JSIAM Letters Vol.7 (2015) pp.37–40 ©2015 Japan Society for Industrial and Applied Mathematics

Improving the convergence behaviour of BiCGSTAB by applying D-norm minimization

Lijiong Su[1], Akira Imakura[1], Hiroto Tadano[1] and Tetsuya Sakurai[1,2]

[1] Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
[2] CREST, JST, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan

E-mail *shinta@mma.cs.tsukuba.ac.jp*

Received September 30, 2014, Accepted February 26, 2015

Abstract

In this article, we deal with the iterative methods for solving unsymmetric linear systems, especially BiCGSTAB. The introduced parameter in BiCGSTAB at each iteration is selected to minimize the 2-norm of the residual vector. Here, we suggest another way to select the parameter by the idea of weighting used in Weighted GMRES. By our procedure, more importance is assigned to the larger entry of the residual vector so that faster convergence can be expected. In numerical experiments, it is shown that our procedure is efficient compared with the original BiCGSTAB.

Keywords linear system, Krylov subspace method, BiCGSTAB, D-norm

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

We are concerned with the numerical solution of the unsymmetric linear system

$$A\boldsymbol{x} = \boldsymbol{b},$$

where $A \in \mathbb{R}^{n\times n}$ is the coefficient matrix, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^n$ are the unknown vector and the right-hand side vector respectively.

Iterative methods based on the Krylov subspace, normally called Krylov subspace methods, are very popular for large and sparse linear systems which arise in real-life applications [1, 2]. In this article, we will focus on BiCGSTAB [3], which is one of the efficient Krylov subspace methods, for solving the unsymmetric linear systems. BiCGSTAB is the product type method based on BiCG [4], where the polynomial of GMRES(1) is used as the so-called stabilization polynomial.

The algorithm of BiCGSTAB consists of a BiCG part and a MR part. Some techniques have been proposed for improving the MR part of BiCGSTAB. One strategy is to improve the stability of the inner product in the BiCG part [5]. Another strategy is to accelerate the convergence of the MR part like BiCGSTAB(ℓ) [6]. In order to improve the convergence of the MR part, BiCGSTAB(ℓ) uses the polynomial of GMRES(ℓ) as the stabilization polynomial instead of the polynomial of GMRES(1). Using more efficient polynomial, BiCGSTAB(ℓ) achieves better performance than BiCGSTAB. In this article, we try to improve the convergence of the MR part in a different way from BiCGSTAB(ℓ) by applying the idea proposed in Weighted GMRES [7].

In Section 2 and Section 3, we give a summary of BiCGSTAB and Weighted GMRES. Our idea is explained in Section 4 and a corresponding algorithm is also given. The numerical examples are shown in Section 5, and finally the conclusion is given in Section 6.

2. BiCGSTAB

Let $\boldsymbol{r} \in \mathbb{R}^n$, the Krylov subspace is defined as

$$K_i(A, \boldsymbol{r}) = \mathrm{span}\{\boldsymbol{r}, A\boldsymbol{r}, A^2\boldsymbol{r}, \dots, A^{i-1}\boldsymbol{r}\}.$$

Krylov subspace methods build up the Krylov subspaces and look for a good approximation within the Krylov subspaces.

BiCG has been proposed by Fletcher for unsymmetric linear system, and the basic idea was originally used by Lanczos [8] to compute eigenvalues of the symmetric matrix. It constructs *bi-orthogonal* bases for the Krylov subspaces corresponding to A and A^{T} by the *two-sided Lanczos procedure* to approximate the solution of the linear system. BiCG requires multiplication by both A and A^{T} on every iteration. In some applications, we only have matrix-vector product of A as a function. In that case BiCG is not applicable.

Let $\boldsymbol{x}_0$ be the initial guess, and $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. The approximate solution of the Krylov subspace method can be written as

$$\boldsymbol{x}_i = \boldsymbol{x}_0 + \boldsymbol{z}_i,$$

where $\boldsymbol{z}_i \in K_i(A, \boldsymbol{r}_0)$. The corresponding residual vector can be written as

$$\boldsymbol{r}_i = \boldsymbol{b} - A\boldsymbol{x}_i = \boldsymbol{r}_0 - A\boldsymbol{z}_i \in K_{i+1}(A, \boldsymbol{r}_0).$$

This means that we can consider the residual vector of Krylov subspace methods as the product of a polynomial of A and the initial residual vector $\boldsymbol{r}_0$.

For BiCG, we can define the residual vector as

$$\boldsymbol{r}_i^{\mathrm{BiCG}} = \varphi_i(A)\boldsymbol{r}_0,$$

Algorithm 1 BiCGSTAB

```
Choose an initial guess $\boldsymbol{x}_0$; $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
Choose $\widetilde{\boldsymbol{r}}$, for example, $\widetilde{\boldsymbol{r}} = \boldsymbol{r}_0$
for $i = 1, 2, \ldots$ do
    $\rho_{i-1} = (\widetilde{\boldsymbol{r}}, \boldsymbol{r}_{i-1})$
    if $\rho_{i-1} = 0$ then
        method fails
    end if
    if $i = 1$ then
        $\boldsymbol{p}_i = \boldsymbol{r}_{i-1}$
    else
        $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
        $\boldsymbol{p}_i = \boldsymbol{r}_{i-1} + \beta_{i-1}(\boldsymbol{p}_{i-1} - \omega_{i-1}\boldsymbol{v}_{i-1})$
    end if
    $\boldsymbol{v}_i = A\boldsymbol{p}_i$
    $\alpha_i = \rho_{i-1}/(\widetilde{\boldsymbol{r}}, \boldsymbol{v}_i)$
    $\boldsymbol{s} = \boldsymbol{r}_{i-1} - \alpha_i\boldsymbol{v}_i$
    If $\|\boldsymbol{s}\|_2$ is small enough: $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_i\boldsymbol{p}_i$ and stop
    $\boldsymbol{t} = A\boldsymbol{s}$
    $\omega_i = (\boldsymbol{t}, \boldsymbol{s})/(\boldsymbol{t}, \boldsymbol{t})$
    if $\omega_i = 0$ then
        method fails
    end if
    $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_i\boldsymbol{p}_i + \omega_i\boldsymbol{s}$
    $\boldsymbol{r}_i = \boldsymbol{s} - \omega_i\boldsymbol{t}$
    Check convergence, continue if necessary
end for
```

where $\varphi_i(A)$ is a polynomial of A of degree i. In [3], van der Vorst redefined the residual vector $\boldsymbol{r}_i^{\mathrm{BiCGSTAB}}$ as

$$\boldsymbol{r}_i^{\mathrm{BiCGSTAB}} = Q_i(A)\boldsymbol{r}_i^{\mathrm{BiCG}} = Q_i(A)\varphi_i(A)\boldsymbol{r}_0, \quad (1)$$

where $Q_i(A)$ is a polynomial of degree i, and defined as

$$Q_i(A) = (I - \omega_i A)Q_{i-1}(A), \quad Q_0(A) = I,$$

where I is the identity matrix. The parameter ω_i is selected to minimize the 2-norm of the residual $\boldsymbol{r}_i^{\mathrm{BiCGSTAB}}$

$$\omega_i = \arg\min_{\omega\in\mathbb{R}} \|(I - \omega A)Q_{i-1}(A)\varphi_i(A)\boldsymbol{r}_0\|_2. \quad (2)$$

By this way of defining, an algorithm called BiCGSTAB (given in Algorithm 1) can be derived. It consists of the BiCG part for updating $\boldsymbol{r}_i^{\mathrm{BiCG}}$, and the MR part for minimizing the 2-norm of $\boldsymbol{r}_i^{\mathrm{BiCGSTAB}}$. BiCGSTAB is expected to have more stable convergence behaviour than BiCG. Also there is an advantage that it does not need the operation of the transpose of the matrix. For the detailed derivation of BiCGSTAB, as well as BiCG and the basic Krylov subspace theorems, one can refer to a comprehensive book by Saad [9].

3. Weighted GMRES

Minimizing the 2-norm of the residual in BiCGSTAB may be considered as the most natural way. However, it is not always the best way for the fast convergence. By making use of the weighting technique used in Weighted GMRES, it is possible to gain faster convergence for BiCGSTAB. To explain our idea, first we give a brief description of Weighted GMRES.

Algorithm 2 GMRES(m)

```
Choose an initial guess $\boldsymbol{x}_0$, let $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, $\beta = \|\boldsymbol{r}_0\|_2$, and $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$
Construct the basis of Krylov subspace $V_m$ by the Arnoldi procedure starting with $\boldsymbol{v}_1$
Define the Henssenberg matrix $H_{(m+1)\times m}$
Compute $\boldsymbol{y}$ by minimizing $\|\beta\boldsymbol{e}_1 - H_i\boldsymbol{y}\|_2$
Obtain the approximate solution $\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m\boldsymbol{y}$
Check convergence, restart if necessary: set $\boldsymbol{x}_0 = \boldsymbol{x}_m$ and $\boldsymbol{r}_0 = \boldsymbol{r}_m$
```

Algorithm 3 Weighted GMRES

```
Choose an initial guess $\boldsymbol{x}_0$, let $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
Choose the vector $\boldsymbol{d}$ such that $\|\boldsymbol{d}\|_2 = \sqrt{n}$, let $D = \mathrm{diag}(\boldsymbol{d})$, set $\widetilde{\beta} = \|\boldsymbol{r}_0\|_D$ and $\widetilde{\boldsymbol{v}}_1 = \boldsymbol{r}_0/\widetilde{\beta}$
Construct the $D$-orthonormal basis $\widetilde{V}_m$ by the weighted Arnoldi's procedure starting with $\widetilde{\boldsymbol{v}}_1$
Define the Henssenberg matrix $\widetilde{H}_{(m+1)\times m}$
Compute $\boldsymbol{y}$ by minimizing $\|\widetilde{\beta}\boldsymbol{e}_1 - \widetilde{H}_m\boldsymbol{y}\|_2$
Obtain the approximate solution $\boldsymbol{x}_m = \boldsymbol{x}_0 + \widetilde{V}_m\boldsymbol{y}$
Check convergence, restart if necessary: set $\boldsymbol{x}_0 = \boldsymbol{x}_m$ and $\boldsymbol{r}_0 = \boldsymbol{r}_m$
```

Definition 1 *Let $\boldsymbol{u} = [u_1, u_2, \ldots, u_n]^{\mathrm{T}}$ and $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]^{\mathrm{T}}$ be two vectors in $\mathbb{R}^n$, and d_i be a positive scalar. Then the D-scalar of $\boldsymbol{u}$ and $\boldsymbol{v}$ is*

$$(\boldsymbol{u}, \boldsymbol{v})_D \equiv \boldsymbol{v}^{\mathrm{T}}D\boldsymbol{u} = \sum_{i=1}^{n} d_i u_i v_i,$$

where d_i is the diagonal entry of the diagonal matrix

$$D = \mathrm{diag}(d_1, d_2, \ldots, d_n). \quad (3)$$

The D-norm is defined associated with the inner product as

$$\|\boldsymbol{u}\|_D \equiv \sqrt{(\boldsymbol{u}, \boldsymbol{u})_D} = \sqrt{\boldsymbol{u}^T D\boldsymbol{u}} = \sqrt{\sum_{i=1}^{n} d_i u_i^2}.$$

Notice that if we let $D = I$, then the D-norm is equivalent with the 2-norm:

$$\|\boldsymbol{u}\|_D \xrightarrow{D=I} \sqrt{\boldsymbol{u}^{\mathrm{T}}I\boldsymbol{u}} = \|\boldsymbol{u}\|_2.$$

The D-norm defined above was applied by Essai [7] to improve the convergence of GMRES [10], whose algorithm is called Weighted GMRES.

GMRES constructs an orthonormal basis $V_m \in \mathbb{R}^{n\times m}$ and a Hessenberg matrix $H_m \in \mathbb{R}^{(m+1)\times m}$ by the *Arnoldi procedure* for the Krylov subspace. The approximate solution is obtained by minimizing the 2-norm of the residual vector:

$$\min_{\boldsymbol{y}\in\mathbb{R}^m} \|\boldsymbol{r}_0 - AV_m\boldsymbol{y}\|_2 = \min_{\boldsymbol{y}\in\mathbb{R}^m} \|\beta\boldsymbol{e}_1 - H_m\boldsymbol{y}\|_2.$$

A restart version of GMRES can be summarized as Algorithm 2.

While in GMRES the 2-norm of the residual vector is minimized, Weighted GMRES minimizes the D-norm of the residual over the Krylov subspace at every restart.

Let $\widetilde{V}_m$ be a D-orthonormal basis and $\widetilde{H}_m$ the corresponding Hessenberg matrix. Then the approximate solution of Weighted GMRES can be computed as

$$\min_{\boldsymbol{y}\in\mathbb{R}^m} \|\boldsymbol{r}_0 - A\widetilde{V}_m\boldsymbol{y}\|_D = \min_{\boldsymbol{y}\in\mathbb{R}^m} \|\widetilde{\beta}\boldsymbol{e}_1 - \widetilde{H}_m\boldsymbol{y}\|_2.$$

By minimizing the D-norm, different entries of the residual vector will get different emphasis. The algorithm of Weighted GMRES can be summarized as Algorithm 3.

4. Applying D-norm minimization

As shown in Section 3, the basic idea of Weighted GMRES is to use D-norm minimization instead of 2-norm minimization. By changing the weight matrix D (3) corresponding to the current residual in each restart, Weighted GMRES shows better convergence behaviour than GMRES(m) [7].

In BiCGSTAB, the 2-norm of the residual is minimized at every iteration. Based on the idea of Weighted GMRES, we propose minimizing the D-norm of the residual in BiCGSTAB instead of 2-norm. By changing the weight matrix D in each iteration corresponding to the current residual similar as Weighted GMRES, we expect that overall convergence behaviour of BiCGSTAB will be improved.

To minimize the D-norm of the residual, first we rewrite (1) as

$$\boldsymbol{r}_i^{\mathrm{BiCGSTAB}} = \boldsymbol{s} - \omega_i\boldsymbol{t},$$

where $\boldsymbol{s} = Q_{i-1}(A)\varphi_i(A)\boldsymbol{r}_0$ and $\boldsymbol{t} = A\boldsymbol{s}$. In order to minimize the D-norm of the residual $\boldsymbol{r}_i^{\mathrm{BiCGSTAB}}$, we should let the parameter ω_i to be

$$\omega_i = \arg\min_{\omega\in\mathbb{R}} \|\boldsymbol{s} - \omega\boldsymbol{t}\|_D = \frac{(\boldsymbol{t},\boldsymbol{s})_D}{(\boldsymbol{t},\boldsymbol{t})_D}. \tag{4}$$

We give the algorithm of BiCGSTAB applied with D-norm minimization of the residual vector in Algorithm 4. Notice that the matrix D can be changed at every iteration.

For the diagonal entries of the matrix D, there is no definite rule on choosing. One possibility is to let

$$D = \mathrm{diag}(\boldsymbol{d}), \quad \boldsymbol{d} = \sqrt{n}\,\frac{|\boldsymbol{r}_i|}{\|\boldsymbol{r}_i\|_2}, \tag{5}$$

where $|\boldsymbol{r}_i| \equiv [|\boldsymbol{r}_i(1)|, |\boldsymbol{r}_i(2)|, \dots, |\boldsymbol{r}_i(n)|]^{\mathrm{T}}$. This is implemented in Weighted GMRES, which realizes the idea of giving greater weight on larger entry. We will also accept this kind of definition in our numerical experiment.

The storage and computational cost required for implementing our algorithm does not increase much compared with the original BiCGSTAB algorithm. It needs additional cost of determining matrix D, one more vector to store the diagonal entries of D and two more pointwise vector multiplications to compute the parameter ω_i.

Notice that if the parameter ω_i becomes zero during the iteration, then BiCGSTAB fails to converge. And even if it does not become zero exactly, it also causes a numerical instability when it is close to zero because of the finite precision calculation. So it is applicable to compute ω_i by (4), when (2) produces zero or small value.

Algorithm 4 BiCGSTAB with D-norm minimization

```
Choose an initial guess $\boldsymbol{x}_0$; $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
Choose $\widetilde{\boldsymbol{r}}$, for example, $\widetilde{\boldsymbol{r}} = \boldsymbol{r}_0$
for $i = 1, 2, \dots$ do
    $\rho_{i-1} = (\widetilde{\boldsymbol{r}}, \boldsymbol{r}_{i-1})$
    if $\rho_{i-1} = 0$ then
        method fails
    end if
    if $i = 1$ then
        $\boldsymbol{p}_i = \boldsymbol{r}_{i-1}$
    else
        $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
        $\boldsymbol{p}_i = \boldsymbol{r}_{i-1} + \beta_{i-1}(\boldsymbol{p}_{i-1} - \omega_{i-1}\boldsymbol{v}_{i-1})$
    end if
    $\boldsymbol{v}_i = A\boldsymbol{p}_i$
    $\alpha_i = \rho_{i-1}/(\widetilde{\boldsymbol{r}}, \boldsymbol{v}_i)$
    $\boldsymbol{s} = \boldsymbol{r}_{i-1} - \alpha_i\boldsymbol{v}_i$
    If $\|\boldsymbol{s}\|_2$ is small enough: $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_i\boldsymbol{p}_i$ and stop
    $\boldsymbol{t} = A\boldsymbol{s}$
    Determine $\boldsymbol{d}$, and let $D = \mathrm{diag}(\boldsymbol{d})$
    $\omega_i = (\boldsymbol{t}, \boldsymbol{s})_D/(\boldsymbol{t}, \boldsymbol{t})_D$
    if $\omega_i = 0$ then
        method fails
    end if
    $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_i\boldsymbol{p}_i + \omega_i\boldsymbol{s}$
    $\boldsymbol{r}_i = \boldsymbol{s} - \omega_i\boldsymbol{t}$
    Check convergence, continue if necessary
end for
```

5. Numerical experiment

In this section we show the results of some numerical examples solved by our algorithm (Algorithm 4) and the BiCGSTAB algorithm (Algorithm 1) without preconditioners. Since our chief concern is the convergence behaviour, the CPU time is not shown. The additional computational cost of our algorithm is negligible compared with the original one.

All the tested real and unsymmetric matrices are from The University of Florida Sparse Matrix Collection [11]. The initial guess $\boldsymbol{x}_0$ is chosen to be the zero vector, the vector $\widetilde{\boldsymbol{r}} = \boldsymbol{r}_0$, and the right-hand side vector is set as $\boldsymbol{b} = [1, 1, \dots, 1]^{\mathrm{T}}$. The stopping criterion is chosen to be $\|\boldsymbol{r}_i\|_2/\|\boldsymbol{b}\|_2 \le 10^{-10}$. The matrix D is determined by (5), and all the computations were performed in MATLAB 2013b.

Fig. 1 illustrates the histories of the relative residual of the matrix ‘cavity01’. The straight line represents the result computed by our algorithm, the star line the original BiCGSTAB. We can see that our algorithm computes the residual more steadily than BiCGSTAB which fluctuates during the iteration. Fig. 2 shows the result of the matrix ‘dw2048’. Again our algorithm needs less iterations to converge, though the difference is not so obvious in contrast to the previous problem.

We give the computation results of all the tested matrices in Table 1. From the table, we can see that for some matrices, BiCGSTAB fails to converge to the desired accuracy while our algorithm obtains the approximate solutions. The efficiency of D-norm is not guaran-

Table 1. Test Results. 'Iter': iterations; 'RR': relative residual; 'TRR': true relative residual $\|\boldsymbol{b} - A\boldsymbol{x}_i\|_2/\|\boldsymbol{b}\|_2$; '-': not converge.

Matrix	n	nonzeros	kind	Our algorithm			BiCGSTAB		
				Iter	RR	TRR	Iter	RR	TRR
cage6	93	785	directed weighted graph	30	9.1e-11	9.1e-11	-	2.0e-04	2.0e-04
cavity01	317	7,280	computational fluid dynamics problem	307	3.0e-11	3.0e-11	471	9.9e-11	9.9e-11
circuit_1	2,624	35,823	circuit simulation problem	863	9.6e-11	9.6e-11	1,757	7.7e-11	7.7e-11
ck400	400	2,860	2D/3D problem	483	8.7e-11	8.7e-11	476	6.2e-11	6.2e-11
dw2048	2,048	10,114	electromagnetics problem	1,873	9.5e-11	9.5e-11	2,065	9.5e-11	9.5e-11
gre_185	185	975	directed weighted graph	571	8.8e-11	8.7e-11	-	5.2e-02	5.2e-02
Pd	8,081	13,036	counter-example problem	189	3.9e-11	1.2e-08	-	5.6e-07	5.6e-07
poisson3Db	85,623	2,374,949	computational fluid dynamics problem	243	8.2e-11	8.2e-11	284	9.1e-11	9.1e-11
rajat09	24,482	105,573	circuit simulation problem	4,199	8.6e-11	8.6e-11	5,146	2.4e-11	2.4e-11
thermal	3,456	66,528	thermal problem	19	5.7e-11	5.7e-11	18	5.5e-11	5.5e-11

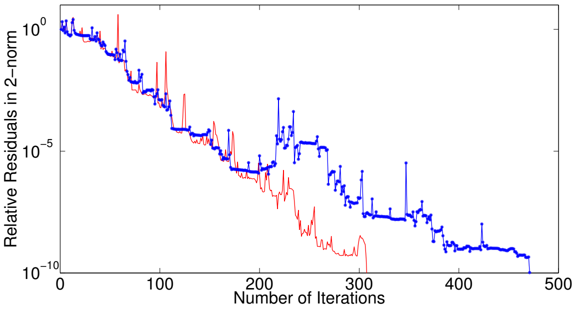

Fig. 1. Histories of relative residual for 'cavity01'. −: BiCGSTAB with D-norm minimization; −∗: BiCGSTAB.

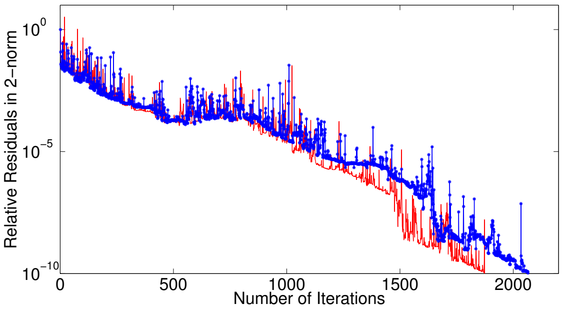

Fig. 2. Histories of relative residual for 'dw2048'. −: BiCGSTAB with D-norm minimization; −∗: BiCGSTAB.

teed. As for 'thermal' and 'ck400' , the iterations needed to converge are almost the same.

6. Conclusions

In this article, we have proposed a new algorithm. From the numerical experiments we are convinced that it is meaningful to apply the D-norm in minimizing the residual vector instead of the 2-norm in BiCGSTAB, though the efficiency is not universal. Our research shows that for the overall convergence behaviour, the 2-norm minimization may not be the best choice for BiCGSTAB.

Our idea of D-norm minimization can also be applied to BiCGSTAB(ℓ) and GPBiCG [12]. However, from our preliminary experiments, the current selection of D does not show much improvement; the details are not shown here due to the limitation of the space. Appropriate selection of D will be one of our future works. For the improvement of the block version of BiCGSTAB, as indicated by Weighted Block GMRES [13], is also under consideration.

Acknowledgments

We would like to thank for the referee's valuable comments. This work was supported in part by MEXT SPIRE, JICFuS, MEXT KAKENHI (No. 22104003), JSPS KAKENHI (Nos. 25870099, 25286097).

References

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

[2] A. Greenbaum, Iterative Methods for Solving Linear Systems, SIAM, Philadelphia, 1997.

[3] 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.

[4] R. Fletcher, Conjugate gradient methods for indefinite systems, Lect. Notes Math., **506** (1976), 73–89.

[5] G. Sleijpen and H. A. van der Vorst, An overview of approaches for the stable computation of hybrid BiCG methods, Appl. Numer. Math., **19** (1995), 235–254.

[6] G. Sleijpen and D. R. Fokkema, BiCGStab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, Electron. Trans. Numer. Anal., **1** (1993), 11–32.

[7] A. Essai, Weighted FOM and GMRES for solving nonsymmetric linear systems, Numer. Algorithms, **18** (1998), 277–292.

[8] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand., **45** (1950), 255–282.

[9] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, Philadelphia, 2003.

[10] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., **7** (1986), 856–869.

[11] The University of Florida Sparse Matrix Collection, `http://www.cise.ufl.edu/research/sparse/matrices/`.

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

[13] A. Imakura, L. Du and H. Tadano, A Weighted Block GMRES method for solving linear systems with multiple right-hand sides, JSIAM Letters, **5** (2013), 65–68.