# An improvement in preconditioned BiCGStab method

Shoji Itoh,[†1] Takahiro Katagiri,[†1] Takao Sakurai,[†2]
Mitsuyoshi Igai,[†3] Satoshi Ohshima,[†1] Hisayasu Kuroda,[†4]
Ken Naono[†2] and Kengo Nakajima[†1]

In order to solve linear equations

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

preconditioned BiCGStab (PBiCGStab) method[3] is often used. PBiCGStab brings well convergence, further, its calculation cost is not a lot and its required memory size is a little. However, conventional representation of PBiCGStab has a mathematical inadvisability on its preconditioning conversion for the shadow residual vector $\boldsymbol{r}_0^\sharp$.

A preconditioned system of eq.(1) is

$$(AK^{-1})(K\boldsymbol{x}) = \boldsymbol{b}, \quad (2)$$

this residual vector is $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$. Here, $K$ ($\approx A$) is a preconditioner. The preconditioned dual system is $(K^{-T}A^T)\boldsymbol{x}^\sharp = K^{-T}\boldsymbol{b}^\sharp$, this (shadow) residual vector is $K^{-T}\boldsymbol{r}^\sharp = K^{-T}\boldsymbol{b}^\sharp - (K^{-T}A^T)\boldsymbol{x}^\sharp$. That is, this preconditioning conversion for shadow residual vector is

$$\tilde{\boldsymbol{r}}_0^\sharp \Rightarrow K^{-T}\boldsymbol{r}_0^\sharp, \quad (3)$$

but conventional typical algorithm[3] is based on

$$\tilde{\boldsymbol{r}}_0^\sharp \Rightarrow \boldsymbol{r}_0^\sharp. \quad (4)$$

Eq.(3) is proper in mathematical, this is common characteristic to Lanczos biorthogonalization type algorithms[2].

Conventional preconditioning conversion for eq.(2) is as follows:

$$\tilde{\boldsymbol{p}} \Rightarrow \boldsymbol{p},\ \tilde{\boldsymbol{t}} \Rightarrow \boldsymbol{t},\ \tilde{\boldsymbol{r}} \Rightarrow \boldsymbol{r},\ \tilde{\boldsymbol{r}}_0^\sharp \Rightarrow \boldsymbol{r}_0^\sharp.$$

On the other hand, our conversion manner for eq.(2) is,

$$\tilde{\boldsymbol{p}} \Rightarrow K\boldsymbol{p},\ \tilde{\boldsymbol{t}} \Rightarrow \boldsymbol{t},$$
$$\tilde{\boldsymbol{z}} \Rightarrow K\boldsymbol{z},\ \tilde{\boldsymbol{r}} \Rightarrow \boldsymbol{r},\ \tilde{\boldsymbol{r}}_0^\sharp \Rightarrow K^{-T}\boldsymbol{r}_0^\sharp,$$

then the following PBiCGStab is obtained.

Algorithm 1. An improved preconditioned BiCGStab method:
$\boldsymbol{x}_0$, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, $\boldsymbol{r}_0^\sharp = K^{-1}\boldsymbol{r}_0$, $\beta_{-1} = 0$,
For $k = 0, 1, 2, \cdots$, until convergence, Do:

$$\boldsymbol{p}_k = K^{-1}\boldsymbol{r}_k + \beta_{k-1}\boldsymbol{z}_{k-1}, \quad (5)$$

$$\alpha_k = \frac{\langle \boldsymbol{r}_0^\sharp, K^{-1}\boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\sharp, K^{-1}A\boldsymbol{p}_k \rangle}, \quad (6)$$

$$\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k,$$
$$K^{-1}\boldsymbol{t}_k = K^{-1}\boldsymbol{r}_k - \alpha_k K^{-1}A\boldsymbol{p}_k, \quad (7)$$

$$\zeta_k = \frac{(AK^{-1}\boldsymbol{t}_k, \boldsymbol{t}_k)}{(AK^{-1}\boldsymbol{t}_k, AK^{-1}\boldsymbol{t}_k)},$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k + \zeta_k K^{-1}\boldsymbol{t}_k,$$
$$\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \zeta_k AK^{-1}\boldsymbol{t}_k,$$
$$\boldsymbol{z}_k = \boldsymbol{p}_k - \zeta_k K^{-1}A\boldsymbol{p}_k, \quad (8)$$
$$\beta_k = \frac{\alpha_k}{\zeta_k} \times \frac{\langle \boldsymbol{r}_0^\sharp, K^{-1}\boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\sharp, K^{-1}\boldsymbol{r}_k \rangle}, \quad (9)$$

End Do

Here, $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ means "scalar product without metric", and $(\boldsymbol{u}, \boldsymbol{v})$ means "inner product with metric".

Alg.1 has two technical interesting items on preconditioning conversion. First, there are three preconditioning operations in alg.1, $K^{-1}\boldsymbol{t}_k$, $K^{-1}\boldsymbol{r}_k$, $K^{-1}A\boldsymbol{p}_k$. But calculating cost of $K^{-1}\boldsymbol{t}_k$ is almost nothing, because $K^{-1}\boldsymbol{t}_k$ is composed of $K^{-1}\boldsymbol{r}_k$ and $K^{-1}A\boldsymbol{p}_k$ by using eq.(7), these latter operations are already calculated by eqs.(9)(6), respectively. Second, eq.(5) is separated in two eqs. (5)(8) in alg.1.

Numerical results by systematic performance evaluation[1] shows advantage of our algorithm. We will show them at the HPCS2011. Our PBiCGStab algorithm will be implemented in Xabclib[4].

**Acknowledgments** This work is partially supported by the grant of "Seamless and Highly-Productive Parallel Programming Environment for High-Performance Computing", High Performance Library Sub Project (2008 – 2011), from MEXT, Japan, and Grant-in-Aid for Scientific Research, 21300017, 21300007.

## References

1) Itoh, S. and Sugihara, M.: Systematic performance evaluation of linear solvers using quality control techniques, Software Automatic Tuning (eds. Naono, K., Teranishi, K., Cavazos, J. and Suda, R.), pp. 135–152, Springer, 2010.
2) Itoh, S. and Sugihara, M.: Preconditioning conversion for Bi-Lanczos based algorithms to solve linear equations, JSIAM Annualmeeting, Meiji Univ., Tokyo, Sep. 2010 (in Japanese).
3) Van der Vorst, Henk 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(2), pp. 631–644, 1992.
4) Xabclib project: http://www.abc-lib.org/Xabclib

†1 The University of Tokyo
†2 Central Research Laboratory, Hitachi, Ltd.
†3 Hitachi ULSI Systems Co., Ltd.
†4 Ehime University