A variant of GPBiCG_AR method with reduction of computational costs

Moe Thuthu and Seiji Fujino
moethu@zeal.cc.kyushu-u.ac.jp
Graduate School of Information Science and Electrical Engineering,
Kyushu University
812-8581
Japan

Abstract

In numerical linear algebra, we have a number of iterative methods based on Krylov subspace method. GPBiCG_AR method which we have proposed is an attractive alternative for solving linear equations with nonsymmetric coefficient matrix. In this paper, we consider a variant of GPBiCG_AR method with reduction of computational costs per single iteration. We refer to it as GPBiCGAR_2 method. Through numerical experiments, we will verify improvement of convergence rate of the variant with safety convergence.

1 Introduction

Many iterative methods based on Krylov subspace method have been proposed. Some of these methods are Genaralized Product type Biconjugate Gradient (GPBiCG), Stabilizes Biconjugate Gradient (BiCGStab), BiCGsafe methods. Form these methods, GPBiCG (abbreviated as GP) method has inherently unstable convergence rate. Then we devised stable variant of GP method. We referred to it as GPBiCG_AR (abbreviated as AR) method [3]. This AR method gain stability of convergence. On the other hand, its convergence rate deteriotes slightly compared with the original GP. Therefore, we devised improved variant of AR method in view of convergence rate by means of alternative computation of parameter ζ_n and η_n. The improved variant of AR method is called as GPBiCGAR_2 method (abbreviated as AR_2). As a result we could resolve two kinds of issue of GP method, e.g. unstability and deterioration of convergence rate.

In this paper, we will discuss the AR_2 method. This method intends to reduce computation times and computational costs of the original AR method. The algorithm of original AR method is constructed based on minimization of the associate residual 2-norm for two acceleration parameters ζ_n, η_n. In the AR_2 method, we impose η_n is zero and ζ_n only is computed when iteration step is even, and both ζ_n and η_n are computed when iteration step is odd. As a result, we could reduce the computational cost per single iteration.

The remainder of this paper is organized as follows: In section 2, GP_2 and AR_2 methods with reduction of computational cost will be introduced. In section 3, we will discuss numerical results and present performance of AR and AR_2 methods. Finally, in section 4, we draw conclusions and future work.

2 GP_2 and AR_2 methods with reduction of computational cost

We consider iterative methods for solving a linear system of equations

$$A\boldsymbol{x} = \boldsymbol{b} \tag{2.1}$$

where $A \in R^{N\times N}$ is a given nonsymmetric matrix, and $\boldsymbol{x}$, $\boldsymbol{b}$ are a solution vector and right-hand side vector, respectively. When A is a large, sparse matrix which arises from realistic problems, the efficient solution of (2.1) is very difficult. This difficulty has led to the development of a rich variety of generalized Conjugate Gradient (CG) type methods having varying degrees of success (see, e.g., [5]).

The bi-conjugate gradient (BiCG) method based of the Lancoz algorithm is a crucial example of a generalized CG method. In many cases, the Lanczos algorithm give some of the fastest solution times and stability of convergence among all generalized CG methods. The Lanczos algorithm, however, is known to break down in some cases. In practice, the occurrence of breakdown can cause failure to irregularly converge to the solution of (2.1). The fact that the Lanczos algorithm perform well in some cases but fail in others heightens the need for further insight and development of the Lanczos type iterative methods. As a result, Zhang [6] proposed GP method. In GP method, acceleration parameters ζ_n and η_n are decided from the local minization of the residual vector of 2-norm $||\boldsymbol{r}_{n+1}||_2 = ||H_{n+1}(\lambda)R_{n+1}(\lambda)||_2$, where where $R_{n+1}(\lambda)$ denotes the residual polynomial of Lanczos algorithm and $H_{n+1}(\lambda)$ denotes the acceleration polynomial for convergence.

On the other hand, acceleration parameters ζ_n and η_n of original AR method are decided from the local minization of the residual vector of 2-norm $||\boldsymbol{a}\_\boldsymbol{r}_n||_2 = ||H_{n+1}(\lambda)R_n(\lambda)||_2$. The residual $\boldsymbol{a}\_\boldsymbol{r}_n$ is written as follows:

$$\boldsymbol{a}\_\boldsymbol{r}_n = \boldsymbol{r}_n - \eta_n A\boldsymbol{z}_{n-1} - \zeta_n A\boldsymbol{r}_n. \tag{2.2}$$

Here $\boldsymbol{r}_n$ is the residual vector and $\boldsymbol{z}_n$ is auxiliary vector. The acceleration parameters ζ_n and η_n of original AR method can be computed as follows:

$$\zeta = \frac{(\boldsymbol{b}_n, \boldsymbol{b}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{a}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{2.3}$$

$$\eta_n = \frac{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}, \tag{2.4}$$

where $\boldsymbol{a}_n = \boldsymbol{r}_n,\ \boldsymbol{b}_n = A\boldsymbol{z}_{n-1},\ \boldsymbol{c}_n = A\boldsymbol{r}_n$. Matrix-vector multiplication of $A\boldsymbol{r}_n$ is computed according to definition of matrix A and vector $\boldsymbol{r}_n$. On the other hand, $A\boldsymbol{z}_n$ is computed using its recurrence. This original AR method can get good convergence and show good performances (see, [3]).

In this paper, GP_2 and AR_2 methods were devised by alternative computation of parameters ζ_n and η_n of original GP and AR methods. In this approach, we set η_n to be zero when the even iteration step. For that reason, the associate residual a_r_n of AR_2 method at even iteration step become as follows:

$$\boldsymbol{a\_r}_n = \boldsymbol{r}_n - \zeta_n A\boldsymbol{r}_n. \tag{2.5}$$

As a result, we can reduce the amount of operations in one iteration. Similarly, algorithm of GP_2 method can be computed as algorithm of AR_2 method. The algoritm of AR_2 methd is shown as follows:

Algorithm 1 AR_2 method

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,
choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0^*,\ \boldsymbol{r}_0) \neq 0$,
set $\beta_{-1} = 0$, compute $A\boldsymbol{r}_0$,
for $n = 0, 1, \cdots$ until $||\boldsymbol{r}_{n+1}|| \leq \varepsilon\ ||\boldsymbol{r}_0||$ do :
begin
$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_{n-1}(\boldsymbol{p}_{n-1} - \boldsymbol{u}_{n-1})$,
$A\boldsymbol{p}_n = A\boldsymbol{r}_n + \beta_{n-1}(A\boldsymbol{p}_{n-1} - A\boldsymbol{u}_{n-1})$,
$\alpha_n = \dfrac{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)}{(\boldsymbol{r}_0^*, A\boldsymbol{p}_n)}$,
$\boldsymbol{a}_n = \boldsymbol{r}_n,\ \boldsymbol{b}_n = A\boldsymbol{z}_{n-1},\ \boldsymbol{c}_n = A\boldsymbol{r}_n$,
if mod $(n, 2) \neq 0$, then
$\zeta_n = \dfrac{(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)}, \eta_n = 0$,
$\boldsymbol{u}_n = \zeta_n A\boldsymbol{p}_n$,
compute $A\boldsymbol{u}_n$,
$\boldsymbol{t}_n = \boldsymbol{r}_n - \alpha_n A\boldsymbol{p}_n$,
$\boldsymbol{z}_n = \zeta_n t_n$,
$A\boldsymbol{z}_n = \zeta_n A\boldsymbol{r}_n - \alpha_n A\boldsymbol{u}_n$,
else
$\zeta_n = \dfrac{(\boldsymbol{b}_n, \boldsymbol{b}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{a}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}$,
$\eta_n = \dfrac{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{a}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{a}_n)}{(\boldsymbol{c}_n, \boldsymbol{c}_n)(\boldsymbol{b}_n, \boldsymbol{b}_n) - (\boldsymbol{b}_n, \boldsymbol{c}_n)(\boldsymbol{c}_n, \boldsymbol{b}_n)}$,
$\boldsymbol{u}_n = \zeta_n A\boldsymbol{p}_n + \eta_n(\boldsymbol{t}_{n-1} - \boldsymbol{r}_n + \beta_{n-1}\boldsymbol{u}_{n-1})$,
compute $A\boldsymbol{u}_n$,
$\boldsymbol{t}_n = \boldsymbol{r}_n - \alpha_n A\boldsymbol{p}_n$,
$\boldsymbol{z}_n = \zeta_n \boldsymbol{r}_n + \eta_n \boldsymbol{z}_{n-1} - \alpha_n \boldsymbol{u}_n$,
$A\boldsymbol{z}_n = \zeta_n A\boldsymbol{r}_n + \eta_n A\boldsymbol{z}_{n-1} - \alpha_n A\boldsymbol{u}_n$,
end if

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha_n \boldsymbol{p}_n + \boldsymbol{z}_n,$$
$$\boldsymbol{r}_{n+1} = \boldsymbol{t}_n - A\boldsymbol{z}_n,$$
compute $A\boldsymbol{r}_{n+1}$,
$$\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{(\boldsymbol{r}_0^*, \boldsymbol{r}_{n+1})}{(\boldsymbol{r}_0^*, \boldsymbol{r}_n)},$$
end

Table 1: Determination parameters η_n and ζ_n of GP, GP_2, AR and AR_2 methods.

Method	Residuals for minimization	Computation of parameters η_n and ζ_n
GP	$\|\|\boldsymbol{r}_{n+1}\|\|_2$	both η_n and ζ_n
GP_2	$\|\|\boldsymbol{r}_{n+1}\|\|_2$	at even iteration: only ζ_n and $\eta_n = 0$ at odd iteration: both η_n and ζ_n
AR	$\|\|\boldsymbol{a}\_\boldsymbol{r}_n\|\|_2$	both η_n and ζ_n
AR_2	$\|\|\boldsymbol{a}\_\boldsymbol{r}_n\|\|_2$	at even iteration: only ζ_n and $\eta_n = 0$ at odd iteration: both η_n and ζ_n

In Table 1, we present how to determine the acceleration parameters η_n and ζ_n of GP, GP_2, AR and AR_2 methods.

Table 2: Computational cost per single iteration of GP, GP_2, AR and AR_2 methods.

method	recurrence	inner product	$\|r_{n+1}\|$	total cost	$A\boldsymbol{v}$
GP	29	7	1	37 (1.03)	2
GP_2	23	4	1	28 (0.78)	2
AR	28	7	1	36 (1.00)	2
AR_2	22.5	4	1	27.5 (0.75)	2

In Table 2, computational costs per single iteration of CG,CG_2, AR and AR_2 methods were shown. "$A\boldsymbol{v}$" meand multiplication matrix A and vector $\boldsymbol{v}$. From this Table 2, we can see clearly that computational cost can be reduced by reducing operations. In this paper, GP, GP_2, AR and AR_2 methods were compared and discussed. In next section, we will discuss convergence properties of GP, GP_2, AR and AR_2 methods through some numerical experiments.

3 Numerical experiments

In this section numerical experiments will be discussed. All computations were done in double precision floating point arithmetics, and performed on HP workstation xw4200 with CPU of Intel(R) Pentium (R) 4, clock of 3.9GHz, main memory of 3GB, OS of Suse Linux version 9.2. Compile option with "-O0 " is used. 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^{-7} of the relative residual 2-norm $||\boldsymbol{r}_{n+1}||_2/||\boldsymbol{r}_0||_2$. The maximum number of iterations is fixed as 10^4. The initial shadow residual $\boldsymbol{r}_0^*$ is set as the initial residual $\boldsymbol{r}_0$ or uniform random number in $[0, 1]$. We examine stability of convergence of GP, GP_2, AR and AR_2 methods. Test matrices are taken from Florida sparse matrix collection[1]. The characteristics of some test matrices are listed in Table 3. In Table 3, "n" means number of dimensions, "nnz" means number of nonzero entries and "$ave.nnz$" means average of nonzero entries per one row.

Table 3: Characteristics of test matrices.

Matrix	n	nnz	$ave.\ nnz$
big	13,209	91,465	6.92
ns3Da	20,414	1,679,599	82.27
ck656	656	3,884	5.92
ex19	12,005	259,879	21.65
stomach	42,930	3,148,656	73.34
2D_bjtcai	27,628	442,898	16.03
af23560	23,560	484,256	20.55
sme3Da	12,504	874,887	69.97
sme3Db	29,067	2,081,063	71.60
epb3	84,617	463,625	5.48
3D_3D	51,448	1,056,610	20.54
ibm	51,448	1,056,610	20.54

In Table 4-7, "Itr." means number of iterations, "Time" means computational time in seconds and "Time ratio" means ratio of computational time to that of AR_2 method. Table 4 shows convergence of AR_2 method is superior to that of the other methods with the initial shadow residual ${r_0}^* = r_0$. Our reduction strategy of operations works very well.

From Table 4, the following observation can be made.

- Our reduction strategy of operations works very well. As a result, AR_2 method can converge with the least iterations and computational time.
- For matrix sme3Da, GP_2 method cannot converge until it reaches at the maximum number of iterations. On the other hand, AR_2 method can get splendid convergence.

After that, Table 5 shows convergence of AR method is superior to that of the other methods with the initial shadow residual ${r_0}^* = r_0$. Then, we can see that AR_2 method takes longer computational time than the original AR and GP methods. Moreover, GP_2

Table 4: Convergence of AR_2 method which is superior to that of the other methods with initial shadow residual ${r_0}^* = r_0$.

Matrix	Method	Itr.	Time [s]	Time Ratio
ex19	AR_2	2004	**13.01**	1.00
	AR	2218	14.82	1.14
	GP2	2639	17.43	1.34
	GP	2237	15.20	1.17
ns3Da	AR_2	675	**23.05**	1.00
	AR	767	26.37	1.14
	GP2	711	24.51	1.06
	GP	739	25.61	1.11
sme3Da	AR_2	3925	**69.52**	1.00
	AR	4028	72.15	1.04
	GP2	max	-	-
	GP	5410	97.91	1.41

Table 5: Convergence of AR method which is superior to that of the other methods with initial shadow residual ${r_0}^* = r_0$.

Matrix	Method	Itr.	Time [s]	Time Ratio
big	AR_2	3318	10.79	1.00
	AR	2235	**7.79**	0.72
	GP_2	4160	14.23	1.32
	GP	2528	9.16	0.85
2D_bjtcai	AR_2	4411	50.21	1.00
	AR	3818	**45.14**	0.90
	GP_2	6080	71.85	1.43
	GP	4058	49.84	0.99
epb3	AR_2	2937	55.17	1.00
	AR	2647	**53.33**	0.97
	GP_2	3048	61.29	1.11
	GP	2866	61.52	1.12

also needs longer computational time than the original AR and GP methods. Because, we impose $\eta_n = 0$ and ζ_n only is computed when iterations step is even in AR_2 and GP_2 methods. As a result, AR_2 and GP_2 methods needs many number of iterations numbers and longer computational time. From Table 5, the following observation can be made.

- We can see the effects of reducing operations technique with the initial shadow residual ${r_0}^*$ = random number. AR_2 method can converge well with the least iterations and computational time.
- Although AR_2, AR and GP_2 methods can converge well, the original GP method

breaks down for matrix 3D_3D.

- For matrix ck656, AR, AR_2 and GP_2 methods converge very fast. However, GP_2 method cannot converge until it reaches at the maximum number of iterations. On the other hand, AR_2 method can get nice convergence.

Table 6: Convergence of AR_2 method which is superior to that of the other methods with initial shadow residual $r_0{}^* =$ random number.

Matrix	Method	Itr.	Time [s]	Time Ratio
stomach	AR_2	146	**12.09**	1.00
	AR	175	15.11	1.25
	GP_2	162	13.98	1.16
	GP	152	13.69	1.13
af23560	AR_2	1798	**21.45**	1.00
	AR	1828	22.52	1.05
	GP_2	1923	23.55	1.10
	GP	1809	22.92	1.07
3D_3D	AR_2	6472	**172.31**	1.00
	AR	6572	180.13	1.05
	GP_2	6509	178.51	1.04
	GP	break	-	-
ck656	AR_2	287	**0.04**	1.00
	AR	295	0.05	1.25
	GP_2	289	0.04	1.00
	GP	max	-	-

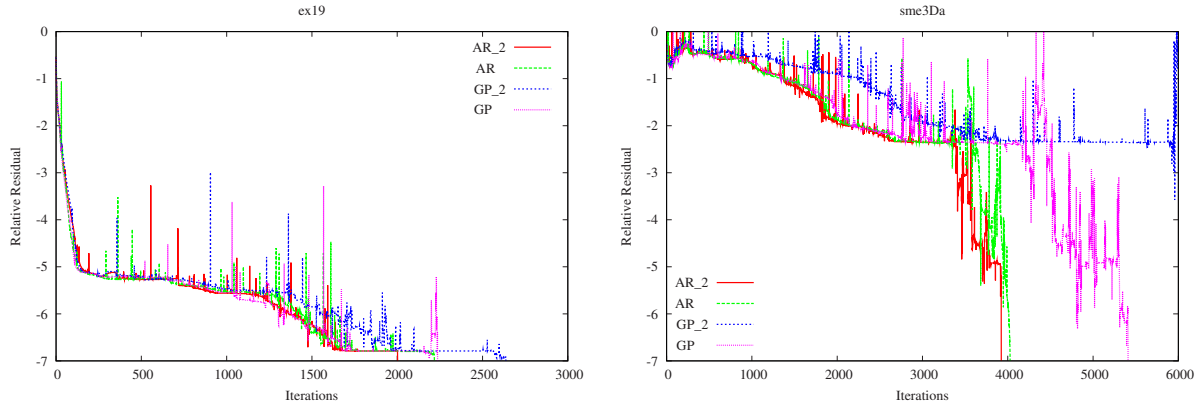
Table 6 shows that convergence of AR_2 method is superior to that of the other methods with the initial shadow residual $r_0{}^* =$ random number. Table 7 also presents that convergence of AR method is superior to that of the other methods with the initial shadow residual $r_0{}^* =$ random number. Even though AR_2 method needs longer computational time than AR method, it can converge well. For matrix sme3Da, GP_2 method cannot get convergence until it reaches at maximum number of iterations of 10^4.

In Fig.1 we demonstrate history of relative residual 2-norm of AR methods and GP methods for two matrices (a)ex19 and (b)sme3Da when the initial shadow residual $r_0^* = r_0$. In Fig.1(a), all methods perform well. AR_2 method (red solid line) shows excellent convergence. As a result, we can see clearly the effects of reducing operations. In Fig.1(b), AR_2 method converges with less number of iterations than that of the original AR method (green dotted line). On the other hand, GP_2 method (blue dotted line) stagnates at the residual level of approximate 10^{-2}.

Fig.2 exhibits history of relative residual of AR methods and GP methods for matrices epb3 and big when the initial shadow residual $r_0^* = r_0$. We can see that original AR method superior to AR_2 method. Because, AR_2 and GP_2 methods need many number of iterations in order to solve linear systems.

Table 7: Convergence of AR method which is superior to that of the other methods with initial shadow residual ${r_0}^* =$ random number.

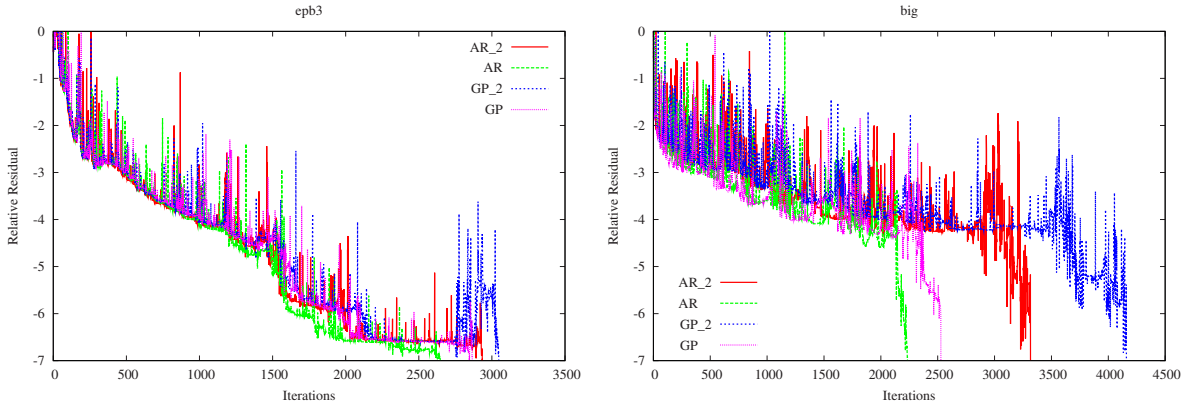
Matrix	Method	Itr.	Time [s]	Time Ratio
2D_bjtcai	AR_2	3796	43.18	1.00
	AR	3074	**36.39**	0.84
	GP_2	3412	40.32	0.93
	GP	3145	38.64	0.89
ibm	AR_2	5892	156.65	1.00
	AR	5144	**141.23**	0.90
	GP_2	6271	172.00	1.10
	GP	5103	144.40	0.92
sme3Da	AR_2	6799	120.42	1.00
	AR	3496	**62.79**	0.52
	GP_2	max	_	_
	GP	4621	83.57	0.69
big	AR_2	2903	9.44	1.00
	AR	1569	**5.48**	0.58
	GP_2	2988	10.22	1.08
	GP	1679	6.11	0.65



(a) matrix: ex19 (b) matrix: sme3Da

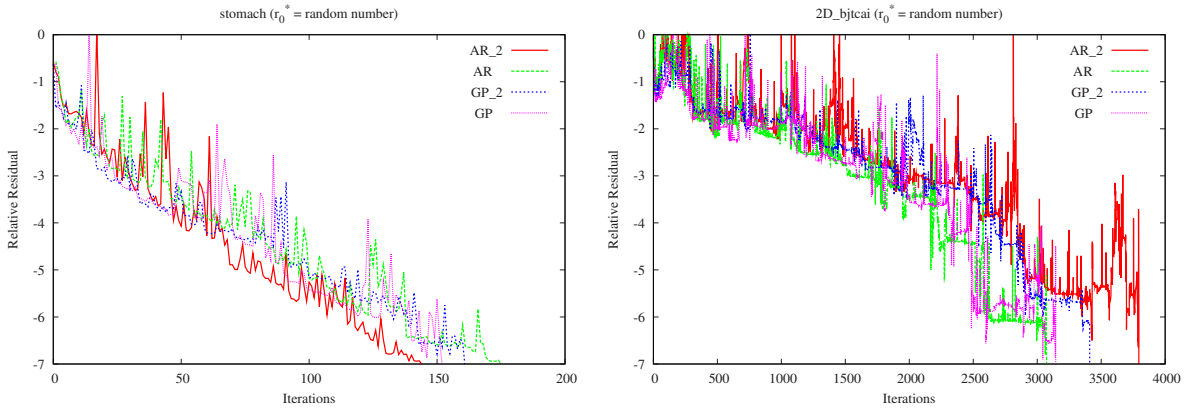
Figure 1: History of relative residual of AR methods and GP methods for matrices ex19 and sme3Da when initial shadow residual $r_0^* = r_0$.

Fig.3 presents history of relative residual of AR methods and GP methods for matrices stomach and 2D_bjtcai when initial shadow residual r_0^* is random number. In Fig.3(a) all methods perform very well. Moreover, AR_2 method converge with the least number of iterations. In Fig.3(b), we see that, though all methods oscillate violently, they can converge. We also understand that the original AR method is superior to AR_2 method.



(a) matrix: epb3 (b) matrix: big

Figure 2: History of relative residual of AR methods and GP methods for matrices epb3 and big when initial shadow residual $r_0^* = r_0$.



(a) matrix: stomach ($r_0^* = \mathrm{rand}$) (b) matrix: 2D_bjtcai($r_0^* = \mathrm{rand}$)

Figure 3: History of relative residual of AR methods and GP methods for matrices stomach and 2D_bjtcai when initial shadow residual r_0^* is random number.

4 Conclusions

In this paper, we proposed a variant of GPBiCG_AR method with reduction of computational costs per single iteration. Through some numerical experiments, we examined effectiveness of the variant of GPBiCG_AR method. As a future work, we will study robustness of the variant.

References

[1] T. Davis' sparse matrix collection of Florida University: http://www.cise.ufl.edu/research/sparse/matrices/

[2] S. Fujino, M. Fujiwara and M. Yoshida: BiCGSafe method based on minimization of associate residual, JSCES 2005. http://save.k.u-tokyo.ac.jp /jsces/trans/trans2005/No20050028.pdf. (in Japanese)

[3] Moe Thuthu, S. Fujino: Stability of GPBiCG_AR method based on minimization of associate residual, ASCM, **5081**(2008), 108-120.

[4] 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., **13**(1992), 631-644.

[5] H.A. van der Vorst: Iterative Krylov preconditionings for large linear systems, Cambridge University Press, Cambridge, 2003.

[6] S.-L. Zhang: GPBi-CG: Generalized product-type preconditionings based on Bi-CG for solving nonsymmetric linear systems, SIAM J. Sci. Comput., **18**(1997), 537-551.