

Bi-CR 法の積型解法について

東京大学大学院工学系研究科 曾我部知広 (Tomohiro Sogabe)

School of Engineering, The University of Tokyo

東京大学大学院工学系研究科 張紹良 (Shao-Liang Zhang)

School of Engineering, The University of Tokyo

1 はじめに

A を $N \times N$ の非対称行列, b を N 次元ベクトルとした大規模な非対称線形方程式

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

を解く数値解法としてクリロフ部分空間法がある. クリロフ部分空間法の代表として共役勾配法 (CG 法 [5]) が挙げられるが, これは正定値対称行列用の解法であり上記のような問題には適していない. 線形方程式 (1) を解くために CG 法を非対称行列用に拡張した Bi-CG 法 [4] が Fletcher により提案されて以降, Bi-CG 法の収束性の加速・安定化を図る手法が数多く提案され, その中で最も成功した改良法の一つに積型と呼ばれる解法 (CGS 法 [10], Bi-CGSTAB 法 [11], GPBi-CG 法 [12]) がある.

一方, CG 法の他に対称行列を係数行列とする線形方程式を解くためのクリロフ部分空間法として, [6] に記述されている共役残差法 (CR 法) がある. 最近, この CR 法を非対称行列用に拡張した Bi-CR 法 [9] が提案され, また Bi-CG 法よりも良い収束の振る舞いを示すことが数値実験で確認されている. そこで本稿では, Bi-CR 法に対して Bi-CG 法と同様に積型解法を導入し, Bi-CR 法の収束性の更なる加速・安定化を試みる.

本稿の構成として, 第 2 節で Bi-CR 法のアルゴリズムとその導出原理について述べる. 第 3 節では, Bi-CR 法の積型解法の枠組みを構築し, Bi-CG 法の積型解法に対応する解法を導出する. 第 4 節で Bi-CG 法と Bi-CR 法のそれぞれの積型解法の性能を数値実験で比較・評価し, 最後に第 5 節でまとめを行なう.

2 Bi-CR 法

非対称線形方程式 (1) を解くための Bi-CR 法のアルゴリズムを以下に示す.

Algorithm 1: The Bi-CR method

x_0^{BCR} is an initial guess, $r_0^{\text{BCR}} = b - Ax_0^{\text{BCR}}$, set $\beta_{-1} = 0$,
 $r_0^{\text{BCR}*}$ is an arbitrary vector, such that $(A^T r_0^{\text{BCR}*}, r_0^{\text{BCR}}) \neq 0$,
 for $n = 0, 1, \dots$ until $\|r_n^{\text{BCR}}\| \leq \epsilon \|b\|$ do:

begin

$$\mathbf{p}_n^{\text{BCR}} = \mathbf{r}_n^{\text{BCR}} + \beta_{n-1} \mathbf{p}_{n-1}^{\text{BCR}}, \quad \mathbf{p}_n^{\text{BCR}*} = \mathbf{r}_n^{\text{BCR}*} + \beta_{n-1} \mathbf{p}_{n-1}^{\text{BCR}*},$$

$$(A\mathbf{p}_n^{\text{BCR}} = A\mathbf{r}_n^{\text{BCR}} + \beta_{n-1} A\mathbf{p}_{n-1}^{\text{BCR}}),$$

$$\alpha_n = \frac{(\mathbf{r}_n^{\text{BCR}*}, A\mathbf{r}_n^{\text{BCR}})}{(A^T \mathbf{p}_n^{\text{BCR}*}, A\mathbf{p}_n^{\text{BCR}})},$$

$$\mathbf{x}_{n+1}^{\text{BCR}} = \mathbf{x}_n^{\text{BCR}} + \alpha_n \mathbf{p}_n^{\text{BCR}},$$

$$\mathbf{r}_{n+1}^{\text{BCR}} = \mathbf{r}_n^{\text{BCR}} - \alpha_n A\mathbf{p}_n^{\text{BCR}}, \quad \mathbf{r}_{n+1}^{\text{BCR}*} = \mathbf{r}_n^{\text{BCR}*} - \alpha_n A^T \mathbf{p}_n^{\text{BCR}*},$$

$$\beta_n = \frac{(\mathbf{r}_{n+1}^{\text{BCR}*}, A\mathbf{r}_{n+1}^{\text{BCR}})}{(\mathbf{r}_n^{\text{BCR}*}, A\mathbf{r}_n^{\text{BCR}})},$$

end

Bi-CR 法では, その n 回目の残差ベクトル $\mathbf{r}_n^{\text{BCR}}$ と探索方向ベクトル $\mathbf{p}_n^{\text{BCR}}$, そして補助ベクトル $\mathbf{r}_n^{\text{BCR}*}$ と $\mathbf{p}_n^{\text{BCR}*}$ がランチョス多項式 R_n [7] と補助多項式 P_n を用いて

$$\mathbf{r}_n^{\text{BCR}} = R_n(A)\mathbf{r}_0^{\text{BCR}}, \quad \mathbf{p}_n^{\text{BCR}} = P_n(A)\mathbf{r}_0^{\text{BCR}}, \quad (2)$$

$$\mathbf{r}_n^{\text{BCR}*} = R_n(A^T)\mathbf{r}_0^{\text{BCR}*}, \quad \mathbf{p}_n^{\text{BCR}*} = P_n(A^T)\mathbf{r}_0^{\text{BCR}*} \quad (3)$$

のように表される. ここで, R_n と P_n は次の交代漸化式を満たす.

$$R_0(\lambda) = 1, \quad P_0(\lambda) = 1, \quad (4)$$

$$R_n(\lambda) = R_{n-1}(\lambda) - \alpha_{n-1} \lambda P_{n-1}(\lambda), \quad (5)$$

$$P_n(\lambda) = R_n(\lambda) + \beta_{n-1} P_{n-1}(\lambda), \quad \text{for } n = 1, 2, \dots \quad (6)$$

式 (5)-(6) のパラメータ $\alpha_{n-1}, \beta_{n-1}$ は, 式 (2) の残差ベクトル $\mathbf{r}_n^{\text{BCR}}$ と探索方向ベクトル $\mathbf{p}_n^{\text{BCR}}$ に対して次の直交条件を課すことにより決められる [9].

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

以上が, Bi-CR 法の理論的な枠組みである.

次に, Bi-CR 法の積型解法を次節で導入するために α_n と β_n の計算式の変形を行なう. 式 (3) の補助ベクトル $\mathbf{r}_n^{\text{BCR}*}$ と $\mathbf{p}_n^{\text{BCR}*}$ は以下のように展開することができる.

$$\mathbf{r}_n^{\text{BCR}*} = R_n(A^T)\mathbf{r}_0^{\text{BCR}*} = \left((-1)^n \prod_{i=0}^{n-1} \alpha_i \right) (A^T)^n \mathbf{r}_0^{\text{BCR}*} + \mathbf{z}_1,$$

$$\mathbf{p}_n^{\text{BCR}*} = P_n(A^T)\mathbf{r}_0^{\text{BCR}*} = \left((-1)^n \prod_{i=0}^{n-1} \alpha_i \right) (A^T)^n \mathbf{r}_0^{\text{BCR}*} + \mathbf{z}_2.$$

ここで $\mathbf{z}_1, \mathbf{z}_2 \in K_n(A^T; \mathbf{r}_0^{\text{BCR}*})$ である. したがって, 直交条件 (7) より Algorithm 1 の α_n と β_n は次のように書換えることができる.

$$\alpha_n = \frac{(\mathbf{r}_n^{\text{BCR}*}, A\mathbf{r}_n^{\text{BCR}})}{(A^T \mathbf{p}_n^{\text{BCR}*}, A\mathbf{p}_n^{\text{BCR}})} = \frac{((A^T)^{n+1} \mathbf{r}_0^{\text{BCR}*}, \mathbf{r}_n^{\text{BCR}})}{((A^T)^{n+1} \mathbf{r}_0^{\text{BCR}*}, A\mathbf{p}_n^{\text{BCR}})}, \quad (8)$$

$$\beta_n = \frac{(\mathbf{r}_{n+1}^{\text{BCR}*}, A\mathbf{r}_{n+1}^{\text{BCR}})}{(\mathbf{r}_n^{\text{BCR}*}, A\mathbf{r}_n^{\text{BCR}})} = -\alpha_n \frac{((A^T)^{n+2} \mathbf{r}_0^{\text{BCR}*}, \mathbf{r}_{n+1}^{\text{BCR}})}{((A^T)^{n+1} \mathbf{r}_0^{\text{BCR}*}, \mathbf{r}_n^{\text{BCR}})}. \quad (9)$$

3 Bi-CR 法の積型解法

本節では, Bi-CG 法の積型解法 (CGS 法, Bi-CGSTAB 法, GPBi-CG 法) に慣らい, Bi-CR 法の積型解法の枠組みを構築し, アルゴリズムを導出する.

3.1 定義

Bi-CR 法の積型解法を設計するにあたり, 新しい残差ベクトル r_n を Bi-CR 法の残差ベクトル r_n^{BCR} と, 最高次項の係数がゼロでない n 次多項式 H_n との積で定義する.

$$r_n = H_n(A)r_n^{\text{BCR}} = b - Ax_n. \quad (10)$$

ここで, $H_n(\lambda)$ は次の交代漸化式を満たすように設計する.

$$H_0(\lambda) = 1, \quad G_0(\lambda) = \zeta_0, \quad (11)$$

$$H_n(\lambda) = H_{n-1}(\lambda) - \lambda G_{n-1}(\lambda), \quad (12)$$

$$G_n(\lambda) = \zeta_n H_n(\lambda) + \eta_n G_{n-1}(\lambda), \quad n = 1, 2, \dots \quad (13)$$

ここで, 特に $\zeta_n = \alpha_n$, $\eta_n = \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n$ とおくと H_n がランチョス多項式 R_n に帰着する.

3.2 残差多項式の計算

交代漸化式 (4)-(6) および (11)-(13) から, 多項式族 $H_n R_n$, $H_n R_{n+1}$, $\lambda G_{n-1} R_{n+1}$, $H_n P_n$, $\lambda H_n P_{n+1}$, $\lambda G_n P_n$, $G_n R_{n+1}$ に関する漸化式は次のように作ることができる.

$$H_{n+1} R_{n+1} = H_n R_{n+1} - \eta_n \lambda G_{n-1} R_{n+1} - \zeta_n \lambda H_n R_{n+1} \quad (14)$$

$$= H_n R_n - \alpha_n \lambda H_n P_n - \lambda G_n R_{n+1}, \quad (15)$$

$$H_n R_{n+1} = H_n R_n - \alpha_n \lambda H_n P_n, \quad (16)$$

$$\begin{aligned} \lambda G_n R_{n+2} &= H_n R_{n+1} - H_{n+1} R_{n+1} \\ &\quad - \alpha_{n+1} \lambda H_n P_{n+1} + \alpha_{n+1} \lambda H_{n+1} P_{n+1}, \end{aligned} \quad (17)$$

$$H_{n+1} P_{n+1} = H_{n+1} R_{n+1} + \beta_n H_n P_n - \beta_n \lambda G_n P_n, \quad (18)$$

$$\lambda H_n P_{n+1} = \lambda H_n R_{n+1} + \beta_n \lambda H_n P_n, \quad (19)$$

$$\begin{aligned} \lambda G_n P_n &= \zeta_n \lambda H_n P_n \\ &\quad + \eta_n (H_{n-1} R_n - H_n R_n + \beta_{n-1} \lambda G_{n-1} P_{n-1}), \end{aligned} \quad (20)$$

$$G_n R_{n+1} = \zeta_n H_n R_n + \eta_n G_{n-1} R_n - \alpha_n \lambda G_n P_n. \quad (21)$$

次に, 6 つの補助ベクトル

$$\begin{aligned} t_n &= H_n(A)r_{n+1}^{\text{BCR}}, & y_n &= AG_{n-1}(A)r_{n+1}^{\text{BCR}}, \\ p_n &= H_n(A)p_n^{\text{BCR}}, & w_n &= AH_n(A)p_{n+1}^{\text{BCR}}, \\ u_n &= AG_n(A)p_n^{\text{BCR}}, & z_n &= G_n(A)r_{n+1}^{\text{BCR}} \end{aligned}$$

を導入すると, 式 (10) 及び式 (14)-(21) から Bi-CR 法の積型解法の残差 r_{n+1} は次の漸化式で計算することができる.

$$r_{n+1} = t_n - \eta_n y_n - \zeta_n A t_n \quad (22)$$

$$= r_n - \alpha_n A p_n - A z_n, \quad (23)$$

$$t_n = r_n - \alpha_n A p_n, \quad (24)$$

$$y_{n+1} = t_n - r_{n+1} - \alpha_{n+1} w_n + \alpha_{n+1} A p_{n+1}, \quad (25)$$

$$p_{n+1} = r_{n+1} + \beta_n (p_n - u_n), \quad (26)$$

$$w_n = A t_n + \beta_n A p_n, \quad (27)$$

$$u_n = \zeta_n A p_n + \eta_n (t_{n-1} - r_n + \beta_{n-1} u_{n-1}), \quad (28)$$

$$z_n = \zeta_n r_n + \eta_n z_{n-1} - \alpha_n u_n. \quad (29)$$

近似解 x_{n+1} の計算は, $r_{n+1} = b - A x_{n+1}$ の関係と式 (23) より

$$x_{n+1} = x_n + \alpha_n p_n + z_n \quad (30)$$

となる.

3.3 α_n と β_n の計算式

多項式 H_n の最高次項の係数は $(-1)^n \prod_{i=0}^{n-1} \zeta_i$ であるので双直交条件 (7) より次の式が成立する.

$$(A^T r_0^{\text{BCR}*}, r_n) = \left((-1)^n \prod_{i=0}^{n-1} \zeta_i \right) ((A^T)^{n+1} r_0^{\text{BCR}*}, r_n^{\text{BCR}}), \quad (31)$$

$$(A^T r_0^{\text{BCR}*}, A p_n) = \left((-1)^n \prod_{i=0}^{n-1} \zeta_i \right) ((A^T)^{n+1} r_0^{\text{BCR}*}, A p_n^{\text{BCR}}). \quad (32)$$

したがって, α_n と β_n は式 (8)-(9) より二つの内積 (31)-(32) を用いて以下のように計算することができる.

$$\alpha_n = \frac{(A^T r_0^{\text{BCR}*}, r_n)}{(A^T r_0^{\text{BCR}*}, A p_n)}, \quad \beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(A^T r_0^{\text{BCR}*}, r_{n+1})}{(A^T r_0^{\text{BCR}*}, r_n)}. \quad (33)$$

3.4 Bi-CR 法の積型解法

漸化式 (22)-(29) および近似解の計算式 (30), そして α_n と β_n の計算式 (33) より以下の Bi-CR 法の積型解法のアルゴリズムが得られる.

Algorithm 2: Product-type methods based on Bi-CR

$\mathbf{x}_0$ is an initial guess, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,
 set $\mathbf{t}_{-1} = \mathbf{w}_{-1} = \mathbf{0}$, $\beta_{-1} = 0$,
 for $n = 0, 1, \dots$ until $\|\mathbf{r}_n\| \leq \epsilon\|\mathbf{b}\|$ do:
 begin
 $\mathbf{p}_n = \mathbf{r}_n + \beta_{n-1}(\mathbf{p}_{n-1} - \mathbf{u}_{n-1})$,
 $\alpha_n = \frac{(A^T \mathbf{r}_0, \mathbf{r}_n)}{(A^T \mathbf{r}_0, A\mathbf{p}_n)}$,
 $\mathbf{y}_n = \mathbf{t}_{n-1} - \mathbf{r}_n - \alpha_n \mathbf{w}_{n-1} + \alpha_n A\mathbf{p}_n$,
 $\mathbf{t}_n = \mathbf{r}_n - \alpha_n A\mathbf{p}_n$,
 compute ζ_n, η_n ,
 $\mathbf{u}_n = \zeta_n A\mathbf{p}_n + \eta_n(\mathbf{t}_{n-1} - \mathbf{r}_n + \beta_{n-1} \mathbf{u}_{n-1})$,
 $\mathbf{z}_n = \zeta_n \mathbf{r}_n + \eta_n \mathbf{z}_{n-1} - \alpha_n \mathbf{u}_n$,
 $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n + \mathbf{z}_n$,
 $\mathbf{r}_{n+1} = \mathbf{t}_n - \eta_n \mathbf{y}_n - \zeta_n A\mathbf{t}_n$,
 $\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(A^T \mathbf{r}_0, \mathbf{r}_{n+1})}{(A^T \mathbf{r}_0, \mathbf{r}_n)}$,
 $\mathbf{w}_n = A\mathbf{t}_n + \beta_n A\mathbf{p}_n$,
 end

Algorithm 2 より Bi-CR 法の積型解法では、反復を行なう前の初期ベクトルの設定時に転置行列と初期残差ベクトルの積が必要となる。Bi-CR 法の積型解法に現れるパラメータ ζ_n, η_n の決め方を次の小節で述べる。

3.5 ζ_n, η_n の決め方

パラメータ ζ_n, η_n の決め方により様々な Bi-CR 法の積型解法が得られる。本小節では、積型解法の導出過程から分かるように、Bi-CG 法の積型解法におけるパラメータの決め方を Bi-CR 法の積型解法にそのまま利用することができるため、代表的な Bi-CG 法の積型解法 (CGS 法, Bi-CGSTAB 法, GPBi-CG 法) に対応する三つの解法を紹介する。

- CRS 法 : $H_n = R_n$ となるように ζ_n と η_n を選ぶ。
- Bi-CRSTAB 法 : $\eta_n = 0$ とし, ζ_n で残差ノルム $\|\mathbf{r}_{n+1}\|$ を局所的に最小化する。
- GPBi-CG 法 : ζ_n と η_n で残差ノルム $\|\mathbf{r}_{n+1}\|$ を局所的に最小化する。

次に、上記の項目に基づいた具体的な計算式を Table 1 に示す。

Table 1. The choice for the product-type methods based on the Bi-CR method.

CRS	$\zeta_n = \alpha_n, \quad \eta_n = \frac{\beta_{n-1}}{\alpha_{n-1}} \alpha_n.$
Bi-CRSTAB	$\zeta_n = \frac{(At_n, t_n)}{(At_n, At_n)}, \quad \eta_n = 0.$
GPBi-CR	$\zeta_n = \frac{(\mathbf{y}_n, \mathbf{y}_n)(At_n, t_n) - (\mathbf{y}_n, t_n)(At_n, \mathbf{y}_n)}{(At_n, At_n)(\mathbf{y}_n, \mathbf{y}_n) - (\mathbf{y}_n, At_n)(At_n, \mathbf{y}_n)},$ $\eta_n = \frac{(At_n, At_n)(\mathbf{y}_n, t_n) - (\mathbf{y}_n, At_n)(At_n, t_n)}{(At_n, At_n)(\mathbf{y}_n, \mathbf{y}_n) - (\mathbf{y}_n, At_n)(At_n, \mathbf{y}_n)},$ If $n = 0$, then $\zeta_n = \frac{(At_n, t_n)}{(At_n, At_n)}, \quad \eta_n = 0.$

本小節の最後に、Bi-CR 法とその積型解法の一反復当たりの演算量を Table 2 に示す。

Table 2. Summary of operations for iteration.

Method	Inner Product	AXPY	Matrix-Vector Product
Bi-CR	2	6	1+1
CRS	2	6	2
Bi-CRSTAB	4	6	2
GPBi-CR	7	12	2

ここで、“1+1”は行列・ベクトル積が一回、転置行列・ベクトル積が一回を意味する。“AXPY”は、ベクトルのスカラー倍とベクトルとの和 ($ax + y$) の回数を意味する。

4 数値実験

本節では、Harwell-Boeing collection [3], NEP collection [1], SPARSKIT collection [8], そして Tim Davis's collection [2] の中から、回路シミュレーション (ADD*, MENPLUS), 有限要素モデリング (CAVITY05), 流体力学 (CDDE1), 電気工学 (DW2048), 化学反応速度論 (FS5414), 油層シミュレーション (ORS*, SHERMAN*), 偏微分方程式 (PDE2961), そして石油工学 (WATT1) の問題に対して Bi-CR 法の積型解法 (CRS 法, Bi-CRSTAB 法, GPBi-CR 法) 及び Bi-CG 法の積型解法 (CGS 法, Bi-CGSTAB 法, GPBi-CG 法) を適用し、性能を評価する。

数値実験は、ALPHA STATION (750MHz) において、FORTRAN コンパイラの倍精度演算で実験を行い、各々の係数行列に対して右辺項は乱数で生成した。また、近似解の初期値は $\mathbf{x}_0 = [0, 0, \dots, 0]^T$ とし、収束判定条件は $\|\mathbf{r}_n\|/\|\mathbf{b}\| \leq 1.0 \times 10^{-12}$ を採用した。

Table 3. Test problems (N = order of matrix, NNZ = nonzeros in matrix) and Its (the number of iterations) for the iterative methods.

Matrix	N	NNZ	Its					
			CGS	CRS	GSTA	RSTA	GPBCG	GPBCR
ADD20	2395	17319	422	383	811	829	568	560
ADD32	4960	23884	73	71	83	100	78	74
CAVITY05	1182	32747	569	576	1064	1377	737	978
CDDE1	961	4681	123	123	128	135	116	139
DW2048	2048	10114	1865	1541	2354	2741	2138	2245
FS5414	541	4285	1343	1390	2555	3032	1709	1932
MENPLUS	17758	126150	1386	1148	2742	2462	1875	1973
ORSIRR1	1030	6858	1046	1157	2684	1996	1767	1714
ORSIRR2	886	5970	874	838	1969	1935	1380	1553
ORSREG1	2205	14133	344	325	653	603	573	537
PDE2961	2961	14585	180	174	169	167	167	171
SHERMAN1	1000	3750	552	510	539	571	572	599
SHERMAN5	3312	20793	2459	1797	3914	5252	3009	4030
WATT1	1856	11360	464	470	287	331	473	1015

Table 4. Test problems (N = order of matrix, NNZ = nonzeros in matrix) and TRR ($\log_{10}$ of the final true relative residual 2-norm) for the iterative methods.

Matrix	N	NNZ	TRR					
			CGS	CRS	GSTA	RSTA	GPBCG	GPBCR
ADD20	2395	17319	-12.23	-12.23	-11.85	-11.95	-11.83	-11.92
ADD32	4960	23884	-12.17	-12.03	-12.00	-12.29	-12.05	-12.18
CAVITY05	1182	32747	-8.87	-11.70	-12.10	-12.19	-11.97	-11.58
CDDE1	961	4681	-12.01	-12.04	-12.01	-12.62	-12.05	-12.27
DW2048	2048	10114	-8.99	-11.07	-11.74	-12.01	-11.32	-12.09
FS5414	541	4285	-6.31	-8.82	-9.56	-9.48	-7.34	-8.20
MENPLUS	17758	126150	-6.78	-11.82	-11.59	-11.62	-9.89	-11.63
ORSIRR1	1030	6858	-9.52	-12.00	-11.77	-11.86	-11.73	-11.77
ORSIRR2	886	5970	-10.03	-12.14	-11.83	-11.79	-11.57	-10.99
ORSREG1	2205	14133	-12.07	-12.05	-12.08	-11.93	-12.08	-11.92
PDE2961	2961	14585	-12.46	-12.11	-12.31	-12.00	-12.16	-13.06
SHERMAN1	1000	3750	-12.15	-12.18	-12.01	-12.04	-12.07	-12.07
SHERMAN5	3312	20793	-3.99	-10.82	-11.74	-11.30	-9.79	-11.04
WATT1	1856	11360	-12.15	-12.15	-12.03	-12.10	-9.17	-8.65

[実験結果]

収束判定条件を満たした時点における各解法の反復回数 (Its) を Table 3 に, 対応する真の相対残差 (TRR: $\log_{10} \|b - Ax_n\|/\|b\|$) を Table 4 に示す. Table 3, 4 の記号である “GSTA” と “RSTA” はそれぞれ “Bi-CGSTAB” と “Bi-CRSTAB” の略であり, “GPBCG” と “GPBCR” はそれぞれ “GPBi-CG” と “GPBi-CR” の略である. Table 3 の網かけ部分は, それぞれ CGS 法と CRS 法, Bi-CGSTAB 法と Bi-CRSTAB 法, GPBi-CG 法と GPBi-CR 法の反復回数で一方が他方の反復回数より約 10 % 以上少ないことを意味する. Table 4 の網かけ部分は, それぞれ CGS 法と CRS 法, GPBi-CG 法と GPBi-CR 法の真の相対残差で一方が他方より精度が一桁以上高いことを意味する. Bi-CGSTAB 法と Bi-CRSTAB 法の真の相対残差は, ほぼ同程度であったので網かけは行なわなかった.

実験結果を, 反復回数と真の相対残差の 2 つの観点で評価する.

Table 3 の反復回数の観点

1. CRS 法は, CGS 法より全体的に速く収束する傾向を示した.
2. Bi-CRSTAB 法と Bi-CGSTAB 法は, 同程度の収束性を示した.
3. GPBi-CR 法は, GPBi-CG 法に比べて収束性がやや悪くなる傾向を示した.

Table 4 の真の相対残差の観点

1. CGS 法の近似解が大きく桁落ちする問題において CRS 法は, 非常に高い精度の近似解を生成した.
2. Bi-CRSTAB 法と Bi-CGSTAB 法は, ほぼ同程度の精度の近似解を生成した.
3. GPBi-CR 法と GPBi-CG 法は, ほぼ同程度の精度の近似解を生成した.

5 まとめ

本稿では, 最近提案された非対称行列用の解法である Bi-CR 法に対してその積型解法を提案しアルゴリズムを導いた. 近年の優れた研究成果として知られている Bi-CG 法の積型解法は, 偏微分方程式から得られた線形方程式に対して極めて有効であることが確認されているため, そのアナロジーとして積型の考え方を Bi-CR 法に適用し収束の加速を試みた. その中で特に CRS 法は, CGS 法に比べ高い収束性を有し, かつ高精度の近似解を生成することから非対称線形方程式を効率良く解くための有力な解法になると期待される.

参考文献

- [1] Bai, Z., Day, D., Demmel, J. and Dongarra, J., A Test Matrix Collection for Non-Hermitian Eigenvalue Problems, Technical Report CS-97-355, Department of Computer Science, University of Tennessee, Knoxville, TN, March, 1997.
- [2] Davis, T., Sparse Matrix Collection, NA Digest, Vol. 94, Issue 42, October 1994.

- [3] Duff, I. S., Grimes, R. G. and Lewis, J. G., User's Guide for the Harwell-Boeing Sparse Matrix Collection, Technical Report RAL-92-086, Rutherford Appleton Laboratory, Chilton, UK, 1992.
- [4] Fletcher, R., Conjugate Gradient Methods for Indefinite Systems, Lecture Notes in Mathematics, 506 (1976), 73-89.
- [5] Hestenes, M. R. and Stiefel, E., Methods of Conjugate Gradients for Solving Linear Systems, J. Res. Nat. Bur. Standards, 49 (1952), 409-436.
- [6] Golub, G. H. and van Loan, C. F., Matrix Computations, 3rd ed., The Johns Hopkins University Press, Baltimore and London, 1996.
- [7] Lanczos, C., Solution of Systems of Linear Equations by Minimized Iterations, J. Res. Nat. Bur. Standards, 49 (1952), 33-53.
- [8] Saad, Y., SPARSKIT: A Basic Tool Kit for Sparse Matrix Computation, Tech. Rep. CSRD TR 1029, CSRD, University of Illinois, Urbana, IL, 1990.
- [9] Sogabe, T. and Zhang, S.-L., Extended Conjugate Residual Methods for Solving Nonsymmetric Linear Systems, Proceedings of International Conference on Numerical Linear Algebra and Optimization, 2003, (submitted).
- [10] Sonneveld, P., CGS: A Fast Lanczos-type Solver for Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., 10 (1989), 36-52.
- [11] van der Vorst, H. A., Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., 13 (1992), 631-644.
- [12] Zhang, S.-L., GPBi-CG: Generalized Product-type Methods Based on Bi-CG for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Comput., 18 (1997), 537-551.