Journal of Computational Physics 228 (2009) 6376–6394

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Lanczos-type variants of the COCR method for complex nonsymmetric linear systems

Yan-Fei Jing [a,*], Ting-Zhu Huang [a], Yong Zhang [a], Liang Li [a], Guang-Hui Cheng [a], Zhi-Gang Ren [a], Yong Duan [a], Tomohiro Sogabe [b], Bruno Carpentieri [c]

[a] *School of Applied Mathematics/Institute of Computational Science, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, PR China*
[b] *Department of Computational Science and Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan*
[c] *CRS4 Bioinformatics Laboratory, Edificio 3, Loc. Piscinamanna, 09010 Pula (CA), Italy*

A R T I C L E   I N F O

Article history:
Received 1 October 2008
Received in revised form 28 April 2009
Accepted 14 May 2009
Available online 20 May 2009

MSC:
65F10

Keywords:
Physical problems
CBiCG
COCR
Complex nonsymmetric matrices
Lanczos-type variants

A B S T R A C T

Motivated by the celebrated extending applications of the well-established complex Biconjugate Gradient (CBiCG) method to deal with large three-dimensional electromagnetic scattering problems by Pocock and Walker [M.D. Pocock, S.P. Walker, The complex Bi-conjugate Gradient solver applied to large electromagnetic scattering problems, computational costs, and cost scalings, IEEE Trans. Antennas Propagat. 45 (1997) 140–146], three Lanczos-type variants of the recent Conjugate *A*-Orthogonal Conjugate Residual (COCR) method of Sogabe and Zhang [T. Sogabe, S.-L. Zhang, A COCR method for solving complex symmetric linear systems, J. Comput. Appl. Math. 199 (2007) 297–303] are explored for the solution of complex nonsymmetric linear systems. The first two can be respectively considered as mathematically equivalent but numerically improved popularizing versions of the BiCR and CRS methods for complex systems presented in Sogabe's Ph.D. Dissertation. And the last one is somewhat new and is a stabilized and more smoothly converging variant of the first two in some circumstances. The presented algorithms are with the hope of obtaining smoother and, hopefully, faster convergence behavior in comparison with the CBiCG method as well as its two corresponding variants. This motivation is demonstrated by numerical experiments performed on some selective matrices borrowed from The University of Florida Sparse Matrix Collection by Davis.

1. Introduction

With respect to "the greatest influence on the development and practice of science and engineering in the 20th century" as written by Dongarra and Sullivan [3], Krylov subspace methods are considered as one of the "Top Ten Algorithms of the Century". Preconditioned Krylov subsace methods are one of the most widespread and extensively accepted techniques for numerical solution of today's large-scale linear systems of the form $Ax = b$. Most of those linear systems arise from various fields of computational science and engineering. Such examples are electromagnetic applications in particular by discretizations of, for instance, Helmholtz equations and Maxwell equations. Successive attempts and efforts have been made in the last half century for generalizations of the well-known Conjugate Gradient (CG) method by Hestenes and Stiefel [4] and the breakthrough Lanczos algorithm by Lanczos [5] for symmetric linear systems, leading to many advances in the development of Krylov subspace methods. The monograph by Brezinski [6] gives an elegant coverage of the intimate relations between

* Corresponding author. Fax: +86 28 83200131.
E-mail addresses: yanfeijing@uestc.edu.cn (Y.-F. Jing), tingzhuhuang@126.com, tzhuang@uestc.edu.cn (T.-Z. Huang).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.022

approximation theory and the Lanczos-type algorithms. A recent excellent and thorough review article by Simoncini and Szyld [7] has investigated the new developments of Krylov subspace methods for the iterative solution of linear systems during the last decade and a half in detail.

One such attractive extension due to Fletcher is what is called the Biconjugate Gradient (BiCG) method [8], which is often used to deal with the nonsymmetric and nonpositive-definite matrices as arising in scattering problems without the penalty of worsened condition number. Jacobs [9,10] extended the BiCG method to complex systems (referred to as the CBiCG method) with certain complex conjugate modifications; see also a similar approach by Sarkar [11] and an analysis of the functional associated with the CG method in complex arithmetic by Markham [12]. Smith et al. [13] applied the CBiCG method to small two-dimensional electromagnetic scattering problems to find it, in general, to be more efficient with respect to convergence behavior compared with the CG method. Later, Pocock and Walker [1] successfully extended the use of the CBiCG method to an isoparametric boundary integral equation formulation for three-dimensional frequency-domain electromagnetic scattering problems. These problems are especially on large and geometrically complex examples, including a 20 wavelength slender dipole, the NASA almond, and a resonant cavity. By the way, Joly and Meurant [14] derived several complex conjugate gradient-like methods from a unified framework to generalize the real case addressed in a Joly's previous paper [15] for solving complex non-Hermitian linear systems. They introduced a variant of the BiCG method different from Jacobs' method (CBiCG). Furthermore, a Biconjugate Gradient Squared (BiCGS) method was derived the same way that the ingenious Conjugate Gradients Squared (CGS) method was developed by Sonneveld [16]. In addition, for the important case where the coefficient matrix is complex symmetric, Clemens and Weiland [17] showed there exists a whole theoretically interesting class of SCBICG(Γ, n) methods. It characterizes each variant of a complex symmetric BiCG method by a set of real-valued coefficients Γ for a polynomial π of degree n which defines the initial pseudo-residual $\tilde{r}_0$ as presented therein. Practically it provides two efficient methods: one is the Conjugate Orthogonal Conjugate Gradient (COCG) method due to van der Vorst and Melissen [18], also appearing in [19,20], and the other BiCGCR method by Clemens and Weiland [21]. As noticed in [18], the COCG method often behaves like the CBiCG method, in terms of number of iterations, but with half the computational cost; refer also to [7]. Li et al. [22] capitalized upon incomplete Cholesky factorization preconditioning techniques combined with the COCG method to solve three-dimensional electromagnetic scattering problems. Clemens and Weiland [21] applied the BiCGCR method with preconditioning techniques for the solution of large sparse complex symmetric systems of linear equations arising from electromagnetic high-voltage problems.

For quite some time, there has been a growing interest in smoothing processes which are known to provide a transition from the results of the first method of a pair of orthogonal and minimal residual methods (or, biorthogonal and quasi-minimal residual methods) to those of the second one. It is with the hope that the residual norm plot becomes smoother and, hopefully, faster decreasing since far outlying iterates and residuals are avoided in the smoothed sequences. Weiss emphasized in his work [23–25] the relationship between certain pairs of orthogonal and minimal residual methods, where the results of the second method can be generated from those of the first method by applying the minimal residual smoothing process. Examples of such pairs are the CG and CR methods, the FOM and GMRES methods, as well as the CGNE and CGNR versions of applying the CG method to the normal equations. For more details of these above pairs, refer to [26]. A somewhat less general, but more specific framework from a different perspective was considered by Gutknecht and Rozložník [27]. By the way, several practitioners discussed a class of techniques known as residual smoothing; see, e.g., [28–32]. A remarkable result shown by Zhou and Walker [31] is that the iterates of the Quasi-Minimal Residual (QMR) method [33] can be obtained from those of the BiCG method as a particular case of residual smoothing. Gutknecht and Rozložník [34] presented a roundoff error analysis of smoothing algorithms to show that the ultimately attainable accuracy of the smoothed iterates, measured in the norm of the corresponding residuals, is, in general, not higher than that of the primary iterates. Around the same time, they [35] deepened the understanding of some of the known relationships in [24,25,31,36–43] between pairs of orthogonal and minimal residual methods (or, biorthogonal and quasi-minimal residual methods) by a significant interpretation of smoothing processes in coordinate space. They also introduced in the same paper a unifying framework for minimal residual and quasi-minimal residual smoothing in order to estimate how much smaller the residuals or quasi-residuals of the minimizing methods can be compared to those of the corresponding Galerkin or Petrov–Galerkin methods. As addressed extensively by Gutknecht and Rozložník [34], although the smoothed residuals do not converge considerably faster than those of the primary method in exact arithmetic, there is a useful consequence of the fact that the ultimate accuracy of the smoothed residuals is on the same level as that of the primary iterates. Thus, smoothing processes can be applied to produce certain smoothed residuals, most notably those of the minimum residual method. Those produced smoothed residuals are ultimately more accurate than those obtained by other, mathematically equivalent algorithms frequently used in practice; see also [7] for review on more QMR-type smoothing procedures and discussions of their sound residual smoothing effects.

Recently, Sogabe and Zhang [2] extended Stiefel's Conjugate Residual (CR) method [44] to the Conjugate A-Orthogonal Conjugate Residual (COCR) method for solving complex symmetric linear systems from the observation of derivations of the CG, CR, and COCG methods. The COCR method is numerically demonstrated to tend to give smoother convergence behavior than the COCG method. And it sometimes converges faster than the QMR method in terms of the number of iterations. Based on one of the simplest derivations of the BiCG method given by van der Vorst [45], the newly published BiCR method by Sogabe et al. [46] also possesses smoother convergence behavior compared to the BiCG method for real nonsymmetric linear systems. Moreover, the BiCR method often converges faster than the BiCG method learned from the involved numerical experiments. For the detailed and other derivations of the COCG and BiCR methods, one may consult Sogabe's Ph.D. Dissertation [47]. In order to improve the performance of the BiCR method, a general framework of the product-type methods was also given in [47] on the basis of the BiCR method. They include the CRS, BiCRSTAB, BiCRSTAB2, GPBiCR methods, among which only the CRS method was precisely described. Also presented in Sogabe's Ph.D. Dissertation was a stabilized CGS (SCGS) method developed to improve the convergence behavior of the CGS method. Except for the COCR method tested on complex symmetric linear systems mentioned above, all the other numerical experiments performed in [47] were focused on real nonsymmetric linear systems. But extensive numerical experiments for complex nonsymmetric linear systems were not made.

Inspired by the elegant achievement thanks to Pocock and Walker [1] mentioned before, the main attention of the present paper is to explore three Lanczos-type variants of the COCR method for the solution of complex nonsymmetric linear systems. The first two can be considered as mathematically equivalent but numerically improved popularizing versions of the BiCR and CRS methods for complex systems, respectively. And the last one is somewhat new and is a stabilized and more smoothly converging variant of the first two in some circumstances. The presented algorithms are with the motivation of obtaining smoother and, hopefully, faster convergence behavior in comparison with the CBiCG method as well as its two evolving variants—the CGS method and one of the most popular methods in use today—the Biconjugate Gradient Stabilized (BiCGSTAB) method by van der Vorst [48], when dealing with large complex nonsymmetric linear systems. Coefficient matrices of the complex nonsymmetric linear systems in our numerical experiments come from four typical and representative physical problems, including an electromagnetics problem (Dehghani/light_in_tissue), a 2D/3D problem (Kim/kim1), an acoustics problem (HB/young1c) and a thermal problem (Bindel/ted_AB_unscaled).

Bearing in mind the essential biorthogonality conditions expressed as (3.25) in [47], we reformulate the BiCR and CRS methods to obtain the first two variants with generalized constraints subspaces. Particularly, a specific initial shadow residual different from the one for the BiCR and CRS methods is selected in the implementation of our methods. The first two presented methods respectively follow similar ways for the derivations of Algorithm 6.17 (Conjugate Gradient) [26] and the CGS method. The expected convergence behaviors of the first two methods are smoother than (at least the same as) those of the BiCR and CRS methods, respectively. Because our first main method developed is strongly related to, as well as defined by, the choice of a general constraints subspace. And the standing point for the choice of the constraints subspace is revealed and suggested by a Biconjugate *A*-Orthonormalizaion Procedure described in the next section. This is a precise and explicit version of Algorithm 3.6: *A*-biorthogonalization process in [47]. Therefore, we name our first method the Biconjugate *A*-Orthogonal Residual (BiCOR) method. Accordingly, the latter two methods are called the Conjugate *A*-Orthogonal Residual Squared (CORS) method and the Biconjugate *A*-Orthogonal Residual Stabilized (BiCORSTAB) method, respectively.

The remainder of the paper is organized as follows. In Section 2, a version of the Biconjugate *A*-Orthonormalizaion Procedure is described first. And then a detailed discussion on the biconjugacy between the associated vectors for the primary and dual systems is given to provide a suggestion of alternative choices for constraints subspaces in the construction of Krylov subspace methods. The broadened constraints subspaces of the one for the BiCR method help to design the BiCOR method in Section 3. For the sake of effectiveness and efficiency enhancement of the BiCOR method, two variants of the BiCOR method—the CORS and BiCORSTAB methods are successively developed in Section 4. In Section 5, numerical experiments are made and a complete analysis is given to illustrate the superiority of our proposed methods to their counterparts related to the CBiCG and BiCR methods. Finally, concluding remarks are given in Section 6.

When it will be helpful, we will use the word "ideally" (or "mathematically") to refer to a result that could hold in exact arithmetic ignoring effects of rounding errors, and "numerically" (or "computationally") to a result of a finite precision computation.

2. A version of Biconjugate *A*-Orthonormalizaion Procedure

Sogabe introduced an *A*-biorthogonalization process in [47] for the sake of another derivation of the BiCR method. In this section, for the purpose of giving some relevant background of the BiCOR method in Section 3, a precise and explicit version of the *A*-biorthogonalization process is described as the Biconjugate *A*-Orthonormalizaion Procedure.

To characterize the procedure, the following notation is introduced first.

Denote the overbar ("–") the conjugate complex of a scalar, vector or matrix and the superscript "*T*" the transpose of a vector or matrix. For a complex nonsymmetric matrix $A = (a_{ij})_{n\times n} \in \mathbb{C}^{n\times n}$, the Hermitian conjugate of A is denoted as

$$A^H \equiv \overline{A}^T = (\bar{a}_{ji})_{n\times n}.$$

The standard Hermitian inner product of two complex vectors $u, v \in \mathbb{C}^n$ is defined as

$$\langle u, v\rangle = u^H v = \sum_{i=1}^{n} \bar{u}_i v_i.$$

The nested Krylov subspace of dimension m generated by A from v is of the form

$$\mathcal{K}_m(A, v) = \mathrm{span}\{v, Av, A^2 v, \dots, A^{m-1} v\}.$$

In addition, e_i denotes the ith column of the appropriate identity matrix.

Instead of pairs of biorthogonal bases formed by the Lanczos Biorthogonalization Prodedure [26,49], the Biconjugate *A*-Orthonormalization Procedure is exploited here to ideally build up a pair of biconjugate *A*-orthonormal bases for the dual

Krylov subspaces $\mathcal{K}_m(A, v_1)$ and $\mathcal{K}_m(A^H, w_1)$, where v_1 and w_1 are chosen initially to satisfy certain conditions. A framework that achieves the above purpose is depicted as follows:

Algorithm 1: A version of Biconjugate *A*-Orthonormalization Procedure

```
1.    Choose $v_1, w_1$, such that $\langle w_1, Av_1\rangle = 1$.
2.    Set $\beta_1 = \delta_1 \equiv 0, w_0 = v_0 \equiv \mathbf{0} \in \mathbb{C}^n$
3.    For $j = 1, 2, \ldots, m$ Do:
4.    $\alpha_j = \langle w_j, A(Av_j)\rangle$
5.    $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
6.    $\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \delta_j w_{j-1}$
7.    $\delta_{j+1} = |\langle \hat{w}_{j+1}, A\hat{v}_{j+1}\rangle|^{\frac{1}{2}}$
8.    $\beta_{j+1} = \frac{\langle \hat{w}_{j+1}, A\hat{v}_{j+1}\rangle}{\delta_{j+1}}$
9.    $v_{j+1} = \frac{\hat{v}_{j+1}}{\delta_{j+1}}$
10.   $w_{j+1} = \frac{\hat{w}_{j+1}}{\beta_{j+1}}$
11.   EndDo
```

Observe that the above algorithm is possible to break down whenever δ_{j+1} vanishes while $\hat{w}_{j+1}$ and $A\hat{v}_{j+1}$ are not equal to $\mathbf{0} \in \mathbb{C}^n$ appearing in line 7. In the interest of counteraction against such breakdowns, refer oneself to remedies such as so-called look-ahead strategies [50–53] which can enhance stability while increasing cost modestly or others, for example [54]. But that is outside the scope of this paper and we shall not pursue that here. For more details, please refer to [26] and the references therein. Throughout the rest of the present paper, suppose there is no breakdown encountered during algorithm implementations because most of our considerations concern the exploration of the three Lanczos-type variants of the COCR method for solving complex nonsymmetric linear systems.

Now the following proposition states some properties of the vectors produced by Algorithm 1, which are analogous to the properties of the *A*-biorthogonalization process in [47].

Proposition 1. *If Algorithm 1 proceeds m steps, then the right and left Lanczos-type vectors v_j, $j = 1, 2, \ldots, m$ and w_i, $i = 1, 2, \ldots, m$, form a biconjugate A-orthonormal system in exact arithmetic, i.e.,*

$$\langle w_i, Av_j\rangle = \delta_{i,j}, \quad 1 \leqslant i, j \leqslant m.$$

Furthermore, denote by $V_m = [v_1, v_2, \ldots, v_m]$ and $W_m = [w_1, w_2, \ldots, w_m]$ the $n \times m$ matrices and by $\underline{T_m}$ the extended tridiagonal matrix of the form

$$\underline{T_m} = \begin{bmatrix} T_m \\ \delta_{m+1} e_m^T \end{bmatrix},$$

where

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{bmatrix},$$

whose entries are the coefficients generated during the algorithm implementation, and in which $\alpha_1, \ldots, \alpha_m$, $\beta_2, \ldots, \beta_m$ are complex while $\delta_2, \ldots, \delta_m$ positive. Then with the Biconjugate A-Orthonormalization Procedure, the following four relations hold:

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T, \tag{1}$$

$$A^H W_m = W_m T_m^H + \bar{\beta}_{m+1} w_{m+1} e_m^T, \tag{2}$$

$$W_m^H A V_m = I_m, \tag{3}$$

$$W_m^H A^2 V_m = T_m. \tag{4}$$

Proof. The biconjugate *A*-orthonormalization of the right and left Lanczos-type vectors v_j, w_i $(i, j = 1, 2, \ldots, m)$ can be shown analogously to the first biorthogonality property of Theorem 3.3.1 in [47] combined with the observation from lines 8–10 in Algorithm 1.

Relations (1) and (2) are matrix reformulations of the following equalities which are readily derived from lines 5–10 in Algorithm 1

$$Av_j = \beta_j v_{j-1} + \alpha_j v_j + \delta_{j+1} v_{j+1}, \quad j = 1, 2, \ldots, m,$$

$$A^H w_j = \delta_j w_{j-1} + \bar{\alpha}_j w_j + \bar{\beta}_{j+1} w_{j+1}, \quad j = 1, 2, \ldots, m.$$

See also the alike relations expressed as (3.65) in [47].

Relation (3) is directly given by the first part of this proposition, i.e., the biconjugate A-orthonormality between the pairs of right and left Lanczos-type vectors. Relation (4) follows by multiplying both sides of (1) by $W_m^H A$ and making use of (3) and the associated biconjugate A-orthonormality between W_m and v_{m+1}. $\square$

Given an initial guess x_0 to the complex nonsymmetric linear systems $Ax = b$, and the associated initial residual $r_0 = b - Ax_0$, we can construct a two-sided algorithm by virtue of Algorithm 1 based on the properties presented in Proposition 1. Consider an oblique projection method onto $K_m(A, v_1)$ and orthogonally to $L_m = A^H K_m(A^H, w_1)$, where $v_1 = \frac{r_0}{\|r_0\|_2}$ and w_1 is chosen arbitrarily such that $\langle w_1, Av_1\rangle \neq 0$. But w_1 is often chosen to be equal to $\frac{Av_1}{\|Av_1\|_2^2}$ subjecting to $\langle w_1, Av_1\rangle = 1$. It is worthy noting that this choice for w_1 plays a significant role in establishing the respective superiority of the BiCOR and CORS methods to the BiCR and CRS methods, as will be seen in the numerical experiments. Thus running Algorithm 1 m steps, we can seek an mth approximate solution x_m from the affine subspace $x_0 + K_m(A, v_1)$ of dimension m, by imposing the Petrov–Galerkin condition

$$b - Ax_m \perp L_m,$$

which can be mathematically written in matrix formulation as

$$\left(A^H W_m\right)^H (b - Ax_m) = 0, \tag{5}$$

where W_m is defined in Proposition 1.

Since the approximate solution can be represented as

$$x_m = x_0 + V_m y_m, \tag{6}$$

where V_m is defined in Proposition 1 and $y_m \in \mathbb{C}^n$ contains the coefficients of the linear combination. By simple substitution and computation with (4)–(6), a tridiagonal system for solving y_m is resulted as

$$T_m y_m = \beta e_1, \tag{7}$$

where T_m is formed in Proposition 1, and $\beta = \|r_0\|_2$.

Consequently, a method for complex nonsymmetric systems making use of Algorithm 1 is briefly given below:

Algorithm 2: Two-sided Biconjugate A-Orthonormalization method

```
1. Compute $r_0 = b - Ax_0$ for some initial guess $x_0$ and set $\beta = \|r_0\|_2$.
2. Start with $v_1 = \frac{r_0}{\beta}$ and choose $w_1$ such that $\langle w_1, Av_1\rangle = 1$, (for example, $w_1 = \frac{Av_1}{\|Av_1\|_2^2}$).
3. Generate the Lanczos-type vectors $v_1, v_2, \ldots, v_m$ and $w_1, w_2, \ldots, w_m$ as well as the tridiagonal matrix $T_m$ by running
   Algorithm 1 $m$ steps.
4. Compute $y_m = T_m^{-1}(\beta e_1)$ and $x_m = x_0 + V_m y_m$.
```

Analogously, we can derive the counterpart of Algorithm 2 for the solution of the corresponding dual system $A^H x^* = b^*$, where the dual approximation x_m^* is sought from the affine subspace $x_0^* + K_m(A^H, w_1)$ of dimension m by satisfying

$$b^* - A^H x_m^* \perp AK_m(A, v_1).$$

Denote $r_0^* = b^* - A^H x_0^*$ and $\beta^* = \|r_0^*\|_2$. If $w_1 = \frac{r_0^*}{\beta^*}$ and v_1 is chosen properly such that $\langle v_1, Aw_1\rangle = 1$, then the counterparts of (5)–(7) are the following:

$$(AV_m)^H (b^* - A^H x_m^*) = 0, \tag{8}$$

$$x_m^* = x_0^* + W_m y_m^*, \tag{9}$$

$$T_m^H y_m^* = \beta^* e_1, \tag{10}$$

where V_m, W_m and T_m are defined in Proposition 1 and $y_m^* \in \mathbb{C}^n$ is the coefficient vector of the dual linear combination.

Assume the LU decomposition of the tridiagonal matrix T_m is

$$T_m = L_m U_m,$$

substituting which into (6), (7) and (9), (10) gives respectively

$$x_m = x_0 + V_m (L_m U_m)^{-1}(\beta e_1) = x_0 + V_m U_m^{-1} L_m^{-1}(\beta e_1) = x_0 + P_m z_m,$$
$$x_m^* = x_0^* + W_m (U_m^H L_m^H)^{-1}(\beta^* e_1) = x_0^* + W_m (L_m^H)^{-1} (U_m^H)^{-1}(\beta^* e_1) = x_0^* + P_m^* z_m^*,$$

where $P_m = V_m U_m^{-1}, z_m = L_m^{-1}(\beta e_1)$, and $P_m^* = W_m (L_m^H)^{-1}, z_m^* = (U_m^H)^{-1}(\beta^* e_1)$.

By observation as does in the derivation of DIOM from IOM Algorithm in [26, Chapter 6], the approximation x_m and the dual approximation x_m^* can be updated, respectively, from x_{m-1} and x_{m-1}^* at each step as

$$x_m = x_{m-1} + \zeta_m p_m,$$
$$x_m^* = x_{m-1}^* + \zeta_m^* p_m^*,$$

where ζ_m and ζ_m^* are coefficients, p_m and p_m^* are the corresponding mth column vectors in P_m and P_m^* defined above, termed as the mth primary and dual direction vectors, respectively.

Then analogous to the second biorthogonality property of Theorem 3.3.1 in [47], the pairs of the primary and dual direction vectors form a biconjugate A^2-orthonormal set, i.e., $\langle p_i^*, A^2 p_j\rangle = \delta_{i,j}$ $(1 \leqslant i,j \leqslant m)$, which follows clearly from

$$(P_m^*)^H A^2 P_m = (W_m(L_m^H)^{-1})^H A^2 V_m U_m^{-1} = L_m^{-1} W_m^H A^2 V_m U_m^{-1} = L_m^{-1} T_m U_m^{-1} = L_m^{-1} L_m U_m U_m^{-1} = I_m,$$

with (4), different from the proof for the second biorthogonality property of Theorem 3.3.1 in [47].

In addition, the mth primary residual vector $r_m = b - Ax_m$ and the mth dual residual vector $r_m^* = b^* - Ax_m^*$ can be represented as

$$r_m = -\delta_{m+1} e_m^T y_m v_{m+1}, \tag{11}$$

$$r_m^* = -\bar{\beta}_{m+1} e_m^T y_m^* w_{m+1}, \tag{12}$$

by simple computation with (1), (2), (6), (7), (9) and (10).

Eqs. (11) and (12) together with (3) reveal that the primary and dual residual vectors satisfy the biconjugate A-orthogonal conditions, i.e., $\langle r_i^*, Ar_j\rangle = 0$, for $i \neq j$, which is exactly the first biorthogonality property of Theorem 3.3.1 in [47].

Now turn back to the oblique projection method mentioned before. It is known that the constraints subspace in an oblique projection method is different from the search subspace and may be totally unrelated to it. This distinction is rather important and gives rise to different types of algorithms [26]. The biconjugate A-orthogonality between the primary and dual residual vectors and the biconjugate A^2-orthonormality between the primary and dual direction vectors reveal and suggest alternative choices for the constraints subspace. This idea helps to devise the BiCOR method in the coming section.

3. A derivation of BiCOR

Given an initial guess x_0 to the considered linear system $Ax = b$, as discussed by Sogabe and Zhang [2] and Sogabe [47], many methods such as the CG, CR, COCG, BiCGCR, COCR and BiCR methods can be unified into the following coupled two-term recurrences by imposing certain conditions:

$$r_0 = b - Ax_0, \quad p_0 = r_0, \tag{13}$$

$$x_{j+1} = x_j + \alpha_j p_j, \tag{14}$$

$$r_{j+1} = r_j - \alpha_j A p_j, \tag{15}$$

$$p_{j+1} = r_{j+1} + \beta_j p_j \quad \text{for } j = 0, 1, \ldots, \tag{16}$$

where $r_j = b - Ax_j$ is the jth residual vector and p_j is the jth search direction vector. Various computational formulas of the involved parameters α_j, β_j $(j = 0, 1, \ldots)$ in the recurrences (15) and (16) may lead to different algorithms. Denoting W the underlying constraints subspace, these parameters can be determined by the following orthogonality conditions:

$$r_{j+1} \perp W \quad \text{and} \quad Ap_{j+1} \perp W. \tag{17}$$

Specifically, W is chosen to be $\mathcal{K}_m(A, r_0)$ and $A\mathcal{K}_m(A, r_0)$ to respectively generate the CG method [4] and the CR method [44] when A is Hermitian positive definite. When A is complex symmetric, $W = K_m(\bar{A}, \bar{r}_0)$ and $W = K_m(\bar{A}, \bar{A}\bar{r}_0)$, respectively, lead to the COCG method [18] and the BiCGCR method [21], while $W = \bar{A}K_m(\bar{A}, \bar{r}_0)$ leading to the COCR method [2]. For a complex nonsymmetric A, $W = K_m(A^H, r_0^*)$ leads to the CBiCG method satisfying $\langle r_0^*, r_0\rangle \neq 0$ [9,10] and $W = A^H K_m\left(A^H, r_0^*\right)$ results in the BiCR method with r_0^* chosen to be, for example, r_0 for its implementation [46,47]. It should be stressed that there were not extensive numerical experiments with complex nonsymmetric linear systems tested for the BiCR method in [46,47].

For concerned complex nonsymmetric linear system

$$Ax = b, \tag{18}$$

similar to [17], an expanded choice as suggested in Section 2, is $W = A^H K_m\left(A^H, r_0^*\right)$, where r_0^* is chosen to be equal to $P(A)r_0$, with $P(t)$ an arbitrary polynomial of certain degree with respect to the variable t and $p_0^* = r_0^*$. It should be noted that the optimal choice for the involved polynomial is in general not easily obtainable and requires some expertise and artifice. This aspect needs further research. When there is no ambiguity or other clarification, a specific default choice for W with $r_0^* = Ar_0$ is adopted in the numerical experiments in comparison with Sogabe's choice for W with $r_0^* = r_0$ in his implementation [46,47]. It is important to note that the scalars α_j, β_j $(j = 0, 1, \ldots)$ in the recurrences (14)–(16) are different from those produced by Algorithm 1. The search direction vectors p_j's here are multiples of the primary direction vectors p_j's defined in Section 2. Analogous to (3.33) and (3.34) in [47], the coupled two-term recurrences for the $(j+1)$th shadow residual vector r_{j+1}^* and the associated $(j+1)$th shadow search direction vector p_{j+1}^* can be augmented by similar relations to (15) and (16) as follows:

$$r_{j+1}^* = r_j^* - \bar{\alpha}_j A^H p_j^*, \tag{19}$$

$$p_{j+1}^* = r_{j+1}^* + \bar{\beta}_j p_j^* \quad \text{for } j = 0, 1, \ldots, \tag{20}$$

where $\bar{\alpha}_j$ and $\bar{\beta}_j$ are the conjugate complex of α_j and β_j in (15) and (16), correspondingly. Then with a certain polynomial $P(t)$ with respect to t, (17) explicitly reads

$$r_{j+1} \perp A^H K_m\left(A^H, r_0^*\right) \quad \text{and} \quad Ap_{j+1} \perp A^H K_m\left(A^H, r_0^*\right) \quad \text{with } r_0^* = P(A) r_0, \tag{21}$$

which can be reinterpreted from a practical point of view as

- the residual vectors r_i's and the shadow residual vectors r_j^*'s are biconjugate A-orthogonal to each other, i.e., $\langle r_j^*, Ar_i\rangle = \langle A^H r_j^*, r_i\rangle = 0$, for $i \neq j$;
- The search direction vectors p_i's and the shadow search direction vectors p_j^*'s form an A^2-biconjugate set, i.e., $\langle p_j^*, A^2 p_i\rangle = \langle A^H p_j^*, Ap_i\rangle = \langle (A^H)^2 p_j^*, p_i\rangle = 0$, for $i \neq j$;

which are the facts already stated in the latter part of Section 2.

Therefore, we possess the conditions to follow a similar way how Algorithm 6.17 (Conjugate Gradient) [26] was derived to determine the scalars α_j and β_j by imposing the corresponding biorthogonality and biconjugacy conditions (21) into (15), (16), (19) and (20). The idea of the derivation of the BiCOR method is essentially the same as one of the BiCR's derivations [47]. But they are different in both the deriving ways and the initial shadow residuals. We use extensively the algorithmic schemes introduced in [55] for descriptions of the present algorithms.

Making the inner product of $A^H r_j^*$ and r_{j+1} as defined by (15) yields

$$\left\langle A^H r_j^*, r_{j+1}\right\rangle = \left\langle A^H r_j^*, r_j - \alpha_j A p_j\right\rangle = 0,$$

with the biconjugate A-orthogonality between r_{j+1} and r_j^*, further resulting in

$$\alpha_j = \frac{\left\langle A^H r_j^*, r_j\right\rangle}{\left\langle A^H r_j^*, Ap_j\right\rangle},$$

where the denominator of the above right-hand side can be further modified with (20) as

$$\left\langle A^H r_j^*, Ap_j\right\rangle = \left\langle A^H p_j^* - \bar{\beta}_{j-1} A^H p_{j-1}^*, Ap_j\right\rangle = \left\langle A^H p_j^*, Ap_j\right\rangle,$$

because p_{j-1}^* and p_j are A^2-biconjugate. Then

$$\alpha_j = \frac{\left\langle A^H r_j^*, r_j\right\rangle}{\left\langle A^H p_j^*, Ap_j\right\rangle} = \frac{\left\langle r_j^*, Ar_j\right\rangle}{\left\langle A^H p_j^*, Ap_j\right\rangle}. \tag{22}$$

Similarly, writing that p_{j+1} as defined by (16) is A^2-biconjugate to p_j^* yields

$$\left\langle p_j^*, A^2 p_{j+1}\right\rangle = \left\langle A^H p_j^*, Ap_{j+1}\right\rangle = \left\langle A^H p_j^*, Ar_{j+1} + \beta_j A p_j\right\rangle = 0,$$

giving

$$\beta_j = -\frac{\left\langle A^H p_j^*, Ar_{j+1}\right\rangle}{\left\langle A^H p_j^*, Ap_j\right\rangle} = -\alpha_j \frac{\left\langle A^H p_j^*, Ar_{j+1}\right\rangle}{\left\langle r_j^*, Ar_j\right\rangle}$$

with α_j computed in (22).

Observe from (19) that

$$-\bar{\alpha}_j A^H p_j^* = r_{j+1}^* - r_j^*$$

and therefore,

$$\beta_j = \frac{\left\langle -\bar{\alpha}_j A^H p_j^*, Ar_{j+1}\right\rangle}{\left\langle r_j^*, Ar_j\right\rangle} = \frac{\left\langle r_{j+1}^* - r_j^*, Ar_{j+1}\right\rangle}{\left\langle r_j^*, Ar_j\right\rangle} = \frac{\left\langle r_{j+1}^*, Ar_{j+1}\right\rangle}{\left\langle r_j^*, Ar_j\right\rangle}, \tag{23}$$

because of the biconjugate A-orthogonality of r_j^* and r_{j+1}. Putting these relations (13)–(16), (19), (20), (22) and (23) together and taking the strategy of reducing the number of matrix-vector multiplications by introducing an auxiliary vector recurrence and changing variables adopted for the BiCR method in [47], together lead to the BiCOR method. It is a mathematically equivalent but numerically improved generalized version of the BiCR method [46,47]. The pseudocode for the left preconditioned BiCOR method with a preconditioner M is given in the following.

Algorithm 3: Left preconditioned BiCOR method

```
1.  Compute $r_0 = b - Ax_0$ for some initial guess $x_0$.
    Choose $r_0^* = P(A)r_0$ such that $\langle r_0^*, Ar_0\rangle \neq 0$, where $P(t)$ is a polynomial in $t$.
    (For example, $r_0^* = Ar_0$).
2.  for $j = 1, 2, \ldots$
3.      solve $Mz_{j-1} = r_{j-1}$
4.      if $j = 1$
5.          solve $M^H z_0^* = r_0^*$;
6.      endif
7.      $\hat{z} = Az_{j-1}$
8.      $\rho_{j-1} = \langle z_{j-1}^*, \hat{z}\rangle$
9.      if $\rho_{j-1} = 0$, method fails
10.       if $j = 1$
11.           $p_0 = z_0$
12.           $p_0^* = z_0^*$
13.           $q_0 = \hat{z}$
14.       else
15.           $\beta_{j-2} = \rho_{j-1}/\rho_{j-2}$
16.           $p_{j-1} = z_{j-1} + \beta_{j-2}p_{j-2}$
17.           $p_{j-1}^* = z_{j-1}^* + \bar{\beta}_{j-2}p_{j-2}^*$
18.           $q_{j-1} = \hat{z} + \beta_{j-2}q_{j-2}$
19.       endif
20.       $q_{j-1}^* = A^H p_{j-1}^*$
21.       solve $M^H u_{j-1}^* = q_{j-1}^*$
22.           $\alpha_{j-1} = \rho_{j-1}/\langle u_{j-1}^*, q_{j-1}\rangle$
23.           $x_j = x_{j-1} + \alpha_{j-1}p_{j-1}$
24.           $r_j = r_{j-1} - \alpha_{j-1}q_{j-1}$
25.           $z_j^* = z_{j-1}^* - \bar{\alpha}_{j-1}u_{j-1}^*$
26.           check convergence; continue if necessary
27. end
```

In Algorithm 3, one can obtain the unpreconditioned BiCOR method with the preconditioner M taken as the identity matrix. Since the BiCOR method is eventually mathematically equivalent to the BiCR method except for a different initial shadow residual, the computational cost for the BiCOR method is approximately the same as that for the BiCR method. For details on the computational cost for the BiCR method as well as its comparison with the BiCG method, see [47]. It should be emphasized that the shadow residual vectors r_j^*'s for both the BiCOR method and the BiCR method are updated in the same form and are only different in their initializations. We will see later in the numerical experiments that the differences in terms of the initial shadow residuals between the two methods can only lead to slightly different smooth convergence behaviors. In general, the BiCOR method practically behaves only a little more smoothly than the BiCR method. But the superiority of the BiCOR method to the BiCR method with respect to smooth convergence behavior can result in dramatically different smooth convergence behaviors in their corresponding variants (see, e.g., Example 1 in Section 5).

4. Two variants of BiCOR

Exploiting the ingenious ideas behind the CGS method [16] and the BiCGSTAB method [48], in this section, we develop two variants of the BiCOR method—the CORS method and the BiCORSTAB method successively. They are with the hope of increasing the effectiveness of the BiCOR method in certain circumstances. Essentially, the idea of the variants of the BiCOR method is the same as that of the BiCR variants—the CRS and BiCRSTAB methods as listed in Table 5.1 of [47]. Here, the CORS method can be regarded as a mathematically equivalent but numerically improved popularizing version of the CRS method for complex systems. And the BiCORSTAB is somewhat new and is described as a whole and precise algorithm while there was no whole and precise algorithm of the BiCRSTAB method given in [47].

First, the CORS method follows exactly a similar way in [16] for the derivation of the CGS method while taking the strategy of reducing the number of matrix-vector multiplications by introducing auxiliary vector recurrences and changing variables adopted for the CRS method in [47].

In Algorithm 3, by simple induction, the polynomial representations of the vectors r_j, r_j^*, p_j, p_j^* at step j can be expressed as follows:

$$r_j = \phi_j(A)r_0, \quad p_j = \pi_j(A)r_0,$$
$$r_j^* = \phi_j(A^H)r_0^*, \quad p_j^* = \pi_j(A^H)r_0^*,$$

where ϕ_j and π_j are Lanczos-type polynomials of degree less than or equal to j satisfying $\phi_j(0) = 1$.

Substituting these corresponding polynomial representations into (22) and (23) gives

$$\alpha_j = \frac{\langle r_j^*, Ar_j \rangle}{\langle A^H p_j^*, Ap_j \rangle} = \frac{\langle \phi_j(A^H) r_0^*, A\phi_j(A) r_0 \rangle}{\langle A^H \pi_j(A^H) r_0^*, A\pi_j(A) r_0 \rangle} = \frac{\langle r_0^*, A\phi_j^2(A) r_0 \rangle}{\langle r_0^*, A^2 \pi_j^2(A) r_0 \rangle},$$

$$\beta_j = \frac{\langle r_{j+1}^*, Ar_{j+1} \rangle}{\langle r_j^*, Ar_j \rangle} = \frac{\langle \phi_{j+1}(A^H) r_0^*, A\phi_{j+1}(A) r_0 \rangle}{\langle \phi_j(A^H) r_0^*, A\phi_j(A) r_0 \rangle} = \frac{\langle r_0^*, A\phi_{j+1}^2(A) r_0 \rangle}{\langle r_0^*, A\phi_j^2(A) r_0 \rangle}.$$

Also, note from (15) and (16) that ϕ_j and π_j can be expressed by the following recurrences:

$$\phi_{j+1}(t) = \phi_j(t) - \alpha_j t \pi_j(t),$$
$$\pi_{j+1}(t) = \phi_{j+1}(t) + \beta_j \pi_j(t).$$

By some algebraic computation with the help of the induction relations between ϕ_j and π_j and the strategy of reducing operations mentioned above, the desired CORS method can be obtained. The pseudocode for the resulting left preconditioned CORS method with a preconditioner M can be represented by the following scheme.

Algorithm 4: Left preconditioned CORS method

```
1.      Compute $r_0 = b - Ax_0$ for some initial guess $x_0$.
        Choose $r_0^* = P(A)r_0$ such that $\langle r_0^*, Ar_0 \rangle \neq 0$, where $P(t)$ is a polynomial in $t$.
        (For example, $r_0^* = Ar_0$).
2.      for $j = 1, 2, \ldots$
3.        solve $Mz_{j-1} = r_{j-1}$
4.        $\hat{r} = Az_{j-1}$
5.        $\rho_{j-1} = \langle r_0^*, \hat{r} \rangle$
6.        if $\rho_{j-1} = 0$, method fails
7.        if $j = 1$
8.          $e_0 = r_0$
9.          solve $Mze_0 = e_0$
10.           $d_0 = \hat{r}$
11.           $q_0 = \hat{r}$
12.         else
13.           $\beta_{j-2} = \rho_{j-1}/\rho_{j-2}$
14.           $e_{j-1} = r_{j-1} + \beta_{j-2}h_{j-2}$
15.           $ze_{j-1} = z_{j-1} + \beta_{j-2}zh_{j-2}$
16.           $d_{j-1} = \hat{r} + \beta_{j-2}f_{j-2}$
17.           $q_{j-1} = d_{j-1} + \beta_{j-2}(f_{j-2} + \beta_{j-2}q_{j-2})$
18.       endif
19.       solve $Mq = q_{j-1}$
20.       $\hat{q} = Aq$
21.       $\alpha_{j-1} = \rho_{j-1}/\langle r_0^*, \hat{q} \rangle$
22.       $h_{j-1} = e_{j-1} - \alpha_{j-1}q_{j-1}$
23.       $zh_{j-1} = ze_{j-1} - \alpha_{j-1}q$
24.       $f_{j-1} = d_{j-1} - \alpha_{j-1}\hat{q}$
25.       $x_j = x_{j-1} + \alpha_{j-1}(2ze_{j-1} - \alpha_{j-1}q)$
26.       $r_j = r_{j-1} - \alpha_{j-1}(2d_{j-1} - \alpha_{j-1}\hat{q})$
27.       check convergence; continue if necessary
28.     end
```

One iteration step of the CORS method involves about as many arithmetical operations as one step of the CRS method listed in Table 5.2 in [47]. Although the two methods are mathematically equivalent, the CORS method can lead to considerably smoother convergence behavior than the CRS method, as well as the CGS and SCGS methods, as illustrated by some figures reflecting the convergence behavior (see, e.g., Example 1 in Section 5). The main reason why the CORS method is sometimes much superior to the CRS method could come from the different choices of the initial shadow residuals, which work through the whole process at each iteration step; see [48] for a related discussion. In such cases, the CORS method gains a lot and sometimes may be amazingly competitive with the BiCGSTAB method (see, e.g., the same example above). However, the CORS method, like the CGS, SCGS, and CRS methods, is based on squaring the residual polynomial. In cases of irregular convergence, this may lead to substantial build-up of rounding errors and worse approximate solutions, or possibly even overflow (see, e.g., Example 3 in Section 5). For discussions on this effect and its consequences, see [7,26,48,55,56].

In order to remedy this difficulty and cure some of the numerical problems that plague the CORS method, to some extent, a more smoothly converging variant of the BiCOR method—the BiCORSTAB method is developed, without giving up the attractive speed of convergence of the CORS method (see, e.g., Example 2 in Section 5). The BiCORSTAB method is a polynomial product variant of the BiCOR method, which is adapted easily from the BiCGSTAB method [26,48,55] by a careful comparison and investigation. Still making use of the strategy of reducing operations utilized for the previous two methods, the pseudocode for the left preconditioned BiCORSTAB algorithm is shown below.

Algorithm 5: Left preconditioned BiCORSTAB method

```
1.    Compute $r_0 = b - Ax_0$ for some initial guess $x_0$.
      Choose $r_0^* = P(A)r_0$ such that $\langle r_0^*, Ar_0\rangle \neq 0$, where $P(t)$ is a polynomial in $t$.
      (For example, $r_0^* = Ar_0$).
2.    for $j = 1, 2, \ldots$
3.        $\hat{r} = Ar_{j-1}$
4.        $\rho_{j-1} = \langle r_0^*, \hat{r}\rangle$
5.        if $\rho_{j-1} = 0$, method fails
6.        if $j = 1$
7.            $p_0 = r_0$
8.            $q_0 = \hat{r}$
9.        else
10.           $\beta_{j-2} = (\rho_{j-1}/\rho_{j-2}) \times (\alpha_{j-2}/\omega_{j-2})$
11.           $p_{j-1} = r_{j-1} + \beta_{j-2}(p_{j-2} - \omega_{j-2}q_{j-2})$
12.           $q_{j-1} = \hat{r} + \beta_{j-2}(q_{j-2} - \omega_{j-2}\hat{q}_{j-2})$
13.       endif
14.       solve $M\hat{p} = p_{j-1}$
15.       $\hat{q}_{j-1} = Aq_{j-1}$
16.       $\alpha_{j-1} = \rho_{j-1}/\langle r_0^*, \hat{q}_{j-1}\rangle$
17.       $s = r_{j-1} - \alpha_{j-1}q_{j-1}$
18.       check norm of $s$; if small enough: set $x_j = x_{j-1} + \alpha_{j-1}\hat{p}$ and stop
19.       solve $M\hat{s} = s$
20.       $t = \hat{r} - \alpha_{j-1}\hat{q}_{j-1}$
21.       $\omega_{j-1} = \langle t, s\rangle / \langle t, t\rangle$
22.       $x_j = x_{j-1} + \alpha_{j-1}\hat{p} + \omega_{j-1}\hat{s}$
23.       $r_j = s - \omega_{j-1}t$
24.       check convergence; continue if necessary for continuation it is necessary that $\omega_{j-1} \neq 0$
25.   end
```

Note that line 18 in the above algorithm will generate "half" iterate each step during the algorithm implementation. So does the BiCGSTAB method; refer to [48,55]. In each iteration, the BiCORSTAB method requires two more additional operations for vector updates than the BiCGSTAB method. Compared to the BiCGSTAB method, the convergence behaviour of the BiCORSTAB method in certain cases is much smoother so that it sometimes produces much more accurate residual vectors (and, hence, better approximate solutions). And it even converges considerably faster than the BiCGSTAB method (see, e.g., Example 2 in Section 5). However, although sometimes the BiCORSTAB method converges a little more smoothly than the BiCGSTAB method, it does not improve the iteration process with respect to efficiency (see, e.g., Examples 1 and 3 in Section 5). In such cases, one may attempt to select other values for the pivotal parameters ω_{j-1} in line 21 of the above algorithm, such like what have been done for the BiCGSTAB2 method by Gutknecht [57] and the BiCGSTAB(l) method by Sleijpen and Fokkema [58] to improve the performance of the BiCORSTAB method. This aspect demands further research. By the way, one can also choose other polynomials to construct the initial shadow residuals in the first line of Algorithm 5 for improvement. By trial and error, the BiCORSTAB method for Example 1 in Section 5 is tested with several different inital shadow residuals to investigate its different performances. This actually offers a feasible direction to improve the BiCORSTAB method.

5. Examples and numerical experiments

Far from being exhaustive, in this section, the applicability of the three Lanczos-type variants of the COCR method is demonstrated for four different, but representative physical problems. The smoothing convergence effects obtained by the BiCOR method and its variants are evaluated in comparison with their counterparts related to the CBiCG and BiCR methods. Numerical comparisons are also made between the present methods and other methods, such as the SCGS method developed by Sogabe in [47], and the GMRES method first proposed by Saad and Schultz [59]. With appropriate preconditioning techniques, all the present methods are very attractive for solving relevant classes of complex nonsymmtric linear systems. But this is not the point we pursue here. All the experiments are performed *without preconditioning techniques*. That is, the preconditioner M in Algorithms 3–5 will be taken as the identity matrix. Refer to the outstanding survey by Benzi

[60] and the distinguished book by Saad [26] on preconditioning techniques for improving the performance and reliability of Krylov subspace methods.

The four sets of test problems as arising from *electromagnetics, discretizations of 2D/3D physical domains, acoustics and thermodynamics*, are described in Table 1. All of them are borrowed in *the MATLAB format from the University of Florida Sparse Matrix Collection provided by Davis* [61], in which the meanings of the column headers of Table 1 can be found. The experiments have been carried out with machine precision 10^{-16} in double precision floating point arithmetic in MATLAB 7.0.4 with a PC-Pentium (R) D CPU 3.00 GHz, 1 GB of RAM. We make comparisons in four aspects: number of iterations (referred to as *Iters*), CPU consuming time in seconds (referred to as *CPU*), $\log_{10}$ of the updated and final true relative residual 2-norms defined respectively as $\log_{10}\|r_n\|_2/\|r_0\|_2$ and $\log_{10}\|b - Ax_n\|_2/\|r_0\|_2$ (referred to as *Relres* and *TRR*). Numerical results in terms of *Iters*, *CPU* and *TRR* are reported by means of tables while convergence histories involved are shown in figures with *Iters* (on the horizontal axis) versus *Relres* (on the vertical axis). The stopping criteria used here is that the 2-norm of the residual be reduced by a factor (referred to as *TOL*) of the 2-norm of the initial residual, i.e., $\|r_n\|_2/\|r_0\|_2 < TOL$, or when *Iters* exceeded the maximal iteration number (referred to as *MAXIT*). Here, we take $MAXIT = 500$. All these tests are started with an initial guess equal to $\mathbf{0} \in \mathbb{C}^n$. Whenever the considered problem contains no right-hand side to the original linear system $Ax = b$, let

Table 1
Structures of the four sets of test problems.

Ex.	Group and name	id	♯ rows	♯ cols	Nonzeros	Problem kind	Sym. (%)
1	Dehghani/light_in_tissue	1873	29,282	29,282	406,084	Electromagnetics	0
2	Kim/kim1	862	38,415	38,415	933,195	2D/3D	0
3	HB/young1c	278	841	841	4089	Acoustics	85
4	Bindel/ted_AB_unscaled	1410	10,605	10,605	522,387	Thermal	0

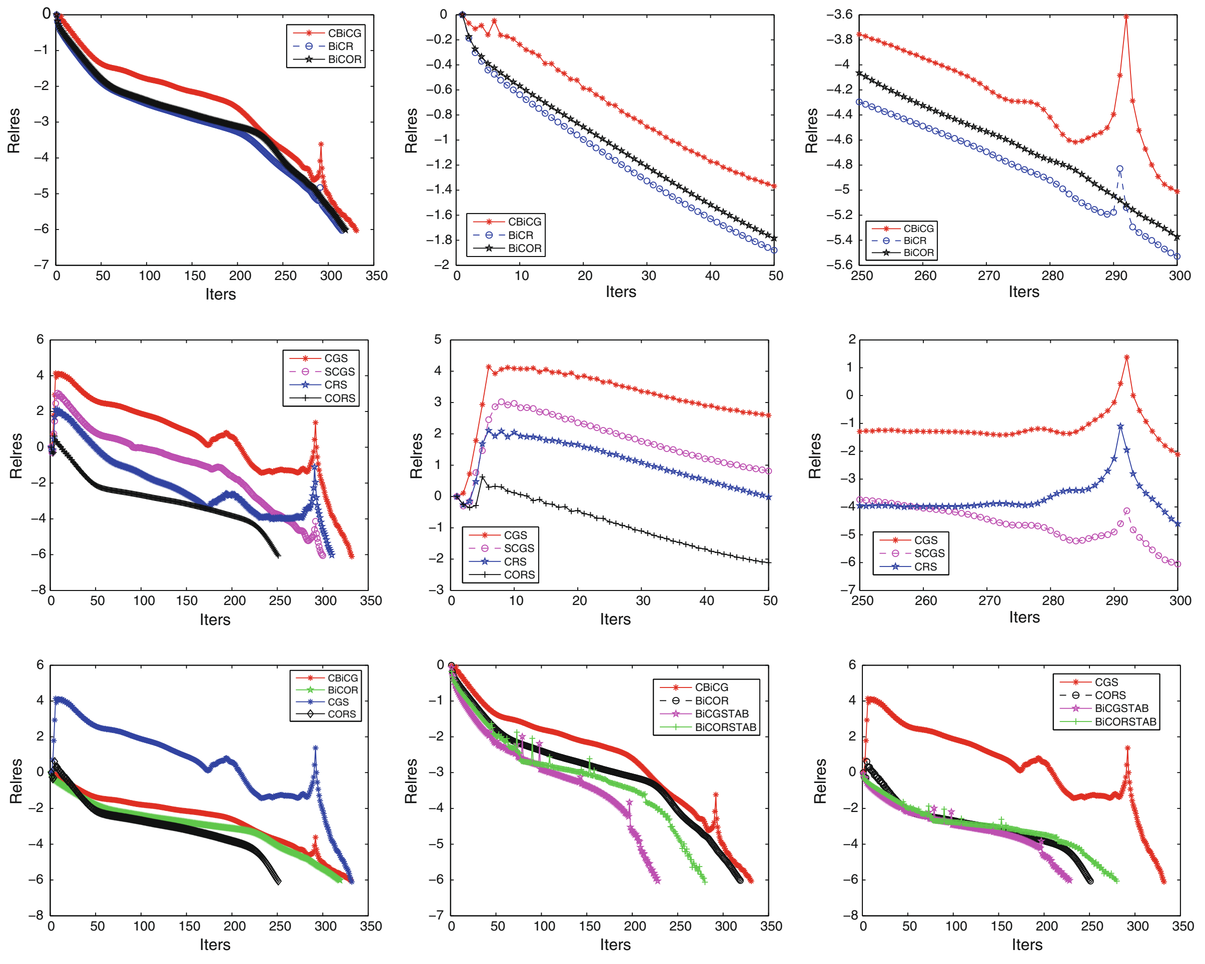


Fig. 1. Convergence histories of Example 1 with $TOL = 10^{-6}$.

$b = Ae$, where e is the $n \times 1$ vector whose elements are all equal to unity, such that $x = (1, 1, \ldots, 1)^T$ is the exact solution. It is stressed again that when there is no ambiguity or other clarification, the specific default choice for W with $r_0^* = Ar_0$ is adopted in the implementation of the proposed three methods in comparison with Sogabe's choice for W with $r_0^* = r_0$ in his implementation [46,47]. This has been already stated in the beginning part of Section 3. A symbol "*" is used to indicate that the method did not meet the required *TOL* before MAXIT or did not converge at all.

5.1. Example 1: Dehghani/light_in_tissue

The first example is one in which the CBiCG method converges quite smoothly except for two spots, into which two particular investigations are given in the above right two plots of Fig. 1. It is observed from the top three plots of Fig. 1 that the BiCR method seems to have about the same "asymptotical" speed of smooth convergence as the CBiCG method (if we discard the first jumping convergence behavior of the CBiCG method but admit its second one) and that the BiCOR method seems a little smoother than the other two.

Since all the CGS, SCGS, CRS, and CORS methods are based on doubling the residual polynomial, they inherit cases of irregular convergence to some extent. Therefore, similar phenomena for the latter four methods with respect to convergence behavior can be found in the middle left corner of Fig. 1. The phenomena are further examined concretely during two stages, one of which is in the beginning 50 *Iters* while the other is after 250 *Iters* displayed respectively in the middle right two plots of Fig. 1. Convergence histories after 250 *Iters* are only compared among the CGS, SCGS and CRS methods in the middle rightmost plot of Fig. 1, because the CORS method converged before 250 *Iters* shown in Table 2. In terms of *Iters* and *CPU*, the first six methods listed in the same table cost roughly the same while the CORS method requires much less, particularly only about 70% of the two terms for the BiCOR method. This surprisingly smoother and faster convergence behavior of the CORS method could come from the superior smooth convergence behavior of the BiCOR method to both the CBiCG and BiCR methods. From the bottom left two plots of Fig. 1, the CORS and BiCORSTAB methods indeed improve the BiCOR method in some aspects. However, it is observed that in this case the BiCGSTAB and BiCORSTAB methods both give more local peaks in the convergence curves than the CBiCG and BiCOR methods shown in the bottom center of Fig. 1. It may be taken as a possible explanation for why the BiCGSTAB and BiCORSTAB methods cost much more *CPU* than the CBiCG and BiCOR methods as listed in Table 2. Notably the convergence history of the CORS method, displayed between those of the BiCGSTAB and BiCORSTAB methods in the bottom rightmost plot of Fig. 1, is considerably smooth. Comparing the efficiency of the CORS and BiCGSTAB methods, we see in Table 2 that the former method requires about 50% of the latter method in terms of *CPU* while only taking about 0.09% more *Iters* than the latter. Therefore, the CORS method in such cases can even be remarkably competitive with the BiCGSTAB method.

It is found in Table 2 that the BiCORSTAB method requires more *Iters* and *CPU* than the BiCGSTAB method, although the two methods have similar convergence behaviors. Notice that the *Iters* here for the BiCGSTAB and BiCORSTAB methods are not integer. The reason is that line 18 in Algorithm 5 will generate "half" iterate each step during the implementation of the BiCORSTAB method. So does the BiCGSTAB method; refer to [48,55]. Such phenomena with respect to the "half" iterate in the other numerical reports for the BiCGSTAB and BiCORSTAB methods can be explained the same way. In order to improve the BiCORSTAB method, we examined the performance of the BiCORSTAB method with different initial shadow residuals by trial and error. Here, we choose four of them for illustration, respectively with $r_0^* = r_0,\ r_0^* = Ar_0,\ r_0^* = r_0 + Ar_0$, and $r_0^* = A^2 r_0$. Note that the second choice with $r_0^* = Ar_0$ is the default for the BiCORSTAB method used usually in the paper. In convenience of comparison, the four cases are named the BiCORSTAB-r, BiCORSTAB-Ar, BiCORSTAB-rAr, BiCORSTAB-A2r, respectively. Here, the BiCORSTAB-Ar method is exactly the BiCORSTAB used acquiescently. From the top left corner of Fig. 2, among the compared four cases, we see that the BiCORSTAB-r method performs best. Then the BiCORSTAB-r method is further taken separately to compare with the BiCGSTAB method in the last three plots of Fig. 2. More smooth convergence properties of the BiCORSTAB-r method can be observed in comparison with the BiCGSTAB method, resulting in a slightly faster convergence rate in terms of *CPU* but *Iters* by comparison of the corresponding results in Tables 2 and 3. This suggests a feasible direction to improve the performance of the BiCORSTAB method.

Finally, we state that all the methods used for this example can achieve the specified accuracy in the final results in terms of *TRR* as shown in Tables 2 and 3. In addition, the BiCORSTAB-A2r method seems to numerically provide a more accurate

Table 2
Comparison results of Example 1 with $TOL = 10^{-6}$.

Method	*Iters*	*CPU*	*TRR*
CBiCG	330	24.6329	−6.0318
BiCR	314	21.1208	−6.0257
BiCOR	318	21.3432	−6.0150
CGS	331	26.3266	−6.1014
SCGS	299	20.1166	−6.0526
CRS	309	22.9095	−6.0254
CORS	**250**	**15.1079**	−6.0645
BiCGSTAB	227.5	28.5066	−6.0295
BiCORSTAB	279.5	30.9885	−6.0611

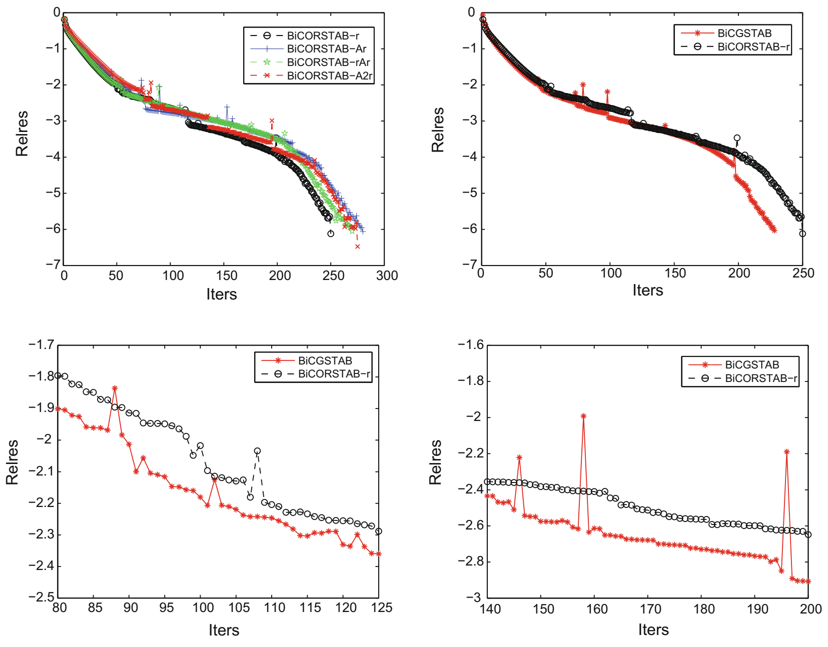


Fig. 2. Convergence histories of BiCORSTAB with different r_0^*'s for Example 1.

Table 3
Numerical results of BiCORSTAB with different r_0^* in Example 1.

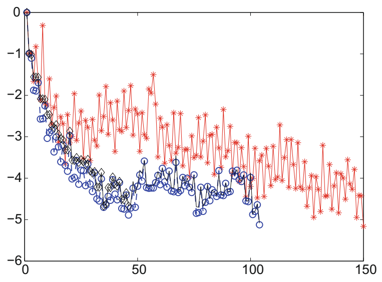
r_0^*	r_0	Ar_0	$r_0 + Ar_0$	A^2r_0
Iters	**249.5**	279.5	269.5	274.5
CPU	**27.9903**	30.9885	30.2427	30.8184
TRR	−6.1199	−6.0611	−6.0518	−6.4744

final solution than the other involved methods in terms of *TRR*, which gives a possible way to improve the accuracy of the final solution.

5.2. Example 2: Kim/kim1

In this situation, the CBiCG method, as observed in the top leftmost plot of Fig. 3, takes on many local peaks in the convergence curve. In such a case, the CGS method would be expected to have some strong effects of irregularities and quite adverse effects on cancellation. See also the highly erratic effects of the CGS method for the next example (Section 5.3). However, in this example, although the local effects in the CGS method are much more violent than those of the CBiCG method as shown in the middle rightmost plot of Fig. 3, these peaks do not seem to delay the convergence of the CGS method. The CGS method converges faster eventually. Comparing the efficiency of the CBiCG and CGS methods, we see in Table 4 that the latter method requires respectively about 40% and 60% in term of *Iters* and *CPU* with $TOL = 10^{-6}$ and $TOL = 10^{-5}$, which are approximately in accord with the theoretical analysis and heuristic arguments in [16]. Similar faster convergence rates with $TOL = 10^{-6}$ are also observed for the CRS and CORS methods compared with the BiCR and BiCOR methods, respectively. Moreover, with $TOL = 10^{-5}$, the CRS and CORS methods require only about 25% of the *Iters* and *CPU* of the BiCR and BiCOR methods. A notable finding with $TOL = 10^{-5}$ is also observed for the faster convergence of the BiCORSTAB method in comparison with the BiCGSTAB method shown in the last two lines of Table 4. Still looking at this table, the BiCOR method requires the same *Iters* as the BiCR method but is a little faster in terms of *CPU*. Similar phenomena exist when comparing the CORS method with the SCGS and CRS methods.

In order to find the reason why these methods perform dramatically differently with *TOLs* different only in one order, the convergence histories of these methods were examined. We found that some of the compared methods such as the CRS, CORS and BiCORSTAB methods stagnated for some time after the 2-norm of the updated residual achieved the accuracy of $TOL = 10^{-5}$ while the others were still in certain oscillation. Since the main concern of this paper are smooth effects obtained by the BiCOR method and its two variants compared with the CBiCG method and its two corresponding variants. Also taking the better performances obtained by our concerned methods with $TOL = 10^{-5}$ into account, we choose $TOL = 10^{-5}$ to illustrate the convergence histories in Fig. 3. An investigation has been made for comparison of the CBiCG, BiCR and BiCOR

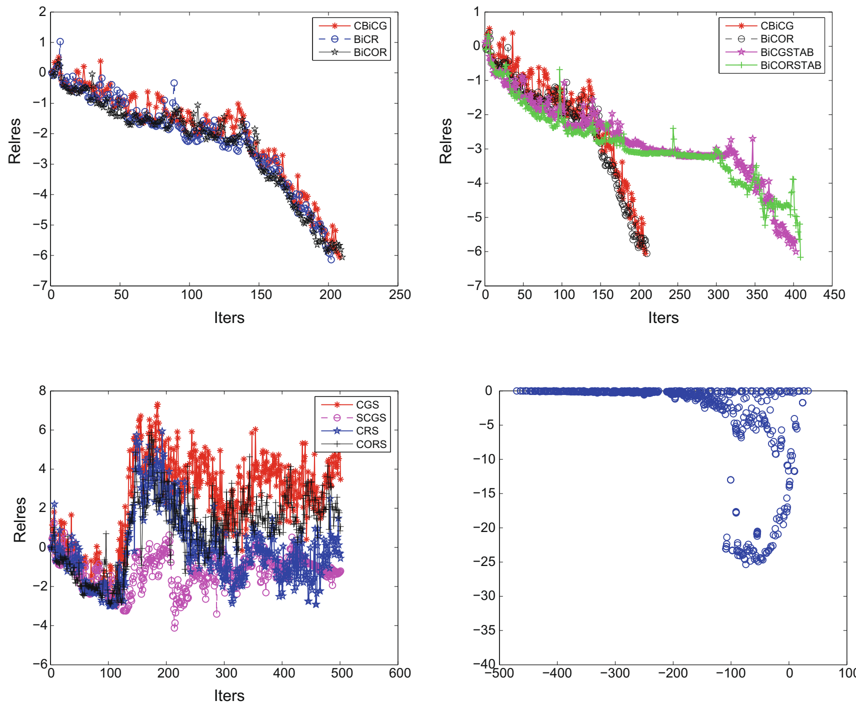


methods by dividing the whole converging process into two stages as shown in the top right two plots of Fig. 3. Similar more smooth convergence properties of the latter two methods can be discovered in comparison with the first one. As was addressed by van der Vorst in [48,62], in such cases, the CBiCG method loses orthogonality among the residuals in a very early phase. Because of the superiority of the BiCR and BiCOR methods to the CBiCG method with respect to smooth convergence behavior, the CRS and CORS methods converge much more smoothly and faster than the CGS method. So does the SCGS method, but it is not as smooth as the CRS and CORS methods. See the middle left two plots in Fig. 3 for details.

Finally, observing the bottom leftmost plot of Fig. 3, in this case with $TOL = 10^{-5}$, the BiCORSTAB method, apart from converging much faster in term of *Iters* and *CPU* than the BiCGSTAB method shown in Table 4, is also apparently more stable. It is concluded here that the BiCORSTAB method may be regarded as a rather efficient and more robust choice for such cases of low accuracy. In addition, looking at Table 4 with $TOL = 10^{-5}$ and the last two plots of Fig. 3, the BiCORSTAB method improves the BiCOR method with respect to smooth and fast convergence behavior and the CORS method with respect to smooth convergence behavior, as we might expect from the purpose of developing the BiCORSTAB method. In addition, with $TOL = 10^{-5}$, this is a typical situation where the CORS method as well as the CRS method is much more efficient than the BiCGSTAB method in terms of *Iters* and *CPU* while keeping similarly "asymptotical" convergence behaviors by observation of Table 4 and the last plot of Fig. 3.

5.3. Example 3: HB/young1c

In the third example, the CBiCG method has small irregularities and does not converge superlinearly, as reflected in first plot of Fig. 4. The BiCOR and BiCR methods seem to have almost the same "asymptotical" speed of convergence as the CBiCG method while the BiCOR method seems a little smoother than the other two. From the second plot in Fig. 4, the BiCORSTAB and BiCGSTAB methods also have similar convergence behaviors. As observed for the CGS method in Example 2, in this case, the CGS, SCGS, CRS and CORS methods decreased far from monotonously, but fluctuated rather strongly and suffered a lot as shown in the third plot of Fig. 4 and Table 5. As stated in [16], the residuals produced by the latter four methods made quite large jumps in the "wrong" direction, which might lead to grievous cancellation, spoiling the solution delivered by the process in terms of *TRR* in Table 5. For related discussions on this effect and its consequences, see also [26,48,62]. At this point, it



is necessary to develop an algorithm to escape from this problem. This can give a necessary support for the development of the BiCGSTAB and BiCORSTAB methods.

Looking again at the second plot in Fig. 4, although the BiCGSTAB and BiCORSTAB methods converge a little more smoothly than the CBiCG and BiCOR methods, they do not improve the efficiency in terms of *Iters* and *CPU* shown in Table 5. We exploited the spectrum of the coefficient matrix as presented in the last plot of Fig. 4. A possible explanation is given by Simoncini and Szyld [7] for when the BiCGSTAB method is not very effective. That is when the spectrum has large imaginary components. So observing the last plot of Fig. 4, it may also give some reason why the BiCORSTAB method does not perform better than the BiCOR method in this case. In such cases, we may fall in favor of the BiCGSTAB(l) method by Sleijpen and Fokkema [58]. As Sleijpen suggested, the BiCGSTAB(l) solver may capture those eigenvalues more effectively to show good performance. Therefore, it may be instructive to later pursue the research direction to select other values for the pivotal parameters ω_{j-1} in line 21 of Algorithm 5 to improve the BiCORSTAB method.

5.4. Example 4: Bindel/ted_AB_unscaled

It is a strange but interesting example for our present methods to respectively perform much more smoothly than the CBiCG method and its two variants, as reflected in Fig. 5. It seemed that all the first seven methods listed in Table 6 were terminated by satisfying the stopping criteria with $TOL = 10^{-6}$ in terms of *TRR*. In fact, the solutions obtained by these methods deviated far from the desired solution after a complete investigation. They all deteriorated seriously in components after about 4200. So the word "fake" is used here to characterize this amazing phenomenon, which is fun to study further.

By the end of this section, we would like to make some further comparisons. The GMRES variant, as it is well known, became the de facto standard for nonsymmetric linear systems [63]. Like all methods satisfying certain minimum residual conditions on nested subspaces, the GMRES method can generate a non-increasing sequence of residual norms [7]. As stressed by Simoncini and Szyld [7], most methods including the GMRES method reviewed there can be employed with no change in case the linear system is complex, with the use of the standard Hermitian inner product. At the end of this section, some comparative experiments between the BiCOR/CORS/BiCORSTAB family and the GMRES method without restarting are performed. For details on the implementation of the GMRES method, refer to Saad [26]. Numerical results with the GMRES method for the above four examples are listed in Table 7 and comparative convergence histories between the BiCOR/CORS/BiCORSTAB family and the GMRES method are illustrated in Fig. 6. Carefully comparing the reports displayed in Table 7 with those in Tables 2,4, 5 and 6 and looking at Fig. 6, the following observations and remarks can be made. For Example 1, the GMRES method costs too much computational time while behaves a little more smoothly than the other three. Both of smooth convergence behavior and efficiency in terms of *Iters* and *CPU* considered, the BiCOR/CORS/BiCORSTAB family is advantageous and the CORS method is the best favourable in such cases. The CORS and BiCORSTAB methods have about the same "asymptotical" speed of convergence as the GMRES method (if we ignore the local peaks in the former two

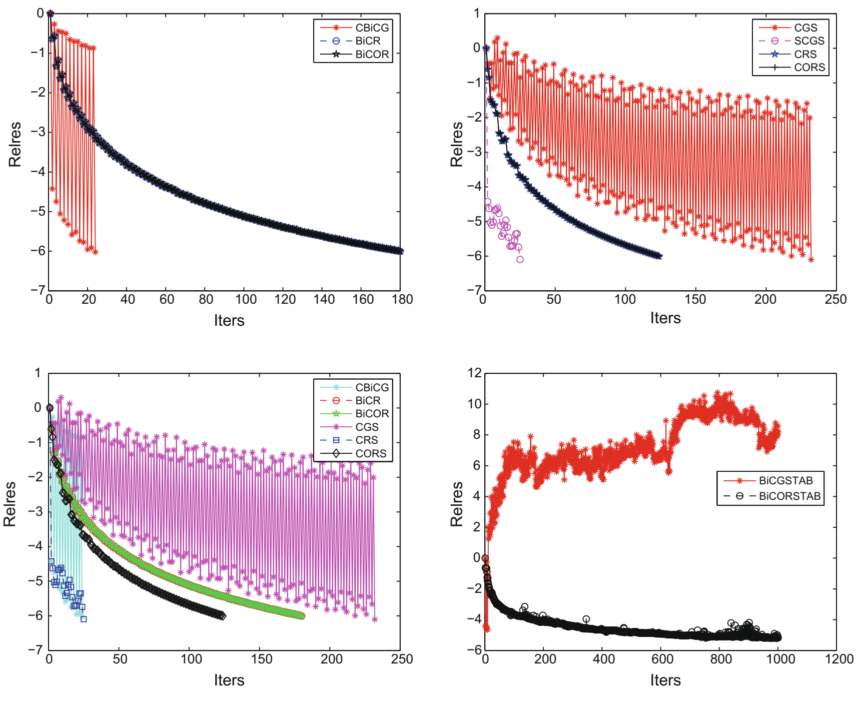


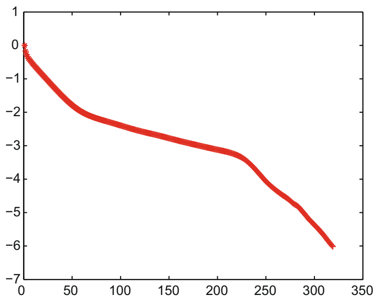
Fig. 5. Fake convergence histories of Example 4 with $TOL = 10^{-6}$.

Table 6
Fake comparison results of Example 4 with $TOL = 10^{-6}$.

Method	*Iters*	*CPU*	*TRR*
CBiCG	23.0000	1.7712	−6.0232
BiCR	179.0000	9.3001	−6.0037
BiCOR	179.0000	9.2359	−6.0037
CGS	231.0000	17.1599	−6.1006
SCGS	24.0000	1.4112	−6.0946
CRS	123.0000	7.0774	−6.0040
CORS	123.0000	6.5109	−6.0040
BiCGSTAB*	3.0000	53.8244	−4.6243
BiCORSTAB*	497.0000	35.4164	−5.2073

Table 7
Numerical results of the four examples with GMRES.

Example	1	2	3	4
TOL	10^{-6}	10^{-5}	10^{-6}	10^{-6}
Iters	314	34	181	17
CPU	132.0694	33.3161	1.3470	1.2046
TRR	−6.0260	−5.0503	−6.0107	−6.0421



methods' curves) in Example 2 while the GMRES costs so much in terms of *CPU*. In Example 3, similar phenomenon is observed for the BiCOR method in comparison with the GMRES method with respect to "asymptotical" convergence behavior. So in cases with quite irregular convergence behaviors, the GMRES method is recommended if storage requirement is allowed. With regard to Example 4, the final solution given by the GMRES method deviated strongly from the desired solution, similarly to what have been observed in the final solution results of the other three methods.

6. Concluding remarks

First and foremost, the Biconjugate *A*-Orthonormalization Procedure outlined here suggests a generalized choice for the constraints subspace. Based on a specific choice in the implementation, the proposed BiCOR method and its two variants have some appealing properties in smoothing effect according to the numerical experiments when solving complex nonsymmetric linear systems. Besides smoother convergence behavior, sometimes they converged much faster than their counterparts related to the CBiCG method as well as the BiCR method in [47]. But the obtained dramatically smoothing convergence behavior is harder to analyze theoretically. Moreover, one is often faced with a quite irregular convergence behavior in many practical situations, as partly illustrated in the numerical experiments.

In addtion, it should be kept in mind that smoothing the residual does not often improve the numerical properties of the primary methods [7,34]. As observed in Tables 2–6, the methods with smoothing effects do not provide a much more accurate final solution than the non-smoothed methods in terms of *TRR*. In fact, what really counts in practice is that a method finds a solution (with a certain stopping criteria) as quickly as possible. The smoothness of the convergence does not always matter from that point of view; see [64].

Nevertheless, what is amazing is that under certain conditions, our present methods may be competitive to the CBiCG method and its two variants as well as Sogabe's methods [46,47] in both aspects of smooth effect and efficiency.

The proposed Lanczos-type variants combined with preconditioning techniques to deal with large electromagnetic problems will be the subject of another future research.

Acknowledgments

The authors would like to thank Professors Knoll and Basil Nyaku and the anonymous reviewers for their insightful comments and constructive and valuable suggestions. One of the reviewers drew our attention to [47], which contributed substantially to this work and added nicely to the presentation of the paper. Their suggested numerous improvements are deeply appreciated, which led to both a clearer presentation of the present methods and a more extensive analysis of the numerical experiments.

This research was supported by NSFC (10771030), 973 Programs (2008CB317110, 2007CB311000), the Scientific and Technological Key Project of the Chinese Education Ministry (107098), the Specialized Research Fund for the Doctoral Program of Higher Education (20070614001), Sichuan Province Project for Applied Basic Research (2008JY0052), the Project of National Defense Key Lab (9140C6902030906) and the Project for Academic Leader and Group of UESTC.

References

[1] M.D. Pocock, S.P. Walker, The complex bi-conjugate gradient solver applied to large electromagnetic scattering problems, computational costs, and cost scalings, IEEE Trans. Antennas Propagat. 45 (1997) 140–146.
[2] T. Sogabe, S.-L. Zhang, A COCR method for solving complex symmetric linear systems, J. Comput. Appl. Math. 199 (2007) 297–303.
[3] J. Dongarra, F. Sullivan, Guest editors' introduction to the top 10 algorithms, Comput. Sci. Eng. 2 (1) (2000) 22–23.
[4] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bureau Standards 49 (1952) 409–436.
[5] C. Lanczos, Solution of systems of linear equations by minized itertions, J. Res. Nat. Bureau Standards 49 (1952) 33–53.
[6] C. Brezinski, Padé Type Approximation and General Orthogonal Polynomials, Birkhäuser-Verlag, Basel/Boston/Stuttgart, 1980.
[7] V. Simoncini, D.B. Szyld, Recent computational developments in Krylov subspace methods for linear systems, Numer. Linear Algebra Appl. 14 (2007) 1–59.
[8] R. Fletcher, Conjugate Gradient Methods for Indefinite Systems, Lecture Notes in Mathematics, vol. 506, Springer-Verlag, Berlin/Heidelberg/New York, 1976. pp. 73–89.
[9] D.A.H. Jacobs, The Exploitation of Sparsity by Iterative Methods, Sparse Matrices and their Uses, I.S. Duff, Springer, 1981, pp. 191–222.
[10] D.A.H. Jacobs, A generalization of the conjugate gradient method to solve complex systems, IMA J. Numer. Anal. 6 (1986) 447–452.
[11] T.K. Sarkar, On the application of the generalized biconjugate gradient method, J. Electromagnet. Waves Appl. 1 (1987) 223–242.
[12] G. Markham, Conjugate gradient type methods for indefinite, asymmetric, and complex systems, IMA J. Numer. Anal. 10 (1990) 155–170.
[13] C.F. Smith, A.F. Peterson, R. Mittra, The biconjugate gradient method for electromagnetic scatterings, IEEE Trans. Antennas Propagat. 38 (1990) 938–940.
[14] P. Joly, G. Meurant, Complex conjugate gradient methods, Numer. Algo. 4 (1993) 379–406.
[15] P. Joly, Méthode de gradient conjugué, Publications du Laboratoire d'Analyse Numérique Université Pierre et Marie Curie, Paris, 1984.
[16] P. Sonneveld, CGS a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 10 (1989) 36–52.
[17] M. Clemens, T. Weiland, Iterative methods for the solution of very large complex-symmetric linear systems of equations in electromagnetics, in: T.A. Manteuffel, S.F. McCormick (Eds.), Eleventh Copper Mountain Conference on Iterative Methods, Part 2, 1996.
[18] H.A. van der Vorst, J.B.M. Melissen, A Petrov–Galerkin type method for solving $Ax = b$, where A is symmetric complex, IEEE Trans. Mag. 26 (1990) 706–708.
[19] J. Cullum, W. Kerner, R.A. Willoughby, A generalized nonsymmetric Lanczos procedure, Comput. Phys. Commu. 53 (1989) 19–48.
[20] J. Cullum, R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Classics in Applied Mathematics, vol. 41, Birkhaüser, Basel, 1985. vol. 1, Theory, vol. 2, Programs, vol. 1 reprinted by SIAM, Philadelphia, PA, 2002.
[21] M. Clemens, T. Weiland, Comparison of Krylov-type methods for complex linear systems applied to high-voltage problems, IEEE Trans. Mag. 5 (1998) 3335–3338.
[22] Liang Li, Ting-Zhu Huang, Yan-Fei Jing, Application of the incomplete Cholesky factorization preconditioned conjugated orthogonal conjugate gradient algorithm to the vector FEM for 3-D electromagnetic scattering problems, Comput. Phys. Commun., submitted for publication.
[23] R. Weiss, Parameter-Free Iterative Linear Solvers, Mathematical Research, vol. 97, Akademie Verlag, Berlin, 1996.
[24] R. Weiss, Convergence behavior of generalized conjugate gradient method, Ph.D. Dissertation, University of Karlsruhe, 1990.
[25] R. Weiss, Properties of generalized conjugate gradient methods, J. Numer. Linear Algebra Appl. 1 (1994) 45–63.
[26] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., The PWS Publishing Company, Boston, 1996. SIAM, Philadelphia, PA, 2003.
[27] Martin H. Gutknecht, Miroslav Rozložník, A framework for generalized conjugate gradient methods – with special emphasis on contributions by Rűdiger Weiss, Appl. Numer. Math. 41 (2002) 7–22.
[28] W. Schönauer, The efficient solution of large linear systems, resulting from the fdm for 3-d PDE's, on vector computers, in: Proceedings of the 1st International colloquium on vector and parallel computing in Scientific Applications, Paris, March 1983, 1983.
[29] B. Hendrickson, R. Leland, An improved spectral graph partitioning algorithm for mapping parallel computations, Technical Report SAND92-1460, UC-405, Sandia National Laboratories, Albuquerque, NM, 1992.
[30] C. Brezinski, M. Redivo-Zaglia, Hybrid procedures for solving systems of linear equations, Numer. Math. 67 (1994) 1–19.

[31] L. Zhou, H.F. Walker, Residual smoothing techniques for iterative methods, SIAM J. Sci. Comput. 15 (1994) 297–312.
[32] R. Weiss, A theoretical overview of Krylov subspace methods, in: W. Schönauer, R. Weiss (Eds.), Special Issue on Iterative Methods for Linear Systems Applied Numerical Methods, 1995, pp. 33–56.
[33] R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.
[34] Martin H. Gutknecht, Miroslav Rozložník, Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers?, BIT 41 (2001) 086–114
[35] Martin H. Gutknecht, Miroslav Rozložník, By how much can residual minization accelerate the convergence of orthogonal residual methods?, Numer Algo. 27 (2001) 189–213.
[36] C.C. Paige, M.A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.
[37] R.W. Freund, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, SIAM J. Sci. Comput. 14 (1993) 470–482.
[38] P.N. Brown, A therical comparison of the Arnoldi and GMRES algorithms, SIAM J. Sci. Statist. Comput. 12 (1991) 58–78.
[39] H.F. Walker, Residual smoothing and peak/plateau behavior in Krylov subspace methods, Appl. Numer. Math. 19 (1995) 279–286.
[40] J. Cullum, Peaks, plateaus, numerical instabilities in a Galerkin/minimal residual pair of methods for solving $Ax = b$, Appl. Numer. Math. 19 (1995) 255–278.
[41] J. Cullum, A. Greenbaum, Relations between Galerkin and norm-minimizing iterative methods for solving linear systems, SIAM J. Matrix Anal. Appl. 17 (1996) 223–247.
[42] H. Sadok, Analysis of the convergence of the minimal and the orthogonal residual methods, Reprint, 1997.
[43] M. Eiermann, O. Ernst, Geometric aspects in the theory of Krylov space methods, Acta Numer. 10 (2001) 251–312.
[44] E. Stiefel, Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme, Comment. Math. Helv. 29 (1955) 157–179.
[45] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, Cambridge, 2003. pp. 97–98.
[46] 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.
[47] T. Sogabe, Extensions of the conjugate residual method, Ph.D. Dissertation, University of Tokyo, 2006.
[48] 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.
[49] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bureau Standards 45 (1950) 255–281.
[50] B. Parlett, D. Taylor, Z.-S. Liu, A look-ahead Lanczos algorithm for unsymmetric matrices, Math. Comput. 44 (1985) 105–124.
[51] B. Parlett, Reduction to tridiagonal form and minimal realizations, SIAM J. Matrix Anal. Appl. 13 (1992) 567–593.
[52] R. Freund, M.H. Gutknecht, N. Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, SIAM J. Sci. Stat. Comput. 14 (1993) 137–158.
[53] M.H. Gutknecht, Lanczos-type solvers for nonsymmetric linear systems of equations, Acta Numer. 6 (1997) 271–397.
[54] D. Day, An efficient implementation of the nonsymmetric Lanczos algorithms, SIAM J. Matrix Anal. Appl. 18 (1997) 566–589.
[55] R. Barrett, M.W. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H.A. van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, second ed., SIAM, Philadelphia, PA, 1994.
[56] G.L.G. Sleijpen, H.A. van der Vorst, An overview of approaches for the stable computation of hybrid BiCG method, Appl. Numer. Math. 19 (1995) 235–254.
[57] M.H. Gutknecht, Variants of BiCGSTAB for matrices with complex spectrum, SIAM J. Sci. Comput. 14 (1993) 1020–1033.
[58] G.L.G. Sleijpen, D.R. Fokkema, BiCGSTAB(l) for linear equations involving unsymmetric matrices with complex spectrum, Electron. Trans. Numer. Anal. 1 (1993) 11–32.
[59] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving a nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[60] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2002) 418–477.
[61] T. Davis, The University of Florida Sparse Matrix Collection, NA Digest 97 (23) (1997).
[62] H.A. van der Vorst, The convergence behavior of preconditioned CG and CG-S in the presence of rounding errors, Lecture Notes Math. 1457 (1990) 126–136.
[63] Y. Saad, H.A. van der Vorst, Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math. 123 (2000) 1–33.
[64] M.H. Gutknecht, Residual smoothing for Krylov space solvers: does it help at all? in: Seminar for Applied Mathematics, ETH Zurich, <http://www.sam.math.ethz.ch/~mhg>.