

BiCGStar 法の提案と性能評価

藤野 清 次^{†1} 牛尾恵浩 (よしひろ)^{†2}

BiCGStar 法では、収束に必要な二つのパラメータの決定を、残差のノルム最小化からではなく、BiCGStar 法では准残差のノルム最小化から求める。また、積型多項式の展開順序についても残差多項式を最初に展開し、収束の安定化を図る。すなわち、BiCGStar 法は、GPBiCG 法とその変形版よりも、一層の収束性の向上と、計算時間の短縮を目指した解法である。本論文では、BiCGStar 法を導出し、アルゴリズムを示す。次に、数値実験を行い、GPBiCG 法と GPBiCG 法の変形版と比較し、提案する BiCGStar 法の収束性能が高いことを検証する。

A proposal and estimation of BiCGStar method

SEIJI FUJINO ^{†1} and YOSHIHIRO USHIO ^{†2}

We propose a new iterative method based on minimization of associate residual. We refer to as BiCG using STabilized Associate Residual (hereafter abbreviated as BiCGStar) method. BiCGStar method is an improvement of the conventional GPBiCG.v2 method with the usual residual. We demonstrate that BiCGStar method outperforms well as compared with the original GPBiCG, GPBiCG.v1 and GPBiCG.v2 methods in view of convergence rate through many numerical experiments.

1. はじめに

非対称行列を係数に持つ連立一次方程式を定常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、BiCG 法から発展した積型 BiCG 法の系統を考える。残差の収束から近似解を求めるが、残差は安定化多項式と残差多項式から得られるが、残差の更新のと

き、どちらの多項式を最初に展開するかにより、得られる解法は異なる。

GPBiCG 法の系統の解法では、安定化多項式を先に展開することで、残差の更新式を得る。一方、BiCGSafe 法や GPBiCG_AR 法では、残差多項式を先に展開している。この違いにより、収束性に差が現れ、解法の特徴を示すことになる。その他にも、どのような補助多項式を用いるかによっても解法は異なる。

ここで、新たに示す BiCGStar 法では、残差多項式を先に展開する。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。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。

本研究では、まず BiCGStar 法の導出について示し、アルゴリズムを示す。次に実験を行い、GPBiCG 法と GPBiCG 法の変形版と比較し、BiCGStar 法の性能を評価する。

2. BiCGStar 法の概要

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) について示す。解くべき連立一次方程式を

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

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、 x , b は、各々次元数 N の解ベクトルと右辺ベクトルである。

2.1 BiCGStar 法の導出

以下では、BiCGStar 法の導出を示す。加速多項式 $H_{k+1}(\lambda)$ を以下の三項漸化式を満たすように設計する。

$$\begin{cases} H_0(\lambda) = 1, & H_1(\lambda) = (1 - \zeta_0\lambda)H_0(\lambda), \\ H_{k+1}(\lambda) = (1 + \eta_k - \zeta_k\lambda)H_k(\lambda) - \eta_k H_{k-1}(\lambda), & k = 1, 2, \dots \end{cases} \quad (2)$$

ここで、 λ は固有値に対応する。また、残差多項式 $R_k(\lambda)$ と補助多項式 $P_k(\lambda)$ は次の交代漸化式を満たす。

$$\begin{aligned} R_0(\lambda) &= 1, \quad P_0(\lambda) = 1, \\ R_k(\lambda) &= R_{k-1}(\lambda) - \alpha_{k-1}\lambda P_{k-1}(\lambda), \end{aligned} \quad (3)$$

$$P_k(\lambda) = R_k(\lambda) - \beta_{k-1}P_{k-1}(\lambda), \quad k = 1, 2, \dots \quad (4)$$

式 (4), (2) より、残差 r_k と補助ベクトル $p := H_k(A)P_k(A)r_0$ を更新し、 r_{k+1} , 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。

^{†1} 九州大学情報基盤研究開発センター Research Institute for Information Technology, Kyushu University

^{†2} 九州大学工学部電気情報工学科 Department of Electrical Engineering and Computer Science, Kyushu University

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

$$H_{k+1}R_{k+1} = H_{k+1}R_k - \alpha_k \lambda H_{k+1}P_k, \quad (5)$$

$H_{k+1}R_k$ の H_{k+1} を展開すると

$$H_{k+1}R_k = (1 + \eta_k)H_kR_k - \zeta_k \lambda H_kR_k - \eta_k H_{k-1}R_k. \quad (6)$$

となる. 同様に, $H_{k+1}P_k$ の H_{k+1} を展開すると,

$$H_{k+1}P_k = (1 + \eta_k)H_kP_k - \zeta_k \lambda H_kP_k - \eta_k H_{k-1}P_k \quad (7)$$

となる. 式 (6), (7) 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} を展開することで,

$$H_kR_{k+1} = H_kR_k - \alpha_k \lambda H_kP_k, \quad (8)$$

$$H_kP_{k+1} = H_kR_{k+1} - \beta_k H_kP_k. \quad (9)$$

式 (5) - (9) 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.

$$H_{k+1}P_{k+1} = H_kR_{k+1} - \beta_k H_{k+1}P_k. \quad (10)$$

ここで, 三つの補助ベクトルを $w_k := H_{k+1}(A)P_k(A)r_0$, $p_k := H_k(A)R_{k+1}(A)r_0$, $u_k := H_k(A)P_{k+1}(A)r_0$ と定義する. このとき, 式 (5) - (9) より, 以下のベクトルの更新式が得られる.

$$a\text{-}r_k = (1 + \eta_k)r_k - \zeta_k Ar_k - \eta_k u_{k-1}, \quad (11)$$

$$w_k = (1 + \eta_k)p_k - \zeta_k Ap_k - \eta_k q_{k-1}, \quad (12)$$

$$u_k = r_k - \alpha_k Ap_k, \quad (13)$$

$$r_{k+1} = a\text{-}r_k - \alpha_k Aw_k, \quad (14)$$

$$q_k = u_k - \beta_k p_k, \quad (15)$$

$$p_{k+1} = r_{k+1} - \beta_k w_k. \quad (16)$$

更新には Ar_k , Ap_k , Aw_k の 3 回の行列ベクトル積を必要とするが, 漸化式

$$Ap_{k+1} = Ar_{k+1} - \beta_k Aw_k. \quad (17)$$

により Ap_{k+1} を更新することで, 行列ベクトル積の計算 1 回が不要となる.

次に, 残差に関する更新式 (11) と (12) そして (14) に対応する近似解の更新式を求める. 式 (11) に対応する近似解の更新式は, ベクトル t_k を $b - At_k = a\text{-}r_k$ を満たすベクトルとすると,

$$b - At_k = (1 + \eta_k)r_k - \zeta_k Ar_k - \eta_k u_{k-1}, \quad (18)$$

$$t_k = (1 + \eta_k)x_k + \zeta_k r_k - \eta_k (x_{k-1} + \alpha_{k-1}u_{k-1}). \quad (19)$$

と得られる. 同様に, 式 (12), (14) に対応する更新式は,

$$v_k = x_k + \alpha_k p_k, \quad (20)$$

$$x_{k+1} = t_k + \alpha_k w_k. \quad (21)$$

と表される.

ここで, 安定化多項式のパラメータは准残差のノルム $\|a\text{-}r_k\|$ を局所的に最小化するように決定する. 式 (11) より, $z_k = u_{k-1} - r_k$ とおくと, 准残差は

$$\|a\text{-}r_k\| = \|r_k - \zeta_k Ar_k - \eta_k z_k\|. \quad (22)$$

と表される. 式 (22) より, ζ_k , η_k は, 正規方程式

$$(Ar_k \ z_k)^T (Ar_k \ z_k) \begin{pmatrix} \zeta_k \\ \eta_k \end{pmatrix} = (Ar_k \ z_k)^T r_k. \quad (23)$$

を解くことにより, 以下で与えられる.

$$\zeta_k = \frac{(z_k, z_k)(Ar_k, r_k) - (Ar_k, z_k)(z_k, r_k)}{(Ar_k, Ar_k)(z_k, z_k) - (Ar_k, z_k)(z_k, Az_k)}, \quad (24)$$

$$\eta_k = \frac{(Ar_k, Ar_k)(z_k, r_k) - (Ar_k, z_k)(Ar_k, r_k)}{(Ar_k, Ar_k)(z_k, z_k) - (Ar_k, z_k)(z_k, Az_k)}. \quad (25)$$

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 $\tilde{r}_0$ と直交するように決定する.

$$u_k = r_k - \alpha_k Ap_k \perp r_0^*, \quad (26)$$

$$Aq_k = Au_k - \beta_k Ap_k \perp r_0^*, \quad (27)$$

より, $(\tilde{r}_0, r_k) - \alpha_k(\tilde{r}_0, Ap_k) = 0$, $(\tilde{r}_0, Au_k) - \beta_k(\tilde{r}_0, Ap_k) = 0$ となる. したがって, α_k , β_k は次式で与えられる.

$$\alpha_k = \frac{(r_0^*, r_k)}{(r_0^*, Ap_k)}, \quad \beta_k = \frac{(r_0^*, Au_k)}{(r_0^*, Ap_k)}. \quad (28)$$

式 (28) 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¹⁾ に倣い, $(r_0^*, Au_k) = (A^T r_0^*, u_k)$ より, $\beta_k = \frac{(A^T r_0^*, u_k)}{(r_0^*, Ap_k)}$ が得られる. $A^T r_0^*$ は反復開始前に計算する.

(b) 文献⁹⁾ に倣い, $\beta_k = -\frac{\alpha_k}{\zeta_k} \cdot \frac{(r_0^*, r_{k+1})}{(r_0^*, r_k)}$ とする. (r_0^*, r_{k+1}) は更新後の r_{k+1} を用いる.

ここでは, 上記の (a) の方法を用いる.

以上の結果から, BiCGStar 法のアルゴリズムが次のように得られる.

2.2 BiCGStar 法のアルゴリズム

Algorithm 1: BiCGStar 法

1. Let $\mathbf{x}_0$ be an initial guess, and put $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
2. Choose $\mathbf{r}_0^*$ such that $(\mathbf{r}_0, \mathbf{r}_0^*) \neq 0$
3. Compute $A\mathbf{r}_0$, $A^T \mathbf{r}_0^*$
4. Set $\beta_{-1} = 0$, $\mathbf{q}_{-1} = \mathbf{v}_{-1} = \mathbf{w}_{-1} = \mathbf{z}_0 = 0$
5. for $k = 0, 1, \dots$, do :
6. begin
7. $\mathbf{p}_k = \mathbf{r}_k - \beta_{k-1} \mathbf{w}_{k-1}$,
8. $A\mathbf{p}_k = A\mathbf{r}_k - \beta_{k-1} A\mathbf{w}_{k-1}$,
9. $\alpha_k = \frac{(\mathbf{r}_0^*, \mathbf{r}_k)}{(\mathbf{r}_0^*, A\mathbf{p}_k)}$,
10. $\zeta_k = \frac{(A\mathbf{r}_k, \mathbf{r}_k)(\mathbf{z}_k, \mathbf{z}_k) - (\mathbf{z}_k, \mathbf{r}_k)(A\mathbf{r}_k, \mathbf{z}_k)}{(A\mathbf{r}_k, A\mathbf{r}_k)(\mathbf{z}_k, \mathbf{z}_k) - (\mathbf{z}_k, A\mathbf{r}_k)(A\mathbf{r}_k, \mathbf{z}_k)}$,
11. $\eta_k = \frac{(A\mathbf{r}_k, A\mathbf{r}_k)(\mathbf{z}_k, \mathbf{r}_k) - (\mathbf{z}_k, A\mathbf{r}_k)(A\mathbf{r}_k, \mathbf{r}_k)}{(A\mathbf{r}_k, A\mathbf{r}_k)(\mathbf{z}_k, \mathbf{z}_k) - (\mathbf{z}_k, A\mathbf{r}_k)(A\mathbf{r}_k, \mathbf{z}_k)}$,
12. (if $k = 0$, then $\zeta_k = \frac{(A\mathbf{r}_k, \mathbf{r}_k)}{(A\mathbf{r}_k, A\mathbf{r}_k)}$, $\eta_k = 0$),
13. $\mathbf{a}\mathbf{r}_k = \mathbf{r}_k - \zeta_k A\mathbf{r}_k - \eta_k \mathbf{z}_k$,
14. $\mathbf{t}_k = (1 + \eta_k)\mathbf{x}_k + \zeta_k \mathbf{r}_k - \eta_k \mathbf{v}_{k-1}$,
15. $\mathbf{w}_k = (1 + \eta_k)\mathbf{p}_k - \zeta_k A\mathbf{p}_k - \eta_k \mathbf{q}_{k-1}$,
16. Compute $A\mathbf{w}_k$,
17. $\mathbf{v}_k = \mathbf{x}_k + \alpha_k \mathbf{p}_k$,
18. $\mathbf{u}_k = \mathbf{r}_k - \alpha_k A\mathbf{p}_k$,
19. $\mathbf{r}_{k+1} = \mathbf{a}\mathbf{r}_k - \alpha_k A\mathbf{w}_k$,
20. $\mathbf{x}_{k+1} = \mathbf{t}_k + \alpha_k \mathbf{w}_k$,
21. if $\frac{\|\mathbf{r}_{k+1}\|_2}{\|\mathbf{r}_0\|_2} \leq \epsilon$ stop,
22. Compute $A\mathbf{r}_{k+1}$,
23. $\beta_k = \frac{(A^T \mathbf{r}_0^*, \mathbf{u}_k)}{(\mathbf{r}_0^*, A\mathbf{p}_k)}$,
24. $\mathbf{q}_k = \mathbf{u}_k - \beta_k \mathbf{p}_k$,
25. $\mathbf{z}_{k+1} = \mathbf{u}_k - \mathbf{r}_{k+1}$,
26. end do

3. 数値実験

数値実験では、GPBiCG 法、GPBiCG_v1_v2 法、BiCGStar 法の合計四つの解法について実験を行った。

3.1 計算機環境と計算条件

計算機環境

- (1) 計算機: Dell Power Edge(CPU: Intel Xeon X5570, クロック周波数: 2.93GHz, メモリ: 24Gbytes, OS: RedHatEnterprise Linux 5.6, ホスト名: dell3)
- (2) プログラム言語: Fortran90, コンパイラ: Intel Fortran Compiler ver. 11.1
- (3) 最適化オプションは “-O3” を使用した。
- (4) 計算は倍精度浮動小数点演算で行い, 時間計測は C 言語の関数 gettimeofday を用いた。

計算条件

- (1) 右辺ベクトル $\mathbf{b}$ は厳密解を $\hat{\mathbf{x}} = (1, 1, \dots, 1)^T$ とし, $\mathbf{b} = A\hat{\mathbf{x}}$ で作成した。
- (2) 収束判定値は相対残差の 2 ノルム: $\|\mathbf{r}_{k+1}\|_2 / \|\mathbf{r}_0\|_2 \leq 10^{-12}$ とした。
- (3) 初期近似解 $\mathbf{x}_0$ はすべて 0, 最大反復回数は 50000 回とした。
- (4) 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。
- (5) 前処理は ILU(0) とし, 加速係数 γ の値は 1.0 とした。
- (6) 初期シャドウ残差 $\mathbf{r}_0^*$ は初期残差 $\mathbf{r}_0$ と同じとした。

3.2 テスト行列

表 1 にテスト行列の特徴を示す。行列はフロリダ大学の疎行列データベース⁶⁾より選出した。また、テスト行列は全部で 10 個、すべ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、“比 1” は原型の GPBiCG 法に対する計算時間の比を示し、“比 2” は原型の GPBiCG 法に対する 1 反復当たりの平均計算時間の比を示す。“比 3” は平均時間の比を表す。

表 2 の結果から、次の観察が得られた。

- (1) 行列 xenon2 において、BiCGStar 法が、反復回数、計算時間共に最も優れている。
- (2) 行列 bcircuit, sme3Da では、BiCGStar 法の反復回数が GPBiCG 法の約 80% と優れた値となった。
- (3) TRR は解法による差がほとんど見られない。

表 3 に各解法の計算時間による順位を示す。

表 1 テスト行列の特徴
Table 1 Specification 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	6.94	流体問題
poisson3Db	85,623	2,374,949	27.74	同上
water_tank	60,740	2,035,281	33.51	同上
bcircuit	68,902	375,558	5.45	回路問題
xenon2	157,464	3,866,688	24.5	要素問題
sme3Da	12,504	874,887	69.97	構造解析
sme3Dc	42,930	3,148,656	73.34	同上
big	13,209	91,465	6.92	重み付きグラフ
ec132	51,993	380,415	7.32	半導体問題
k3plates	11,107	378,927	34.12	音響問題

表 3 の結果から、次の観察が得られる。

- (1) BiCGStar 法が七つの行列に対して最も速い結果になった。
- (2) その他の解法では、GPBiCG_v2 法が優れていた。

図 1(a)-(d) に各解法の四つの行列に対する残差履歴を示す。

図 1 に示したグラフから、以下の観察が得られた。

- (1) 図 1(a) より、行列 xenon2 では、BiCGStar 法が最も速く収束した。
- (2) その他の行列でも、BiCGStar 法は平均的に速く収束した。

4. ま と め

本稿では、准残差と三項漸化式に基づいた BiCGStar 法を示した。准残差を使用することで、収束性が変化することが得られた。数値実験により、BiCGStar 法が GPBiCG 法、三項漸化式を使用する二つの変形版 GPBiCG 法と比較して、収束性が優れているという結論が得られた。

参 考 文 献

- 1) K. Abe, G.L.G. Sleijpen, Solving linear equations with a stabilized GPBiCG method, to be published in Applied Numerical Mathematics, (<http://dx.doi.org/10.1016/j.apnum.2011.06.010>).
- 2) S. Fujino, M. Fujiwara and M. Yoshida: BiCGSafe method based on minimization of associate residual, Transactions of JSCES, No.20050028, 2005. (in Japanese)

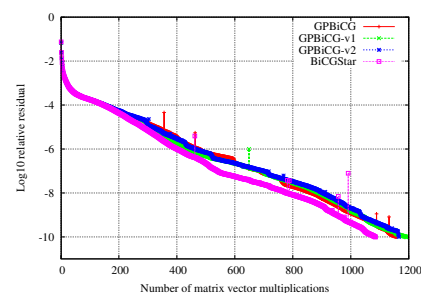
表 2 各反復解法の性能比較
Table 2 Comparison of performance of some iterative methods.

行列	解法	反復回数	比 1	合計計算時間 [s]	比 2	平均時間 [ms]	比 3	TRR
atmosmodd	GPBiCG	308	1.00	26.430	1.00	85.614	1.00	-10.00
	GPBiCG_v1	296	0.96	25.550	0.97	86.111	1.01	-10.10
	GPBiCG_v2	283	0.92	24.424	0.92	86.085	1.01	-10.00
	BiCGStar	296	0.96	22.408	0.85	75.493	0.88	-10.01
poisson3Db	GPBiCG	286	1.00	4.832	1.00	16.853	1.00	-10.00
	GPBiCG_v1	297	1.04	5.059	1.05	16.993	1.01	-10.00
	GPBiCG_v2	288	1.01	4.871	1.01	16.872	1.00	-10.00
	BiCGStar	284	0.99	4.658	0.96	16.359	0.97	-10.04
water_tank	GPBiCG	2,034	1.00	18.955	1.00	9.316	1.00	-10.19
	GPBiCG_v1	2,062	1.01	19.342	1.02	9.376	1.01	-10.03
	GPBiCG_v2	1,953	0.96	18.369	0.97	9.401	1.01	-10.02
	BiCGStar	2,054	1.01	18.507	0.98	9.006	0.97	-10.02
bcircuit	GPBiCG	15,567	1.00	57.084	1.00	3.667	1.00	-10.02
	GPBiCG_v1	12,477	0.80	47.796	0.84	3.830	1.04	-9.60
	GPBiCG_v2	16,966	1.09	63.879	1.12	3.765	1.03	-9.86
	BiCGStar	12,855	0.83	44.325	0.78	3.448	0.94	-9.77
xenon2	GPBiCG	1,158	1.00	23.018	1.00	19.863	1.00	-10.00
	GPBiCG_v1	1,192	1.03	24.010	1.04	20.129	1.01	-10.02
	GPBiCG_v2	1,166	1.01	23.680	1.03	20.295	1.02	-10.00
	BiCGStar	1,085	0.94	20.491	0.89	18.870	0.95	-10.00
sme3Da	GPBiCG	4,099	1.00	15.354	1.00	3.745	1.00	-11.30
	GPBiCG_v1	3,583	0.87	13.731	0.89	3.831	1.02	-9.53
	GPBiCG_v2	2,859	0.70	10.798	0.70	3.776	1.01	-9.86
	BiCGStar	3,322	0.81	12.272	0.80	3.693	0.99	-9.86
sme3Dc	GPBiCG	5,366	1.00	98.413	1.00	18.338	1.00	-9.99
	GPBiCG_v1	6,600	1.23	121.121	1.23	18.350	1.00	-9.34
	GPBiCG_v2	4,742	0.88	86.980	0.88	18.340	1.00	-9.18
	BiCGStar	4,770	0.89	86.315	0.88	18.093	0.99	-9.45
big	GPBiCG	1,745	1.00	1.033	1.00	0.591	1.00	-10.03
	GPBiCG_v1	1,761	1.01	1.081	1.05	0.614	1.04	-9.66
	GPBiCG_v2	1,776	1.02	1.084	1.05	0.610	1.03	-9.67
	BiCGStar	1,791	1.03	1.020	0.99	0.569	0.96	-10.52
ec132	GPBiCG	377	1.00	1.014	1.00	2.682	1.00	-10.00
	GPBiCG_v1	370	0.98	1.050	1.04	2.832	1.06	-10.09
	GPBiCG_v2	347	0.92	0.978	0.96	2.810	1.05	-10.01
	BiCGStar	373	0.99	0.939	0.93	2.512	0.94	-10.12
k3plates	GPBiCG	4,892	1.00	6.340	1.00	1.296	1.00	-10.05
	GPBiCG_v1	4,774	0.98	6.242	0.98	1.307	1.01	-10.00
	GPBiCG_v2	4,417	0.90	5.749	0.91	1.301	1.00	-10.04
	BiCGStar	4,282	0.88	5.398	0.85	1.260	0.97	-10.04

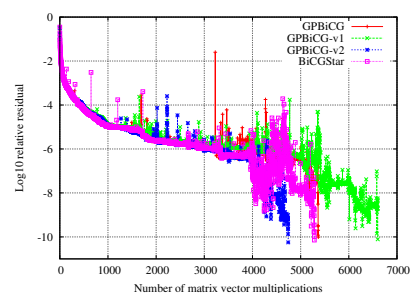
表 3 各解法の計算時間による順位

Table 3 Ranking of iterative methods in view of computational time

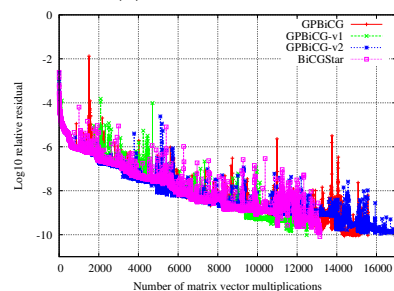
解法	順位				合計	合計 順位
	1	2	3	4		
GPBiCG	1	2	4	3	29	3
GPBiCG-v1	0	1	4	5	34	4
GPBiCG-v2	2	4	2	2	24	2
BiCGStar	7	3	0	0	13	1
合計	10	10	10	10	-	-



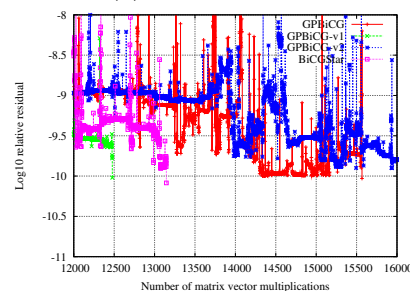
(a) 行列 : xenon2



(b) 行列 : sme3Dc



(c) 行列 : bcircuit



(d) 行列 : bcircuit の拡大図

図 1 各解法の三つの行列に対する残差履歴

Fig. 1 History of relative residual 2-norm of iterative methods for three matrices.

- 3) Moethuthu, S. Fujino: Stability of GPBiCG AR method based on minimization of associate residual, Journal of ASCM, pp.108-120, 2008.
- 4) 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.
- 5) G.L.G. Sleijpen, P. Sonneveld, M. B. van Gijzen: Bi-CGSTAB as an induced dimension reduction method: Applied Numerical Math., Vol.60, pp.1100-1114, 2010.
- 6) University of Florida Sparse Matrix Collection:
<http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- 7) H.A. van der Vorst: 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.
- 8) H.A. van der Vorst; Iterative Krylov preconditionings for large linear systems, Cambridge University Press, Cambridge, 2003.
- 9) S.-L. Zhang: 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.
- 10) S.-L. Zhang, M. Natori, C.-H. Jin: Product-type Krylov-Subspace Methods for Solving Nonsymmetric Linear Systems, Vol.989, 92-102, 1997.