Variants of BiCGSafe and BiCGStab methods stemmed from consideration on product of two kinds of polynomials

Seiji Fujino*, Takashi Sekimoto**
*Research Institute for Information Technology, Kyushu University
** Ryoyu Systems Co. Ltd.

Abstract: A numer of iterative methods based on Lanczos polynomial added with auxiliary polynomial were proposed independently after appearance of CGS and BiCGStab methods. Then strategy of combination of two polynomials was generalized as a form of product of two polynomials in 1997. However, the optimization of product of polynomials remains as an open problem. The first solution among neive realization of product-type iterative methods was made partly by Fujino *et al*. in 2005 owing to 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 refered 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 and GPBiCGSafe methods in 2009 and 2011 one after another.

In this article, we propose both variant of BiCGSafe method and variant of BiCGStab method stemmed from consideration on product of polynomials. We present performance of some GPBiCG type and BiCGSafe type methods with ILU(0) preconditioning for non-symmetric matrices Freescale1 and bcircuit derived from Florida Sparse Matrix Collection in Table 1. The stopping criterion is relative residual 2-norm of $||\boldsymbol{r}_{k+1}||_2/||\boldsymbol{r}_0||_2 \leq 10^{-12}$. "TRR" means the True Relative Residual for the approximate solution. "pre-t.", "itr-t." and "tot-t." mean time of preconditioning, that of iterations and total time in seconds, respectively. The bold figures mean the fastest time for each matrix. From the result shown in Table 1, we can see that the proposed **variant_1** and **variant_2 of BiCGSafe method** outperform well compared with GPBiCG type methods from the viewpoint of cpu-time and safety convergence.

Table 1: Performance of some GPBiCG type and BiCGSafe type methods with ILU(0) preconditioning without extra fi ll-ins.

(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**