Improvement of the accuracy of the approximate solution of the Block BiCR method

Hiroto Tadano¹, Youichi Ishikawa² and Akira Imakura¹

¹ Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
² FBS Corporation, 7 Kanda Mitoshirocho, Chiyoda-ku, Tokyo 101-0053, Japan

E-mail tadano@es.tsukuba.ac.jp

Received April 1, 2014, Accepted July 10, 2014

Abstract
Block Krylov subspace methods are efficient solvers for linear systems with multiple right-hand sides in terms of the number of iterations and computational time. As one of Block Krylov subspace methods, the Block BiCR method has been proposed by Zhang et al. in 2013. This method often shows a smooth convergence behavior compared with the Block BiCG method. However, the accuracy of the approximate solution generated by the Block BiCR method often deteriorates. In this paper we propose a modified Block BiCR method in order to improve the accuracy of the approximate solutions.

Keywords Block Krylov subspace methods, linear systems, multiple right-hand sides

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

1. Introduction
In this paper we consider solving linear systems with multiple right-hand sides:

\[ AX = B, \]

where \( A \in \mathbb{C}^{n \times n} \) is a non-Hermitian matrix, and \( X, B \in \mathbb{C}^{n \times L} \). Linear systems (1) appear in many applications such as lattice quantum chromodynamics (QCD) calculation [1], and the eigensolver based on the contour integral [2]. In these applications, it is necessary to compute a high accuracy approximate solution of (1).

Numerical methods for (1) are roughly divided into the direct methods and the iterative methods. As iterative methods for (1), Block Krylov subspace methods such as the Block Bi-Conjugate Gradient (BiCG) method [3], the Block BiCGSTAB method [4], and the Block GMRES method [5] have been proposed. Block Krylov subspace methods are efficient solvers for (1) in terms of the number of iterations and computational time.

As one of Block Krylov subspace methods, the Block Bi-Conjugate Residual (BiCR) method [6] has been proposed by Zhang et al. This method is a natural extension of the BiCR method [7] for linear systems with single right-hand side proposed by Sogabe et al. The Block BiCR method often shows smooth convergence behavior compared with the Block BiCG method. However, the accuracy of the approximate solution generated by the Block BiCR method may deteriorate due to an error matrix that arises from the matrix multiplication with respect to the coefficient matrix \( A \). In this paper, a modified Block BiCR method is proposed in order to improve the accuracy of the approximate solutions. Moreover, this method is stabilized by using the residual orthonormalization technique [8].

This paper is organized as follows. In Section 2, the Block BiCG method and the Block BiCR method are briefly described. In Section 3, the influence of the error matrix on the accuracy of the approximate solution is analyzed. The modified Block BiCR method is proposed in Section 4. In Section 5, the modified Block BiCR method is stabilized by the residual orthonormalization technique. Section 6 provides the results of numerical experiments to show the efficiency of the proposed method. This paper is concluded in Section 7.

2. The Block BiCG method and the Block BiCR method
Let \( X_{k+1} \in \mathbb{C}^{n \times L} \) be a \((k+1)\)th approximate solution of the linear systems (1). \( X_{k+1} \) is computed so that the following condition is satisfied.

\[ X_{k+1} = X_0 + Z_{k+1}, \quad Z_{k+1} \in \mathbb{C}^{n \times L} \]

\[ \kappa_{k+1}^\square(A; R_0). \]

Here, \( R_0 = B - AX_0 \) is an initial residual, and \( \kappa_{k+1}^\square(A; R_0) \) is a block Krylov subspace defined as follows:

\[ \kappa_{k+1}^\square(A; R_0) = \left\{ \sum_{j=0}^{k} A^j R_0 \gamma_j \mid \gamma_j \in \mathbb{C}^{L \times L} \ (j = 0, 1, \ldots, k) \right\}. \]

The \((k+1)\)th residual \( R_{k+1} = B - AX_{k+1} \) of the Block BiCG method [3] and the Block BiCR method [6] is computed by the following recurrence relations.

\[ R_0 = P_0 = B - AX_0 \in \kappa_1^\square(A; R_0), \]

\[ R_{k+1} = R_k - AP_k \alpha_k \in \kappa_{k+2}^\square(A; R_0), \quad (2) \]
In (6), the matrix enclosed by the brackets denotes the corresponding residual \( R_{k+1} = P_{k+1} R_k \beta_k \in \mathbb{C}^{L \times L} \) of the linear system \( A^H X = B \) computed simultaneously in order to compute \( \alpha_k \) and \( \beta_k \). The matrices \( \tilde{R}_{k+1} \) and \( \tilde{P}_{k+1} \) are computed by the following recurrence relations:

\[
\begin{align*}
\tilde{R}_0 &= \tilde{P}_0 = \tilde{B} - A^H \tilde{X}_0 \in \mathbb{C}^{L \times L}, \\
\tilde{R}_{k+1} &= \tilde{R}_k - A^H \tilde{P}_k \tilde{\alpha}_k \in \mathbb{C}^{L \times L}, \\
\tilde{P}_{k+1} &= \tilde{P}_{k+1} + \tilde{P}_k \tilde{\beta}_k \in \mathbb{C}^{L \times L},
\end{align*}
\]

(4)

The matrices \( \alpha_k, \beta_k, \tilde{\alpha}_k, \) and \( \tilde{\beta}_k \) are determined by imposing a bi-orthogonality condition shown in Table 1. Figs. 1 and 2 show the algorithm of the Block BiCG method and of the Block BiCR method, respectively. In these algorithms, \( \| \cdot \|_F \) denotes the Frobenius norm of a matrix, and \( \varepsilon \) is a sufficiently small value defined by users.

The Block BiCR method requires three matrix multiplications \( AP_k, AR_k, \) and \( A^H \tilde{P}_k \) in each iteration. The matrices \( AR_k \) and \( A^H \tilde{P}_k \) are computed by the explicit matrix multiplication. To reduce the computational complexity, the matrix \( AP_k \) is computed by the recurrence relation.

3. The influence of the error matrix on the approximate solution

In this section, the influence of the error matrix on the approximate solution is analyzed. Expanding (2) and (3), the \((k+1)\)th approximate solution \( X_{k+1} \) and the corresponding residual \( R_{k+1} \) are rewritten as follows:

\[
\begin{align*}
X_{k+1} &= X_0 + \sum_{j=0}^{k} P_j \alpha_j, \\
R_{k+1} &= R_0 - \sum_{j=0}^{k} (AP_j) \alpha_j.
\end{align*}
\]

(5)

(6)

In (6), the matrix enclosed by the brackets denotes the matrix computed in advance.

From (5) and (6), the relationship between the true residual \( B - AX_{k+1} \) and the residual \( R_{k+1} \) can be obtained as follows:

\[
B - AX_{k+1} = R_{k+1} + E_{k+1},
\]

(7)

\[ X_0 \in \mathbb{C}^{n \times L} \] is an initial guess, \( \text{Compute} \ R_0 = B - AX_0, \)
\( \text{Choose} \ R_0 \in \mathbb{C}^{n \times L}, \)
\( \text{Set} \ P_0 = R_0, \)
\( \text{For} \ k = 0, 1, \ldots \text{until} \ |R_k|_F / |B|_F \leq \varepsilon \text{ do:} \)
\( \text{Solve} \ (P_k U_k) \alpha_k = \tilde{R}_k \text{ for} \ \alpha_k, \)
\( \text{Solve} \ (P_k V_k) \beta_k = \tilde{R} \text{ for} \ \beta_k, \)
\( \text{End For} \)

Fig. 1. Algorithm of the Block BiCG method.

\[ E_{k+1} = \sum_{j=0}^{k} [(AP_j) \alpha_j - (P_j \alpha_j)]. \]

Theoretically, the error matrix \( E_{k+1} \) does not exist because the residual \( R_{k+1} \) computed by the recurrence relation is equal to the true residual \( B - AX_{k+1} \). However, in numerical computation, the matrix \( E_{k+1} \) appears because \( (AP_j) \alpha_j \neq (P_j \alpha_j) \).

4. Modification of the Block BiCR method for improving the accuracy of the approximate solution

As mentioned in the previous section, the error matrix \( E_{k+1} \) of (7) has an influence on the accuracy of the approximate solution. In order to improve the accuracy of the approximate solution, we need to reduce the influence of the error matrix \( E_{k+1} \).

The error matrix \( E_{k+1} \) includes the matrix \( AP_{k+1} \). In the Block BiCR method, the matrix \( AP_k \) is computed by the recurrence relation. The accuracy of the approximate solution may deteriorate if the matrix \( AP_k \) computed by the recurrence relation differs appreciably from one computed by the explicit matrix multiplication. In this section, the Block BiCR method is modified to compute the matrix \( AP_k \) by the explicit matrix multiplication without the increase of computational complexity.

In the modified Block BiCR method, the matrix \( A^H \tilde{P}_k \)
X₀ ∈ Cᴺ×L is an initial guess, Compute R₀ = B − AX₀, Choose R₀ ∈ Cᴺ×L, Set P₀ = R₀, P₀ = R₀, U₀ = AR₀, V₀ = A⁺R₀, For k = 0, 1, ..., until ∥Rₖ∥F/∥B∥F ≤ ε do: Solve (Uₖ₅)Vₖ₅ = Vₖ₅Hₖ₅Rₖ₅ for αₖ, Solve (Uₖ₅)Vₖ₅ = Vₖ₅Hₖ₅Rₖ₅ for βₖ, Solve (Rₖ₅)Vₖ₅ = Rₖ₅Hₖ₅Vₖ₅ for βₖ, Pₖ₁ = Rₖ₅ + Pₖ₅βₖ, Pₖ₊₁ = Rₖ₅₊₁ + Pₖ₅βₖ, Uₖ₊₁ = Vₖ₊₁ + Ûₖ₊₁βₖ, Uₖ₊₁ = APₖ₊₁, End For

Fig. 3. Algorithm of the modified Block BiCR method.

is computed by the recurrence relation instead of computing APₖ by the explicit matrix multiplication. Since the matrix A⁺Rₖ is required to compute A⁺Pₖ by the recurrence relation, A⁺Rₖ is computed by the explicit matrix multiplication. The matrix ARₖ is not required by using the following algorithm:

$$\tilde{R}_k^H A \tilde{R}_k = (A^H \tilde{R}_k)^H R_k, \quad (A R_k)^H \tilde{R}_k = R_k^H A^H \tilde{R}_k.$$  

Fig. 3 shows the algorithm of the modified Block BiCR method.

5. Improvement of numerical stability

The residual norms of Block Krylov subspace methods may not converge due to numerical instability when the number of right-hand sides is large. This numerical instability comes from the loss of linear independence among column vectors of n × L matrices which appear in the methods. In this section, numerical stability of the modified Block BiCR method is improved by the residual orthonormalization. This approach is also used in [8].

The residual Rₖ and the matrix \( \tilde{R}_k \) are factored as Rₖ = QₖSₖ and \( \tilde{R}_k = \tilde{Q}_k \tilde{S}_k \) by the QR factorization, respectively. The matrices Qₖ and Qₖ satisfy the identity matrix of order L, and \( \xi_k, \tilde{\xi}_k \in \mathbb{C}^{L \times L} \). From (2) and (4), the following equations can be obtained.

$$Q_{k+1} \tilde{\xi}_{k+1} = Q_k - A \tilde{S}_k \alpha'_k,$$

$$\tilde{Q}_{k+1} \tilde{\xi}_{k+1} = \tilde{Q}_k - A^H \tilde{S}_k \tilde{\alpha}'_k.$$  

Here, \( \tau_k \equiv \xi_{k+1} \xi_{k+1}^{-1}, \tau_{k+1} \equiv \xi_{k+1} \xi_{k}^{-1}, \alpha'_k \equiv \xi_k \alpha_k \xi_k^{-1}, \tilde{\alpha}'_k \equiv \tilde{\xi}_k \tilde{\alpha}_k \tilde{\xi}_k^{-1} \).

The algorithm of the modified Block BiCR method with residual orthonormalization is shown in Fig. 4. The matrices \( \beta'_k \) and \( \tilde{\beta}_k \) in this algorithm are defined as \( \beta_k \equiv \xi_k \tilde{\xi}_k \tau_k \) and \( \tilde{\beta}_k \equiv \xi_k \tilde{\beta}_k \tau_k \), respectively. Since the Frobenius norm of \( \tilde{R}_k \) satisfies \( \| R_k \|_F = \| \xi_k \|_F \|, \) the residual norm is monitored by \( \| \xi_k \|_F \) instead of \( \| R_k \|_F \).

6. Numerical experiments

In this section we evaluate the performance of the Block BiCG method, the Block BiCR method and the modified Block BiCR method through some numerical experiments. In numerical experiments, the methods with residual orthonormalization are used to improve numerical stability. In the rest of this paper, we call these methods “Block BiCGrQ”, “Block BiCRrQ”, and “modified Block BiCRrQ”, respectively.

Test matrices used in numerical experiments are S5H12, pde2961, poisson3Da, and waveguide3D [9]. The size n, the number of nonzero elements (NNZ), and structure of test matrices are shown in Table 2. All experiments are carried out in double precision arithmetic on CPU: Intel Xeon X5650 2.67GHz, Memory: 24GB DDR3 1333MHz, Software: MATLAB R2014a.

The initial solution X₀ and the matrix R₀ are set as X₀ = O and \( \tilde{R}_0 = R_0 \), respectively. The right-hand side B is given by the MATLAB function \texttt{rand}. The iteration is stopped if the condition \( \| R_k \|_F / \| B \|_F \leq 10^{-14} \) is satisfied.

Table 3 shows the results of the Block BiCGrQ method, the Block BiCRrQ method, and the modified Block BiCRrQ method. “#Iter.”, “Res.”, and “True Res.” denote the number of iterations, the relative residual norm \( \| R_k \|_F / \| B \|_F \), and the true relative residual norm \( \| B - A X_k \|_F / \| B \|_F \), respectively.

From Table 3, we can see that the number of iterations and the computational time of three methods are almost the same. However, the true residual norms of the modified Block BiCRrQ method are smaller than that of the Block BiCRrQ method when the number L of right-hand sides is large. Hence, the modified Block BiCRrQ method can generate the higher accuracy approximate solutions than the Block BiCRrQ method.

Fig. 5 shows the true relative residual histories for S5H12 in the case of L = 32. The true relative residual norm of the Block BiCR method stagnates around 3.0 ×
10^{-11}. On the other hand, that of the modified Block BiCR method decreases to 4.3 \times 10^{-13}.

7. Conclusion

In this paper, we have proposed the modified Block BiCR method in order to improve the accuracy of the approximate solutions of the Block BiCR method [6]. Through some numerical experiments, we verified that the proposed method can generate the higher accuracy approximate solutions than the Block BiCR method.

Acknowledgments

This work was supported by Strategic Programs for Innovative Research (SPIRE) Field 5 “The origin of matter and the universe”, MEXT KAKENHI (No. 22104003), JSPS KAKENHI (Nos. 25870099, 25286097).

References

[1] PACS-CS Collaboration, S. Aoki et al., 2+1 flavor lattice QCD toward the physical point, arXiv:0807.1661.v1 [heplat], 2008.
[2] T. Ikegami, T. Sakurai and U. Nagashima, A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method, J. Comput. Appl. Math., 233 (2010), 1927–1936.
[3] D. P. O’Leary, The block conjugate gradient algorithm and related methods, Linear Algebra 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, Electron. Trans. Numer. Anal., 16 (2010), 129–142.
[5] B. Vital, Étude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur, Ph.D. Thesis, Université de Rennes I, Rennes, 1990.
[6] J. Zhang and J. Zhao, A novel class of block methods based on the block AA^T-Lanczos bi-orthogonalization process for matrix equations, Int. J. Comput. Math., 90 (2013), 341–359.
[7] T. Sogabe, M. Sugihara and S.-L. Zhang, An extension of the conjugate residual method to nonsymmetric linear systems, J. Comput. Appl. Math., 230 (2009), 103–113.
[8] A. A. Dubrulle, Retooling the method of block conjugate gradients, Electron. Trans. Numer. Anal., 12 (2001), 216–233.
[9] The University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices/.

Table 3. Results of the Block BICGrQ method, the Block BiCRrQ method, and the modified Block BiCRrQ method.

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |

| L     | #Iter. | Time/L [s] | Res.    | True Res. |
|-------|--------|------------|---------|-----------|
| 1     | 15     | 37.1       | 2.1     | 2.1       |
| 8     | 62     | 27.9       | 2.2     | 2.2       |
| 32    | 134    | 27.9       | 2.2     | 2.2       |