を，中間ベクトル $\boldsymbol{y}_n$ と $\boldsymbol{u}_n$ を取り上げ細かく観察する．すなわち，それらのベクトルの定義式と式変形は次のように各々書き表せる．下線部分が降順の漸化式 (10) の部分である．

$$\boldsymbol{y}_n := \underline{\lambda G_n} R_{n+2} \tag{27}$$

$$= \underline{(H_n - H_{n+1})}R_{n+2}$$

$$= H_n(R_{n+1} - \alpha_{n+1}P_{n+1}) - H_{n+1}(R_{n+1} - \alpha_{n+1}P_{n+1}) \tag{28}$$

$$\boldsymbol{u}_n := \lambda G_n P_n \tag{29}$$

$$= \lambda P_n(\zeta_n H_n + \eta_n G_{n-1})$$

$$= \zeta_n\lambda H_n P_n + \eta_n\lambda G_{n-1}P_n$$

$$= \zeta_n\lambda H_n P_n + \eta_n\lambda G_{n-1}(R_n + \beta_{n-1}P_{n-1})$$

$$= \zeta_n\lambda H_n P_n + \eta_n(\underline{\lambda G_{n-1}}R_n + \beta_{n-1}\lambda G_{n-1}P_{n-1})$$

$$= \zeta_n\lambda H_n P_n + \eta_n(\underline{(H_{n-1} - H_n)}R_n + \beta_{n-1}\lambda G_{n-1}P_{n-1}) \tag{30}$$

2.2 GPBiCG_AR 法

GPBiCG 法の算法中の式 (23) において，降順な漸化式を使わないようにした算法が GPBiCG_AR 法である．以下に，GPBiCG_AR 法の算法を示す．

GPBiCG_AR 法の算法

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,

Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0,\; \beta_{-1} = 0$,

set $\boldsymbol{p}_{-1} = \boldsymbol{t}_{-1} = \boldsymbol{u}_{-1} = \boldsymbol{z}_{-1} = 0$,

for $n = 0, 1, \cdots$ until $||\boldsymbol{r}_n|| \leq \varepsilon\, ||\boldsymbol{r}_0||$ do :

begin

$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}(\boldsymbol{p}_{n-1} - \boldsymbol{u}_{n-1}), \tag{31}$$

$$A\boldsymbol{p}_n = A\boldsymbol{r}_n + \beta_{n-1}(A\boldsymbol{p}_{n-1} - A\boldsymbol{u}_{n-1}), \tag{32}$$

$$\alpha_n = \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_n)}, \tag{33}$$

$$\boldsymbol{a}_n = \boldsymbol{r}_n, \quad \boldsymbol{b}_n = A\boldsymbol{z}_{n-1}, \quad \boldsymbol{c}_n = A\boldsymbol{r}_n, \tag{34}$$

$$\zeta_n = \frac{(\boldsymbol{b}_n, \boldsymbol{b}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{a}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{35}$$

$$\eta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{36}$$

$$(\text{if } n = 0, \text{ then } \zeta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)}, \ \eta_n = 0) \tag{37}$$

$$\boldsymbol{u}_n = \zeta_n A\boldsymbol{p}_n + \eta_n(\boldsymbol{t}_{n-1} - \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{u}_{n-1}), \tag{38}$$

$$\boldsymbol{t}_n = \boldsymbol{r}_n - \alpha_n A\boldsymbol{p}_n, \tag{39}$$

$$\boldsymbol{z}_n = \zeta_n \boldsymbol{r}_n + \eta_n \boldsymbol{z}_{n-1} - \alpha_n \boldsymbol{u}_n, \tag{40}$$

$$A\boldsymbol{z}_n = \zeta_n A\boldsymbol{r}_n + \eta_n A\boldsymbol{z}_{n-1} - \alpha_n A\boldsymbol{u}_n, \tag{41}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \boldsymbol{p}_n + \boldsymbol{z}_n, \tag{42}$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{t}_n - A\boldsymbol{z}_n, \tag{43}$$

$$\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{n+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}, \tag{44}$$

end

前小節で考察したように，降順な漸化式を使わないようにするためには，中間ベクトル $\boldsymbol{y}_n$ を使用しなければよいが，パラメター ζ_n と η_n を決めることができなくなる．そこで，GPBiCG 法のように残差ベクトルの 2 ノルムの最小化からではなく，准残差ベクトルの 2 ノルムの最小化からパラメータの値を定めた算法が GPBiCG_AR 法である．この施策により，中間ベクトル $\boldsymbol{y}_n$(および $\boldsymbol{w}_n$) を使う必要がなくなったが，中間ベクトル $\boldsymbol{u}_n$ はそのまま残ることになった．

2.3 阿部らの変形版 GPBiCG 法と漸化式

阿部らの考案した算法は，降順な漸化式 (10) の使用を避ける点では本研究と同じである．しかし，漸化式 (10) を適用して得られた 3 項漸化式 (11), (12) はそのまま使用するという点が我々と異なる．

すなわち，阿部，Sleijpen の変形版 GPBiCG (以下では，AS_GPBiCG(_v1-v4) と略す) 法の残差ベクトル $\boldsymbol{r}_{n+1}$ は次のように構成される．

$$\begin{aligned} H_{n+1}R_{n+1} \ = \ & (1+\eta_n)H_nR_{n+1} - \zeta_n AH_nR_{n+1} \\ & -\eta_n H_{n-1}R_{n+1} \end{aligned} \tag{45}$$

しかし，この展開では，降順な漸化式の使用の影響がすでに算法全体に広がっている可能性がある．

2.4 GPBiCGSafe 法

すでに述べたように，GPBiCG_AR 法において，中間ベクトル $\boldsymbol{u}_n$ の更新において降順な漸化式が 1 度だけ使用されている．そこで，ベクトル $\boldsymbol{u}_n$ をベクトル $\boldsymbol{z}_n$ を用いて次のように書き表す．この式変形 (下線を付けた部分) により降順な漸化式 (10) の使用が消滅した．

$$\begin{aligned} \boldsymbol{u}_n \ &:= \ \lambda G_n P_n \qquad (46) \\ &= \ \lambda P_n(\zeta_n H_n + \eta_n G_{n-1}) \\ &= \ \zeta_n \lambda H_n P_n + \eta_n \lambda G_{n-1}(R_n + \beta_{n-1}P_{n-1}) \\ &= \ \zeta_n \lambda H_n P_n \\ &\quad +\eta_n(\underline{\lambda G_{n-1}R_n} + \beta_{n-1}\lambda G_{n-1}P_{n-1}) \\ &= \ \zeta_n A\boldsymbol{p}_n + \eta_n(\underline{A\boldsymbol{z}_{n-1}} + \beta_{n-1}A\boldsymbol{u}_{n-1}) \qquad (47) \end{aligned}$$

このように，降順な漸化式 (10) を使わないよう改良した算法を **GPBiCGSafe 法**と呼ぶ．また，$A\boldsymbol{z}_{n-1}$ は従来の GPBiCG_AR 法でも使用されていたため計算量は増えない．付録に **GPBiCGSafe 法**の算法を示す．

以上の考察より，降順な漸化式の数に注目してその個数をまとめたものを **Table 1** に示す．

Table 1 The number of reverse-ordered recurrence.

method	reverse-ordered recurrence
GPBiCG	2
AS_GPBiCG_v1, -_v2	2
GPBiCG_AR	1
GPBiCGSafe	0
BiCGSafe	0

2.5 ILU(0) 前処理

ILU(0) 前処理は，係数行列を，

$$A \ = \ (\tilde{L} + I)(\tilde{U} + \tilde{D}) + R \tag{48}$$

と分解し，

$$K \ = \ (\tilde{L} + I)(\tilde{U} + \tilde{D}) \tag{49}$$

とする前処理である．ここで，$\tilde{L}$, $\tilde{D}$, $\tilde{U}$ は ILU(0) 分解によって生成された狭義下三角行列, 対角行列, 狭義上三角行列を各々意味し，R は $\tilde{L}$, $\tilde{D}$, $\tilde{U}$ のスパース性を保つための誤差行列を意味する．ILU(0) 前処理では，不完全分解過程において非零要素を除いた位置に現れるフィルインを考慮しない．**Table 2** に ILU(0) 前処理つき積型反復法における計算量を示す．

3. 数 値 実 験

3.1 計算機環境と計算条件

計算はすべて倍精度浮動小数点演算で行った．計算機は Nehalem (CPU：Intel Xenon X5570, クロック周波数：2.93GHz, メモリ：24Gbytes, OS：RedHat Enterprise Linux 5.6) を用いた．プログラムは Fortran90 を用いて実装し，最適化オプションは “-O3” を使用した．右辺項ベクトル $\boldsymbol{b}$ は厳密解を $\hat{\boldsymbol{x}} \ = \ (1, 1, ..., 1)^T$

Table 2 Computational costs of six kinds of methods with ILU(0) preconditioning per iteration.

method	$K^{-1}\boldsymbol{v}$ ($\times 2nz$)	$A\boldsymbol{v}$ ($\times 2nz$)	$(\boldsymbol{u},\boldsymbol{v})$ ($\times 2n$)	$\boldsymbol{u}\pm\boldsymbol{v}$ ($\times n$)	$\alpha\boldsymbol{v}$ ($\times n$)
GPBiCG	2	2	7	17	14
AS_GPBiCG_v1	2	2	8	15	16
AS_GPBiCG_v2	2	2	8	15	15
GPBiCG_AR	2	2	7	15	13
GPBiCGSafe	2	2	7	14	13
BiCGSafe	2	2	7	14	13

とし，$\boldsymbol{b} = A\hat{\boldsymbol{x}}$ で作成した．収束判定条件は相対残差の 2 ノルム：$||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2 \leq 10^{-10}$ とした．初期近似解 $\boldsymbol{x}_0$ はすべて 0 とした．行列は予め対角スケーリングによって対角項をすべて 1.0 に正規化した．最大反復回数は 10000 回とした．調べた反復法は GPBiCG 法，AS_GPBiCG_v1 法，AS_GPBiCG_v2 法，GPBiCG_AR 法，GPBiCGSafe 法，BiCGSafe 法の計 6 種類とし，前処理としてフィルインを考慮しない ILU(0) 分解を選んだ．初期シャドウ残差ベクトル $\boldsymbol{r}_0^\star$ には初期残差ベクトル $\boldsymbol{r}_0(=\boldsymbol{b}-A\boldsymbol{x}_0)$ を代入した．

3.2 テスト問題

Table 3 に本実験で使用する 17 種類のテスト行列の特徴を示す．行列 air-cfl5 は Manchester 大学 F. Costen 研究室，行列 waseda は早稲田大学若尾研究室提供の行列で，その他 15 種類の行列はフロリダ大学の疎行列データベースから選出した[5]．表中の “N” は次元数，“NNZ” は総非零要素数，“ave. NNZ” は 1 行当りの平均非零要素数を各々表す．

3.3 実験結果

Table 4-5 に 5 種類の ILU(0) 前処理つき積型反復法の収束性を示す．表中の “max” は最大反復回数までで収束しなかったことを意味し，“itr.(iteration)”，“ave.(average) time”，“TRR” は反復回数，平均反復時間，真の相対残差 (True Relative Residual) の常用対数 $\log_{10}\left(||\boldsymbol{b}-A\boldsymbol{x}_{n+1}||_2/||\boldsymbol{b}-A\boldsymbol{x}_0||_2\right)$ の値を意味する．最大反復回数に達するまでに収束しかつ近似解に対する真の相対残差の 2 ノルムの大きさ TRR が $10^{-9.60}$ 以上の場合は**偽収束** (表中の TRR の数字に括弧をつけた) と判定した．**Table 6** に 5 種類の解法における収束 (未収束) ケース，偽収束 (spurious) ケースと最速ケースを示す．**Table 7** に 6 種類の反復解法の速度順位とその総合順位を示す．ただし，成績表の集計について，計算時間が同じ場合は同一順位にした．また収束しなかった場合は順位を 6 位にした．例えば，行列 k3plates において GPBiCG 法と AS_GPBiCG_v2 法は両方とも 6 位である．**Table 4-7** の観察から以下の知見が得られる．

Table 3 Specifications of test matrices.

matrix	N	NNZ	ave. NNZ	analytical field
air-cfl5	1,536,000	19,435,428	12.65	hydro
atmosmodd	1,270,432	8,814,880	6.94	dynamic
poisson3Db	85,623	2,374,949	27.74	
raefsky3	21,200	1,488,768	70.22	
water_tank	60,740	2,035,281	33.51	
bcircuit	68,902	375,558	5.45	electrical
Freescale1	3,428,755	17,052,626	4.97	circuit
memplus	17,758	126,150	7.1	
sme3Da	12,504	874,887	69.97	structural
sme3Db	29,067	2,081,063	71.6	
sme3Dc	42,930	3,148,656	73.34	
appu	14,000	1,853,104	132.36	directed
big	13,209	91,465	6.92	weighted graph
ecl32	51,993	380,415	7.32	semiconductor
epb3	84,617	463,625	5.48	thermal
k3plates	11,107	378,927	34.12	acoustics
waseda	19,060	24,377,548	1278.99	electromagnetics

- GPBiCG_AR 法と GPBiCGSafe 法，BiCGSafe 法は全ての行列で収束したが，残り 3 種類の解法は反復回数が最大回数に達したり，収束しても偽収束現象が発生したときがあった．
- AS_GPBiCG_v1 法では行列 bcircuit，sme3Dc の 2 ケース，AS_GPBiCG_v2 法では行列 sme3Db の 1 ケースで偽収束が発生した．また GPBiCG 法で行列 sme3Db の 1 ケースで偽収束が発生した．
- 17 種類のテスト行列において，最も速く収束したケースが多い解法は GPBiCGSafe 法の 8 ケースである．
- GPBiCG 法の最も速く収束したケースが零に対し，AS_GPBiCG_v1，_2 法は 1 ケースと 2 ケースであるが，GPBiCG 法よりも計算時間が長いケースが多い．
- 行列 bcircuit において，GPBiCGSafe 法の計算時間は GPBiCG_AR 法の計算時間より約 10%ほど速い．

4. ま と め

本論文では，積型反復法で降順な漸化式で定義される多項式 $\{G_n(A)\}$ の存在に着目し，GPBiCG_AR 法において降順な漸化式を使用しない GPBiCGSafe 法を提案した．数値実験の結果，提案法が収束性において優れていることがわかった．

参 考 文 献

1) S.-L. Zhang, GPBi-CG: Generalized product-type methods preconditionings based on Bi-CG for solving non-symmetric linear systems, SIAM J. Sci. Comput.,

pp.537-551, 1997.
2) 藤野 清次, 藤原 牧, 吉田 正浩, 準残差の最小化に基づく BiCGSafe 法の収束性について,Transactions of JSCES, Vol.2005, 20050028, 2005.
3) Moethuthu, S. Fujino: Stability of GPBiCG_AR method based on minimization of associate residual, Journal of ASCM, pp.108-120, 2008.
4) K. Abe, G.L.G. Sleijpen, Solving linear equations with a stabilized GPBiCG method, 日本応用数理学会 第 14 回 環瀬戸内応用数理研究部会, 岡山理科大学, Jan., 2011.
5) University of Florida Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/index.html

Table 4 Performance of six kinds of iterative methods.

matrix	method	itr.	time [sec.]	ave.time [msec.]	TRR
air-cfl5	GPBiCG	21	7.032	258.238	-10.11
	AS_GPBiCG_v1	21	7.205	266.143	-10.11
	AS_GPBiCG_v2	21	7.190	264.952	-10.11
	GPBiCGAR	21	6.776	245.952	-10.06
	GPBiCGSafe	21	**6.749**	244.190	-10.06
	BiCGSafe	21	6.787	246.429	-10.06
atmosmodd	GPBiCG	89	14.157	153.955	-10.08
	AS_GPBiCG_v1	90	14.821	159.667	-10.03
	AS_GPBiCG_v2	90	14.716	158.533	-10.10
	GPBiCGAR	90	13.341	143.078	-10.09
	GPBiCGSafe	90	**13.247**	142.122	-10.24
	BiCGSafe	93	13.685	142.215	-10.15
poisson3Db	GPBiCG	86	3.481	33.884	-10.12
	AS_GPBiCG_v1	84	3.440	34.250	-10.07
	AS_GPBiCG_v2	84	3.424	34.060	-10.08
	GPBiCGAR	84	3.383	33.548	-10.06
	GPBiCGSafe	84	3.369	33.405	-10.23
	BiCGSafe	83	**3.354**	33.506	-10.05
raefsky3	GPBiCG	129	1.759	12.047	-10.04
	AS_GPBiCG_v1	133	1.803	12.023	-10.80
	AS_GPBiCG_v2	125	**1.711**	12.032	-10.23
	GPBiCGAR	134	1.803	11.948	-10.65
	GPBiCGSafe	134	1.806	11.940	-10.36
	BiCGSafe	140	1.882	11.964	-10.28
water_tank	GPBiCG	270	5.260	18.641	-10.02
	AS_GPBiCG_v1	285	5.537	18.628	-10.27
	AS_GPBiCG_v2	276	5.344	18.551	-10.02
	GPBiCGAR	279	5.338	18.351	-10.29
	GPBiCGSafe	269	**5.154**	18.305	-10.01
	BiCGSafe	275	5.260	18.305	-10.29
bcircuit	GPBiCG	6995	55.321	7.905	-10.01
	AS_GPBiCG_v1	4765	38.645	8.104	(-9.41)
	AS_GPBiCG_v2	4553	36.275	7.961	-10.18
	GPBiCGAR	3594	27.715	7.702	-10.52
	GPBiCGSafe	3239	**24.851**	7.663	-10.14
	BiCGSafe	4114	32.091	7.793	-10.21
Freescale1	GPBiCG	1681	675.306	400.659	-10.09
	AS_GPBiCG_v1	1567	659.221	419.536	-10.05
	AS_GPBiCG_v2	2135	890.542	416.282	-9.94
	GPBiCGAR	1347	**500.602**	370.317	-10.02
	GPBiCGSafe	1384	512.137	368.749	-10.11
	BiCGSafe	1409	531.936	376.251	-10.11
memplus	GPBiCG	251	0.390	1.510	-10.04
	AS_GPBiCG_v1	257	0.416	1.572	-10.04
	AS_GPBiCG_v2	243	0.398	1.588	-10.03
	GPBiCGAR	251	0.379	1.458	-10.08
	GPBiCGSafe	244	**0.363**	1.443	-10.13
	BiCGSafe	248	0.377	1.476	-10.09

Table 5 Performance of six kinds of iterative methods.(cont'd)

matrix	method	itr.	time [sec.]	ave.time [msec.]	TRR
sme3Da	GPBiCG	1498	12.038	7.836	-10.22
	AS_GPBiCG_v1	1090	8.860	7.850	-10.23
	AS_GPBiCG_v2	1295	10.446	7.836	-9.68
	GPBiCGAR	1039	8.387	7.782	-10.04
	GPBiCGSafe	1072	8.604	7.745	-10.07
	BiCGSafe	949	**7.664**	7.759	-10.16
sme3Db	GPBiCG	1192	27.262	22.193	(-9.59)
	AS_GPBiCG_v1	1096	25.162	22.220	-10.06
	AS_GPBiCG_v2	1389	31.587	22.162	(-9.12)
	GPBiCGAR	831	**19.137**	22.060	-10.07
	GPBiCGSafe	1005	22.955	22.040	-10.01
	BiCGSafe	924	21.149	22.018	-10.04
sme3Dc	GPBiCG	2599	94.239	35.755	-10.80
	AS_GPBiCG_v1	2207	80.437	35.854	(-8.64)
	AS_GPBiCG_v2	1756	64.182	35.812	-9.96
	GPBiCGAR	1628	59.232	35.582	-10.17
	GPBiCGSafe	1603	**58.482**	35.669	-10.04
	BiCGSafe	1614	58.713	35.575	-10.01
appu	GPBiCG	27	1.765	18.296	-10.32
	AS_GPBiCG_v1	26	1.744	18.346	-10.05
	AS_GPBiCG_v2	26	**1.739**	18.346	-10.05
	GPBiCGAR	29	1.787	18.207	-10.22
	GPBiCGSafe	29	1.789	18.172	-10.20
	BiCGSafe	29	1.796	18.379	-10.31
big	GPBiCG	936	1.267	1.346	-10.00
	AS_GPBiCG_v1	831	1.161	1.390	-10.30
	AS_GPBiCG_v2	948	1.304	1.368	-10.49
	GPBiCGAR	769	**1.012**	1.308	-10.14
	GPBiCGSafe	803	1.056	1.305	-10.34
	BiCGSafe	812	1.082	1.324	-10.36
ecl32	GPBiCG	125	0.775	5.952	-10.11
	AS_GPBiCG_v1	126	0.805	6.167	-10.21
	AS_GPBiCG_v2	121	0.752	5.992	-10.15
	GPBiCGAR	124	0.755	5.831	-10.13
	GPBiCGSafe	123	**0.741**	5.797	-10.20
	BiCGSafe	124	0.754	5.847	-10.06
epb3	GPBiCG	89	0.784	8.528	-10.12
	AS_GPBiCG_v1	89	0.790	8.629	-10.73
	AS_GPBiCG_v2	89	0.900	9.820	-10.20
	GPBiCGAR	94	0.807	8.362	-10.02
	GPBiCGSafe	90	0.767	8.267	-10.06
	BiCGSafe	89	**0.722**	7.899	-10.15
k3plates	GPBiCG	max	-	-	-9.69
	AS_GPBiCG_v1	7675	24.443	3.180	-10.30
	AS_GPBiCG_v2	max	-	-	-9.54
	GPBiCGAR	5580	**17.357**	3.104	-10.33
	GPBiCGSafe	6391	19.845	3.100	-10.08
	BiCGSafe	5680	17.672	3.105	-10.94
waseda	GPBiCG	28	138.850	178.071	-10.44
	AS_GPBiCG_v1	28	138.874	177.714	-10.53
	AS_GPBiCG_v2	28	138.957	177.679	-10.53
	GPBiCGAR	28	138.835	177.357	-10.03
	GPBiCGSafe	28	**138.834**	178.143	-10.03
	BiCGSafe	28	138.874	179.393	-10.03

Table 6 Summary of performance of six kinds of preconditioned iterative methods.

method	convergence*	non-conv.	fastest cases
GPBiCG	16(1)/17	1/17	0/17
AS_GPBiCG_v1	17(2)	0	0
AS_GPBiCG_v2	16(1)	1	2
GPBiCG_AR	**17**	0	4
GPBiCGSafe	**17**	0	**8**
BiCGSafe	**17**	0	3

* The numbers in parenthesis mean spurious convergence for the approximate solutions $\boldsymbol{x}_{n+1}$.

Table 7 The ranking of convergence time and overall for six kinds of preconditioned iterative methods.

method	ranking							total	overall
	1	2	3	4	5	6	total	score	ranking
GPBiCG	0	2	3	3	**6**	3	17	73	4
AS_GPBiCG_v1	0	1	1	**8**	1	**6**	17	78	5
AS_GPBiCG_v2	2	0	0	4	4	**7**	17	80	6
GPBiCG_AR	4	**5**	**5**	1	2	0	17	43	2
GPBiCGSafe	**8**	4	3	0	2	0	17	**35**	**1**
BiCGSafe	3	**6**	**5**	1	0	2	17	46	3
total	17	18	17	17	15	18	-	-	

付録

Fig. A.1 に積型反復法の成長過程を示す．各枠内には重要なキーワード，解法名，発表年などを記した．

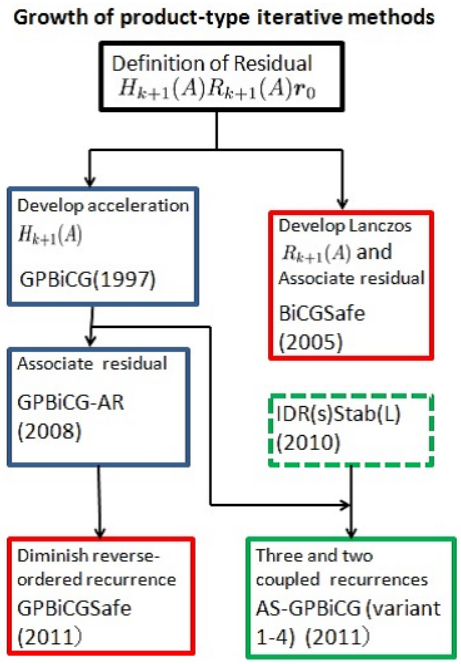

Fig. A.1 Growth of product-type iterative methods.

以下に，前処理つき GPBiCGSafe 法の算法を示す．

前処理つき GPBiCGSafe 法の算法

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,

Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0$, $\beta_{-1} = 0$,

set $\boldsymbol{p}_{-1} = \boldsymbol{u}_{-1} = \boldsymbol{z}_{-1} = 0$,

for $n = 0, 1, \cdots$ until $||\boldsymbol{r}_n|| \leq \varepsilon\ ||\boldsymbol{r}_0||$ do :

begin

$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}(\boldsymbol{p}_{n-1} - \boldsymbol{u}_{n-1}), \tag{A.1}$$

$$AK^{-1}\boldsymbol{p}_n = AK^{-1}\boldsymbol{r}_n + \beta_{n-1}(AK^{-1}\boldsymbol{p}_{n-1} - AK^{-1}\boldsymbol{u}_{n-1}), \tag{A.2}$$

$$\alpha_n = \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}{(\boldsymbol{r}_0^*, AK^{-1}\boldsymbol{p}_n)}, \tag{A.3}$$

$$\boldsymbol{a}_n = \boldsymbol{r}_n, \boldsymbol{b}_n = AK^{-1}\boldsymbol{z}_{n-1}, \boldsymbol{c}_n = AK^{-1}\boldsymbol{r}_n, \tag{A.4}$$

$$\zeta_n = \frac{(\boldsymbol{b}_n, \boldsymbol{b}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{a}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{A.5}$$

$$\eta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{A.6}$$

$$(\text{if } n = 0, \text{ then } \zeta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)},\ \eta_n = 0) \tag{A.7}$$

$$\boldsymbol{u}_n = \zeta_n AK^{-1}\boldsymbol{p}_n + \eta_n(AK^{-1}\boldsymbol{z}_n + \beta_{n-1}\boldsymbol{u}_{n-1}), \tag{A.8}$$

$$\boldsymbol{t}_n = \boldsymbol{r}_n - \alpha_n AK^{-1}\boldsymbol{p}_n, \tag{A.9}$$

$$\boldsymbol{z}_n = \zeta_n \boldsymbol{r}_n + \eta_n \boldsymbol{z}_{n-1} - \alpha_n \boldsymbol{u}_n, \tag{A.10}$$

$$AK^{-1}\boldsymbol{z}_n = \zeta_n AK^{-1}\boldsymbol{r}_n + \eta_n AK^{-1}\boldsymbol{z}_{n-1} - \alpha_n AK^{-1}\boldsymbol{u}_n, \tag{A.11}$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n K^{-1}\boldsymbol{p}_n + K^{-1}\boldsymbol{z}_n, \tag{A.12}$$

$$\boldsymbol{r}_{n+1} = \boldsymbol{t}_n - AK^{-1}\boldsymbol{z}_n, \tag{A.13}$$

$$\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{n+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}, \tag{A.14}$$

end