Proceedings of the World Congress on Engineering 2012 Vol I
WCE 2012, July 4 - 6, 2012, London, U.K.

A Proposal of Variants of BiCGSafe Method for Solving Linear Systems in Realistic Problems

Takashi Sekimoto*, Seiji Fujino†

Abstract — **We consider on efficient and stable solution for linear systems by product-type iterative methods. By means of improvement on ordering of Lanczos and stability polynomials which construct algorithm of product-type BiCG method, we derive new algorithm based on BiCGSafe method with safety convergence. We will apply the proposed BiCGSafe method to solve linear systems which appeared in a variety of realistic problems. Through numerical experiments, we will make clear that the proposed variants of BiCGSafe method have an excellent convergence rate.**

Keywords: product-type iterative methods, BiCGSafe method, Lanczos polynomial, preconditioning

1 Introduction

After appearance of CGS[5] and BiCGStab[7] methods, a number of iterative methods based on Lanczos polynomial added with auxiliary polynomial were proposed independently [8]. Then strategy of combination of two polynomials was generalized as a form of product of two polynomials in 1997[9]. However, the optimization of product of polynomials remains as an open problem. The first solution among naive realization of product-type iterative methods was made partly by Fujino *et al*. in 2005 [2] because of adoption of associate residual in place of residual for decision of undetermined two parameters. They found out a clue in the ordering of developing of polynomials. As a result, they succeeded fairly in reduction of instability of convergence. They referred BiCGSafe method in view of safety convergence. With the same strategy, i.e., adoption of associate residual, variants of GPBiCG method were produced such as GPBiCG_AR in 2009 [3].

On the other hand, different approach exists also, e.g., an approach of two-term recurrence proposed by prof. Rutishauser in 1959 [4]. Recently K. Abe and G. Sleijpen [1] applied it to improvement of some variants of GPBiCG method. However, some variants succeeded in, some variants may failed out in view of convergence rate and stability of convergence.

In this paper, we derive variants of BiCGSafe method [2] by means of considering on ordering of Lanczos and stability polynomials which construct algorithm of product-type BiCG method. This paper is organized as follows. In section 2 we describe derivation of variants of BiCG method, and estimate the computational cost. In section 3, safety convergence of variants of BiCGSafe method will be demonstrated by numerical experiments, and in section 4 finally we will draw some conclusions.

2 Derivation of variants of BiCGSafe method

We will focus on the iteration solution of a linear system of equations

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

in which A is a non-singular real $n \times n$ matrix and $\boldsymbol{b}$ is a given real n-vector. Starting from some initial guess $\boldsymbol{x}_0$ for the solution, BiCG(Bi-Conjugate Gradient) method generates a sequence $\boldsymbol{x}_k$ with the property that the kth residual $\boldsymbol{r}_k := \boldsymbol{b} - A\boldsymbol{x}_k$ lies in the Krylov subspace generated by A from $\boldsymbol{r}_0$, i.e.,

$$\boldsymbol{r}_k \in K_{k+1} := \mathrm{span}(\boldsymbol{r}_0, A\boldsymbol{r}_0, \cdots, A^k\boldsymbol{r}_0). \tag{2}$$

The sequence of residual polynomials R_k are defined by

$$\boldsymbol{r}_k := R_k(A)\boldsymbol{r}_0 \tag{3}$$

and polynomials $R_k(\lambda)$ are refered to as the so-called Lanczos polynomials.

In the standard treatment of the BiCG method, second polynomials $P_k(\lambda)$ play a role, and are defined by

$$\boldsymbol{p}_k := P_k(A)\boldsymbol{r}_0. \tag{4}$$

We note that the basic recurrence relations between $R_k(\lambda)$ and $P_k(\lambda)$ hold as follows:

$$R_0(\lambda) = 1, \quad P_0(\lambda) = 1, \tag{5}$$
$$R_{k+1}(\lambda) = R_k(\lambda) - \alpha_k \lambda P_k(\lambda), \tag{6}$$

*Ryoyu System Engineering Co., Ltd., 5-1-6 Komatsudori, Hyogo-ku, Kobe 652-0865, Japan, Email:takashi_sekimoto@rsg.kobe.mhi.co.jp
†Research Institute for Information Technology, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan, Email: fujino@kyushu-u.ac.jp

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$P_{k+1}(\lambda) = R_{k+1}(\lambda) + \beta_k P_k(\lambda), \quad k = 1, 2, \ldots. \quad (7)$$

Then we can introduce the three-term recurrence relations for Lanczos polynomials $R_k(\lambda)$ only by eliminating $P_k(\lambda)$ from eqns. (5) and (7) as follows:

$$\begin{aligned} R_0(\lambda) &= 1,\ R_1(\lambda) = (1 - \alpha_0\lambda)R_0(\lambda), && (8)\\ R_{k+1}(\lambda) &= (1 + \frac{\beta_{k-1}}{\alpha_{k-1}}\alpha_k - \alpha_k\lambda)R_k(\lambda) \\ &\quad - \frac{\beta_{k-1}}{\alpha_{k-1}}\alpha_k R_{k-1}(\lambda), \quad k = 1, 2, \ldots. && (9)\end{aligned}$$

S.-L. Zhang [9] discovered that an often excellent convergence property can be gained by choosing for stability polynomials $H_k(\lambda)$ that are built up in the three-term recurrence of form as polynomial $R_k(\lambda)$ in eqns. (8) and (9) by adding suitable undetermined parameters ζ_k and η_k as follows:

$$\begin{aligned} H_0(\lambda) &= 1,\ H_1(\lambda) = (1 - \zeta_0\lambda)H_0(\lambda), && (10)\\ H_{k+1}(\lambda) &= (1 + \eta_k - \zeta_k\lambda)H_k(\lambda) \\ &\quad -\eta_k H_{k-1}(\lambda),\ k = 1, 2, \ldots. && (11)\end{aligned}$$

By reconstruction of eqn. (9) using the stability polynomials $H_k(\lambda)$ and $G_k(\lambda)$, we have the following coupled two-term recursion of the form as

$$\begin{aligned} H_0(\lambda) &= 1,\ G_0(\lambda) = \zeta_0, && (12)\\ H_k(\lambda) &= H_{k-1}(\lambda) - \lambda G_{k-1}(\lambda), && (13)\\ G_k(\lambda) &= \zeta_k H_k(\lambda) + \eta_k G_{k-1}(\lambda),\ k = 1, 2, \ldots. && (14)\end{aligned}$$

3 Algorithm of variants of BiCGSafe method

We derived the algorithm of BiCGSafe method by developing firstly Lanczos polynomial $R_{k+1}(A)$ in the residual vector $\boldsymbol{r}_k := H_{k+1}(A)R_{k+1}(A)\boldsymbol{r}_0$. In this section, we consider on developing of three auxiliary polynomials of $H_{k+1}(\lambda)P_{k+1}(\lambda)$, $\lambda G_k(\lambda)P_k(\lambda)$ and $G_k(\lambda)R_{k+1}(\lambda)$. For reduction of computational cost, we develop firstly $P_k(\lambda)$ in the update formula of auxiliary polynomial $\lambda G_k(\lambda)P_k(\lambda)$ as below.

$$\begin{aligned} \lambda G_k(\lambda)P_k(\lambda) &= \lambda G_k(\lambda)(R_k(\lambda) + \beta_{k-1}P_{k-1}(\lambda)) \\ &= \lambda G_k(\lambda)R_k(\lambda) + \beta_{k-1}\lambda(\zeta_k H_k(\lambda) \\ &\quad +\eta_k G_{k-1}(\lambda))P_{k-1}(\lambda) \\ &= \lambda G_k(\lambda)R_k(\lambda) \\ &\quad +\beta_{k-1}(\zeta_k\lambda H_k(\lambda)P_{k-1}(\lambda) \\ &\quad +\eta_k\lambda G_{k-1}(\lambda)P_{k-1}(\lambda)). && (15)\end{aligned}$$

We develop the following auxiliary polynomials $\lambda G_k(\lambda)R_k(\lambda)$ and $\lambda_k H_k(\lambda)P_{k-1}(\lambda)$.

$$\begin{aligned} \lambda G_k(\lambda)R_k(\lambda) &= \lambda(\zeta_k(\lambda)H_k(\lambda) + \eta_k G_{k-1}(\lambda))R_k(\lambda) \\ &= \zeta_k\lambda H_k(\lambda)R_k(\lambda) \\ &\quad +\eta_k\lambda G_{k-1}(\lambda)R_k(\lambda), && (16)\\ \lambda H_k(\lambda)P_{k-1}(\lambda) &= \lambda(H_{k-1}(\lambda) - \lambda G_{k-1}(\lambda))P_{k-1}(\lambda) \\ &= \lambda H_{k-1}(\lambda)P_{k-1}(\lambda) \\ &\quad -\lambda(\lambda G_{k-1}(\lambda)P_{k-1}(\lambda)). && (17)\end{aligned}$$

Here we introduce the following auxiliary vectors.

$$\begin{aligned} \boldsymbol{p}_k &:= H_k(A)P_k(A)\boldsymbol{r}_0, && (18)\\ \boldsymbol{u}_k &:= AG_k(A)P_k(A)\boldsymbol{r}_0, && (19)\\ \boldsymbol{z}_k &:= G_k(A)R_{k+1}(A)\boldsymbol{r}_0, && (20)\\ \boldsymbol{y}_{k+1} &:= AG_k(A)R_{k+1}(A)\boldsymbol{r}_0, && (21)\\ \boldsymbol{q}_k &:= AG_k(A)R_k(A)\boldsymbol{r}_0, && (22)\\ \boldsymbol{t}_k &:= AH_k(A)P_{k-1}(A)\boldsymbol{r}_0. && (23)\end{aligned}$$

We get three auxiliary vectors by substituting auxiliary vectors $\boldsymbol{q}_k$, $\boldsymbol{t}_k$ into eqns. (15)-(17).

$$\begin{aligned} \boldsymbol{u}_k &= \boldsymbol{q}_k + \beta_{k-1}(\zeta_k\boldsymbol{t}_k + \eta_k\boldsymbol{u}_{k-1}), && (24)\\ \boldsymbol{q}_k &= \zeta_k A\boldsymbol{r}_k + \eta_k\boldsymbol{y}_k, && (25)\\ \boldsymbol{t}_k &= A\boldsymbol{p}_{k-1} - A\boldsymbol{u}_{k-1} && (26)\end{aligned}$$

Moreover, we can modify $A\boldsymbol{p}_k$ and $\boldsymbol{y}_{k+1}$ by the auxiliary vectors $\boldsymbol{q}_k$ and $\boldsymbol{t}_k$ as below.

$$\begin{aligned} A\boldsymbol{p}_k &= A\boldsymbol{r}_k + \beta_{k-1}(A\boldsymbol{p}_{k-1} - A\boldsymbol{u}_{k-1}) \\ &= A\boldsymbol{r}_k + \beta_{k-1}\boldsymbol{t}_k, && (27)\\ \boldsymbol{y}_{k+1} &= \zeta_k A\boldsymbol{r}_k + \eta_k\boldsymbol{y}_k - \alpha_k A\boldsymbol{u}_k \\ &= \boldsymbol{q}_k - \alpha_k A\boldsymbol{u}_k. && (28)\end{aligned}$$

Similarly we can modify the residual vector $\boldsymbol{r}_{k+1}$ by using the auxiliary vectors $\boldsymbol{q}_k$ and $\boldsymbol{t}_k$.

$$\begin{aligned} \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k - \boldsymbol{y}_{k+1} && (29)\\ &= \boldsymbol{r}_k - \alpha_k\boldsymbol{t}_k - \boldsymbol{q}_k. && (30)\end{aligned}$$

On the other hand, by using the update formula for associate residual $\boldsymbol{a\_r}_k := H_{k+1}(A)R_k(A)\boldsymbol{r}_0$,

$$\begin{aligned} H_{k+1}(\lambda)R_k(\lambda) &= H_k(\lambda)R_k(\lambda) - \zeta_k\lambda H_k(\lambda)R_k(\lambda) \\ &\quad -\eta_k\lambda G_{k-1}(\lambda)R_k(\lambda) && (31)\end{aligned}$$

we determine two parameters ζ_k, η_k from the local minimization of 2-norm of the associate residual vector $\boldsymbol{a\_r}_k$ as follows:

$$||\boldsymbol{a\_r}_k||_2 = ||\boldsymbol{r}_k - \zeta_k A\boldsymbol{r}_k - \eta_k\boldsymbol{y}_k||_2. \quad (32)$$

From the above derivation, we can present the algorithms as follows: We refer to these algorithms as BiCGSafe_variant_1 (abberivated as BiCGSafe_var_1) and BiCGSafe_variant_2 (abberivated as BiCGSafe_var_2) methods. We show difference between BiCGSafe_var_1 method and BiCGSafe_var_2 method in the update formula for residual vector $\boldsymbol{r}_{k+1}$ as eqns. (48) and (49). The underlined update procedures differ from that of the algorithm of BiCGSafe method.

ISBN: 978-988-19251-3-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithms of BiCGSafe_var_1(var_2) methods

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
Choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*, \boldsymbol{r}_0) \neq 0$, $\beta_{-1} = 0$,
for $k = 0, 1, \cdots$ until $||\boldsymbol{r}_k|| \leq \varepsilon\ ||\boldsymbol{r}_0||$

$$\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_{k-1}(\boldsymbol{p}_{k-1} - \boldsymbol{u}_{k-1}), \quad (33)$$

Compute $A\boldsymbol{r}_k$, (34)

$$\underline{A\boldsymbol{p}_k = A\boldsymbol{r}_k + \beta_{k-1}\boldsymbol{t}_k}, \quad (35)$$

$$\alpha_k = \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_k)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_k)}, \quad (36)$$

$$\boldsymbol{a}_k = \boldsymbol{r}_k, \quad \boldsymbol{b}_k = \boldsymbol{y}_k, \quad \boldsymbol{c}_k = A\boldsymbol{r}_k, \quad (37)$$

$$\zeta_k = \frac{(\boldsymbol{b}_k, \boldsymbol{b}_k)(\boldsymbol{c}_k, \boldsymbol{a}_k) - (\boldsymbol{b}_k, \boldsymbol{a}_k)(\boldsymbol{c}_k, \boldsymbol{b}_k)}{(\boldsymbol{c}_k, \boldsymbol{c}_k)(\boldsymbol{b}_k, \boldsymbol{b}_k) - (\boldsymbol{b}_k, \boldsymbol{c}_k)(\boldsymbol{c}_k, \boldsymbol{b}_k)}, \quad (38)$$

$$\eta_k = \frac{(\boldsymbol{c}_k, \boldsymbol{c}_k)(\boldsymbol{b}_k, \boldsymbol{a}_k) - (\boldsymbol{b}_k, \boldsymbol{c}_k)(\boldsymbol{c}_k, \boldsymbol{a}_k)}{(\boldsymbol{c}_k, \boldsymbol{c}_k)(\boldsymbol{b}_k, \boldsymbol{b}_k) - (\boldsymbol{b}_k, \boldsymbol{c}_k)(\boldsymbol{c}_k, \boldsymbol{b}_k)}, \quad (39)$$

$$(\text{if } k = 0, \text{ then } \zeta_k = \frac{(\boldsymbol{c}_k, \boldsymbol{a}_k)}{(\boldsymbol{c}_k, \boldsymbol{c}_k)}, \ \eta_k = 0) \quad (40)$$

$$\underline{\boldsymbol{q}_k = \zeta_k A\boldsymbol{r}_k + \eta_k \boldsymbol{y}_k}, \quad (41)$$

$$\underline{\boldsymbol{u}_k = \boldsymbol{q}_k + \beta_{k-1}(\zeta_k \boldsymbol{t}_k + \eta_k \boldsymbol{u}_{k-1})}, \quad (42)$$

$$\boldsymbol{z}_k = \zeta_k \boldsymbol{r}_k + \eta_k \boldsymbol{z}_{k-1} - \alpha_k \boldsymbol{u}_k, \quad (43)$$

Compute $A\boldsymbol{u}_k$, (44)

$$\underline{\boldsymbol{y}_{k+1} = \boldsymbol{q}_k - \alpha_k A\boldsymbol{u}_k}, \quad (45)$$

$$\underline{\boldsymbol{t}_{k+1} = A\boldsymbol{p}_k - A\boldsymbol{u}_k}, \quad (46)$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k + \boldsymbol{z}_k, \quad (47)$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k - \boldsymbol{y}_{k+1}, \quad (48)$$

$$(= \boldsymbol{r}_k - \alpha_k \boldsymbol{t}_k - \boldsymbol{q}_k), \quad (49)$$

$$\beta_k = \frac{\alpha_k}{\zeta_k} \cdot \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{k+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_k)}, \quad (50)$$

end for

4 Numerical experiments

All computations were done in double precision floating point arithmetic, and performed on Dell PowerEdge R210 II(CPU: Intel Xeon E3-1220, clock: 3.1GHz, memory: 8Gbytes, OS: Scientific Linux 6.0). Compiler of Intel Fortran Compiler version 11.0 is used. All codes were compiled with the "-O3" optimization option. The right-hand side $\boldsymbol{b}$ was imposed from the physical load conditions. The stopping criterion for successful convergence of the iterative methods is less than 10^{-12} of the relative residual 2-norm $||\boldsymbol{r}_{k+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$. In all cases the iteration was started with the initial guess solutions $\boldsymbol{x}_0 = \boldsymbol{0}$. The maximum number of iterations is fixed as 10^4. Matrices are normalized with diagonal scaling. The initial shadow residual $\boldsymbol{r}_0^*$ is equal to the initial residual $\boldsymbol{r}_0 (= \boldsymbol{b} - A\boldsymbol{x}_0)$. All matrices are taken from Florida sparse matrix collection[6].

We examined performance and stability of convergence of GPBiCG, GPBiCG_v1 (Abe-Sleijpen GPBiCG variant-1), GPBiCG_v2 (Abe-Sleijpen GPBiCG variant-2), BiCGSafe, BiCG-Safe_var_1 and BiCGSafe_var_2 methods with ILU(0) preconditioning without extra fill-ins.

Table 1 shows the numerical results of GPBiCG, GP-BiCG_v1, GPBiCG_v2, BiCGSafe, BiCGSafe_var_1 and BiCGSafe_var_2 methods. "pre-t." means computation time of making preconditioner, "itr-t." means iteration time and "tot-t." means total time in seconds. "ratio" means the ratio of computation time of GPBiCG method to that of each iterative method. "max" denotes non-convergence until iterations reach at the maximum iteration counts. "TRR" (True Relative Residual) means a value of $||\boldsymbol{b} - A\boldsymbol{x}_{n+1}||_2/||\boldsymbol{b} - A\boldsymbol{x}_0||_2$ for the approximate solution $\boldsymbol{x}_{n+1}$. The bold figures implies the fastest case for each matrix.

We can see the following facts from Table 1.

- BiCGSafe_var_1 and BiCGSafe_var_2 methods outperform among the tested iterative methods in view of both computation time for successful convergence.
- Performance of GPBiCG_v1 and GPBiCG_v2 methods is poor.
- Though TRR of BiCGSafe method for matrix bcircuit is slightly poor, TRRs of BiCGSafe- _var_1 and BiCGSafe_var_2 methods are good.

5 Conclusions

We derived algorithms of two variants of BiCGSafe method, i.e., BiCGSafe_var_1 and BiCGSafe _var_2 methods. Moreover we examined performance and robustness of convergence of these iterative methods through numerical experiments. As a result, it turned out that BiCGSafe_var_1 and BiCGSafe_var_2 methods work well from the viewpoint of convergence rate and accuracy of the approximate solution.

Acknowledgments:

The authors appreciate sincerely Ryoyu System Engineering Co., Ltd. for giving opportunity of having a talk at World Congress on Engineering 2012 at London, U.K., 4-6 July, 2012.

References

[1] Abe, K., Sleijpen, G.L.G., Solving linear equations with a STABilized GPBiCG method, Proc. of 14th Kansetouchi workshop of JSIAM, pp.29-34, Okayama University of Science, January, 2011.

1976.

[2] Fujino, S., Fujiwara, M., Yoshida, M.: A proposal of preconditioned BiCGSafe method with safe conver-

Table 1: Performance of some GPBiCG type and BiCGSafe type methods with ILU(0) preconditioning.

(a)matrix: Freescale1

method	iterations	pre-t.	itr-t.	tot-t.	$\log_{10}$(TRR)	ratio
GPBiCG	3475	1.105	1075.170	1076.274	-12.00	1.00
GPBiCG_v1	3113	1.091	996.007	997.097	-10.60	0.93
GPBiCG_v2	3441	1.077	1094.850	1095.930	-10.54	1.02
BiCGSafe	2227	1.125	648.762	649.887	-12.03	0.60
BiCGSafe_var_1	2120	1.135	624.497	**625.632**	-12.31	**0.58**
BiCGSafe_var_2	2287	1.094	673.811	674.904	-12.07	0.63

(b)matrix: bcircuit

method	iterations	pre-t.	itr-t.	tot-t.	$\log_{10}$(TRR)	ratio
GPBiCG	7369	0.023	45.496	45.519	-11.76	1.00
GPBiCG_v1	7665	0.022	47.852	47.874	-10.34	1.05
GPBiCG_v2	max	0.021	62.912	62.933	-10.17	1.38
BiCGSafe	6273	0.021	37.294	37.315	(-9.55)	-
BiCGSafe_var_1	3601	0.023	21.348	21.371	-11.98	0.47
BiCGSafe_var_2	3322	0.022	19.906	**19.928**	-12.02	**0.44**

(c)matrix: big

method	iterations	pre-t.	itr-t.	tot-t.	$\log_{10}$(TRR)	ratio
GPBiCG	1033	0.003	1.170	1.173	-11.32	1.00
GPBiCG_v1	1023	0.005	1.220	1.225	-10.96	1.04
GPBiCG_v2	1027	0.006	1.209	1.215	-10.56	1.04
BiCGSafe	823	0.004	0.906	0.910	-12.12	0.78
BiCGSafe_var_1	825	0.006	0.922	0.928	-12.25	0.79
BiCGSafe_var_2	770	0.005	0.863	**0.868**	-12.33	**0.74**

gence, Proc. of The 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, CD-ROM, Paris, France, 2005.7.

[3] Moethuthu, Fujino, S.,: Stability of GPBiCG_AR method based on minimization of associate residual, Journal of ASCM, pp.108-120, 2008.

[4] Rutishauser, H., Theory of gradient method, in: Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Value Problems, in: Mitt. Inst. Angew. Math. ETH Zürich, Birkhäuser, Basel, pp.24-49, 1959.

[5] Sonneveld, P., CGS: a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statis. Comput., Vol.10, pp.36-52, 1989.

[6] University of Florida Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/ matrices/index.html

[7] 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. Comput., 13, pp.631-644, 1992.

[8] van der Vorst, H.A., Iterative Krylov Methods for Large Linear Systems, Cambdrige University Press, Cambdrige, 2003.

[9] Zhang, S.-L., GPBi-CG: Generalized product-type methods preconditionings based on Bi-CG for solving nonsymmetric linear systems, SIAM J. Sci. Comput.,vol.18, pp.537-551, 1997.