JSIAM Letters Vol.1 (2009) pp.44–47 ©2009 Japan Society for Industrial and Applied Mathematics

Block BiCGGR: a new Block Krylov subspace method for computing high accuracy solutions

Hiroto Tadano[1], Tetsuya Sakurai[1] and Yoshinobu Kuramashi[2]

Department of Computer Science, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan[1]
Graduate School of Pure and Applied Sciences, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan[2]

E-mail {*tadano, sakurai*}*@cs.tsukuba.ac.jp, kuramasi@het.ph.tsukuba.ac.jp*

Received March 19, 2009, Accepted June 15, 2009

Abstract

In this paper, the influence of errors which arise in matrix multiplications on the accuracy of approximate solutions generated by the Block BiCGSTAB method is analyzed. In order to generate high accuracy solutions, a new Block Krylov subspace method named "Block BiCGGR" is also proposed. Some numerical experiments illustrate that the Block BiCGGR method can generate high accuracy solutions compared with the Block BiCGSTAB method.

Keywords Block Krylov subspace methods, Block BiCGSTAB, linear systems with multiple right hand sides, high accuracy solutions

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

1. Introduction

Linear systems with multiple right hand sides

$$AX = B, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}, B, X \in \mathbb{C}^{n\times L}$, appear in many scientific applications such as lattice quantum chromodynamics (lattice QCD) calculation of physical quantities [1], an eigensolver using contour integration [2]. To solve these linear systems for X, some Block Krylov subspace methods (e.g., Block BiCG [3], Block BiCGSTAB [4], Block QMR [5]) have been proposed.

Block Krylov subspace methods can compute approximate solutions of linear systems with multiple right hand sides efficiently compared with Krylov subspace methods for single right hand side [5]. However, the gap between the residual generated by the recursion of the Block BiCGSTAB method and the true residual may arise. In this paper, the gap which arises in the Block BiCGSTAB method is analyzed. Then, a new Block Krylov subspace method named "Block BiCGGR" for reducing the gap is also proposed.

This paper is organized as follows. In Section 2, a matrix-valued polynomial and an operation are defined. The Block BiCGSTAB method is briefly described in Section 3. In Section 4, the influence of errors which arise in matrix multiplications on the accuracy of approximate solutions of the Block BiCGSTAB method. In Section 5, the Block BiCGGR method is proposed for reducing the gap between the residual generated by the recursion and the true residual. Then the true residual of the Block BiCGGR method is also evaluated. In Section 6, the accuracy of approximate solutions generated by both methods is verified by numerical experiments. The paper is concluded in Section 7.

2. Matrix-valued polynomial

Let $\mathcal{M}_k(z)$ be a matrix-valued polynomial of degree k defined by

$$\mathcal{M}_k(z) \equiv \sum_{j=0}^{k} z^j M_j,$$

where $M_j \in \mathbb{C}^{L\times L}$ and $z \in \mathbb{C}$. The operation $\circ$ is used in this paper for the multiplication

$$\mathcal{M}_k(A) \circ V \equiv \sum_{j=0}^{k} A^j V M_j,$$

where $V \in \mathbb{C}^{n\times L}$. This operation satisfies the following properties [4].

Proposition 1 *Let $\mathcal{M}(z)$ and $\mathcal{N}(z)$ be matrix-valued polynomials of degree k and let V and ξ be an $n \times L$ matrix and an $L \times L$ matrix, respectively. Then, the following properties are satisfied.*

(*i*) $(\mathcal{M}(A) \circ V)\xi = (\mathcal{M}\xi)(A) \circ V$,

(*ii*) $(\mathcal{M} + \mathcal{N})(A) \circ V = \mathcal{M}(A) \circ V + \mathcal{N}(A) \circ V$.

3. The Block BiCGSTAB method

The $(k+1)$th residual $R_{k+1} \in \mathbb{C}^{n\times L}$ of the Block BiCGSTAB method is defined by

$$R_{k+1} = B - AX_{k+1} \equiv (\mathcal{Q}_{k+1}\mathcal{R}_{k+1})(A) \circ R_0, \tag{2}$$

where $R_0 = B - AX_0$ is an initial residual. The matrix-valued polynomial $\mathcal{R}_{k+1}(z)$ of degree $(k+1)$ which appears in (2) can be computed by the following recursions

$$\mathcal{R}_0(z) = \mathcal{P}_0(z) = I_L,$$
$$\mathcal{R}_{k+1}(z) = \mathcal{R}_k(z) - z\mathcal{P}_k(z)\alpha_k,$$
$$\mathcal{P}_{k+1}(z) = \mathcal{R}_{k+1}(z) + \mathcal{P}_k(z)\beta_k,$$

$X_0 \in \mathbb{C}^{n\times L}$ is an initial guess,
Compute $R_0 = B - AX_0$,
Set $P_0 = R_0$,
Choose $\tilde{R}_0 \in \mathbb{C}^{n\times L}$,
For $k = 0, 1, \ldots,$ **until** $\|R_k\|_{\mathrm{F}} \le \varepsilon \|B\|_{\mathrm{F}}$ **do:**

$$V_k = AP_k,$$
Solve $(\tilde{R}_0^{\mathrm{H}} V_k)\alpha_k = \tilde{R}_0^{\mathrm{H}} R_k$ for α_k,
$$T_k = R_k - V_k\alpha_k,$$
$$Z_k = AT_k,$$
$$\zeta_k = \mathrm{Tr}\left[Z_k^{\mathrm{H}} T_k\right] / \mathrm{Tr}\left[Z_k^{\mathrm{H}} Z_k\right],$$
$$X_{k+1} = X_k + P_k\alpha_k + \zeta_k T_k,$$
$$R_{k+1} = T_k - \zeta_k Z_k,$$
Solve $(\tilde{R}_0^{\mathrm{H}} V_k)\beta_k = -\tilde{R}_0^{\mathrm{H}} Z_k$ for β_k,
$$P_{k+1} = R_{k+1} + (P_k - \zeta_k V_k)\beta_k,$$
End

Fig. 1. Algorithm of the Block BiCGSTAB method.

where $\mathcal{P}_{k+1}(z)$ is an auxiliary matrix-valued polynomial of degree $(k+1)$, I_L is an $L\times L$ identity matrix, α_k and β_k are $L\times L$ complex matrices. The polynomial $\mathcal{Q}_{k+1}(z)$ of degree $(k+1)$ is defined as follows:

$$\mathcal{Q}_0(z) = 1,$$
$$\mathcal{Q}_{k+1}(z) = (1 - \zeta_k z)\mathcal{Q}_k(z),$$

where $\zeta_k \in \mathbb{C}$. The residual R_{k+1} can be computed by the following recursions,

$$R_{k+1} = T_k - \zeta_k AT_k, \tag{3}$$
$$P_{k+1} = R_{k+1} + (P_k - \zeta_k AP_k)\beta_k,$$
$$T_k = R_k - AP_k\alpha_k, \tag{4}$$

where matrices P_{k+1} and T_k are defined by $P_{k+1} \equiv (\mathcal{Q}_{k+1}\mathcal{P}_{k+1})(A)\circ R_0$ and $T_k \equiv (\mathcal{Q}_k\mathcal{R}_{k+1})(A)\circ R_0$, respectively. The Proposition 1 is used to derive the above recursions. From the Eqs. (2), (3), and (4), recursion for the approximate solution X_{k+1} can be obtained by

$$X_{k+1} = X_k + P_k\alpha_k + \zeta_k T_k. \tag{5}$$

The $L\times L$ matrices α_k and β_k are determined so that bi-orthogonal conditions:

$$\tilde{R}_0^{\mathrm{H}} A^j(\mathcal{R}_k(A)\circ R_0) = O_L, \quad j = 0, 1, \ldots, k{-}1, \tag{6}$$
$$\tilde{R}_0^{\mathrm{H}} A^{j+1}(\mathcal{P}_k(A)\circ R_0) = O_L, \quad j = 0, 1, \ldots, k{-}1, \tag{7}$$

are satisfied. Here, $\tilde{R}_0$ is an $n\times L$ arbitrary nonzero matrix, O_L is an $L\times L$ zero matrix, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix. Typically, $\tilde{R}_0$ is set to R_0, or given by random numbers. The scalar parameter ζ_k is determined so that $\|R_{k+1}\|_{\mathrm{F}}$ is minimized. Fig. 1 shows the algorithm of the Block BiCGSTAB method. Here, $\mathrm{Tr}[\,\cdot\,]$ denotes the trace of a matrix, and $\varepsilon > 0$ is a sufficiently small value for the stopping criterion.

4. Evaluation of the true residual of the Block BiCGSTAB method

In this section, it is assumed that computation errors arise in the multiplications with $\alpha_0, \alpha_1, \ldots, \alpha_k$ which appear in the Block BiCGSTAB method. The influence of these errors on the true residual of the Block BiCGSTAB method is considered. A matrix enclosed by a symbol $\langle\,\cdot\,\rangle$ denotes the perturbed matrix. Throughout this section, it is assumed that no calculation errors arise except for multiplications with $\alpha_0, \alpha_1, \ldots, \alpha_k$.

The perturbed matrices $\langle P_j\alpha_j\rangle$ and $\langle (AP_j)\alpha_j\rangle$ are required for the computation of X_{j+1} and R_{j+1}, respectively. These matrices can be written as follows:

$$\langle P_j\alpha_j\rangle = P_j\alpha_j + F_j, \tag{8}$$
$$\langle (AP_j)\alpha_j\rangle = AP_j\alpha_j + G_j, \tag{9}$$

where F_j and G_j denote error matrices.

From the Eqs. (5) and (8), X_{k+1} is written as

$$X_{k+1} = X_k + \langle P_k\alpha_k\rangle + \zeta_k T_k$$
$$= X_0 + \sum_{j=0}^{k}(P_j\alpha_j + \zeta_j T_j) + \sum_{j=0}^{k} F_j. \tag{10}$$

By using the Eq. (9), the residual R_{k+1} generated by the recursion (3) is also written as

$$R_{k+1} = R_k - \langle (AP_k)\alpha_k\rangle - \zeta_k AT_k$$
$$= R_0 - \sum_{j=0}^{k}(AP_j\alpha_j + \zeta_j AT_j) - \sum_{j=0}^{k} G_j. \tag{11}$$

By using the Eqs. (10) and (11), the true residual $B - AX_{k+1}$ of the Block BiCGSTAB method is given by

$$B - AX_{k+1} = R_0 - \sum_{j=0}^{k}(AP_j\alpha_j + \zeta_j AT_j) - \sum_{j=0}^{k} AF_j$$
$$= R_{k+1} + \sum_{j=0}^{k} E_j, \tag{12}$$

where the matrix E_j is defined by $E_j \equiv G_j - AF_j$. From the Eqs. (8) and (9), the matrix E_j can be written as

$$E_j = \langle (AP_j)\alpha_j\rangle - A\langle P_j\alpha_j\rangle.$$

The error matrices $E_0, E_1, \ldots, E_k$ appear in (12) when the computation errors arise in the multiplications with $\alpha_0, \alpha_1, \ldots, \alpha_k$. The Eq. (12) implies that the true residual $B - AX_{k+1}$ of the Block BiCGSTAB method approaches to $\sum_{j=0}^{k} E_j$ when the residual norm $\|R_{k+1}\|_{\mathrm{F}}$ is sufficiently small.

5. The Block BiCGGR method

The error matrices F_j and G_j generate the gap between the residual and the true residual. To negate the influence of these matrices, a condition $G_j = AF_j$ should be satisfied. In this section, a new Block Krylov subspace method is proposed to reduce the gap.

5.1 Construction of an algorithm

There are two ways of constructing the recursion for the residual $R_{k+1} = (\mathcal{Q}_{k+1}\mathcal{R}_{k+1})(A)\circ R_0$. In the Block BiCGSTAB method, firstly, the polynomial $\mathcal{Q}_{k+1}$ is expanded. In this case, as shown in the Eq. (12), the true residual $B - AX_{k+1}$ is not equal to the residual R_{k+1}

generated by the recursion. In the proposed method, firstly, the polynomial $\mathcal{R}_{k+1}$ is expanded for computing $\mathcal{Q}_{k+1}\mathcal{R}_{k+1}$. The recursion of this polynomial is given by

$$\mathcal{Q}_{k+1}\mathcal{R}_{k+1} = \mathcal{Q}_k\mathcal{R}_k - \zeta_k z\mathcal{Q}_k\mathcal{R}_k - z\mathcal{Q}_{k+1}\mathcal{P}_k\alpha_k.$$

The polynomials $\mathcal{Q}_{k+1}\mathcal{P}_k$ and $\mathcal{Q}_{k+1}\mathcal{P}_{k+1}$ are computed by the following recursions:

$$\mathcal{Q}_{k+1}\mathcal{P}_k = \mathcal{Q}_k\mathcal{P}_k - \zeta_k z\mathcal{Q}_k\mathcal{P}_k,$$
$$\mathcal{Q}_{k+1}\mathcal{P}_{k+1} = \mathcal{Q}_{k+1}\mathcal{R}_{k+1} + \mathcal{Q}_{k+1}\mathcal{P}_k\beta_k.$$

From the above recursions, the residual R_{k+1} and auxiliary matrices can be computed by

$$R_{k+1} = R_k - \zeta_k AR_k - AU_k, \tag{13}$$

$$P_{k+1} = R_{k+1} + U_k\alpha_k^{-1}\beta_k, \tag{14}$$

$$S_k = P_k - \zeta_k AP_k,$$

where $S_k \equiv (\mathcal{Q}_{k+1}\mathcal{P}_k)(A) \circ R_0$ and $U_k \equiv S_k\alpha_k$. From the Eqs. (2) and (13), X_{k+1} can be computed by

$$X_{k+1} = X_k + \zeta_k R_k + U_k. \tag{15}$$

In the proposed method, the generation of the gap between the residual and the true residual can be avoided by computing the multiplication of S_k and α_k before the computation of X_{k+1} and R_{k+1}.

Matrices α_k and β_k are determined so that the bi-orthogonality conditions (6) and (7) are satisfied. From the Eq. (6), the matrix α_k can be computed by

$$\alpha_k = (\tilde{R}_0^{\mathrm{H}} AP_k)^{-1}\tilde{R}_0^{\mathrm{H}} R_k. \tag{16}$$

By the bi-orthogonality condition (7) and the relation

$$\tilde{R}_0^{\mathrm{H}} R_{k+1} = -\zeta_k\tilde{R}_0^{\mathrm{H}} AT_k,$$

the matrix β_k can be obtained by

$$\beta_k = (\tilde{R}_0^{\mathrm{H}} AP_k)^{-1}\tilde{R}_0^{\mathrm{H}} R_{k+1}/\zeta_k. \tag{17}$$

The matrix $\gamma_k \equiv \alpha_k^{-1}\beta_k$ is appeared in the Eq. (14). By using the Eqs. (16) and (17), γ_k can be obtained by

$$\gamma_k = (\tilde{R}_0^{\mathrm{H}} R_k)^{-1}\tilde{R}_0^{\mathrm{H}} R_{k+1}/\zeta_k.$$

If the parameter ζ_k is determined so that $\|R_{k+1}\|_{\mathrm{F}}$ is minimized, then extra multiplications with A are required in the proposed method. To avoid the multiplications with A, the parameter ζ_k is computed by

$$\zeta_k = \mathrm{Tr}\left[(AR_k)^{\mathrm{H}} R_k\right]/\mathrm{Tr}\left[(AR_k)^{\mathrm{H}} AR_k\right].$$

In the proposed method, the three multiplications with A are required in each iteration. To reduce the number of multiplications with A, the matrix AP_{k+1} is computed by the following recursion

$$AP_{k+1} = AR_{k+1} + AU_k\gamma_k.$$

5.2 *Evaluation of the true residual*

Similar to the previous section, assume that no calculation errors arise except for the multiplications with $\alpha_0, \alpha_1, \ldots, \alpha_k$. The multiplication with α_j is appeared in the computation of $U_j = S_j\alpha_j$. By using the symbol $\langle\cdot\rangle$, the perturbed matrix $\langle S_j\alpha_j\rangle$ is represented as

$$\langle S_j\alpha_j\rangle = S_j\alpha_j + H_j, \tag{18}$$

$X_0 \in \mathbb{C}^{n\times L}$ is an initial guess,
Compute $R_0 = B - AX_0$,
Set $P_0 = R_0$ and $V_0 = W_0 = AR_0$,
Choose $\tilde{R}_0 \in \mathbb{C}^{n\times L}$,
For $k = 0, 1, \ldots,$ **until** $\|R_k\|_{\mathrm{F}} \le \varepsilon\|B\|_{\mathrm{F}}$ **do:**

Solve $(\tilde{R}_0^{\mathrm{H}} V_k)\alpha_k = \tilde{R}_0^{\mathrm{H}} R_k$ for α_k,

$$\begin{aligned}
\zeta_k &= \mathrm{Tr}\left[W_k^{\mathrm{H}} R_k\right]/\mathrm{Tr}\left[W_k^{\mathrm{H}} W_k\right],\\
S_k &= P_k - \zeta_k V_k,\\
U_k &= S_k\alpha_k,\\
Y_k &= AU_k,\\
X_{k+1} &= X_k + \zeta_k R_k + U_k,\\
R_{k+1} &= R_k - \zeta_k W_k - Y_k,\\
W_{k+1} &= AR_{k+1},
\end{aligned}$$

Solve $(\tilde{R}_0^{\mathrm{H}} R_k)\gamma_k = \tilde{R}_0^{\mathrm{H}} R_{k+1}/\zeta_k$ for γ_k,

$$\begin{aligned}
P_{k+1} &= R_{k+1} + U_k\gamma_k,\\
V_{k+1} &= W_{k+1} + Y_k\gamma_k,
\end{aligned}$$

End

Fig. 2. Algorithm of the Block BiCGGR method.

where H_j is an error matrix. From the Eqs. (15) and (18), the approximate solution X_{k+1} is written as

$$\begin{aligned}
X_{k+1} &= X_k + \zeta_k R_k + \langle S_k\alpha_k\rangle\\
&= X_0 + \sum_{j=0}^{k}(\zeta_j R_j + S_j\alpha_j) + \sum_{j=0}^{k} H_j.
\end{aligned}\tag{19}$$

By using the Eqs. (13) and (19), R_{k+1} is represented as

$$\begin{aligned}
R_{k+1} &= R_k - \zeta_k AR_k - A\langle S_k\alpha_k\rangle\\
&= R_0 - \sum_{j=0}^{k}(\zeta_j AR_j + AS_j\alpha_j) - \sum_{j=0}^{k} AH_j\\
&= B - A\left[X_0 + \sum_{j=0}^{k}(\zeta_j R_j + S_j\alpha_j) + \sum_{j=0}^{k} H_j\right]\\
&= B - AX_{k+1}.
\end{aligned}$$

By regarding the matrices H_j and AH_j as F_j and G_j, it is confirmed that the proposed method satisfies $E_j = G_j - AF_j = O$. Since the proposed method can reduce the gap between the residual and the true residual, this method is named "Block Bi-Conjugate Gradient Gap-Reducing (Block BiCGGR)". The algorithm of the Block BiCGGR method is shown in Fig. 2.

6. Numerical experiments

Test matrices used in numerical experiments were PDE900, JPWH991, and CONF5.4-00L8X8-1000 [6]. The size and the number of nonzero elements of these matrices are shown in Table 1. The coefficient matrix of CONF5.4-00L8X8-1000 is constructed by $I_n - \kappa D$, where D is an $n \times n$ non-Hermitian matrix and κ is a real valued parameter. This parameter was set to 0.1782.

The initial solution X_0 was set to the zero matrix. The shadow residual $\tilde{R}_0$ was given by a random number generator. The right hand side B of (1) was given by $B = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_L]$, where $\boldsymbol{e}_j$ is a jth unit vector. The

Table 1. The size and the number of nonzero elements of test matrices. NNZ denotes the number of nonzero elements.

Matrix name	Size	NNZ
PDE900	900	4,380
JPWH991	991	6,027
CONF5.4-00L8X8-1000	49,152	1,916,928

Table 2. Results of the Block BiCGSTAB method.

L	#Iter.	Time/L [s]	Res.	True Res.
PDE900				
1	53	0.0096	4.8×10^{-15}	4.8×10^{-15}
2	46	0.0067	1.1×10^{-15}	2.0×10^{-13}
4	41	0.0031	4.8×10^{-15}	1.8×10^{-12}
JPWH991				
1	56	0.0159	5.7×10^{-15}	1.2×10^{-14}
2	49	0.0083	8.3×10^{-15}	4.1×10^{-13}
4	43	0.0034	6.3×10^{-15}	5.9×10^{-12}
CONF5.4-00L8X8-1000				
1	555	13.9408	8.9×10^{-15}	9.5×10^{-15}
2	452	7.5609	7.3×10^{-15}	2.5×10^{-13}
4	406	6.1544	8.7×10^{-15}	2.8×10^{-13}

value ε for the stopping criterion was set to 1.0×10^{-14}.

All experiments were carried out in double precision arithmetic on CPU: Intel Core 2 Duo 2.4GHz, Memory: 4GBytes, Compiler: Intel Fortran ver. 10.1, Compile option: `-O3 -xT -openmp`. The multiplication with the coefficient matrix was parallelized by OpenMP.

The results of the Block BiCGSTAB method are shown in Table 2. In this Table, #Iter., Res, and True Res. denote the number of iterations, the relative residual norm $\|R_k\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$, and the true relative residual norm $\|B - AX_k\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$, respectively.

As shown in Table 2, the relative residual norms of the Block BiCGSTAB method were satisfied the convergence criterion. However, the true residual norms did not reach 10^{-14} when $L = 2, 4$.

The relation between the true relative residual norm and $\|\sum_{j=0}^{k} E_j\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$ for JPWH991 with $L = 4$ is shown in Fig. 3. The true relative residual norm became almost equal to the value $\|\sum_{j=0}^{k} E_j\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$. The Eq. (12) was verified through this numerical example.

The results of the Block BiCGGR method are shown in Table 3. The true relative residual norms reached 10^{-14} except for JPWH991 with $L = 1$. By using the Block BiCGGR method, the gap between the residual and the true residual can be reduced compared with the Block BiCGSTAB method.

7. Conclusions

In this paper, we have evaluated the true residual of the Block BiCGSTAB method when the computation errors arise in the multiplications with $\alpha_0, \alpha_1, \ldots, \alpha_k$. We have shown that the true residual of this method approaches to the sum of error matrices when the residual norm is sufficiently small. Then, we have proposed the Block BiCGGR method for reducing the gap between the residual and the true residual. Through some numerical experiments, we have verified that the Block BiCGGR method generates the high accuracy solutions compared with the Block BiCGSTAB method.

Table 3. Results of the Block BiCGGR method.

L	#Iter.	Time/L [s]	Res.	True Res.
PDE900				
1	53	0.0107	3.2×10^{-15}	3.3×10^{-15}
2	46	0.0051	1.1×10^{-15}	1.4×10^{-15}
4	45	0.0031	5.5×10^{-15}	5.6×10^{-15}
JPWH991				
1	52	0.0134	8.4×10^{-15}	1.3×10^{-14}
2	51	0.0082	3.7×10^{-15}	6.1×10^{-15}
4	44	0.0035	1.5×10^{-15}	2.3×10^{-15}
CONF5.4-00L8X8-1000				
1	555	14.2714	7.4×10^{-15}	8.5×10^{-15}
2	456	8.1093	5.6×10^{-15}	6.7×10^{-15}
4	386	6.0348	7.4×10^{-15}	8.6×10^{-15}

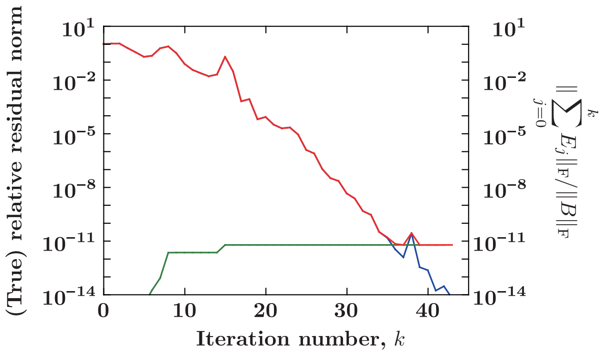

Fig. 3. Relation between the true relative residual norm of Block BiCGSTAB and $\|\sum_{j=0}^{k} E_j\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$ (JPWH991, $L = 4$). —— : true relative residual norm $\|B - AX_k\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$, —— : relative residual norm $\|R_k\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$, —— : $\|\sum_{j=0}^{k} E_j\|_{\mathrm{F}}/\|B\|_{\mathrm{F}}$.

Acknowledgments

This work was supported by Grant-in-Aid for Young Scientists (Start-up) (No. 20800009).

References

[1] PACS-CS Collaboration, S. Aoki *et al.*, $2 + 1$ Flavor Lattice QCD toward the Physical Point, arXiv:0807.1661v1 [hep-lat], 2008.

[2] T. Sakurai, H. Tadano, T. Ikegami, and U. Nagashima, A parallel eigensolver using contour integration for generalized eigenvalue problems in molecular simulation, Tech. Rep. CS–TR–08–14, Univ. of Tsukuba, 2008.

[3] D. P. O'Leary, The block conjugate gradient algorithm and related methods, Lin. Alg. Appl., **29** (1980), 293–322.

[4] A. El Guennouni, K. Jbilou and H. Sadok, A block version of BiCGSTAB for linear systems with multiple right-hand sides, Elec. Trans. Numer. Anal., **16** (2003), 129–142.

[5] R. W. Freund and M. Malhotra, A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, Lin. Alg. Appl., **254** (1997), 119–157.

[6] Matrix Market, `http://math.nist.gov/MatrixMarket/`