Journal of Computational and Applied Mathematics 234 (2010) 985–994

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

BiCR variants of the hybrid BiCG methods for solving linear systems with nonsymmetric matrices

Kuniyoshi Abe[a,*], Gerard L.G. Sleijpen[b]

[a] Faculty of Economics and Information, Gifu Shotoku University, Nakauzura, Gifu 500-8288, Japan

[b] Department of Mathematics, Utrecht University, P.O. Box 80.010, 3508 TA Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 1 October 2008
Received in revised form 27 February 2009

Keywords:
Linear systems
Krylov subspace method
Hybrid bi-conjugate gradient method
Bi-conjugate residual method
Nonsymmetric matrices

ABSTRACT

We propose Bi-Conjugate Residual (BiCR) variants of the hybrid Bi-Conjugate Gradient (BiCG) methods (referred to as the hybrid BiCR variants) for solving linear systems with nonsymmetric coefficient matrices. The recurrence formulas used to update an approximation and a residual vector are the same as those used in the corresponding hybrid BiCG method, but the recurrence coefficients are different; they are determined so as to compute the coefficients of the residual polynomial of BiCR. From our experience it appears that the hybrid BiCR variants often converge faster than their BiCG counterpart. Numerical experiments show that our proposed hybrid BiCR variants are more effective and less affected by rounding errors. The factor in the loss of convergence speed is analyzed to clarify the difference of the convergence between our proposed hybrid BiCR variants and the hybrid BiCG methods.

1. Introduction

In this paper, we deal with Krylov subspace methods for solving a large sparse linear system

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where A stands for an n-by-n matrix, and $\boldsymbol{b}$ is an n-vector. The Bi-Conjugate Gradient (BiCG) method [5] is a well-known generic Krylov subspace method for solving this problem, and a number of hybrid BiCG methods such as the Conjugate Gradient Squared method (CGS) [17], the Bi-Conjugate Gradient STABilized method (BiCGSTAB) [19], the BiCGStab2 method [8], the Generalized Product-type method derived from BiCG (GPBiCG) [20] and the BiCGstab(l) method [15] have been developed to improve the convergence.

The Conjugate Residual (CR) [4] method has been known as a Krylov subspace method derived from the minimum residual approach [7] for symmetric matrices. The Bi-Conjugate Residual (BiCR) method [16] has been proposed as a generalization of CR for nonsymmetric matrices. It has been reported that the oscillations in the residual norms of CR and BiCR are smaller than those of BiCG, and that the residual norms of CR and BiCR tend to converge faster than those of BiCG [16]. We expect to see similar advantages in BiCR variants of the hybrid BiCG methods, which have not been previously proposed in an international journal.

Therefore, following [1] we propose BiCR variants of the hybrid BiCG methods (referred to as the hybrid BiCR variants). In other words, the BiCG part, which is a component of the residual polynomials of the hybrid BiCG methods, is replaced with BiCR. The recurrence formulas used to update an approximation and a residual vector are the same as those used in the

[*] Corresponding author. Tel.: +81 58 278 0711.
E-mail address: abe@gifu.shotoku.ac.jp (K. Abe).

0377-0427/$ – see front matter Crown Copyright © 2009 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.03.003

corresponding hybrid BiCG method, but the recurrence coefficients are different; they are determined so as to compute the coefficients of the residual polynomial of BiCR. From our experience it appears that the hybrid BiCR variants often converge faster than their BiCG counterpart. Numerical experiments show that our proposed BiCR variants are more effective and less affected by rounding errors. The factor in the loss of convergence speed is analyzed to clarify the difference of the convergence between our proposed hybrid BiCR variants and the hybrid BiCG methods.

In the following section, the outline of the hybrid BiCG methods is described. In Section 3, the mathematical properties of BiCR and the description of the recurrence coefficients of BiCR are given. In Section 4, the hybrid BiCR variants are proposed. In Section 5, the factor in the loss of convergence speed is analyzed. Numerical experiments on model problems with nonsymmetric matrices demonstrate that the hybrid BiCR variants converge faster and are more effective than the original hybrid BiCG methods.

2. Hybrid BiCG methods

Let $\boldsymbol{x}_0$ and $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ denote the initial guess and the corresponding initial residual, respectively. Then, the residual vector $\boldsymbol{r}_k^{\mathrm{BCG}}$ generated by BiCG is expressed by $\boldsymbol{r}_k^{\mathrm{BCG}} = R_k(A)\boldsymbol{r}_0$, where $R_k(\lambda)$ is the residual polynomial of BiCG. It is a multiple of the so-called Bi-Lanczos polynomial [18], which satisfies the recurrence relation

$$\begin{cases} R_0(\lambda) = 1, \\ R_1(\lambda) = 1 - \alpha_0\lambda, \\ R_{k+1}(\lambda) = \left(1 + \alpha_k \dfrac{\beta_{k-1}}{\alpha_{k-1}} - \alpha_k\lambda\right) R_k(\lambda) - \alpha_k \dfrac{\beta_{k-1}}{\alpha_{k-1}} R_{k-1}(\lambda), \quad k = 1, 2, \ldots, \end{cases} \tag{2}$$

for certain coefficients α_k and β_{k-1}.

The residual vectors of the hybrid BiCG methods are expressed as

$$S_k(A)\boldsymbol{r}_k^{\mathrm{BCG}}$$

by combining a polynomial $S_k(\lambda)$ of degree k together with BiCG. The polynomial $S_k(\lambda)$ is selected to make the residual of BiCG converge toward zero faster. Although a number of hybrid BiCG methods are known, this paper deals with only the CGS, BiCGSTAB, GPBiCG and BiCGstab(l) methods.

When the relation $S_k(\lambda) = R_k(\lambda)$ is valid for the polynomial $S_k(\lambda)$, the residual vector $R_k(A)\boldsymbol{r}_k^{\mathrm{BCG}}$ of CGS can be derived. The residual vector of BiCGSTAB is expressed by $Q_k(A)\boldsymbol{r}_k^{\mathrm{BCG}}$. Here, the polynomial $Q_k(\lambda)$ is the Generalized Minimal RESidual (1) (GMRES(1)) [13] or Generalized Conjugate Residual (1) (GCR(1)) [4] polynomial. $H_k(A)\boldsymbol{r}_k^{\mathrm{BCG}}$ stands for the residual vector of GPBiCG, where $H_k(\lambda)$ is generated by a three-term recurrence formula similar to one in (2) [20]. Moreover, the residual vector of BiCGstab(l) is equal to $T_k(A)\boldsymbol{r}_k^{\mathrm{BCG}}$, where the polynomial $T_k(\lambda)$ is a product of GMRES(l) polynomials for k, which is a multiple of l.

BiCG and the BiCG part in CGS, BiCGSTAB, GPBiCG and BiCGstab(l) are theoretically equivalent to one another. Therefore, the hybrid BiCR variants can be derived by replacing the recurrence coefficients α_k and β_k of BiCG with the recurrence coefficients α_k and β_k of BiCR. In Section 4, we propose BiCR variants of the hybrid BiCG methods in which the BiCG part $\boldsymbol{r}_k^{\mathrm{BCG}}$ has been replaced by the residual vector of BiCR.

3. BiCR for nonsymmetric matrices

In this section, we introduce BiCR for nonsymmetric matrices. It uses the same recurrence formulas as BiCG for updating the approximation and the residual vector.

The residual vector of BiCR is expressed by

$$\boldsymbol{r}_k = R_k(A)\boldsymbol{r}_0$$

with the polynomial R_k as defined by (2). To update the residual vector $\boldsymbol{r}_k$, we introduce a new auxiliary vector $\boldsymbol{p}_k$, that, for some polynomial P_k of degree k, can be expressed as

$$\boldsymbol{p}_k = P_k(A)\boldsymbol{r}_0.$$

Then, the two sequences of polynomials $R_k(\lambda)$ and $P_k(\lambda)$ are mutually interlocked with the following recurrence relations.

$$R_{k+1}(A) = R_k(A) - \alpha_k A P_k(A),$$
$$P_{k+1}(A) = R_{k+1}(A) + \beta_k P_k(A), \quad k = 1, 2, \ldots.$$

Furthermore, vectors $\boldsymbol{r}_k^*$ and $\boldsymbol{p}_k^*$, which need to be updated in the BiCR algorithm as well as in the BiCG algorithm, are expressed by $\boldsymbol{r}_k^* = R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*$ and $\boldsymbol{p}_k^* = P_k(A^{\mathrm{T}})\boldsymbol{r}_0^*$. Here, $\boldsymbol{r}_0^*$ is called an initial shadow residual, and any vector can be used.

We now will derive expressions for the recurrence coefficients α_k and β_k. We use that the vectors $\boldsymbol{p}_k$ and $\boldsymbol{p}_k^*$ are basis vectors of $K_k(A, \boldsymbol{r}_0)$ and $K_k(A^{\mathrm{T}}, \boldsymbol{r}_0^*)$, respectively, and are derived from the $A^{\mathrm{T}}A$-Lanczos Bi-orthogonalization algorithm [9–11], where $K_k(A, \boldsymbol{r}_0)$ and $K_k(A^{\mathrm{T}}, \boldsymbol{r}_0^*)$ denote the k-th Krylov subspace. From this approach, we learn that $\boldsymbol{p}_k$ and $\boldsymbol{p}_k^*$ satisfy

$$(A\boldsymbol{p}_i, A^{\mathrm{T}}\boldsymbol{p}_j^*) = 0, \quad (i \neq j). \tag{3}$$

Using the property (3) and induction, we have the orthogonality for vectors $\boldsymbol{r}_k$ and $A^{\mathrm{T}}\boldsymbol{p}_i^*$

$$(\boldsymbol{r}_k, A^{\mathrm{T}}\boldsymbol{p}_i^*) = 0, \quad (i < k). \tag{4}$$

Combining the conditions (3) and (4) yields

$$(\boldsymbol{r}_k, A^{\mathrm{T}}\boldsymbol{r}_i^*) = 0, \quad (i < k). \tag{5}$$

From the properties (3)–(5), we obtain the recurrence coefficients α_k and β_k.

$$\alpha_k = \frac{(A\boldsymbol{r}_k, \boldsymbol{r}_k^*)}{(A\boldsymbol{p}_k, A^{\mathrm{T}}\boldsymbol{p}_k^*)}, \qquad \beta_k = \frac{(A\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1}^*)}{(A\boldsymbol{r}_k, \boldsymbol{r}_k^*)}. \tag{6}$$

We would like to remark that Sogabe et al. have proposed BiCR [16]. Their approach is different from ours based on $A^{\mathrm{T}}A$-Lanczos Bi-orthogonalization algorithm, but the expressions for the recurrence coefficients α_k and β_k are the same.

4. BiCR variants of hybrid BiCG methods

This section derives the recurrence coefficients α_k and β_k computed in BiCR variants of the hybrid BiCG methods as in [1], and the algorithms are proposed.

The residuals $\boldsymbol{r}_k^{\mathrm{hyb}}$ of the hybrid BiCR variants are defined as

$$\boldsymbol{r}_k^{\mathrm{hyb}} := S_k(A)\boldsymbol{r}_k = S_k(A)R_k(A)\boldsymbol{r}_0. \tag{7}$$

In other words, the residual vector of the hybrid BiCR variants is expressed by a product of the residual of BiCR and the polynomial $S_k(\lambda)$, which has been used in the original hybrid BiCG methods for making the convergence fast. A new auxiliary vector $\boldsymbol{p}_k^{\mathrm{hyb}}$, which is a product of the auxiliary vector $\boldsymbol{p}_k$ of BiCR and the polynomial $S_k(\lambda)$, is introduced by

$$\boldsymbol{p}_k^{\mathrm{hyb}} := S_k(A)\boldsymbol{p}_k = S_k(A)P_k(A)\boldsymbol{r}_0. \tag{8}$$

Now the recurrence coefficients α_k and β_k of the hybrid BiCR variants are determined as follows. First, we define the inner product ρ_k' as

$$\begin{aligned}\rho_k' &:= (AS_k(A)R_k(A)\boldsymbol{r}_0, \boldsymbol{r}_0^*) = (AR_k(A)\boldsymbol{r}_0, S_k(A^{\mathrm{T}})\boldsymbol{r}_0^*) \\ &= \left(AR_k(A)\boldsymbol{r}_0, (-1)^k \prod_{i=0}^{k-1} \zeta_i (A^{\mathrm{T}})^k \boldsymbol{r}_0^* + \boldsymbol{q}_1\right),\end{aligned}$$

where $\boldsymbol{q}_1 \in K_{k-1}(A^{\mathrm{T}}, \boldsymbol{r}_0^*)$, and ζ_i $(i = 0, \dots, k-1)$ is the leading coefficient for the polynomial $S_k(\lambda)$. The vectors $\boldsymbol{r}_k$ and $\boldsymbol{r}_k^*$ satisfy the condition (5). In other words, $AR_k(A)\boldsymbol{r}_0$ is orthogonal to $(A^{\mathrm{T}})^i\boldsymbol{r}_0^*$ $(i < k)$. In addition, the relation $R_k(A^{\mathrm{T}}) = (-1)^k \prod_{i=0}^{k-1} \alpha_i (A^{\mathrm{T}})^k \boldsymbol{r}_0^* + \boldsymbol{q}_2$ holds, where $\boldsymbol{q}_2 \in K_{k-1}(A^{\mathrm{T}}, \boldsymbol{r}_0^*)$. Hence, ρ_k' can be written as

$$\rho_k' = \frac{\prod_{i=0}^{k-1} \zeta_i}{\prod_{i=0}^{k-1} \alpha_i} (AR_k(A)\boldsymbol{r}_0, R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*). \tag{9}$$

Using the inner product ρ_k', the recurrence coefficient β_k of BiCR can be rewritten as

$$\beta_k = \frac{(A\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1}^*)}{(A\boldsymbol{r}_k, \boldsymbol{r}_k^*)} = \frac{(AR_{k+1}(A)\boldsymbol{r}_0, R_{k+1}(A^{\mathrm{T}})\boldsymbol{r}_0^*)}{(AR_k(A)\boldsymbol{r}_0, R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)} = \frac{\alpha_k}{\zeta_k}\frac{\rho_{k+1}'}{\rho_k'}. \tag{10}$$

Eqs. (9) and (10), together with $\boldsymbol{r}_k^{\mathrm{hyb}}$ defined in (7), yield the recurrence coefficient β_k of the hybrid BiCR variants as follows:

$$\beta_k = \frac{\alpha_k}{\zeta_k}\frac{(AS_{k+1}(A)R_{k+1}(A)\boldsymbol{r}_0, \boldsymbol{r}_0^*)}{(AS_k(A)R_k(A)\boldsymbol{r}_0, \boldsymbol{r}_0^*)} = \frac{\alpha_k}{\zeta_k}\frac{(A\boldsymbol{r}_{k+1}^{\mathrm{hyb}}, \boldsymbol{r}_0^*)}{(A\boldsymbol{r}_k^{\mathrm{hyb}}, \boldsymbol{r}_0^*)}. \tag{11}$$

However, the use of Eq. (11) to update β_k of the hybrid BiCR variants entails a high computational cost; therefore, in practice, the recurrence coefficient β_k of the hybrid BiCR variants is updated by

$$\beta_k = \frac{\alpha_k}{\zeta_k}\frac{(\boldsymbol{r}_{k+1}^{\mathrm{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}{(\boldsymbol{r}_k^{\mathrm{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}.$$

Since $\zeta_k = \alpha_k$ holds for the BiCR variant of CGS, the recurrence coefficient β_k is simplified to $\beta_k = \frac{(\boldsymbol{r}_{k+1}^{\mathrm{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}{(\boldsymbol{r}_k^{\mathrm{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}$.

Since the recurrence coefficient α_k of BiCR is given by the first expression in (6), it can be converted to the recurrence coefficient α_k of the hybrid BiCR variants using $\boldsymbol{r}_k^{\text{hyb}}$ and $\boldsymbol{p}_k^{\text{hyb}}$, which are defined in (7) and (8), using the same analogy to derive the recurrence coefficients as was used in [17,19]. This is shown in the following.

$$\begin{aligned}\alpha_k &= \frac{(A\boldsymbol{r}_k, \boldsymbol{r}_k^*)}{(A\boldsymbol{p}_k, A^{\mathrm{T}}\boldsymbol{p}_k^*)} = \frac{(AR_k(A)\boldsymbol{r}_0, R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)}{(AP_k(A)\boldsymbol{r}_0, A^{\mathrm{T}}P_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)} = \frac{(AR_k(A)\boldsymbol{r}_0, R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)}{(AP_k(A)\boldsymbol{r}_0, A^{\mathrm{T}}R_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)} \\ &= \frac{(AR_k(A)\boldsymbol{r}_0, S_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)}{(AP_k(A)\boldsymbol{r}_0, A^{\mathrm{T}}S_k(A^{\mathrm{T}})\boldsymbol{r}_0^*)} = \frac{(AS_k(A)R_k(A)\boldsymbol{r}_0, \boldsymbol{r}_0^*)}{(AS_k(A)P_k(A)\boldsymbol{r}_0, A^{\mathrm{T}}\boldsymbol{r}_0^*)} \\ &= \frac{(A\boldsymbol{r}_k^{\text{hyb}}, \boldsymbol{r}_0^*)}{(A\boldsymbol{p}_k^{\text{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}. \end{aligned} \tag{12}$$

Since the computational costs of the hybrid BiCR variants per iteration incurred with (12) are high, in practice, the recurrence coefficient α_k of the hybrid BiCR variants is computed by

$$\alpha_k = \frac{(\boldsymbol{r}_k^{\text{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}{(A\boldsymbol{p}_k^{\text{hyb}}, A^{\mathrm{T}}\boldsymbol{r}_0^*)}.$$

Consequently, the algorithms of BiCR variants of the hybrid BiCG methods can be obtained by replacing the shadow residual $\boldsymbol{r}_0^*$ with $A^{\mathrm{T}}\boldsymbol{r}_0^*$; $\boldsymbol{r}_0^*$ is used in the denominator and the numerator of the expressions to compute the parameters α_k and β_k in the original hybrid BiCG algorithms. Sleijpen et al. have shown in [6] that the convergence behavior of one hybrid BiCG method (variant of CGS) is improved when the initial shadow residual in the hybrid BiCG method is set to $A^{\mathrm{T}}\boldsymbol{r}_0$ or to the product of A^{T} and a random vector. The BiCR variant algorithms of CGS, BiCGSTAB, GPBiCG and BiCGstab(l) will be referred to as the Conjugate Residual Squared (CRS) method, the BiCR STABilized (BiCRSTAB) method, the Generalized Product-type (GPBiCR) method derived from BiCR and the BiCRstab(l) method, respectively. The hybrid BiCR variants have the same computational costs as the original hybrid BiCG methods.

Preconditioning can be included by replacing Eq. (1) with

$$(K_1^{-1}AK_2^{-1})(K_2\boldsymbol{x}) = K_1^{-1}\boldsymbol{b},$$

and applying the algorithms to the preconditioned systems $\tilde{A}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}$ with $\tilde{A} = K_1^{-1}AK_2^{-1}$, $\tilde{\boldsymbol{x}} = K_2\boldsymbol{x}$ and $\tilde{\boldsymbol{b}} = K_1^{-1}\boldsymbol{b}$, where $A \approx K = K_1K_2$ holds and K is a preconditioner. Since the focus of this paper is convergence and stability of the BiCR variant algorithms, we will not further discuss preconditioning.

5. The factor in the loss of convergence speed

In the left panel of Fig. 1 we see that the hybrid BiCR variants can be much more effective than the BiCG counterpart. For the origin of the linear system that is solved for this figure, we refer to Section 6.1. The convergence plots show the number of matrix-vector products (on the horizontal axis) versus the relative residual 2-norms ($\log_{10} \|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$). The iteration was started with random vector $\boldsymbol{x}_0$. The initial shadow residual was set to $\boldsymbol{r}_0^* = \boldsymbol{r}_0$.

In order to find out whether we would have obtained the same improvement in exact arithmetic, our numerical experiments are carried out using quadruple-precision floating-point arithmetic (the previous experiments, as all others in this paper, were done in double-precision floating-point arithmetic). The results are displayed in the right panel of Fig. 1. We can observe that the improvements in performance are not as impressive as before: with increased precision arithmetic, the numerical results of BiCRSTAB and BiCGSTAB are comparable. We therefore conclude that the improvements of the hybrid BiCR variants have to be contributed to better stability properties. It is also remarkable that, in contrast to BiCGSTAB, BiCRSTAB appears to achieve almost optimal convergence already in double-precision floating-point arithmetic. In this section, we analyze the effect of rounding errors to the hybrid BiCR variants and the hybrid BiCG methods.

The residual vectors of the hybrid BiCR variants and the hybrid BiCG methods are expressed by the product of the polynomial $S_k(\lambda)$ and the residual vectors of BiCR and BiCG, respectively. Thus the following two-term recurrence formulas can be considered to be used as a part of the hybrid BiCR variants and the hybrid BiCG methods for updating the BiCR and BiCG part. In other words, (13) is connected with the convergence behaviors of our proposed hybrid BiCR variants and the original hybrid BiCG methods.

$$\boldsymbol{p}_{k+1} = \boldsymbol{r}_k - \beta_k\boldsymbol{p}_k, \qquad \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_kA\boldsymbol{p}_{k+1}, \tag{13}$$

where $\alpha_k = \frac{(\boldsymbol{r}_k,\boldsymbol{r}_0^*)}{(A\boldsymbol{p}_{k+1},\boldsymbol{r}_0^*)}$ ($\frac{(\boldsymbol{r}_k,A^{\mathrm{T}}\boldsymbol{r}_0^*)}{(A\boldsymbol{p}_{k+1},A^{\mathrm{T}}\boldsymbol{r}_0^*)}$ in the case of the hybrid BiCR variants) and $\beta_k = \alpha_{k-1}\frac{(\boldsymbol{r}_k,\boldsymbol{r}_0^*)}{(\boldsymbol{r}_{k-1},\boldsymbol{r}_0^*)}$ ($\alpha_{k-1}\frac{(\boldsymbol{r}_k,A^{\mathrm{T}}\boldsymbol{r}_0^*)}{(\boldsymbol{r}_{k-1},A^{\mathrm{T}}\boldsymbol{r}_0^*)}$ in the case of the hybrid BiCR variants).

We use scaled quantities in our analysis because, for convergence, it is important to know how much *directions* (i.e., the normalized variants) of the vectors $\boldsymbol{p}_k$ and $\boldsymbol{r}_k$ are affected by rounding errors. Now let $(\boldsymbol{r}_k, \boldsymbol{r}_0^*)$ and $(A\boldsymbol{p}_k, \boldsymbol{r}_0^*)$ ($(\boldsymbol{r}_k, A^{\mathrm{T}}\boldsymbol{r}_0^*)$ and $(A\boldsymbol{p}_k, A^{\mathrm{T}}\boldsymbol{r}_0^*)$ in the case of the hybrid BiCR variants) be denoted by ρ_k and σ_k, respectively. The recurrence coefficients α_k and β_k can be written by $\alpha_k = \frac{\rho_k}{\sigma_{k+1}}$ and $\beta_k = \frac{\rho_k}{\sigma_k}$. The scaled ρ_k and σ_k are defined as $\hat{\rho}_k := \frac{\rho_k}{\|\boldsymbol{r}_k\|\|\boldsymbol{r}_0^*\|}$ and $\hat{\sigma}_k := \frac{\sigma_k}{\|A\boldsymbol{p}_k\|\|\boldsymbol{r}_0^*\|}$, which

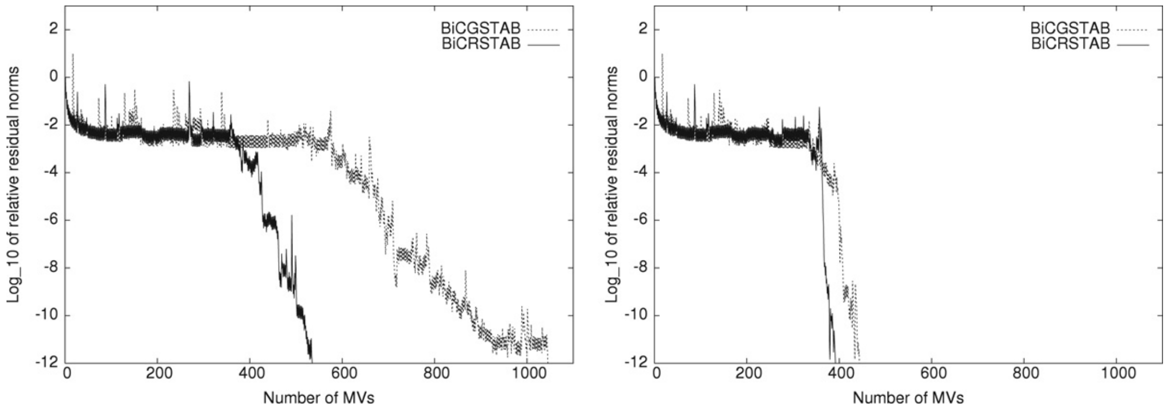

Fig. 1. Convergence histories of BiCGSTAB and BiCRSTAB for $(\gamma, \beta) = (100, -50)$ in double-precision (on the left) and quadruple-precision (on the right) floating-point arithmetic.

represent the angles between the initial shadow residual vector $\boldsymbol{r}_0^*$ and the vectors $\boldsymbol{r}_k$ and $A\boldsymbol{p}_k$, respectively [14]. Moreover, let $\hat{\boldsymbol{p}}_{k+1}$ and $\hat{\boldsymbol{r}}_{k+1}$ denote the normalized vectors of $\boldsymbol{p}_{k+1}$ and $\boldsymbol{r}_{k+1}$, respectively. Eq. (13) can be converted as follows:

$$\begin{aligned}\hat{\boldsymbol{p}}_{k+1} &:= \frac{\boldsymbol{p}_{k+1}}{\|\boldsymbol{p}_{k+1}\|} = \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{p}_{k+1}\|}\left(\frac{\boldsymbol{r}_k}{\|\boldsymbol{r}_k\|} - \frac{\boldsymbol{p}_k}{\|\boldsymbol{p}_k\|}\frac{\rho_k}{\|\boldsymbol{r}_k\|}\frac{\|\boldsymbol{p}_k\|}{\sigma_k}\right)\\ &= \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{p}_{k+1}\|}\left(\hat{\boldsymbol{r}}_k - \hat{\boldsymbol{p}}_k\frac{\|\boldsymbol{p}_k\|}{\|A\boldsymbol{p}_k\|}\frac{\hat{\rho}_k}{\hat{\sigma}_k}\right),\\ \hat{\boldsymbol{r}}_{k+1} &:= \frac{\boldsymbol{r}_{k+1}}{\|\boldsymbol{r}_{k+1}\|} = \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_{k+1}\|}\left(\frac{\boldsymbol{r}_k}{\|\boldsymbol{r}_k\|} - \frac{A\boldsymbol{p}_{k+1}}{\|A\boldsymbol{p}_{k+1}\|}\frac{\rho_k}{\|\boldsymbol{r}_k\|}\frac{\|A\boldsymbol{p}_{k+1}\|}{\sigma_{k+1}}\right)\\ &= \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_{k+1}\|}\left(\hat{\boldsymbol{r}}_k - \widehat{A\boldsymbol{p}}_{k+1}\frac{\hat{\rho}_k}{\hat{\sigma}_{k+1}}\right).\end{aligned}$$

ρ_k in finite precision arithmetic floating point can be written by

$$fl(\rho_k) = \rho_k + (|\boldsymbol{r}_k|, |\boldsymbol{r}_0^*|)n\xi = \rho_k + \|\boldsymbol{r}_k\|\|\boldsymbol{r}_0^*\|n\xi,$$

where the ξ is some value in $[-\mathbf{u}, \mathbf{u}]$, and $\mathbf{u}$ is the machine precision. The ξ at different locations may have different values but in $[-\mathbf{u}, \mathbf{u}]$. Similarly, σ_k in finite precision arithmetic floating point is expressed by

$$fl(\sigma_k) = \sigma_k + \|A\boldsymbol{p}_k\|\|\boldsymbol{r}_0^*\|n\xi.$$

Below, for ease of notation, with $fl(\cdot)$ we refer to the exact quantity, obtained by the defining expressions, in which ρ_k and σ_k have been replaced by $fl(\rho_k)$ and $fl(\sigma_k)$. For instance, since $\hat{\rho}_k$ and $\hat{\sigma}_k$ are equal to $\frac{\rho_k}{\|\boldsymbol{r}_k\|\|\boldsymbol{r}_0^*\|}$ and $\frac{\sigma_k}{\|A\boldsymbol{p}_k\|\|\boldsymbol{r}_0^*\|}$, respectively, we have

$$fl(\hat{\rho}_k) = \frac{fl(\rho_k)}{\|\boldsymbol{r}_k\|\,\|\boldsymbol{r}_0^*\|} = \hat{\rho}_k + n\xi, \qquad fl(\hat{\sigma}_k) = \frac{fl(\sigma_k)}{\|A\boldsymbol{p}_k\|\,\|\boldsymbol{r}_0^*\|} = \hat{\sigma}_k + n\xi. \tag{14}$$

The convergence behaviors of our proposed hybrid BiCR variants and the original hybrid BiCG methods are influenced by rounding errors that arise from ρ_k and σ_k. We analyze the effect of only these errors at the updated vectors $\boldsymbol{p}_{k+1}$ and $\boldsymbol{r}_{k+1}$. By using (14), $fl(\frac{\hat{\rho}_k}{\hat{\sigma}_k})$ and $fl(\frac{\hat{\rho}_k}{\hat{\sigma}_{k+1}})$ can be written as follows:

$$fl\left(\frac{\hat{\rho}_k}{\hat{\sigma}_k}\right) = \frac{\hat{\rho}_k + n\xi}{\hat{\sigma}_k + n\xi} = \frac{\hat{\rho}_k}{\hat{\sigma}_k} + \left(\frac{1}{|\hat{\sigma}_k|} + \frac{|\hat{\rho}_k|}{|\hat{\sigma}_k|^2}\right)n\xi, \tag{15}$$

$$fl\left(\frac{\hat{\rho}_k}{\hat{\sigma}_{k+1}}\right) = \frac{\hat{\rho}_k + n\xi}{\hat{\sigma}_{k+1} + n\xi} = \frac{\hat{\rho}_k}{\hat{\sigma}_{k+1}} + \left(\frac{1}{|\hat{\sigma}_{k+1}|} + \frac{|\hat{\rho}_k|}{|\hat{\sigma}_{k+1}|^2}\right)n\xi. \tag{16}$$

Eqs. (15) and (16) yield $fl(\hat{\boldsymbol{p}}_{k+1})$ and $fl(\hat{\boldsymbol{r}}_{k+1})$:

$$fl(\hat{\boldsymbol{p}}_{k+1}) = \hat{\boldsymbol{p}}_{k+1} + \hat{\boldsymbol{p}}_k\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{p}_{k+1}\|}\frac{\|\boldsymbol{p}_k\|}{\|A\boldsymbol{p}_k\|}\left(\frac{1}{|\hat{\sigma}_k|} + \frac{|\hat{\rho}_k|}{|\hat{\sigma}_k|^2}\right)n\xi, \tag{17}$$

$$fl(\hat{\boldsymbol{r}}_{k+1}) = \hat{\boldsymbol{r}}_{k+1} + \widehat{A\boldsymbol{p}}_k\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_{k+1}\|}\left(\frac{1}{|\hat{\sigma}_{k+1}|} + \frac{|\hat{\rho}_k|}{|\hat{\sigma}_{k+1}|^2}\right)n\xi. \tag{18}$$

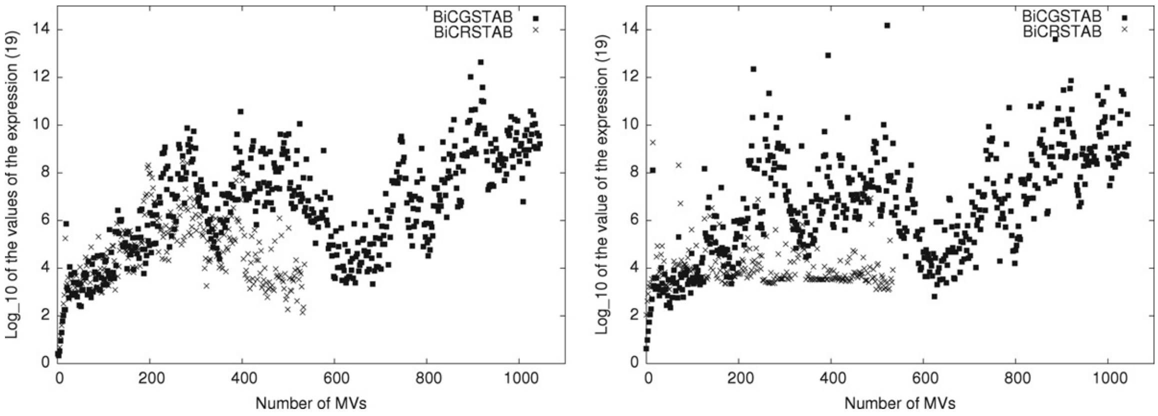

Fig. 2. Values of the first (on the left) and second (on the right) expressions of (19) when solved by BiCGSTAB and BiCRSTAB for $(\gamma, \beta) = (100, -50)$.

If the values of (19) are large, (17) and (18) show that the directions of $\boldsymbol{p}_{k+1}$ and $\boldsymbol{r}_{k+1}$ are affected by large errors. (Note that the parameter α_k is computed such that $\boldsymbol{r}_{k+1}$ and $A\boldsymbol{p}_k$ are linearly independent). We expect that this will lead to the loss of convergence speed.

$$\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{p}_{k+1}\|}\frac{\|\boldsymbol{p}_k\|}{\|A\boldsymbol{p}_k\|}\left(\frac{1}{|\hat{\sigma}_k|}+\frac{|\hat{\rho}_k|}{|\hat{\sigma}_k|^2}\right), \qquad \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_{k+1}\|}\left(\frac{1}{|\hat{\sigma}_{k+1}|}+\frac{|\hat{\rho}_k|}{|\hat{\sigma}_{k+1}|^2}\right). \tag{19}$$

In Fig. 2, we display the values of the first and second expressions in (19) for the example used in Fig. 1. The plots show the number of matrix-vector products (on the horizontal axis) versus common logarithm of (19).

From Fig. 2 we can observe the following: the value of the second expression in (19) calculated by BiCRSTAB becomes about 10^9, while the value calculated by BiCGSTAB reaches 10^{14}. The gap of the values between BiCRSTAB and BiCGSTAB is large. Moreover, the values of the first expression in (19) provided by BiCRSTAB and BiCGSTAB become about 10^8 and 10^{10}, respectively. The results show that rounding errors that arise from ρ_k and σ_k lead to very inaccurate residual directions in some steps (only one digit is accurate versus six digits for BiCRSTAB), which explain the loss of convergence speed.

We would like to remark that the quantities that determine the loss of convergence speed are isolated (see (19)), but we do not reveal how the values of (19) are related to the initial shadow residual $\boldsymbol{r}_0^*$ or $A^{\mathrm{T}}\boldsymbol{r}_0^*$.

6. Numerical experiments

In this section, we present numerical experiments on model problems with nonsymmetric matrices, and reveal the merits of the hybrid BiCR variants.

Numerical calculations were carried out in double-precision floating-point arithmetic on a PC with an Intel Xeon 3.06 GHz processor equipped with a Fujitsu Fortran compiler in Section 6.1 and on a PC with an AMD Opteron 148 2.2 GHz processor equipped with a GNU Fortran compiler in Section 6.2. The iterations of solvers were stopped when the relative residual norms (i.e., $\|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$) become 10^{-12}.

6.1. *Example 1*

As shown in [12], applying 5-point central differences to the two-dimensional convection–diffusion equation

$$-u_{xx} - u_{yy} + \gamma(xu_x + yu_y) + \beta u = f(x, y)$$

over the unit square $\Omega = (0, 1) \times (0, 1)$ with the Dirichlet boundary conditions $u|_{\partial u} = 0$ yields a linear system with a nonsymmetric matrix. The mesh size is chosen as $101(= M + 1)$ in both directions of Ω, so that the resulting system has the $M^2 \times M^2$ coefficient matrix. The right-hand side vector $\boldsymbol{b}$ is taken by substituting a vector $\hat{\boldsymbol{x}} = (1, \ldots, 1)^{\mathrm{T}}$ into the equation $\boldsymbol{b} = A\hat{\boldsymbol{x}}$. The linear system with a nonsymmetric coefficient matrix is solved by CGS, CRS, BiCGSTAB, BiCRSTAB, GPBiCG, GPBiCR, BiCGstab(2) and BiCRstab(2), and then the convergence behaviors between CGS and CRS, BiCGSTAB and BiCRSTAB, GPBiCG and GPBiCR, and BiCGstab(2) and BiCRstab(2) are compared. The numerical computation was carried out for $(\gamma, \beta) = (50, -30)$, $(50, -50)$, $(100, -30)$ and $(100, -50)$. The iteration was started with random vector $\boldsymbol{x}_0$. The initial shadow residual was set to $\boldsymbol{r}_0^* = \boldsymbol{r}_0$.

Table 1 shows the number of matrix-vector products (abbreviated by *MVs*) and the computation time (abbreviated as *Time* (second)) required to obtain the successful convergence, and the explicitly computed residual norm $\log_{10}(\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2/\|\boldsymbol{b} - A\boldsymbol{x}_0\|_2)$ (abbreviated as *True res.*) at the final iteration. The data written in bold face in Table 1 indicates the best result of each method in four cases with $(\gamma, \beta) = (50, -30)$, $(50, -50)$, $(100, -30)$ and $(100, -50)$. We display the convergence

Table 1
Number of MVs, computation time and explicitly computed relative residual norm for (50, −30) (on the upper left), (50, −50) (on the upper right), (100, −30) (on the lower left) and (100, −50) (on the lower right).

(50, −30)	MVs	Time	True res.	(50, −50)	MVs	Time	True res.
CGS	468	0.270	−12.0	CGS	496	0.273	−11.8
CRS	412	0.264	−12.4	CRS	**422**	**0.261**	−12.2
BiCGSTAB	682	0.344	−12.3	BiCGSTAB	712	0.365	−12.0
BiCRSTAB	486	0.254	−12.0	BiCRSTAB	452	0.242	−12.8
GPBiCG	740	0.632	−12.2	GPBiCG	556	0.485	−12.2
GPBiCR	588	0.502	−12.2	GPBiCR	482	0.413	−12.0
BiCGSTAB(2)	660	0.404	−11.4	BiCGstab(2)	652	0.417	−12.1
BiCRSTAB(2)	496	0.306	−12.1	BiCRstab(2)	516	0.327	−11.8
(100, −30)	MVs	Time	True res.	(100, −50)	MVs	Time	True res.
CGS	536	0.296	−11.8	CGS	532	0.308	−11.6
CRS	560	0.352	−11.5	CRS	490	0.309	−12.1
BiCGSTAB	1738	0.867	−12.0	BiCGSTAB	1046	0.513	−12.4
BiCRSTAB	**572**	**0.300**	−12.3	BiCRSTAB	536	0.275	−12.5
GPBiCG	816	0.717	−12.4	GPBiCG	846	0.769	−12.0
GPBiCR	572	0.491	−12.0	GPBiCR	**570**	**0.490**	−12.5
BiCGSTAB(2)	1120	0.689	−12.6	BiCGSTAB(2)	684	0.425	−12.1
BiCRSTAB(2)	**548**	**0.338**	−12.5	BiCRSTAB(2)	588	0.362	−12.0

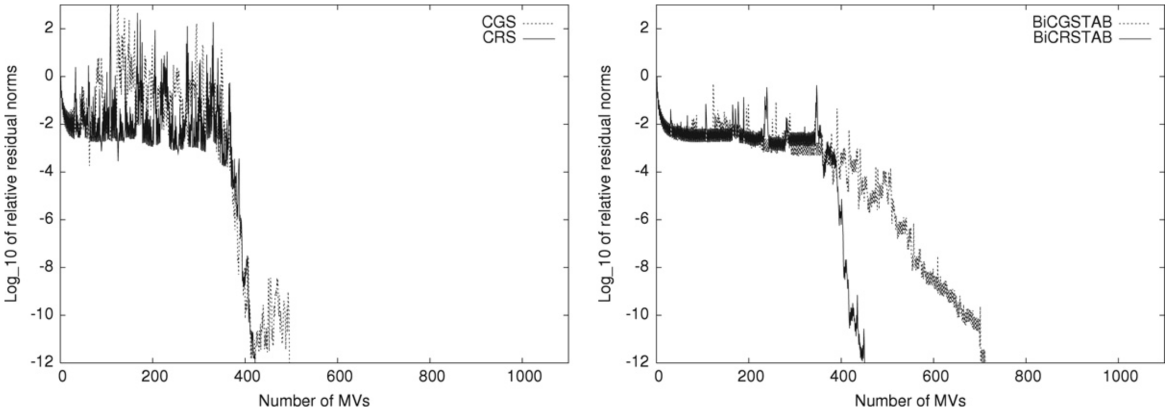

Fig. 3. Convergence histories of CGS, CRS (on the left) and BiCGSTAB, BiCRSTAB (on the right) for $(\gamma, \beta) = (50, -50)$.

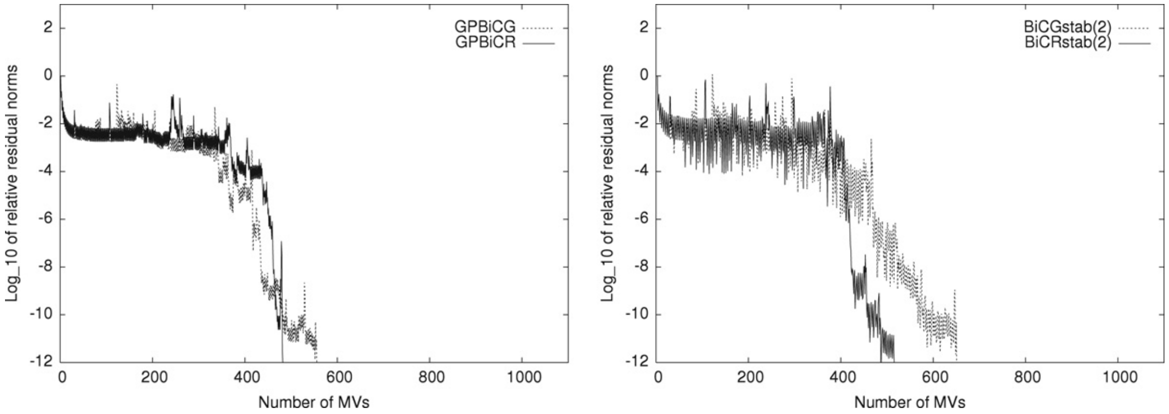

Fig. 4. Convergence histories of GPBiCG, GPBiCR (on the left) and BiCGstab(2), BiCRstab(2) (on the right) for $(\gamma, \beta) = (50, -50)$.

behavior for the parameters $(\gamma, \beta) = (50, -50)$ and $(100, -50)$ in Figs. 3–6, respectively. The convergence plots show the number of MVs (on the horizontal axis) versus the relative residual 2-norms ($\log_{10} \|\boldsymbol{r}_k\|_2/\|\boldsymbol{r}_0\|_2$).

From Table 1 and Figs. 3–6, we can observe the following: the number of MVs and computation time required to obtain successful convergence when using CRS are at most 85% and as long as 95% of those required for successful convergence using CGS. The number of MVs and computation time required to obtain successful convergence when using BiCRSTAB or BiCRstab(2) are at most 33% or 49% of those required for successful convergence using BiCGSTAB or BiCGstab(2). The number

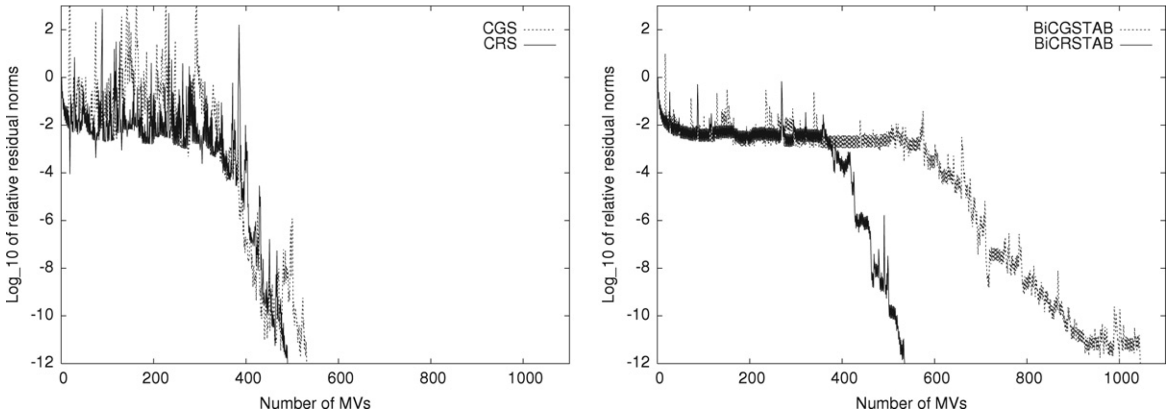

Fig. 5. Convergence histories of CGS, CRS for (on the left) and BiCGSTAB, BiCRSTAB (on the right) for $(\gamma, \beta) = (100, -50)$.

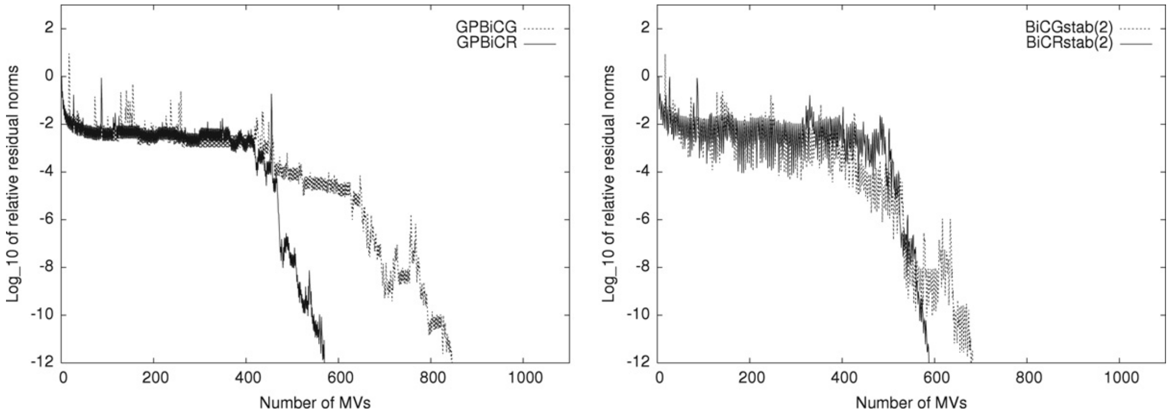

Fig. 6. Convergence histories of GPBiCG, GPBiCR (on the left) and BiCGstab(2), BiCRstab(2) (on the right) for $(\gamma, \beta) = (100, -50)$.

Table 2
Characteristics of coefficient matrices.

Matrices	*N*	*NNZ*	Ave. *NNZ*
ADD20	2 395	13 651	5.5
MEMPLUS	17 758	99 147	5.6

of MVs and computation time required to obtain successful convergence when using GPBiCR are at most 67% and as long as 64% of those required for successful convergence using GPBiCG. Consequently, our proposed hybrid BiCR variants converge faster and are more effective than the original hybrid BiCG methods.

We note that GMRES(20) and GMRES(50) did not converge, although we continued the iterations till 3000 MVs. Moreover, the numbers of MVs required to obtain successful convergence when using GMRES(100) are 1097, 1290, 953 and 1248, and the computation times required to obtain successful convergence using GMRES(100) are 3.01 second, 3.50 second, 2.56 second and 3.50 second, for $(\gamma, \beta) = (50, -30)$, $(50, -50)$, $(100, -30)$ and $(100, -50)$, respectively.

6.2. Example 2

We use matrices ADD20 (circuit simulation problem) and MEMPLUS (circuit simulation problem) from Tim Davis' Sparse Matrix Collection [3] and Matrix Market [2]. Table 2 shows the dimensions of the matrices (indicated by N), the number of nonzero entries (abbreviated as *NNZ*) and the average number of nonzero entries in each row (abbreviated as *Ave. NNZ*). The right-hand side vector is set as $\boldsymbol{b} = (1, \ldots, 1)^{\mathrm{T}}$. The linear systems with coefficient matrices described in Table 2 are solved by CGS, CRS, BiCGSTAB, BiCRSTAB, GPBiCG, GPBiCR, BiCGstab(2) and BiCRstab(2), and then the convergence behaviors between CGS and CRS, BiCGSTAB and BiCRSTAB, GPBiCG and GPBiCR, and BiCGstab(2) and BiCRstab(2) are compared. The iteration was started with random vector $\boldsymbol{x}_0$. The initial shadow residual was set to $\boldsymbol{r}_0^* = \boldsymbol{r}_0$.

Table 3 shows the number of matrix-vector products (abbreviated as *MVs*) and the computation time (abbreviated as *Time* (second)) required to obtain the successful convergence, and the explicitly computed residual norm $\log_{10}(\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2/\|\boldsymbol{b} -$

Table 3
Number of MVs, computation time and explicitly computed relative residual norm (displayed in order of ADD20 and MEMPLUS).

ADD20	MVs	Time	True res.	MEMPLUS	MVs	Time	True res.
CGS	878	0.223	−11.9	CGS	2838	6.18	−7.9
CRS	782	0.199	−11.9	CRS	**2476**	**5.13**	−11.4
BiCGSTAB	1932	0.487	−11.7	BiCGSTAB	6310	13.07	−11.1
BiCRSTAB	**1640**	**0.415**	−11.6	BiCRSTAB	5706	11.78	−11.1
GPBiCG	1266	0.403	−11.7	GPBiCG	4140	10.58	−11.1
GPBiCR	1244	0.394	−11.7	GPBiCR	**3834**	**9.71**	−11.0
BiCGstab(2)	1796	0.479	−11.8	BiCGstab(2)	5136	10.81	−11.1
BiCRstab(2)	1568	0.418	−11.6	BiCRstab(2)	**4124**	**8.74**	−11.2

$\boldsymbol{A}\boldsymbol{x}_0\|_2$) (abbreviated as *True res.*) at the final iteration. The data written in bold face in Table 3 indicates the best result of each method in two problems with the matrices ADD20 and MEMPLUS.

From Table 3 we can observe the following: the number of MVs and computation time required to obtain successful convergence when using CRS are at most 87% and as long as 83% of those required for successful convergence using CGS. The number of MVs and computation time required to obtain successful convergence when using BiCRSTAB, GPBiCR or BiCRstab(2) are at most 85%, 92% or 80% of those required for successful convergence using BiCGSTAB, GPBiCG or BiCGstab(2). Moreover, the approximate solution solved by CRS is more accurate than that solved by CGS on the problem with the matrix MEMPLUS. Consequently, our proposed hybrid BiCR variants converge faster and are more effective than the original hybrid BiCG methods.

7. Concluding remarks

We have derived BiCR variants of the hybrid BiCG methods by replacing the BiCG part in the residual polynomial of the hybrid BiCG methods with BiCR. The recurrence formulas used to update an approximation and a residual vector in the hybrid BiCR variant are the same as those used in the corresponding hybrid BiCG method, but the recurrence coefficients α_k and β_k are different; they are determined so as to compute the coefficients of the residual polynomial of BiCR. Moreover, we have analyzed the factor in the loss of convergence speed. The numerical experiments reveal the following merits of our proposed hybrid BiCR variants.

(1) BiCR variants such as CRS, BiCRSTAB, GPBiCR and BiCRstab(l) converge faster than the hybrid BiCG methods.
(2) The hybrid BiCR variants are less affected by rounding errors than the original hybrid BiCG methods.
(3) The convergence behaviors of our proposed hybrid BiCR variants and the hybrid BiCG methods are influenced by rounding errors that arise from ρ_k and σ_k when updating the residual vector $\boldsymbol{r}_k$ rather than the vector $\boldsymbol{p}_k$.

Acknowledgments

We would like to express our thanks to Professor S. Fujino (Kyushu university), Professor S.-L. Zhang, Dr. T. Sogabe (Nagoya university) and Dr. M. B. van Gijzen (Delft university of Technology) for their valuable suggestions. We would like to thank the reviewers for their helpful and useful suggestions. This research was partly supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research(C), 18560064, 2008.

References

[1] K. Abe, T. Sogabe, S. Fujino, S.-L. Zhang, A product-type Krylov subspace method based on conjugate residual method for nonsymmetric coefficient matrices, Trans. IPSJ 48 (2007) 11–21. (in Japanese).
[2] R. Boisvert, R. Pozo, K. Remington, B. Miller, R. Lipman, Matrix Market: http://math.nist.gov/MatrixMarket/.
[3] T. Davis, Sparse Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/.
[4] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357.
[5] R. Fletcher, Conjugate gradient methods for indefinite systems, Lecture Notes in Mathematics 506 (1976) 73–89.
[6] D.R. Fokkema, G.L.G. Sleijpen, H.A. van der Vorst, Generalized conjugate gradient squared, J. Comput. Appl. Math. 71 (1996) 125–146.
[7] H.G. Golub, H.A. van der Vorst, Closer to the Solution: Iterative Linear Solvers, in: I.S. Duff, G.A. Watson (Eds.), The State of the Art in Numerical Analysis, Clarendon Press, Oxford, 1997, pp. 63–92.
[8] M.H. Gutknecht, Variants of BiCGStab for matrices with complex spectrum, SIAM J. Sci. Comput. 14 (1993) 1020–1033.
[9] M.H. Gutknecht, Lanczos-type solvers for nonsymmetric linear systems of equations, Acta Numer. 6 (1997) 217–397.
[10] C. Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Natl. Bur. Stand. 49 (1952) 33–53.
[11] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition, SIAM, Philadelphia, 2003.
[12] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Stat. Comput. 14 (1993) 461–469.
[13] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[14] G.L.G. Sleijpen, H.A van der Vorst, Maintaining convergence properties of BiCGstab methods in finite precision arithmetic, Numer. Algorithms 10 (1995) 202–223.
[15] G.L.G. Sleijpen, H.A. van der Vorst, D.R. Fokkema, BiCGstab(l) and other hybrid Bi-CG methods, Numer. Algorithms 7 (1994) 75–109.
[16] T. Sogabe, M. Sugihara, S.-L. Zhang, An extension of the conjugate residual method to nonsymmetric linear systems, J. Comput. Appl. Math. 226 (2009) 103–113.

[17] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Comput. 10 (1989) 36–52.

[18] E.L. Stiefel, Kernel polynomial in linear algebra and their numerical applications, in: Further Contributions to the Determination of Eigenvalues, in: NBS Applied Math. Ser., vol. 49, 1958, pp. 1–22.

[19] 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.

[20] 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.