准残差と三項漸化式に基づく BiCGStar 法の性能評価

牛尾　恵浩[2]　村上　啓一[3]　藤野　清次[4]

概要：大規模な非対称疎行列を係数に持つ，線形な連立一次方程式 $A\boldsymbol{x} = \boldsymbol{b}$ を反復法で解くことを考える．積型反復法の中で広く知られた解法の一つに，GPBiCG 法がある．しかし，GPBiCG 法にはいくつかの問題点が存在する．そこで，GPBiCG 法を，准残差と三項漸化式の考え方による改良によって得られた，新しい BiCGStar 法を提案する．そして，数値実験を通して，BiCGStar 法の収束性などの性能を評価する．

キーワード：積型反復法，GPBiCG 法，准残差，三項漸化式，BiCGStar 法

Improvement of convergence rate for GPBiCG method by means of stabilized associate residual

Ushio Yoshihiro[2]　Murakami Keiichi[3]　Fujino Seiji[4]

Abstract: We consider to solve a linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$ by iterative method. In product-type iterative methods, the residual vector $\boldsymbol{r}_k := \boldsymbol{b} - A\boldsymbol{x}$ is obtained by multiplying the Lanczos polynomial and acceleration polynomial. Though GPBiCG method is one of the popular product-type iterative methods, this method has some issues. There, we improve GPBiCG method based on the associate residual and three term recurrence. We propose BiCGStar method from above improvements. BiCGStar method is a new product-type iterative method. We demonstrate the performance of BiCGStar method on convergence through numerical experiments.

Keywords: product-type iterative method, GPBiCG method, associate residual, three-term recurrence, BiCGStar method

1. はじめに

非対称行列を係数に持つ連立一次方程式を反復法で解くことを考える．その場合に用いられる解法として，積型反復法を考える．積型反復法では，連立一次方程式の近似解は，残差の収束から求めることができる．このとき，残差は安定化多項式と Lanczos 多項式から得られるが，安定化多項式にどのような漸化式を採用するかによって，得られる解法は異なる．例えば，積型反復法で良く知られている GPBiCG 法 [10] では，加速多項式として $H_k(\lambda)$ と $G_k(\lambda)$ による漸化式を採用している．一方，BiCGSafe 法 [2] では，交代二項漸化式を採用している．これら採用する漸化式の違いにより，収束性に差が現れ，解法の特徴を示す一つの要因となっている．

ここで，新たに示す BiCGStar 法では，安定化多項式に三項漸化式を採用する．また，収束に必要な加速パラメータの決定を，一般的には残差の最小化から得るが，BiCGStar 法では，准残差の最小化から求めるようにアルゴリズムを導出した．BiCGStar 法は，GPBiCG 法の変形版と同様に，収束性の向上と，それに伴う計算時間の短縮が得られるのではないかと期待できる．

本論文では，まず BiCGStar 法の導出について示し，ア

1 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
2 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, School of Engineering, Kyushu University
3 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
4 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

ルゴリズムを示す．そこで，前回まではできなかった，近似解ベクトルの詳しい導出を行った．また，Lanczos 多項式のパラメータ β_k の導出の背景，目的を明確にした．次に数値実験を行い，GPBiCG 法と GPBiCG 法の変形版と比較し，BiCGStar 法の性能を評価する．数値実験では，行列の数を 10 種類から 20 種類に増やし，より広い分野の問題を取り扱った．

2. BiCGStar 法の概要

2.1 BiCGStar 法の導出

本論文では，安定化多項式により准残差を求めることによって得られる BiCGStar 法 (BiCG method using STabilized Associate Residual) を提案する．解くべき連立一次方程式を

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

とする．ここで，行列 A は大きさ $N \times N$ の非対称行列であり，$\boldsymbol{x}$, $\boldsymbol{b}$ は，それぞれ次元数 N の解ベクトルと右辺ベクトルである．また，式 (1) に対して，反復法の第 k 反復目の残差は $\boldsymbol{r}_k := \boldsymbol{b} - A\boldsymbol{x}_k$ で定義される．以下では，BiCGStar 法の導出を示す．安定化多項式 $H_{k+1}(\lambda)$ を以下の三項漸化式を満たすように設計する．

$$\left\{\begin{array}{lcl} H_0(\lambda) & = & 1, \quad H_1(\lambda) = (1-\zeta_0\lambda)H_0(\lambda), \\ H_{k+1}(\lambda) & = & (1+\eta_k-\zeta_k\lambda)H_k(\lambda) \\ & & \quad -\eta_k H_{k-1}(\lambda), \quad k = 1, 2, \ldots \end{array}\right. \tag{2}$$

ここで，λ は固有値に対応する．GPBiCG 法では，この多項式に補助多項式として $G_k(\lambda)$ を組み込んでいたが，今回提案する BiCGStar 法では使用しない．また，Lanczos 多項式 $R_k(\lambda)$ と補助多項式 $P_k(\lambda)$ は次の交代漸化式を満たす．

$$R_0(\lambda) = 1, \ P_0(\lambda) = 1,$$
$$R_k(\lambda) = R_{k-1}(\lambda) - \alpha_{k-1}\lambda P_{k-1}(\lambda), \tag{3}$$
$$P_k(\lambda) = R_k(\lambda) - \beta_{k-1}P_{k-1}(\lambda), \ k = 1, 2, \ldots. \tag{4}$$

式 (4)，(2) より，残差

$$\boldsymbol{r}_k = H_k(A)R_k(A)\boldsymbol{r}_0 \tag{5}$$

と補助ベクトル

$$\boldsymbol{p} := H_k(A)P_k(A)\boldsymbol{r}_0 \tag{6}$$

を更新し，$\boldsymbol{r}_{k+1}$，$\boldsymbol{p}_{k+1}$ を得る手順を導く．以下，簡単のため $R_k(\lambda)$，$H_k(\lambda)$ を各々 R_k，H_k と略記する．

まず，$H_{k+1}R_{k+1}$ の R_k を展開する事で次式が得られる．

$$H_{k+1}R_{k+1} = H_{k+1}R_k - \alpha_k\lambda H_{k+1}P_k, \tag{7}$$

$H_{k+1}R_k$ の H_{k+1} を展開すると次式が得られる．

$$H_{k+1}R_k = (1+\eta_k)H_kR_k - \zeta_k\lambda H_kR_k \tag{8}$$
$$\qquad -\eta_k H_{k-1}R_k. \tag{9}$$

となる．ここで，准残差ベクトルを

$$\boldsymbol{a\_r}_k := H_{k+1}(A)R_k(A)\boldsymbol{r}_0 \tag{10}$$

と定義する．$H_{k+1}P_k$ についても同様に展開すると，

$$H_{k+1}P_k = (1+\eta_k)H_kP_k - \zeta_k\lambda H_kP_k \tag{11}$$
$$\qquad -\eta_k H_{k-1}P_k \tag{12}$$

となる．式 (9)，(12) より，H_kR_{k+1}，H_kP_{k+1} を更新する必要がある．R_{k+1}，P_{k+1} を展開する事で，次の式が得られる．

$$H_kR_{k+1} = H_kR_k - \alpha_k\lambda H_kP_k, \tag{13}$$
$$H_kP_{k+1} = H_kR_{k+1} - \beta_kH_kP_k. \tag{14}$$

式 (7) - (14) より，H_kR_k，H_kP_k から $H_{k+1}R_{k+1}$ が得られることがわかる．また，$H_{k+1}P_{k+1}$ は次式で得られる．

$$H_{k+1}P_{k+1} = H_kR_{k+1} - \beta_kH_{k+1}P_k. \tag{15}$$

補助ベクトルを

$$\boldsymbol{w}_k := H_{k+1}(A)P_k(A)\boldsymbol{r}_0, \tag{16}$$
$$\boldsymbol{u}_k := H_k(A)R_{k+1}(A)\boldsymbol{r}_0, \tag{17}$$
$$\boldsymbol{q}_k := H_k(A)P_{k+1}(A)\boldsymbol{r}_0 \tag{18}$$

と定義すると，式 (7) - (15) より，以下のベクトルの更新式が得られる．

$$\boldsymbol{r}_{k+1} = \boldsymbol{a\_r}_k - \alpha_kA\boldsymbol{w}_k, \tag{19}$$
$$\boldsymbol{a\_r}_k = (1+\eta_k)\boldsymbol{r}_k - \zeta_kA\boldsymbol{r}_k - \eta_k\boldsymbol{u}_{k-1}, \tag{20}$$
$$\boldsymbol{w}_k = (1+\eta_k)\boldsymbol{p}_k - \zeta_kA\boldsymbol{p}_k - \eta_k\boldsymbol{q}_{k-1}, \tag{21}$$
$$\boldsymbol{u}_k = \boldsymbol{r}_k - \alpha_kA\boldsymbol{p}_k, \tag{22}$$
$$\boldsymbol{q}_k = \boldsymbol{u}_k - \beta_k\boldsymbol{p}_k, \tag{23}$$
$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} - \beta_k\boldsymbol{w}_k. \tag{24}$$

更新には $A\boldsymbol{r}_k$，$A\boldsymbol{p}_k$，$A\boldsymbol{w}_k$ の 3 回の行列ベクトル積を必要とするが，漸化式

$$A\boldsymbol{p}_{k+1} = A\boldsymbol{r}_{k+1} - \beta_kA\boldsymbol{w}_k \tag{25}$$

により $A\boldsymbol{p}_{k+1}$ を更新することで，行列ベクトル積 1 回が不要となる．

次に，近似解の更新式を求める．残差ベクトルの定義式 $\boldsymbol{r}_{k+1} := \boldsymbol{b} - A\boldsymbol{x}_{k+1}$ より，

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A^{-1}(\boldsymbol{b} - \boldsymbol{r}_{k+1}) \\ &= A^{-1}(\boldsymbol{b} - \boldsymbol{a\_r}_k + \alpha_kA\boldsymbol{w}_k) \\ &= A^{-1}(\boldsymbol{b} - \boldsymbol{a\_r}_k) + \alpha_k\boldsymbol{w}_k \end{aligned} \tag{26}$$

と求められる．ここで,

$$t_k = A^{-1}(b - a\_r_k) \tag{27}$$

とおく．この式 (27) の両辺に左から A を掛け, 整理すると,

$$\begin{aligned} b - At_k &= a\_r_k \\ &= (1+\eta_k)r_k - \zeta_k A r_k - \eta_k u_{k-1} \end{aligned} \tag{28}$$

となる．この式 (28) に二つの式

$$r_k = b - Ax_k, \tag{29}$$

$$u_{k-1} = r_{k-1} - \alpha_{k-1}Ap_{k-1} \tag{30}$$

を代入する．さらに，この代入で得られた r_{k-1} に

$$r_{k-1} = b - Ax_{k-1} \tag{31}$$

を代入し整理すると，次式が得られる．

$$t_k = (1+\eta_k)x_k + \zeta_k r_k \tag{32}$$

$$-\eta_k(x_{k-1} + \alpha_{k-1}p_{k-1}). \tag{33}$$

ここで,

$$v_{k-1} = x_{k-1} + \alpha_{k-1}p_{k-1} \tag{34}$$

とおくと,

$$t_k = (1+\eta_k)x_k + \zeta_k r_k - \eta_k v_{k-1} \tag{35}$$

となる．よって，x_{k+1} の更新式

$$x_{k+1} = t_k + \alpha_k w_k \tag{36}$$

が得られる．

次に，安定化多項式のパラメータは准残差のノルム $||a\_r_k||$ を局所的に最小化するように決定する．式 (20) より，簡略化のため，$z_k = u_{k-1} - r_k$ とおく．准残差のノルムは,

$$||a\_r_k|| = ||r_k - \zeta_k A r_k - \eta_k z_k|| \tag{37}$$

と表される．式 (37) より，$\phi = \frac{1}{2}||r_k - \zeta_k A r_k - \eta_k z_k||^2$ とすると，$\nabla\phi = 0$ より，次の正規方程式が得られる．

$$(Ar_k\ z_k)^T(Ar_k\ z_k)\begin{pmatrix}\zeta_k \\ \eta_k\end{pmatrix} = (Ar_k\ z_k)^T r_k. \tag{38}$$

この式より，ζ_k，η_k は，以下で与えられる．

$$\zeta_k = \frac{(z_k,z_k)(Ar_k,r_k) - (Ar_k,z_k)(z_k,r_k)}{(Ar_k,Ar_k)(z_k,z_k) - (Ar_k,z_k)(z_k,Az_k)}, \tag{39}$$

$$\eta_k = \frac{(Ar_k,Ar_k)(z_k,r_k) - (Ar_k,z_k)(Ar_k,r_k)}{(Ar_k,Ar_k)(z_k,z_k) - (Ar_k,z_k)(z_k,Az_k)}. \tag{40}$$

一方，α_k，β_k は，双直交条件より $H_k(A)R_{k+1}(A)r_0$ と $AH_k(A)P_{k+1}(A)r_0$ が初期シャドウ残差 r_0^* と直交するように決定する [1].

$$u_k = r_k - \alpha_k Ap_k \perp r_0^*, \tag{41}$$

$$Aq_k = Au_k - \beta_k Ap_k \perp r_0^* \tag{42}$$

より,

$$(r_0^*, r_k) - \alpha_k(r_0^*, Ap_k) = 0, \tag{43}$$

$$(r_0^*, Au_k) - \beta_k(r_0^*, Ap_k) = 0 \tag{44}$$

となる．したがって，α_k，β_k は次式で与えられる．

$$\alpha_k = \frac{(r_0^*, r_k)}{(r_0^*, Ap_k)}, \quad \beta_k = \frac{(r_0^*, Au_k)}{(r_0^*, Ap_k)}. \tag{45}$$

式 (45) による β_k の更新は行列ベクトル積 Au_k を必要とする．行列ベクトル積の計算を避けるため，β_k の計算方法として以下の 2 通りを考える．

(a) 文献 [10] に倣い,

$$\beta_k = -\frac{\alpha_k}{\zeta_k}\cdot\frac{(r_0^*, r_{k+1})}{(r_0^*, r_k)} \tag{46}$$

とする．(r_0^*, r_{k+1}) は更新後の r_{k+1} を用いる．

(b) 文献 [4] で提案された手法を用いる．

$$(r_0^*, Au_k) = (A^T r_0^*, u_k) \tag{47}$$

と変形することより，次式

$$\beta_k = \frac{(A^T r_0^*, u_k)}{(r_0^*, Ap_k)} \tag{48}$$

が得られる．式中の $A^T r_0^*$ は反復開始前に計算する．

どちらの方法でも β_k を計算することができる．そこで，今回の数値実験では，(b) の方法を用いる．以上の考察により，BiCGStar 法のアルゴリズムが得られる．

Algorithm 1: BiCGStar 法

1. Let x_0 be an initial guess, and put $r_0 = b - Ax_0$
2. Choose r_0^* such that $(r_0, r_0^*) \neq 0$
3. Compute $A^T r_0^*$
4. Set $\beta_{-1} = 0,\ q_{-1} = v_{-1} = w_{-1} = z_0 = \mathbf{0}$
5. for $k = 0, 1, \ldots,$ do :
6. begin
7. $p_k = r_k - \beta_{k-1}w_{k-1}$,
8. Compute Ar_k,
9. $Ap_k = Ar_k - \beta_{k-1}Aw_{k-1}$,
10. $\alpha_k = \dfrac{(r_0^*, r_k)}{(r_0^*, Ap_k)}$,
11. $\zeta_k = \dfrac{(Ar_k,r_k)(z_k,z_k) - (z_k,r_k)(Ar_k,z_k)}{(Ar_k,Ar_k)(z_k,z_k) - (z_k,Ar_k)(Ar_k,z_k)}$,
12. $\eta_k = \dfrac{(Ar_k,Ar_k)(z_k,r_k) - (z_k,Ar_k)(Ar_k,r_k)}{(Ar_k,Ar_k)(z_k,z_k) - (z_k,Ar_k)(Ar_k,z_k)}$,
13. (if $k = 0$, then $\zeta_k = \dfrac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$, $\eta_k = 0$),
14. $a\_r_k = r_k - \zeta_k Ar_k - \eta_k z_k$,
15. $t_k = (1+\eta_k)x_k + \zeta_k r_k - \eta_k v_{k-1}$,

$\boldsymbol{w}_k = (1+\eta_k)\boldsymbol{p}_k - \zeta_k A\boldsymbol{p}_k - \eta_k \boldsymbol{q}_{k-1}$,

Compute $A\boldsymbol{w}_k$,

$\boldsymbol{v}_k = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$,

$\boldsymbol{u}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k$,

$\boldsymbol{r}_{k+1} = \boldsymbol{a\_r}_k - \alpha_k A\boldsymbol{w}_k$,

$\boldsymbol{x}_{k+1} = \boldsymbol{t}_k + \alpha_k \boldsymbol{w}_k$,

if $\dfrac{||\boldsymbol{r}_{k+1}||_2}{||\boldsymbol{r}_0||_2} \leq \epsilon$ stop,

$\beta_k = \dfrac{(A^T\boldsymbol{r}_0^*, \boldsymbol{u}_k)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_k)}$,

$\boldsymbol{q}_k = \boldsymbol{u}_k - \beta_k \boldsymbol{p}_k$,

$\boldsymbol{z}_{k+1} = \boldsymbol{u}_k - \boldsymbol{r}_{k+1}$,

end do

3. 数値実験

3.1 計算機環境と計算条件

計算は倍精度浮動小数点演算で行った．計算機は Dell Power Edge(CPU: Intel Xeon X5570, クロック周波数: 2.93GHz, メモリ: 24Gbytes, OS: Red Hat Enterprise Linux 5.6，ホスト名：dell3) を用いた．プログラムは Fortran90 を用いて実装し，コンパイラは Intel Fortran Compiler ver. 11.1 を用いた．最適化オプションは“ -O3 ”を使用した．時間計測は C 言語の関数 gettimeofday を用いた．右辺項ベクトル $\boldsymbol{b}$ は厳密解を $\hat{\boldsymbol{x}} = (1, 1, ..., 1)^T$ とし，$\boldsymbol{b} = A\hat{\boldsymbol{x}}$ で作成した．収束判定値は相対残差の 2 ノルム：$||\boldsymbol{r}_{k+1}||_2/||\boldsymbol{r}_0||_2 \leq 10^{-10}$ とした．初期近似解 $\boldsymbol{x}_0$ はすべて 0，最大反復回数は 50000 回とした．行列は予め対角スケーリングによって対角項を 1 に正規化した．前処理は ILU(0) とし，加速係数 γ の値は 1.0 とした．初期シャドウ残差 $\boldsymbol{r}_0^*$ は初期残差 $\boldsymbol{r}_0$ と同じとした．今回の数値実験に使用する解法は，GPBiCG 法，GPBiCG_v1-_v2 法，BiCGStar 法の合計 4 つである．

3.2 テスト行列

表 1 にテスト行列の特徴を示す．行列は Florida 大学の疎行列データベース [7] より選出した．ただし，行列 air-cfl5 は Manchester 大学 F. Costen 専任講師から提供を受けた．また，テスト行列は全部で 20 個であり，全て非対称行列である．

3.3 実験結果

表 2 に各解法の実験結果を示す．ここで，“TRR” は“ TRR ”は初期近似解と第 k+1 反復目の近似解との真の相対残差 (True Relative Residual) の 2 ノルム $||\boldsymbol{b} - A\boldsymbol{x}_{k+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$ の常用対数の値を意味する．“比” は，各解法の GPBiCG 法に対する計算時間の比を表す．

表 1 テスト行列の特徴
Table 1 Specifications of test matrices.

行列	次元数	非零要素数	平均非零要素数	解析分野問題分野
atmosmodd	1,270,432	8,814,880	6.9	流体問題
air-cfl5	1,536,000	19,435,428	19.5	
poisson3Db	85,623	2,374,949	27.7	
water_tank	60,740	2,035,281	33.5	
raefsky3	21,200	1,488,768	70.2	
raefsky2	3,242	293,551	90.5	
bcircuit	68,902	375,558	5.5	回路問題
xenon2	157,464	3,866,688	24.5	要素問題
af23560	23,560	460,598	19.6	構造解析
sme3Da	12,504	874,887	70.0	
sme3Dc	42,930	3,148,656	73.3	
epb3	84,617	463,625	5.5	熱問題
epb1	14,734	95,053	6.5	
language	399,130	1,216,334	3.0	重み付きグラフ
big	13,209	91,465	6.9	
wang3	26,064	177,168	6.8	半導体問題
wang4	26,068	177,196	6.8	
ecl32	51,993	380,415	7.3	
torso3	259,156	4,429,042	17.0	2D/3D
k3plates	11,107	378,927	34.1	音響問題

表 2 より，以下の観察が得られる．

(1) 行列 k3plates において，BiCGStar 法の反復回数が GPBiCG 法より約 13%少なく，計算時間も約 14%短いという結果が得られた．

(2) 多く行列において，計算時間は，反復回数に比例した値が得られた．ただし，行列 epb3 などにおいては，反復回数と計算時間が比例しない結果も得られた．

(3) どの解法も, 多くの行列で収束が確認されている．ただし，行列によっては，TRR が 10^{-10} まで到達していない解法もある．例えば, 行列 big においては，GPBiCG 法，BiCGStar 法は TRR が 10^{-10} 以下であるのに対し，GPBiCG_v1,_v2 法では，TRR が 10^{-10} よりわずかに大きい．

次に，**図 1** に各解法の残差履歴を示す．(a)，(b)，(c) にそれぞれ行列 xenon2，sme3Dc，k3plates における各解法の残差履歴を示す．さらに，(d) に行列 k3plates における各解法の残差履歴の拡大図を示す．

図 1 より，次の観察が得られる．

(1) (a) より，行列 xenon2 では，BiCGStar 法が，同じ反復回数では，ほぼ常に他の 3 つの解法よりも残差が小さいという結果を示した．

(2) (b) より，行列 sme3Dc では，BiCGStar 法は GPBiCG_v2 と同様に，GPBiCG 法，GPBiCG_v2 法よ

表 2 各解法の実験結果
Table 2 The result of iterative methods.

(a) 行列 atmosmodd から sme3Da
(a)matrix atmosmodd ∼ sme3Da

行列	解法	反復回数	計算時間 [s]	比	TRR
atmosmodd	GPBiCG	308	25.271	1.00	-10.00
	GPBiCG_v1	296	25.300	1.00	-10.10
	GPBiCG_v2	**283**	24.363	0.96	-10.00
	BiCGStar	296	**22.137**	0.88	-10.01
air-cfl5	GPBiCG	27	4.001	1.00	-10.23
	GPBiCG_v1	27	4.191	1.05	-10.26
	GPBiCG_v2	27	4.189	1.05	-10.25
	BiCGStar	27	**3.830**	0.96	-10.32
poisson3Db	GPBiCG	286	4.797	1.00	-10.00
	GPBiCG_v1	297	5.047	1.05	-10.00
	GPBiCG_v2	288	4.878	1.02	-10.00
	BiCGStar	**284**	**4.696**	0.98	-10.04
water_tank	GPBiCG	2,034	18.835	1.00	-10.19
	GPBiCG_v1	2,062	19.457	1.03	-10.03
	GPBiCG_v2	**1,953**	**18.389**	0.98	-10.02
	BiCGStar	2,054	18.654	0.99	-10.02
raefsky3	GPBiCG	**4,380**	**25.816**	1.00	-10.02
	GPBiCG_v1	4,514	26.997	1.05	-10.00
	GPBiCG_v2	4,511	27.200	1.05	-10.00
	BiCGStar	4,408	26.017	1.01	-10.00
raefsky2	GPBiCG	**337**	**0.279**	1.00	-10.71
	GPBiCG_v1	360	0.302	1.08	-10.70
	GPBiCG_v2	339	0.285	1.02	-10.16
	BiCGStar	343	0.299	1.07	-10.05
bcircuit	GPBiCG	15,567	57.154	1.00	-10.02
	GPBiCG_v1	**12,477**	48.127	0.84	-9.60
	GPBiCG_v2	16,966	68.919	1.21	-9.86
	BiCGStar	12,855	**44.796**	0.78	-9.77
xenon2	GPBiCG	1,158	22.741	1.00	-10.00
	GPBiCG_v1	1,192	24.008	1.06	-10.02
	GPBiCG_v2	1,166	23.582	1.04	-10.00
	BiCGStar	**1,076**	**20.446**	0.90	-10.01
af23560	GPBiCG	2,309	4.446	1.00	-10.05
	GPBiCG_v1	2,288	4.561	1.03	-10.02
	GPBiCG_v2	**2,151**	4.307	0.97	-10.03
	BiCGStar	2,302	**4.303**	0.97	-10.04
sme3Da	GPBiCG	4,099	15.628	1.00	-11.30
	GPBiCG_v1	3,583	13.663	0.87	-9.53
	GPBiCG_v2	**2,859**	**10.930**	0.70	-9.86
	BiCGStar	3,322	12.552	0.80	-9.86

(b) 行列 sme3Dc から k3plates
(b)matrix sme3Dc ∼ k3plates

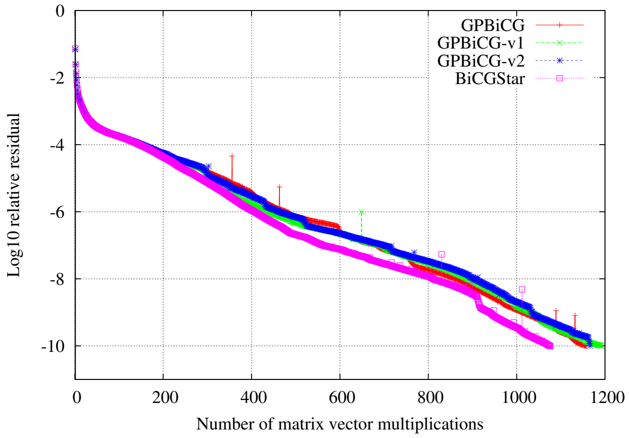
行列	解法	反復回数	計算時間 [s]	比	TRR
sme3Dc	GPBiCG	5,366	98.386	1.00	-9.99
	GPBiCG_v1	6,600	121.771	1.24	-9.34
	GPBiCG_v2	**4,742**	87.497	0.89	-9.18
	BiCGStar	4,770	**87.126**	0.89	-9.45
epb3	GPBiCG	3,447	13.981	1.00	-10.27
	GPBiCG_v1	3,157	13.119	0.94	-10.00
	GPBiCG_v2	**2,982**	12.356	0.88	-10.06
	BiCGStar	3,335	**11.943**	0.85	-10.00
epb1	GPBiCG	413	**0.231**	1.00	-10.11
	GPBiCG_v1	420	0.251	1.09	-10.01
	GPBiCG_v2	424	0.260	1.13	-10.03
	BiCGStar	**404**	**0.231**	1.00	-10.45
language	GPBiCG	27	**0.647**	1.00	-11.02
	GPBiCG_v1	27	0.697	1.08	-10.78
	GPBiCG_v2	27	0.697	1.08	-10.57
	BiCGStar	29	0.662	1.02	-10.16
big	GPBiCG	**1,745**	**1.010**	1.00	-10.03
	GPBiCG_v1	1,761	1.078	1.07	-9.66
	GPBiCG_v2	1,776	1.118	1.11	-9.67
	BiCGStar	1,791	1.030	1.02	-10.52
wang3	GPBiCG	**151**	**0.160**	1.00	-10.46
	GPBiCG_v1	154	0.170	1.06	-10.56
	GPBiCG_v2	156	0.176	1.10	-10.65
	BiCGStar	154	0.168	1.05	-10.15
wang4	GPBiCG	145	**0.152**	1.00	-10.02
	GPBiCG_v1	144	0.159	1.05	-10.02
	GPBiCG_v2	147	0.166	1.09	-10.06
	BiCGStar	**143**	0.157	1.03	-10.05
ecl32	GPBiCG	377	1.013	1.00	-10.00
	GPBiCG_v1	370	1.053	1.04	-10.09
	GPBiCG_v2	**347**	1.017	1.00	-10.01
	BiCGStar	373	**0.971**	0.96	-10.12
torso3	GPBiCG	75	1.942	1.00	-10.18
	GPBiCG_v1	73	1.953	1.01	-10.10
	GPBiCG_v2	**72**	1.933	1.00	-10.09
	BiCGStar	73	**1.832**	0.94	-10.15
k3plates	GPBiCG	4,892	6.323	1.00	-10.05
	GPBiCG_v1	4,774	6.265	0.99	-10.00
	GPBiCG_v2	4,417	5.761	0.91	-10.04
	BiCGStar	**4,282**	**5.452**	0.86	-10.04

りも停滞が短い．

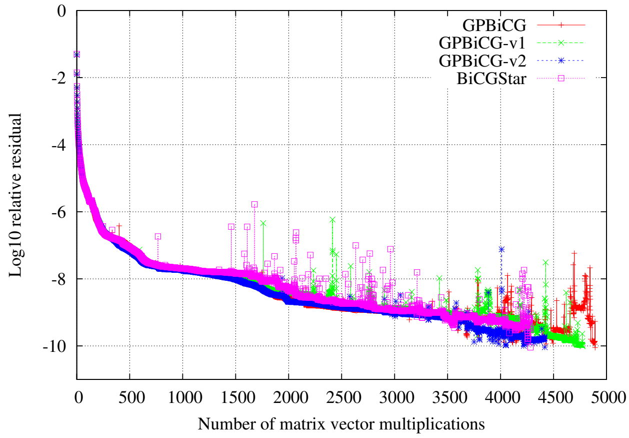
(3) (c)，(d) より，GPBiCG 法は，収束の直前に，残差履歴に振動が見られる．一方 BiCGStar 法はほぼ振動することなく収束した．

次に，**表 3** に各解法の計算時間による順位を示す．20 種類のテスト行列に対する，各解法の計算時間による順位付けを行った．

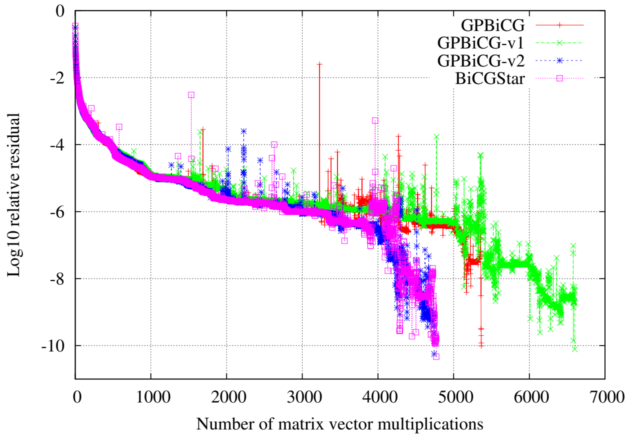
表 3 より，BiCGStar 法が合計 12 個の行列で最も速いと



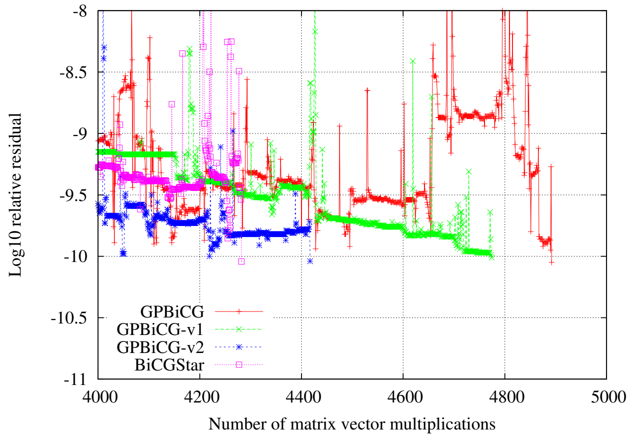
(a) 行列 xenon2
(a)matrix xenon2



(c) 行列 k3plates
(c)matrix k3plates



(b) 行列 sme3Dc
(b)matrix sme3Dc



(d) 行列 k3plates における各解法の残差履歴の拡大図
(d)enlarged figure of matrix bcircuit

図 1 各解法の残差履歴
Fig. 1 The history of the residual 2-norms of iterative methods.

表 3 各解法の計算時間による順位
Table 3 The ranking of computational time for iterative methods.

解法	順位				合計	総合
	1	2	3	4		順位
GPBiCG	7	5	5	3	44	2
GPBiCG_v1	0	1	9	10	69	4
GPBiCG_v2	2	6	6	6	56	3
BiCGStar	**12**	7	1	0	**29**	**1**
合計	21	19	21	19	-	-

いう結果が得られた．

4. まとめ

本論文では，准残差と三項漸化式に基づいた BiCGStar 法を提案した．さらに，数値実験を通して，BiCGStar 法の収束性の高さを示した．また，BiCGStar 法が，より多くの分野で有効性を持っていることが実証できた．

参考文献

[1] K. Abe, G.L.G. Sleijpen: Solving linear equations with a stabilized GPBiCG method, to be published in Appl. Numer. Math., (http://dx.doi.org/10.1016/j.apnum.2011.06.010).
[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，日本計算工学会論文集，No.20050028, 2005.
[3] Moethuthu, S. Fujino: Stability of GPBiCG AR method based on minimization of associate residual, J. ASCM, pp.108-120, 2008.
[4] K. Murakami, S. Fujino: A proposal of a product type iterative method using associate residual for parallel computer, International workshop on HPC (High performance computing), Krylov Subspace method and its applications pp.23-26, Beppu, Oita, 01/2013.
[5] H. Rutishauser: Theory of gradient method, Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Value Problems, Mitt. Inst. Angew. Math. ETH Zürich, Birkhäuser, Basel, pp.24-49, 1959.
[6] G.L.G. Sleijpen, P. Sonneveld, M. B. van Gijzen: Bi-CGSTAB as an induced dimension reduction method, Appl. Numer. Math., Vol.60, pp.1100-1114, 2010.
[7] University of Florida Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/index.html
[8] H.A. van der Vorst: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsym-

metric linear systems, SIAM J. Sci. Stat. Comput., Vol.13, pp.631-644, 1992.
[9] H.A. van der Vorst: Iterative Krylov preconditionings for large linear systems, Cambridge University Press, Cambridge, 2003.
[10] S.-L. Zhang: GPBi-CG: Generalized product-type based on Bi-CG for solving nonsymmetric linear systems, SIAM J. Sci. Comput., Vol.18, pp.537-551, 1997.
[11] S.-L. Zhang, M. Natori, C.-H. Jin: Product-type Krylov-Subspace Methods for Solving Nonsymmetric Linear Systems, Vol.989, pp.92-102, 1997.