

## BiCG 法による助走区間つき積型反復法の収束性評価

日下部雄三 阿部邦美† 尾上勇介 藤野清次‡  
(九州大学大学院システム情報科学府 † 岐阜聖徳学園大学 ‡ 九州大学情報基盤研究開発センター)

積型反復法では BiCG 法で生成されるシャドウ残差が収束性に大きな影響を与える。本研究では積型反復法の初期シャドウ残差に数回反復させた BiCG 法のシャドウ残差を与える助走区間つき積型反復法を提案し、数値実験によりその有用性を明らかにする。

### Evaluation of convergence of Product-type of Iterative method with Approach run using BiCG method

Yuzo Kusakabe Kuniyoshi Abe† Yusuke Onoue Seiji Fujino‡  
(Graduate School of Information Science and Electrical Engineering, Kyushu University  
†Gifu Shotoku Gakuen University ‡Research Institute for Information Technology, Kyushu University)

On product-type of iterative methods, shadow residual generated by BiCG method has large influence on the convergence. In this research, we suggest Product-type of Iterative method with Approach run which gives shadow residual of BiCG method iterated for some steps to initial shadow residual of product-type of iterative method, and show the validity through some numerical experiments.

## 1 はじめに

非対称行列を係数行列に持つ大規模な連立一次方程式の求解においてクリロフ部分空間法が有効であることが知られている。また、クリロフ部分空間法の中に積型反復法と呼ばれる反復法がある。積型反復法では BiCG (Bi-Conjugate Gradient) 法で生成されるシャドウ残差が収束性に大きな影響を持つことが知られている。しかし、BiCG 法では残差、シャドウ残差に対して更新が必要であり、計算コストが高い。また、CGS (CG Squared) 法、BiCGStab (BiCG Stabilized) 法等の積型反復法では加速パラメータを計算する内積の中に  $r_0^*$  の形でのみ存在する。このため、初期シャドウ残差の与え方が重要となる。

本稿では BiCG 法により生成されるシャドウ残差を生かし、積型反復法の収束性を向上させることを目的とする。数回反復させた BiCG 法のシャドウ残差を積型反復法の一つである BiCGStab 法の初期シャドウ残差に設定し、その収束性の向上を数値実験を通して実証する。

本稿の構成は次の通りである。第 2 節では従来の積型反復法である BiCG 法と BiCGStab 法についてその概要と算法を記述する。第 3 節では本稿で提案する助走区間つき積型反復法の概要と助走

区間つき BiCGStab 法の算法を記述する。第 4 節で数値実験を通して助走区間つき積型反復法の有用性を実証する。第 5 節でまとめと今後の課題について述べる。

## 2 従来の積型反復法

この節では従来の積型反復法である BiCG 法、BiCGStab 法について述べる。

### 2.1 BiCG 法

解くべき連立一次方程式を

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

とする。ここで、 $A$  は  $N \times N$  の実数非対称行列、 $x$  は次元数  $N$  の解ベクトル、 $b$  は次元数  $N$  の右辺ベクトルとする。初期近似解を  $x_0$  とし、初期残差  $b - Ax_0$  を  $r_0$  とする。このとき、 $r_0$  から形成されたクリロフ部分空間を以下のように表す。

$$K_n(A; r_0) = \text{span}\{r_0, Ar_0, \dots, A^{n-1}r_0\}. \quad (2)$$

クリロフ部分空間  $K_n(A; r_0)$  から以下に示す空間条件 **KS** (Krylov Subspace condition) を満たす

ように近似解  $\mathbf{x}_n$  を作り出す.

$$\mathbf{x}_n = \mathbf{x}_0 + \mathbf{z}_n, \quad \mathbf{z}_n \in K_n(A; \mathbf{r}_0). \quad (3)$$

このとき, 近似解  $\mathbf{x}_n$  に対応する残差  $\mathbf{r}_n$  は次のように表される.

$$\mathbf{r}_n := \mathbf{b} - A\mathbf{x}_n = \mathbf{r}_0 - A\mathbf{z}_n \in K_n(A; \mathbf{r}_0). \quad (4)$$

また, BiCG 法では残差ベクトル  $\mathbf{r}_n$ , シャドウ残差ベクトル  $\mathbf{r}_n^*$  は Lanczos 多項式  $R(\lambda)$  を用いて以下のように計算される.

$$\mathbf{r}_n = R_n(A)\mathbf{r}_0, \quad (5)$$

$$\mathbf{r}_n^* = R_n(A^T)\mathbf{r}_0^*. \quad (6)$$

ただし,  $A^T$  は係数行列  $A$  の転置行列である. 残差ベクトル  $\mathbf{r}_n$ , シャドウ残差ベクトル  $\mathbf{r}_n^*$  の更新に使用される加速パラメータ  $\alpha, \beta$  の計算には以下の双直交条件 Bi-OR を用いる.

$$\mathbf{r}_n \perp K_n(A^T; \mathbf{r}_0^*). \quad (7)$$

よって上記の式と条件により以下の BiCG 法の算法が得られる.

#### BiCG 法の算法

```

Let  $\mathbf{x}_0$  be an initial guess,
set  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ,
choose  $\mathbf{r}_0^*$  such that  $(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0$ ,
set  $\beta_{-1} = 0$ ,
for  $n = 0, 1, \dots$ , until  $\frac{\|\mathbf{r}_{n+1}\|_2}{\|\mathbf{r}_0\|_2} \leq \varepsilon$  do
begin
 $\mathbf{p}_n = \mathbf{r}_n + \beta_{n-1}\mathbf{p}_{n-1}$ 
 $\mathbf{p}_n^* = \mathbf{r}_n^* + \beta_{n-1}\mathbf{p}_{n-1}^*$ 
 $\alpha_n = \frac{(\mathbf{r}_n^*, \mathbf{r}_n)}{(\mathbf{p}_n^*, A\mathbf{p}_n)}$ 
 $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n\mathbf{p}_n$ 
 $\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n A\mathbf{p}_n$ 
 $\mathbf{r}_{n+1}^* = \mathbf{r}_n^* - \alpha_n A^T\mathbf{p}_n^*$ 
 $\beta_n = \frac{(\mathbf{r}_{n+1}^*, \mathbf{r}_{n+1})}{(\mathbf{r}_n^*, \mathbf{r}_n)}$ 
end.
```

## 2.2 BiCGStab 法

BiCG 法が式 (5) を用いて残差ベクトルの生成するのに対し, BiCGStab 法では以下の式で残差

ベクトルを生成する.

$$\mathbf{r}_n = Q_n(A)R_n(A)\mathbf{r}_0 \quad (8)$$

$$= Q_n(A)\mathbf{r}_n^{\text{BiCG}}. \quad (9)$$

上記の多項式  $Q_n(\lambda)$  は以下のように定義される.

$$Q_0(\lambda) = 1, \quad (10)$$

$$Q_{n+1}(\lambda) = (1 - \zeta_k \lambda)Q_n(\lambda), \quad k = 0, 1, \dots, (11)$$

BiCGStab 法では式 (11) のパラメータ  $\zeta_k$  は各反復で  $\|\mathbf{r}_{n+1}\|_2$  が最小になるように決定される.

よって上記の式と条件により以下の BiCGStab 法の算法が得られる.

#### BiCGStab 法の算法

```

Let  $\mathbf{x}_0$  be an initial guess,
set  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ ,
choose  $\mathbf{r}_0^*$  such that  $(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0$ ,
set  $\beta_{-1} = 0$ ,
for  $n = 0, 1, \dots$ , until  $\frac{\|\mathbf{r}_{n+1}\|_2}{\|\mathbf{r}_0\|_2} \leq \varepsilon$  do
begin
 $\mathbf{p}_n = \mathbf{r}_n + \beta_{n-1}(\mathbf{p}_{n-1} - \zeta_{n-1}A\mathbf{p}_{n-1})$ ,
 $\alpha_n = \frac{(\mathbf{r}_0^*, \mathbf{r}_n)}{(\mathbf{r}_0^*, A\mathbf{p}_n)}$ ,
 $\mathbf{t}_n = \mathbf{r}_n - \alpha_n A\mathbf{p}_n$ ,
 $\zeta_n = \frac{(A\mathbf{t}_n, \mathbf{t}_n)}{(A\mathbf{t}_n, A\mathbf{t}_n)}$ ,
 $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n\mathbf{p}_n + \zeta_n\mathbf{t}_n$ ,
 $\mathbf{r}_{n+1} = \mathbf{t}_n - \zeta_n A\mathbf{t}_n$ ,
 $\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(\mathbf{r}_0^*, \mathbf{r}_{n+1})}{(\mathbf{r}_0^*, \mathbf{r}_n)}$ ,
end.
```

## 3 助走区間つき積型反復法

この節では本稿で提案する助走区間つき積型反復法について述べる.

### 3.1 助走区間つき BiCGStab 法

BiCG 法では近似解ベクトル  $\mathbf{x}_k$  を求めるための加速パラメータ計算において残差ベクトル  $\mathbf{r}_n$ , シャドウ残差ベクトル  $\mathbf{r}_n^*$  が使用される. しかし,

$r_n^*$  は加速パラメータ  $\alpha, \beta$  の計算にのみ使用され、その更新には係数行列の転置行列  $A^T$  の行列・ベクトル積計算が必要であり計算コストが高い。

一方、BiCGStab 法では残差ベクトル  $r_n$  の生成式を式 (8) のように設定し、シャドウ残差ベクトル  $r_n^*$  の更新が不要になっている。しかし、加速パラメータ  $\alpha, \beta$  を計算する内積の中に初期シャドウ残差  $r_0^*$  が残っており、かつ初期シャドウ残差  $r_0^*$  が満たすべき条件は  $(r_0^*, r_0) \neq 0$  のみであるため、収束性を向上させる初期シャドウ残差  $r_0^*$  の設定が難しい。

そこで、BiCGStab 法では加速パラメータ  $\alpha, \beta$  の計算に BiCG 法と同様に双直交条件 Bi-OR を用いることと BiCG 法で生成されるシャドウ残差が収束性に大きな影響を与えるに着眼し、BiCGStab 法の初期シャドウ残差  $r_0^{*(\text{BiCGStab})}$  に数回反復させた BiCG 法のシャドウ残差  $r_m^{*(\text{BiCG})}$  を与える助走区間つき BiCGStab 法を提案する。すなわち、BiCGStab 法における初期近似解、初期残差、初期シャドウ残差を

$$x_0^{(\text{BiCGStab})} = x_m^{(\text{BiCG})}, \quad (12)$$

$$r_0^{(\text{BiCGStab})} = b - Ax_0^{(\text{BiCGStab})}, \quad (13)$$

$$r_0^{*(\text{BiCGStab})} = r_m^{*(\text{BiCG})} \quad (14)$$

と設定する。ただし、“ $m$ ” は助走に用いる BiCG 法の反復回数を意味する。この設定を用いるのは BiCG 法による助走により双直交条件 Bi-OR をある程度満たすシャドウ残差ベクトル  $r_n^*$  を得て BiCGStab 法における加速パラメータ  $\alpha, \beta$  の計算の精度が向上することを期待するためである。

以下に助走区間つき BiCGStab 法の算法を示す。

## 助走区間つき BiCGStab 法の算法

```

Let  $x_0^{(\text{BiCG})}$  be an initial guess,
and set  $r_0^{(\text{BiCG})} = b - Ax_0^{(\text{BiCG})}$ ,
choose  $r_0^{*(\text{BiCG})}$  such that  $(r_0^*, r_0) \neq 0$ ,
iterate BiCG for  $m$  steps to get  $x_m^{(\text{BiCG})}$ ,
set  $x_0^{(\text{BiCGStab})} = x_m^{(\text{BiCG})}$ 
set  $r_0^{(\text{BiCGStab})} = b - Ax_0^{(\text{BiCGStab})}$ ,
choose  $r_0^{*(\text{BiCGStab})}$ ,
such that  $(r_0^{*(\text{BiCGStab})}, r_0^{(\text{BiCGStab})}) \neq 0$ ,
set  $\beta_{-1} = 0$ ,
for  $n = 0, 1, \dots$ , until  $\frac{\|r_{n+1}\|_2}{\|r_0\|_2} \leq \varepsilon$  do :
begin
 $p_n = r_n + \beta_{n-1}(p_{n-1} - \zeta_{n-1}Ap_{n-1})$ ,
 $\alpha_n = \frac{(r_0^*, r_n)}{(r_0^*, Ap_n)}$ ,
 $t_n = r_n - \alpha_n Ap_n$ ,
 $\zeta_n = \frac{(At_n, t_n)}{(At_n, At_n)}$ ,
 $x_{n+1} = x_n + \alpha_n p_n + \zeta_n t_n$ ,
 $r_{n+1} = t_n - \zeta_n At_n$ ,
 $\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(r_0^*, r_{n+1})}{(r_0^*, r_n)}$ ,
end

```

## 4 数値実験

### 4.1 計算機環境

計算機環境は以下の通りである。

1. 計算はすべて倍精度浮動小数点演算で行った。
2. 計算機は HITACHI SR11000 モデル J1 (CPU: POWER5, クロック周波数: 1.9GHz, メモリ: 128Gbytes, OS: AIX 5.3, ホスト名: muromi) を使用した。
3. コンパイラは最適化 Fortran90 を使い、最適化オプションは -64 -Oss -nolimit -noscope -noparallel を使用した。
4. 時間の計測には日立のサービスサブルーチンの時間計測関数 xclock を使用した。

## 4.2 計算条件

計算条件は次の通りである。右辺項ベクトル $b$ は厳密解を $\hat{x} = (1, \dots, 1)^T$ とし、 $b = A\hat{x}$ として作成した。収束判定条件は相対残差の2ノルム： $\|r_{k+1}\|_2 / \|r_0\|_2 \leq 10^{-7}$ とした。初期近似解 $x_0$ はすべて0とした。行列は予め対角スケール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5000回とした。BiCG法、BiCGStab法では初期シャドウ残差には $r_0^* = r_0$ 、 $r_0^* = A^T r_0$ の2通りの与え方を用いた。助走区間つきBiCGStab法ではBiCG法における初期シャドウ残差には $r_0^* = r_0$ 、 $r_0^* = A^T r_0$ の2通り、BiCGStab法における初期シャドウ残差には $r_0^* = r_0$ 、 $r_0^* = A^T r_0$ の2通りの計4通りの与え方を用いた。助走区間つきBiCGStab法ではBiCG法の反復回数は1, 2, 5, 10回の4通りを調べた。

## 4.3 テスト問題

31個のテスト行列 [1] の特徴を表1に示す。

表1: テスト行列の特徴

行列	次元数	非零要素数	非零要素 / 行	解析分野
bridge	64,461	4,373,817	67.85	構造解析
olafu	16,416	1,015,156	61.84	
poisson3Da	13,514	352,762	26.10	
poisson3Db	85,623	2,374,949	27.74	熱交換器
xenon1	48,600	1,181,120	24.30	
xenon2	157,464	3,866,688	24.56	
sme3Da	12,504	874,887	69.97	流体力学
sme3Db	29,067	2,081,063	71.60	
sme3Dc	42,930	3,148,656	73.34	
ecl32	51,993	380,415	7.32	回路解析
epb1	14,734	95,053	6.45	
epb2	25,228	175,027	6.94	
epb3	84,617	463,625	5.48	電磁気学
ex10hs	2,548	57,308	22.49	
ex11	16,614	1,096,948	66.03	
fidap007	1,633	54,487	33.37	回路解析
fidap010	2,410	54,816	22.75	
fidap015	6,867	96,421	14.04	
fidapm37	9,152	765,944	83.69	電磁気学
li	22,695	1,215,181	53.54	
raefsky2	3,242	294,276	90.77	
raefsky3	21,200	1,488,768	70.22	電磁気学
raefsky4	19,779	1,328,611	67.17	
venkat01	62,424	1,717,792	27.52	
venkat50	62,424	1,717,777	27.52	回路解析
memplus	17,758	126,150	7.10	
wang3	26,064	177,168	6.80	
wang4	26,068	177,196	6.80	電磁気学
torso2	115,967	1,033,473	8.91	
torso3	259,156	4,429,042	17.09	
Zhao1	33,861	166,453	4.92	

## 4.4 実験結果

表2, 3に各行列における各解法の最短計算時間を各々示す。ただし、表2, 3中の時間の単位はすべて秒とした。表2, 3中の“Stab”, “助走 Stab”, “m”はBiCGStab法、助走区間つきBiCGStab法、助走区間のBiCGの反復回数を各々意味する。また、表2, 3中の“max”, “break”は最大反復回数で収束しなかったこと、計算中のオーバーフローにより強制終了したことを各々意味する。また、各解法が各行列において収束した場合の真の相対残差の値は収束判定条件である $10^{-7}$ 以下であった。

表2: 各行列における各解法の最短計算時間

行列	解法	$r_0^*$ [BiCG]	m	$r_0^*$ [Stab]	反復回数	計算時間
bridge	BiCG	$A^T r_0$	-	-	924	24.25
	Stab	-	-	$r_0$	807	20.49
	助走 Stab	$A^T r_0$	1	$r_0$	<b>700</b>	17.82
olafu	BiCG	$A^T r_0$	-	-	max	26.58
	Stab	-	-	$A^T r_0$	max	28.00
	助走 Stab	$A^T r_0$	10	$A^T r_0$	max	28.02
poisson3Da	BiCG	$A^T r_0$	-	-	134	0.46
	Stab	-	-	$r_0$	105	0.33
	助走 Stab	$A^T r_0$	2	$r_0$	<b>97</b>	0.31
poisson3Db	BiCG	$A^T r_0$	-	-	299	7.65
	Stab	-	-	$A^T r_0$	221	5.34
	助走 Stab	$A^T r_0$	5	$r_0$	<b>210</b>	4.98
xenon1	BiCG	$A^T r_0$	-	-	816	6.78
	Stab	-	-	$r_0$	562	4.59
	助走 Stab	$A^T r_0$	1	$r_0$	<b>524</b>	4.29
xenon2	BiCG	$A^T r_0$	-	-	949	28.21
	Stab	-	-	$r_0$	632	18.46
	助走 Stab	$A^T r_0$	1	$r_0$	<b>619</b>	17.96
sme3Da	BiCG	$r_0$	-	-	<b>1779</b>	11.69
	Stab	-	-	$r_0$	3357	21.89
	助走 Stab	$A^T r_0$	10	$r_0$	2230	14.50
sme3Db	BiCG	$r_0$	-	-	<b>2210</b>	39.74
	Stab	-	-	$r_0$	2795	48.15
	助走 Stab	$r_0$	5	$r_0$	3210	54.34
sme3Dc	BiCG	$r_0$	-	-	<b>3224</b>	94.14
	Stab	-	-	$r_0$	max	144.30
	助走 Stab	$A^T r_0$	5	$r_0$	max	143.36
venkat01	BiCG	$A^T r_0$	-	-	150	1.77
	Stab	-	-	$A^T r_0$	79	0.93
	助走 Stab	$r_0$	1	$A^T r_0$	<b>77</b>	0.90
venkat50	BiCG	$r_0$	-	-	max	58.98
	Stab	-	-	$r_0$	63	break
	助走 Stab	$A^T r_0$	10	$A^T r_0$	84	break
ecl32	BiCG	$A^T r_0$	-	-	454	2.20
	Stab	-	-	$r_0$	<b>309</b>	1.35
	助走 Stab	$A^T r_0$	1	$A^T r_0$	310	1.36
epb1	BiCG	$A^T r_0$	-	-	552	0.66
	Stab	-	-	$r_0$	<b>312</b>	0.33
	助走 Stab	$A^T r_0$	2	$r_0$	333	0.34
epb2	BiCG	$A^T r_0$	-	-	305	0.64
	Stab	-	-	$r_0$	137	0.26
	助走 Stab	$A^T r_0$	1	$A^T r_0$	<b>128</b>	0.24
epb3	BiCG	$A^T r_0$	-	-	2555	15.65
	Stab	-	-	$r_0$	<b>1609</b>	8.99
	助走 Stab	$r_0$	10	$r_0$	1647	8.57
ex10hs	BiCG	$A^T r_0$	-	-	<b>635</b>	0.27
	Stab	-	-	$r_0$	2322	0.93
	助走 Stab	$r_0$	2	$r_0$	989	0.40

表 3 : 各行列における各解法の最短計算時間

行列	解法	$r_0^*$		m	$r_0^*$		反復回数	計算時間
		[BiCG]	[Stab]		[Stab]	[Stab]		
ex11	BiCG	$A^T r_0$	-	-	-	-	1046	5.99
	Stab	-	-	-	$A^T r_0$	-	908	5.51
	助走 Stab	$A^T r_0$	5	$A^T r_0$	-	-	<b>854</b>	5.17
fidap007	BiCG	$A^T r_0$	-	-	-	-	max	1.71
	Stab	-	-	-	$r_0$	-	max	1.73
	助走 Stab	$A^T r_0$	2	$r_0$	-	-	max	1.73
fidap010	BiCG	$A^T r_0$	-	-	-	-	<b>1929</b>	0.76
	Stab	-	-	-	$r_0$	-	max	1.90
	助走 Stab	$A^T r_0$	2	$r_0$	-	-	max	1.90
fidap015	BiCG	$A^T r_0$	-	-	-	-	1	break
	Stab	-	-	-	$r_0$	-	1	break
	助走 Stab	$A^T r_0$	1	$r_0$	-	-	2	break
fidapm37	BiCG	$A^T r_0$	-	-	-	-	<b>4170</b>	15.76
	Stab	-	-	-	$r_0$	-	max	20.15
	助走 Stab	$r_0$	10	$r_0$	-	-	max	20.15
li	BiCG	$A^T r_0$	-	-	-	-	176	1.31
	Stab	-	-	-	$r_0$	-	223	1.72
	助走 Stab	$r_0$	2	$r_0$	-	-	<b>165</b>	1.13
raefsky2	BiCG	$r_0$	-	-	-	-	364	0.53
	Stab	-	-	-	$r_0$	-	311	0.48
	助走 Stab	$r_0$	1	$r_0$	-	-	<b>300</b>	0.46
raefsky3	BiCG	$A^T r_0$	-	-	-	-	max	36.75
	Stab	-	-	-	$r_0$	-	2469	19.70
	助走 Stab	$A^T r_0$	2	$r_0$	-	-	<b>2397</b>	19.17
raefsky4	BiCG	$A^T r_0$	-	-	-	-	max	34.03
	Stab	-	-	-	$r_0$	-	max	36.02
	助走 Stab	$A^T r_0$	10	$A^T r_0$	-	-	max	36.07
memplus	BiCG	$A^T r_0$	-	-	-	-	<b>114</b>	0.18
	Stab	-	-	-	$r_0$	-	170	0.27
	助走 Stab	$A^T r_0$	10	$r_0$	-	-	139	0.22
wang3	BiCG	$A^T r_0$	-	-	-	-	210	0.44
	Stab	-	-	-	$r_0$	-	<b>123</b>	0.24
	助走 Stab	$A^T r_0$	1	$A^T r_0$	-	-	128	0.24
wang4	BiCG	$A^T r_0$	-	-	-	-	196	0.41
	Stab	-	-	-	$r_0$	-	117	0.22
	助走 Stab	$A^T r_0$	2	$r_0$	-	-	<b>113</b>	0.22
torso2	BiCG	$A^T r_0$	-	-	-	-	30	0.39
	Stab	-	-	-	$r_0$	-	<b>18</b>	0.20
	助走 Stab	$A^T r_0$	1	$r_0$	-	-	19	0.22
torso3	BiCG	$A^T r_0$	-	-	-	-	<b>77</b>	2.98
	Stab	-	-	-	$A^T r_0$	-	82	3.02
	助走 Stab	$r_0$	2	$A^T r_0$	-	-	83	2.77
Zhao1	BiCG	$A^T r_0$	-	-	-	-	34	0.09
	Stab	-	-	-	$A^T r_0$	-	<b>30</b>	0.07
	助走 Stab	$A^T r_0$	2	$A^T r_0$	-	-	31	0.07

図 1 に行列 bridge, 図 2 に行列 xenon1, 図 3 に行列 sme3Da における各解法の相対残差履歴を示す。各行列に対する各解法のパラメータは表 2, 3 の値を用いた。

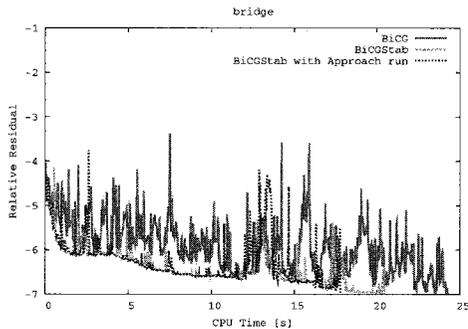


図 1 : 行列 bridge における各解法の相対残差履歴

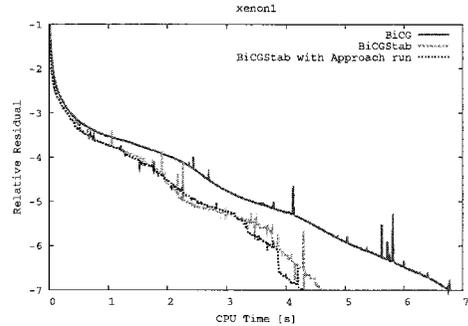


図 2 : 行列 xenon1 における各解法の相対残差履歴

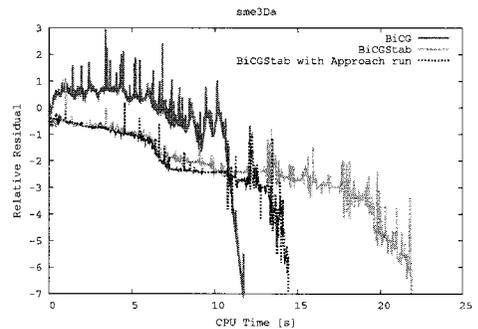


図 3 : 行列 sme3Da における各解法の相対残差履歴

表 4, 5 に各解法における初期シャドウ残差ごとの最速収束行列数を示す。表 6 に各解法における最速収束行列数を示す。

表 4 : BiCG 法, BiCGStab 法における初期シャドウ残差ごとの最速収束行列数

解法	$r_0^*$ [BiCG]		$r_0^*$ [Stab]		合計	
	$r_0$	$A^T r_0$	$r_0$	$A^T r_0$	収束数	未収束
BiCG	4	<b>21</b>	-	-	25	6
Stab	-	-	<b>18</b>	5	23	8

表 5 : 助走区間つき BiCGStab 法における初期シャドウ残差ごとの最速収束行列数

解法	$r_0^*$ [BiCG]		$r_0^*$ [Stab]		合計	
	$r_0$	$A^T r_0$	$r_0$	$A^T r_0$	収束数	未収束
助走 Stab	$r_0$	5	2	-	23	8
	$A^T r_0$	<b>11</b>	5	-		

表 6 : 各解法における最速収束行列数

解法	行列数
BiCG	8
Stab	6
助走 Stab	12
未収束	5

#### 4.5 考察

- BiCG 法においては 31 個の行列中 25 個の行列が収束し, 25 個の行列中 21 個の行列が初期シャドウ残差に  $A^T r_0$  を与えた場合の方が速く収束した. これは  $A^T$  を初期シャドウ残差ベクトルに掛けることにより双対クリロフ部分空間を張るベクトルのノルムの大きさが変化したためと考えられる [2][3].
- BiCGStab 法においては 31 個の行列中 23 個の行列が収束し, 23 個の行列中 18 個の行列が初期シャドウ残差に  $r_0$  を与えた場合の方が速く収束した. これは BiCGStab 法の初期シャドウ残差に  $A^T$  を掛けた場合, BiCGStab 法が BiCRStab 法 [4] と等価となり, 残差の振動が抑制されるが収束に時間がかかるようになったためと考えられる.
- 助走区間つき BiCGStab 法においては 31 個の行列中 23 個の行列が収束し, 23 個の行列中 11 個の行列が BiCG 法の初期シャドウ残差に  $A^T r_0$ , BiCGStab 法の初期シャドウ残差に  $r_0$  を与えた場合の方が速く収束した. これにより助走区間つき BiCGStab 法においても上記の単独の各解法と同様に各初期シャドウ残差ベクトルの設定は BiCG 法では  $A^T r_0$ , BiCGStab 法では  $r_0$  とした方がよいことがわかった.
- 31 個中 8 個の行列で BiCG 法が最も速く収束し, 31 個中 6 個の行列で BiCGStab 法が最も速く収束し, 31 個中 12 個の行列で助走

区間つき BiCGStab 法が最も速く収束した. これにより BiCG 法による助走を利用することが BiCGStab 法の収束性を向上させるために有効であることがわかった.

## 5 まとめと今後の課題

BiCG 法において生成されるシャドウ残差を積型反復法の初期シャドウ残差に利用する助走区間つき積型反復法を提案した. 数値実験により BiCG 法による助走区間つき BiCGStab 法は従来の BiCG 法, BiCGStab 法と比較して速く収束することがわかった. また, 収束頑強性は BiCGStab 法と同程度であることがわかった.

今後の課題として助走区間つき BiCGStab 法においては初期近似解, 初期シャドウ残差ベクトル, BiCG 法の反復回数との与え方をより詳しく調べる必要がある. また, 他の積型反復法に BiCG 法による助走区間をつけた場合の収束性の変化を調べる必要がある.

## 参考文献

- [1] Univ. of Florida Sparse Matrix Collection : <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- [2] 鬼束崇博, 藤野清次, 阿部邦美 :  $A^T$  に対する Richardson 反復を利用した Krylov 部分空間法の設計, 日本応用数理学会平成 20 年度研究部会連合発表会, 首都大学東京, 3 月, 2008.
- [3] 藤野清次, 阿部邦美, 鬼束崇博, 尾上勇介: BiCG 法と CGS 法の初期シャドウ残差  $r_0^*$  の与え方と収束特性, 第 37 回数値解析シンポジウム, 講演予稿集, 秋田県たざわこ芸術村, 6 月, 2008.
- [4] 阿部邦美, 曾我部知広, 藤野清次, 張紹良: 非対称行列用共役残差法に基づく積型反復解法, 情報処理学会論文誌, Vol. 41, No. 6, 2007, pp. 1234-1241.