Applied Mathematics, **2012, 3, 331-335**
http://dx.doi.org/10.4236/am.2012.34050 Published Online April 2012 (http://www.SciRP.org/journal/am)

Flexible GPBi-CG Method for Nonsymmetric Linear Systems*

Jia-Min Wang[1], Tong-Xiang Gu[2#]
[1]Chinese Academy of Engineering Physics, Beijing, China
[2]Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, China
Email: #txgu@iapcm.ac.cn

Received November 7, 2011; revised February 22, 2012; accepted February 29, 2012

ABSTRACT

We present a flexible version of GPBi-CG algorithm which allows for the use of a different preconditioner at each step of the algorithm. In particular, a result of the flexibility of the variable preconditioner is to use any iterative method. For example, the standard GPBi-CG algorithm itself can be used as a preconditioner, as can other Krylov subspace methods or splitting methods. Numerical experiments are conducted for flexible GPBi-CG for a few matrices including some nonsymmetric matrices. These experiments illustrate the convergence and robustness of the flexible iterative method.

Keywords: Krylov Subspace Method; Flexible Preconditioning; Inner-Outer Iteration; GPBi-CG

1. Introduction

Krylov subspace methods are the iterative choice for solving linear system of the form

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

where the matrix $\boldsymbol{A}$ is assumed to be nonsingular. The strength of Krylov subspace methods are most apparent when combined with a preconditioner. We only consider right preconditioning in the paper. Thus one solves the equivalent linear system

$$AM^{-1}(Mx) = b. \quad (2)$$

The preconditioner M is selected to be close to the matrix $\boldsymbol{A}$. And the matrix AM^{-1} is never formed explicitly. Instead, when $M^{-1}v = z$ is needed, one solves the corresponding system

$$Mz = v. \quad (3)$$

In this paper, we present a flexible version of GPBi-CG, which allows the preconditioner M vary from one iteration to another. Let us denote the matrix $\boldsymbol{M}_n$ the preconditioner used in the nth iteration. The need to allow for a variable preconditioner arises when the solution of (2) is not obtained exactly (say, by a direct method), but is approximated by a second (inner) iterative method.

In recent years, several flexible variants of Krylov subspace methods have been established successfully. They include flexible CG, which is applied on a symmetric positive definite matrix [1], flexible GMRES [2], flexible QMR [3], variable preconditioned GCR [4], flexible BiCG and flexible Bi-CGSTAB [5]. Preconditioning as this form is called flexible preconditioning, also known as variable or inexact preconditioning.

The paper is organized as follows. In the next section, we design the FGPBi-CG algorithm, which is a flexible version of GPBi-CG [6]. In Section 3, some numerical experiments will be conducted to illustrate the convergence of the algorithm. Furthermore, in some cases, it is shown that FGPBi-CG can achieve convergence to a tolerance when GPBi-CG is not convergent or even FBi-CGSTAB suffers stagnation. Finally we make some concluding remarks in Section 4.

Throughout the paper, x_0 is the initial approximation, $r_0 = b - Ax_0$ is the initial residual, and the norm used is 2-norm.

2. Flexible GPBi-CG Method

We describe the basic idea of variable preconditioning and how it is incorporated with the algorithm GPBi-CG in this section.

The expression $M^{-1}v$ is calculated at each iteration of the conventional preconditioned Krylov subspace methods. The object of preconditioning is to change the original coefficient matrix A into another matrix close to identity, *i.e.* $AM^{-1} \approx I$. Consequently, the following property that $M^{-1}v$ approximates $A^{-1}v$ can be verified easily.

*The project is partly supported by the NSF of China (No. 61170309, No. 60973151, and No. 91130024) and the major project of scientific and technical development of China Academy of Engineering Physics (2011A0202012).
#Corresponding author.

$$M^{-1}v \approx A^{-1}v .$$

Thus, we consider obtaining an approximation of $A^{-1}v$ instead of computing $M^{-1}v$. That is, the following system (4) is roughly solved by an iterative method to a certain degree of accuracy that is not sufficient.

$$Az = v . \tag{4}$$

Here, an approximation for the system (4) does not need to be solved at the same precision at each iteration. A stopping criteria has been established to make the preconditioner to be changed at each iteration. Different inner-loop can be applied to the system (4) including Krylov subspace methods and stationery iterative methods.

The GPBi-CG algorithm proposed by Zhang [6], uses an unified way to derive a class generalizations of Bi-CG. By choosing different coefficients, namely, the following ζ and η, the GPBi-CG algorithm will be reduced to other methods based on Bi-CG includes the well known CGS, Bi-CGSTAB, Bi-CGSTAB2.

Next we present a flexible version of GPBi-CG, which needs only some small modification of the GPBi-CG code.

ALGORITHM (FGP-BiCG with right preconditioner) x_0 is an initial guess, $r_0 = b - Ax_0$; r_0^* is an arbitrary vector, such that $\left(r_0^*, r_0\right) \neq 0$, e.g., $r_0^* = r_0$; and set $t_{-1} = w_{-1} = 0$, $\beta_{-1} = 0$;

For $n = 0,1,\cdots$ until $\|r_n\| \le \varepsilon \|b\|$ **do**

$$p_n = r_n + \beta_{n-1}\left(p_{n-1} - u_{n-1}\right)$$

solve $M_n \hat{p} = p_n$

$$\alpha_n = \left(r_0^*, r_n\right) / \left(r_0^*, A\hat{p}\right)$$

$$y_n = t_{n-1} - r_n - \alpha_n w_{n-1} + \alpha_n A\hat{p}$$

$$t_n = r_n - \alpha_n A\hat{p}$$

solve $M_n \hat{t} = t_n$

$$\zeta_n = \frac{\left(y_n, y_n\right)\left(A\hat{t}, t_n\right) - \left(y_n, t_n\right)\left(A\hat{t}, y_n\right)}{\left(A\hat{t}, A\hat{t}\right)\left(y_n, y_n\right) - \left(y_n, A\hat{t}\right)\left(A\hat{t}, y_n\right)}$$

$$\eta_n = \frac{\left(A\hat{t}, A\hat{t}\right)\left(y_n, t_n\right) - \left(y_n, A\hat{t}\right)\left(A\hat{t}, t_n\right)}{\left(A\hat{t}, A\hat{t}\right)\left(y_n, y_n\right) - \left(y_n, A\hat{t}\right)\left(A\hat{t}, y_n\right)}$$

(if $n = 0$, then $\zeta_n = \dfrac{\left(A\hat{t}, t_n\right)}{\left(A\hat{t}, A\hat{t}\right)}, \eta_n = 0$)

$$u_n = \zeta_n A\hat{p} + \eta_n\left(t_{n-1} - r_n + \beta_{n-1}u_{n-1}\right)$$

$$z_n = \zeta_n r_n + \eta_n z_{n-1} - \alpha_n u_n$$

solve $M_n \hat{z} = z_n$

$$x_{n+1} = x_n + \alpha_n \hat{p} + \hat{z}$$

$$r_{n+1} = t_n + \eta_n y_n + \zeta_n A\hat{t}$$

$$\beta_n = \frac{\alpha_n}{\zeta_n}\frac{\left(r_0^*, r_{n+1}\right)}{\left(r_0^*, r_n\right)}$$

$$w_n = A\hat{t} + \beta_n A\hat{p}$$

Enddo

Noted that if we replace M_n with *M*, a fixed preconditioner, the above algorithm will be reduced to the standard GPBi-CG method with right preconditioner.

3. Numerical Experiments

In this section, we report some numerical experiments to show the convergence behaviors of FGPBi-CG. In all cases the iteration was started with $x_0 = \left(0, 0, \cdots, 0\right)$, and the outer-loop is stopped when the relative residual norm $\|r_n\| / \|r_0\| \le 10^{-14}$. In the following examples, we use stopping criterion for inner-loop as:

1) $\|r_n\| / \left\|Az_{k+1}^{(l)}\right\| \le \delta$;

2) The maximum number of iterations of inner loop $l = N_{\max}$.

Here, $z_{k+1}^{(l)}$ denotes the *l*-th approximation when computing $Az = v$ at *k*-th steps of the outer-loop.

3.1. Examples for Toeplitz Matrix

In the first example, we consider a Toeplitz matrix of order 200 with a parameter γ.

$$\boldsymbol{A} = \begin{pmatrix} 4 & 0 & 1 & 0.7 & & \\ \gamma & 4 & 0 & 1 & 0.7 & \\ & \gamma & 4 & 0 & 1 & \ddots \\ & & \gamma & 4 & 0 & \ddots \\ & & & \gamma & 4 & \ddots \\ & & & & \ddots & \ddots \end{pmatrix}$$

In this experiment, we choose γ to be 3.79 and the inner iteration stopping criteria to be the maximum iteration is $N_{\max} = 50$ and relative residuals range from $\delta = 10^{-3}$ to 10^{-6}. We can see from the **Figure 1** that GPBi-CG converges faster than that of Bi-CGSTAB. When the standard GPBi-CG algorithm performs well, the flexible version of GPBi-CG is also convergent, but it need more computation. The results can be seen in **Table 1**. In the table, "FG(B)" denotes FGPBi-CG with preconditioning Bi-CGSTAB, and so on, while "MV" represents the number of matrix-vector multiplication, "OIt" denotes the number of outer iteration.

From **Figure 1** and **Table 1**, we can see that for the problem that GPBi-CG and Bi-CGSTAB method can convergent fast, FGPBi-CG and FBi-CGSTAB will not gain too much. While FGPBi-CG(GPBi-CG) and FBi-CGSTAB(GPBi-CG) will be faster than FGPBi-CG(Bi-CGSTAB) and FBi-CGSTAB(Bi-CGSTAB) re-

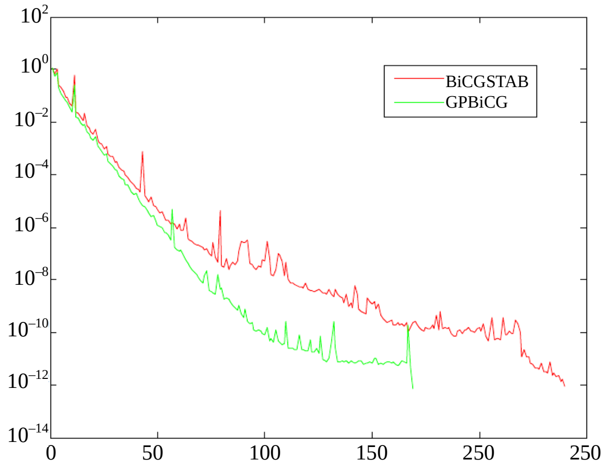

Figure 1. Convergence history of Bi-CGSTAB and GPBi-CG for Toeplitz matrix 1.

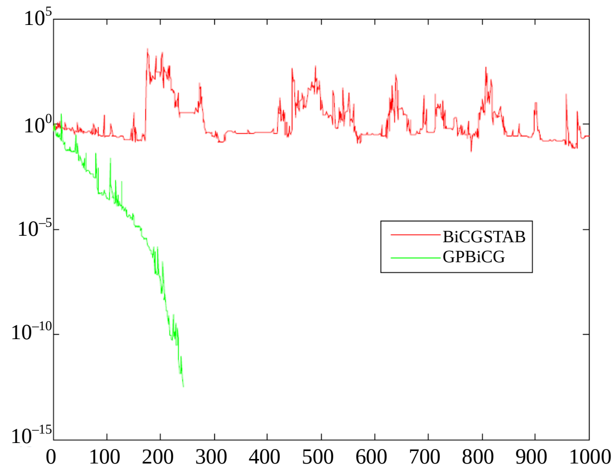

Figure 2. Convergence history of Bi-CGSTAB and GPBi-CG for Toeplitz matrix 2.

Table 1. Performance comparison: γ = 3.79.

δ	FG(B)		FG(G)		FB(B)		FB(G)	
	MV	OIt	MV	OIt	MV	OIt	MV	OIt
10^{-3}	550	3	506	3	356	3	316	3
10^{-4}	454	2	416	2	294	2	258	3
10^{-5}	1102	4	528	2	736	4	332	2
10^{-6}	574	2	550	2	374	2	350	2

Table 2. Performance comparison: γ = 1.9.

δ	FG(B)		FG(G)		FB(B)		FB(G)	
	MV	OIt	MV	OIt	MV	OIt	MV	OIt
10^{-3}	3926	13	1158	5	2626	13	758	5
10^{-4}	3926	13	904	3	2626	13	574	3
10^{-5}	3926	13	896	3	2626	13	604	3
10^{-6}	3926	13	906	3	2626	13	606	2

spectively. Because GPBi-CG used in the inner iteration usually converges faster than Bi-CGSTAB.

Now, we consider another Toeplitz matrix of order 200 with a parameter γ as following.

$$A=\begin{pmatrix} 2 & 1 & & & & \\ 0 & 2 & 1 & & & \\ \gamma & 0 & 2 & 1 & & \\ & \gamma & 0 & 2 & 1 & \\ & & \gamma & 0 & 2 & \ddots \\ & & & \ddots & \ddots & \ddots \end{pmatrix}$$

In this experiment, we choose γ to be 1.9 and the inner iteration stopping criteria to be the maximum iteration is $N_{\max}=50$ and relative residuals range from $\delta=10^{-3}$ to 10^{-6}.

We can see from the **Figure 2** and **Table 2** that GPBi-CG is convergent while Bi-CGSTAB is not. As a result, when Bi-CGSTAB is used as an inner iteration, the number of matrix vector multiplication is not influenced by the inner relative residual stopping criteria, *i.e.*, the inner iteration is not terminated until it reaches $N_{\max}$ (the largest iterative number of inner loop). For this reason, GPBi-CG used as the inner iteration performs better.

3.2. Examples for Model Problem

We consider the finite difference discretization of the partial differential equation ([3,5,7])

$$-\delta u+\gamma\left(xu_x+yu_y\right)+\beta u=f \tag{5}$$

on a unit square, where f is such that the exact solution to the discretized equation $Ax=b$ is $x=(1,1,\cdots,1)$. The parameters β and γ are chosen to have a nonsymmetric matrix. In our experiment, $\beta=10$ and $\gamma=100$ or $\gamma=1000$. The mesh is chosen of equal size in both dimension (32 nodes), and the corresponding matrix is thus of order 1024.

In the first example, we take $\beta=10$, $\gamma=100$ and the inner iteration stopping criteria is $N_{\max}$ ranges from 30 to 70 and relative residuals is 10^{-6}.

For this choice, **Figure 3** and **Table 3** show that both Bi-CGSTAB and GPBi-CG perform quite well for this example, flexible versions of these algorithms are also convergent. If the inner-loop stopping criterion is $N_{\max}$ = 40 and $\delta=10^{-6}$, the FBi-CGSTAB(Bi-CGSTAB) needs 316 matrix vector multiplications to reach the prescribed tolerance, faster than that of Bi-CGSTAB.

In the next experiment, we choose $\beta=10$, $\gamma=1000$ and the inner iteration stopping criteria is $N_{\max}$ ranges from 90 to 1000 and relative residuals is 10^{-9}.

Neither Bi-CGSTAB nor GPBi-CG without preconditioning is convergent for this problem (see **Figure 4**).

We see from **Table 4** that FGPBi-CG and FBi-CGSTAB with flexible precondition converges, but the cost of FGPBi-CG is about one and a half that of FBi-

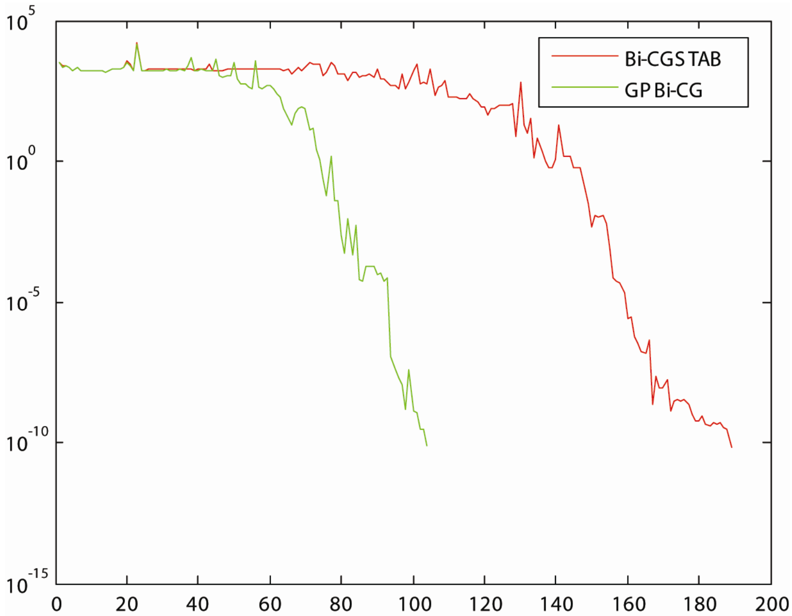

Figure 3. Convergence history of Bi-CGSTAB and GPBi-CG for $\beta = 10$, $\gamma = 100$.

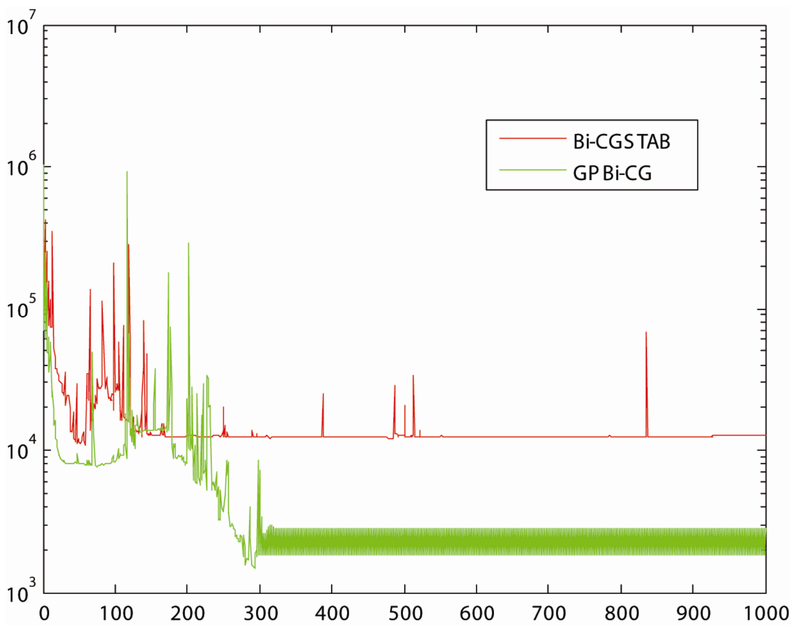

Figure 4. Convergence history of Bi-CGSTAB and GPBi-CG for $\beta = 10$ and $\gamma = 1000$.

Table 3. Performance comparison: $\beta = 10$, $\gamma = 100$.

N_{max}	FG(B)		FG(G)		FB(B)		FB(G)	
	MV	OIt	MV	OIt	MV	OIt	MV	OIt
30	910	5	1274	7	482	4	488	4
40	446	2	726	3	316	2	486	3
50	484	2	520	2	334	2	348	2
60	498	2	512	2	350	2	358	2
70	530	2	582	2	378	2	402	2

Table 4. Performance comparison: $\beta = 10$; $\gamma = 1000$.

N_{max}	FB(G)		FG(G)	
	MV	OIt	MV	OIt
90	2534	7	9576	18
140	3372	6	6736	8
170	2728	4	4088	4
200	3208	4	4808	4
300	4720	4	12,606	7
500	4836	3	7972	3
1000	4880	2	6870	2

CGSTAB, because there are three inner-loops in the FGPBi-CG algorithm rather than two in FBi-CGSTAB.

And from this table, we see FGPBi-CG (GPBi-CG) converges for most of the inner-loop stopping criteria. When appropriate stopping criteria is used in the inner iteration, the flexible version will be a good choice.

4. Conclusion

We have formulated a flexible version of GPBi-CG for the large sparse nonsymmetric linear systems. The preconditioning is carried out by roughly solving $Az = v$ by an iterative method to a certain degree of precision. In our proposal, the iteration for solving $Az = v$ is stopped according to satisfy a certain accuracy of approximation or the maximum number of iterations, so the preconditioner is changed at each outer iteration. Our numerical experiments show that FGPBi-CG is a viable alternative to GPBi-CG. And some examples show that FGPBi-CG is convergent when GPBi-CG suffers from stagnation.

REFERENCES

[1] Y. Notay, "Flexible Conjugate Gradients," *SIAM Journal on Scientific Computing*, Vol. 22, No. 4, 2000, pp. 1444-1460. doi:10.1137/S1064827599362314

[2] Y. Saad, "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific Computing*, Vol. 14, No. 2, 1993, pp. 461-469. doi:10.1137/0914028

[3] D. B. Szyld and J. A. Vogel, "FQMR: A Flexible Quasi-Minimal Residual Method with Inexact Preconditioning," *SIAM Journal on Scientific Computing*, Vol. 23, No. 2, 2001, pp. 363-380. doi:10.1137/S106482750037336X

[4] K. Abe and S.-L. Zhang, "A Variable Preconditioning Using the SOR Method For GCR-Like Methods," *International Journal of Numerical Analysis and Modeling*, Vol. 2, No.2, 2005, pp. 147-161.

[5] J. A. Vogel, "Flexible BiCG and Flexible Bi-CGSTAB for Nonsymmetric Linear Systems," *Applied Mathematics and Computation*, Vol. 188, No. 1, 2007, pp. 226-233. doi:10.1016/j.amc.2006.09.116

[6] S.-L. Zhang, "GPBi-CG: Generalized Product-Type Methods Baseed on Bi-CG for Solving Nonsymentric Linear Systems," *SIAM Journal on Scientific Computing*, Vol. 18, No. 2, 1997, pp. 537-551. doi:10.1137/S1064827592236313

[7] Y. Saad, "Iterative Method for Solving Linear Systems," 2nd Edition, SIAM, Philadelphia, 2003. doi:10.1137/1.9780898718003

